

Technische Universität München

Fakultät für Informatik

Eine Cache–optimale Implementierung eines Navier–Stokes
Lösers unter besonderer Berücksichtigung physikalischer
Erhaltungssätze

Diplomarbeit
Tobias Weinzierl

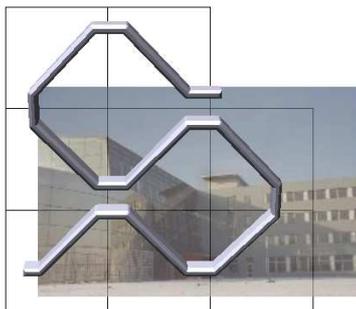
Aufgabensteller : Univ.–Prof. Dr. Dr. hc Christoph Zenger
Betreuer : Dr. Miriam Mehl
Dr. Frank Günther
Abgabedatum : 15. Februar 2005

Ich erkläre hiermit, dass ich diese Arbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe.

München, den 3. Februar 2005

.....

Vorwort



Es gibt eine Vielzahl von Möglichkeiten, die numerische Simulation kontinuierlicher deterministischer Modelle, wie sie durch partielle Differentialgleichungen gegeben sind, zu motivieren. Ganz nach Geschmack des Autors und der in der Arbeit behandelten Differentialgleichung muss dazu von dem Schiff Alinghi bis zum Dach des Münchner Olympiastadions irgendein probater Aufhänger herhalten. Tatsache ist, dass der Bereich der numerischen Simulation im wissenschaftlichen Rechnen ein interdisziplinäres Gebiet mit den drei Hauptsäulen *Mathematik*, *Informatik* und *Anwendungsdomäne* ist.

In dieser Arbeit wird die Simulation von Fluiden thematisiert, sie ist also dem Bereich der numerischen Strömungssimulation zuzuordnen. Eine Motivation der Behandlung von Fluiden zu geben erscheint dabei aufgrund der Fülle der Publikationen¹ zur Relevanz dieses Problems als nicht unbedingt essentiell. Konkret bietet die Simulation laminarer inkompressibler Fluide in der Zeit jedoch die Möglichkeit, gleich eine ganze Fülle an interessanten Aspekten des wissenschaftlichen Rechnens zu thematisieren.

So weisen die Navier–Stokes Gleichungen aus mathematischer Sicht eine Reihe von Eigenschaften auf, die theoretisch sehr interessant sind. Dabei ist als Beispiel die Reynoldszahl–abhängige Hybrideigenschaft zwischen elliptischem und nichtlinearem Problem zu einem Zeitpunkt zu nennen. Die Zeitabhängigkeit an sich offeriert ebenfalls ein breites Betätigungsfeld. Dem Numeriker bietet sich dabei die Möglichkeit, für nahezu das komplette Sammelsurium an Techniken, von den Finiten–Differenzen, der Finite–Elemente–Methode über die Mehrgittermethodik zur adaptiven Fehlerschätzern, sinnvolle Anwendungsfelder zu finden.

Bei der Umsetzung der mathematischen Erkenntnisse sieht sich der Informatiker mit der Notwendigkeit einer effizienten Implementierung bezüglich aktueller Rechnerarchitekturen konfrontiert. Die Zeitabhängigkeit der Differentialgleichungen bringt darüber hinaus unabdingbar die Problematik großer Datenmengen mit sich. Letztendlich ist eine praxisrelevante Behandlung der Problematik

¹Man betrachte alleine die Literaturliste von [Gresho 98] oder [Griebel 94]

also nur unter Einsatz des Wissens um Hochleistungsarchitekturen und probater Parallelisierungsstrategien möglich.

Historie und Ziel

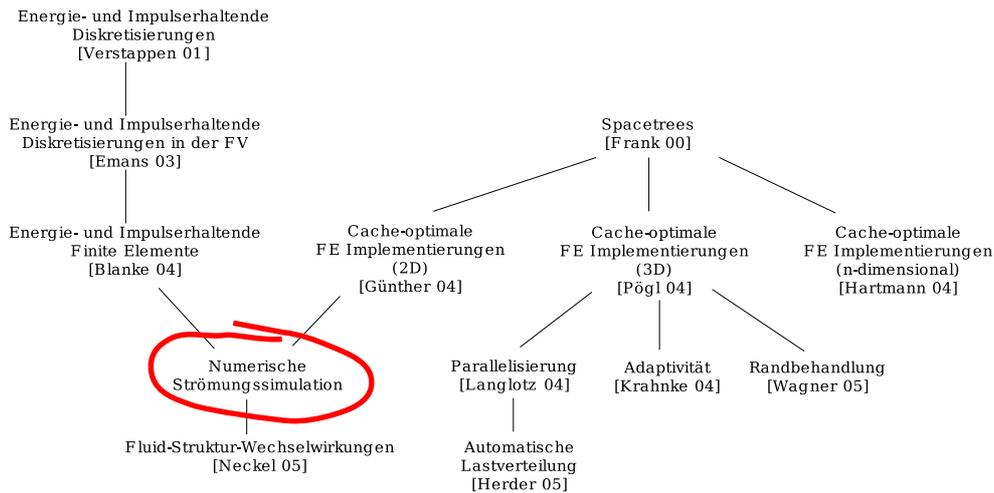


Abbildung 0.1: Verschiedene wichtige Arbeiten (ohne Anspruch auf Vollständigkeit).

In dieser Arbeit werden einige interessante Aspekte aus obiger unstrukturierter Aufzählung herausgegriffen. Dabei steht der Text in der Tradition eines ganzen Pulks themenverwandter Werke. Diese Historie ist entsprechend zu würdigen und kommt über die Literaturverweise zum Ausdruck.

Die Gleichungen der Strömungsmechanik, die Gegenstand dieser Arbeit sind, lassen sich zurückführen auf physikalische Erhaltungssätze in der Natur. Explizit genannt sind dies Massen-, Energie- und Impulserhaltung. Wird die Massenerhaltung im hier vorliegenden Modell explizit sichergestellt, so sind die anderen Erhaltungssätze nur noch implizit in den die Fluide beschreibenden Gleichungen gegeben. In [Verstappen 01] werden nun die diskrete Energie und der diskrete Impuls für eine beliebige Diskretisierung dieser Gleichungen untersucht. Soll das numerische Ergebnis akkurat sein, so müssen trivialerweise auch die korrespondierenden diskreten Erhaltungssätze gelten. Aus diesen leitet nun [Verstappen 01] Anforderungen an die Diskretisierungsmatrizen ab. In [Blanke 04] wurden nun Funktionenräume konstruiert, bei denen die Matrizen zur Finite-Elemente-Methode genau die postulierten Erhaltungseigenschaften aufweisen. Diese Elemente kommen in dieser Arbeit zur Anwendung.

Der andere Haupteinfluss geht auf die Beobachtung zurück, dass man Spacetreer-basierte Datenstrukturen [Frank 00] mittels raumfüllender Kurven [Sagan 94] linearisieren kann. Aufbauend auf dieser Linearisierung haben nun [Günther 04] für den zweidimensionalen und [Pögl 04] für den dreidimensionalen Fall ein Konzept entworfen, wie sich die Finite-Elemente-Methode effizient bezüglich heutiger Rechnerarchitekturen realisieren lässt. Schlüsseltechnik ist dabei die exklusive Verwendung von Stacks als Datenstrukturen, die die Eigenschaft der Cache-Optimalität aufweisen. Wie bei der Untersuchung neuer numerischer Ansätze üblich haben sich beide Autoren dabei

schwerpunktmäßig auf die Untersuchung der Poisson–Gleichung konzentriert. Eine Beschäftigung mit praxisrelevanten oder gar zeitabhängigen Differentialgleichung fand nicht statt. Dieser Schritt wird erst mit der hier vorliegenden Arbeit vollzogen.

Ziel ist es also, erstmals einen FE Ansatz zur Lösung instationärer Navier–Stokes Gleichungen zu schaffen, der den physikalischen Erhaltungssätzen per se genügt. Die Realisierung soll dabei Cache–optimal im Sinne von [Günther 04] erfolgen. Diese Anforderungen sind auch direkt dem Titel der Arbeit zu entnehmen. Nicht dem Titel zu entnehmen ist, dass numerische Standardtechniken selbstverständlich voll und ganz einzubeziehen sind. Insbesondere sei hier die Mehrgittertechnik und die Unterstützung adaptiver Gitter genannt. Resultat ist ein neuartiger Algorithmus, der alle diese Anforderungen erfüllt und damit Standardverfahren zur Lösung der Fluidgleichungen überlegen ist.

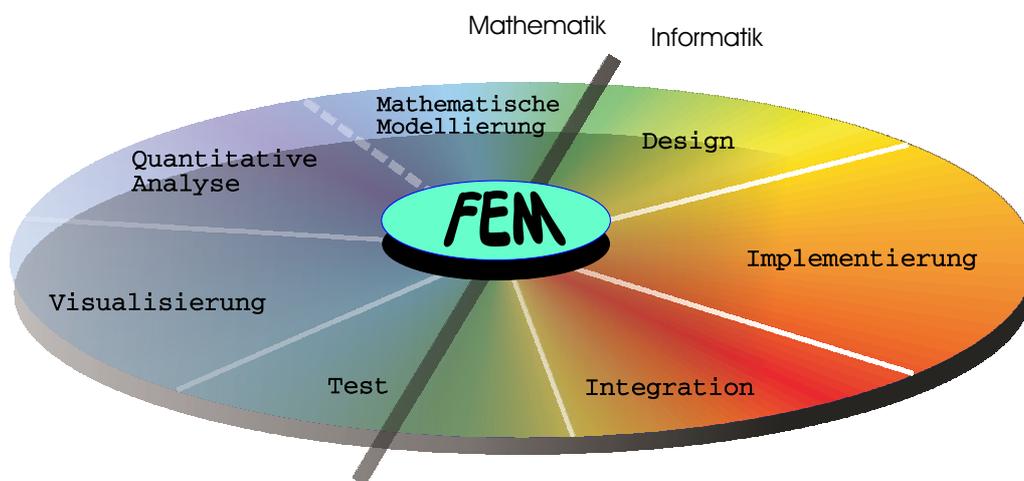


Abbildung 0.2: Der typische Ablauf der numerischen Simulation und Programmevolution.

Diese Arbeit liefert hierzu den theoretischen Unterbau, der zur Integration von [Blanke 04] und [Günther 04] noch zu leisten war. Sie zeigt, dass es möglich ist, die Konzepte aus [Günther 04] dahingehend zu erweitern, dass eine Fluidsimulation durchführbar ist, die sich mit allen aktuellen Ansätzen messen kann. Insbesondere bedeutet das nachzuweisen, dass durch die hochspezialisierten Ansätze aus [Günther 04] nicht die Möglichkeit verbaut wurde, bestimmte Techniken wie versetzte Gitter zu implementieren. An einigen Stellen ist es darüber hinaus sogar gelungen, die aus [Günther 04] übernommene Systematik im Detail zu verbessern.

Nun ist jede einzelne Behauptung natürlich auch an Problemen aus der Praxis zu validieren. Das jedoch kann und soll diese Arbeit allein aufgrund des beschränkten Umfangs nicht mehr leisten. Jedoch befindet der Autor sich in der glücklichen Lage, gleich auf eine aufbauende Arbeit verweisen zu können: In [Neckel 05] werden bereits einfache Fluid–Struktur–Wechselwirkungen mit dem neuen Ansatz behandelt. Damit kann der Praxisteil entsprechend knapp ausfallen.

Fasst man nun die Entwicklung und die Evolution numerischer Software als iterativen Prozess auf, so lässt sich diese Arbeit eindeutig einordnen: Thematisiert werden die Bereiche *mathematische Modellierung*, *Design*, *Implementierung* und *Integration*. Die dem folgenden Teile sind großteils in [Neckel 05] zu finden. Die Abbildung 0.2 visualisiert diese Tatsache. Naturgemäß liegt der Schwerpunkt der hier vorliegenden schriftlichen Arbeit auf ersten beiden oben genannten Gebieten.

Inhalt

Voraussetzungen

Obwohl diese Arbeit der Informatik Fakultät entstammt, ist ein gewisses Grundwissen um die Finite-Elemente-Methode zum Verständnis dieser Arbeit unabdingbar. Insbesondere findet sich hier keine Einführung mehr in diese Thematik, stattdessen sei auf [Braess 02, Knabner 00] verwiesen. Desweiteren seien Grundlagenkenntnisse der Mehrgittermethodik [Griebel 92, Griebel 94, Trottenberg 00] und der numerischen linearen Algebra [Stoer 90] vorausgesetzt.

Die physikalischen Grundlagen werden analog allenfalls angerissen und dürfen als bekannt vorausgesetzt werden. Bei Interesse seien hier [Gresho 98, Griebel 95] und natürlich im Umfeld des hier behandelten Themenkreises [Neckel 05] genannt. Der Informatikanteil ist mit Grundkenntnissen und der Lektüre von [Günther 04] oder [Pögl 04] gut zu bewältigen².

Inhaltliche Konzeption

Die Arbeit durchzieht der rote Faden, von einer Status-Quo-Analyse ausgehend mathematisches Modell über die mathematisch handwerklichen Aspekte bis zu Implementierungsdetails darstellen zu wollen.

Einleitend werden dazu in Kapitel 1 die behandelten Gleichungen präsentiert. Dort findet sich auch eine ausführlichere Darstellung der Themenschwerpunkte. In Kapitel 2 findet sich dann ein Umriss der bekannten Ansätze, Strömungsmechanik mit der FEM zu betreiben. Diese werden dargestellt und auch evaluiert.

Ab Kapitel 3 wird dann der neu entwickelte Lösungsalgorithmus präsentiert. Grundlegend hierfür ist die Existenz der in [Blanke 04] entwickelten Elemente, weshalb es vernünftig erschien, diese innerhalb dieses Themenblocks einzuführen. Nachdem dort alle mathematischen Grundlagen gelegt worden sind, kann in Kapitel 4 die mathematische Handwerksarbeit geleistet werden. Konkret leite man dort die beteiligten Matrizen der FEM her. Kapitel 5 umfasst dann einen kurzen Ausflug in die numerische lineare Algebra — es werden Standardgleichungslöser rekapituliert — bevor in Kapitel 6 endlich der neue Algorithmus und der Ansatz von [Günther 04] integriert werden. Den inhaltlichen Abschluss bildet Kapitel 7 mit der Konstruktion eines Mehrgitterverfahrens für das neu entwickelte Programm.

Aufbau und Anhang

Den Einzelkapiteln steht jeweils eine kurze Einleitung und Inhaltsübersicht voran, so dass das Querlesen erleichtert wird. Am Ende eines jeden Themenblockes findet sich desweiteren eine kurze Stichwortsammlung der angesprochenen Themen und Aspekte.

Die Gesamtarbeit wird immer wieder unterbrochen von sogenannten Exkursen, die neben tatsächlichen thematischen Ausflügen technische Details beinhalten. Als Beispiel ist hier die Definition der mathematischen Bezeichner auf Seite 3 zu nennen. Diese Exkurse tauchen im Inhalt nicht auf — schließlich heißen sie ja Exkurse und haben damit mit dem eigentlichen Inhalt nichts zu tun — eine Übersicht gibt jedoch unter anderem der Index.

Es wurde versucht allzu technische Details in den Anhang auszulagern. So finden sich beispielsweise alle größeren Matrizen dort. Man kann über den Sinn der Angabe derselben streiten. Tatsache ist, dass die Angabe von Matrizen in [Gresho 98] bei der Erstellung dieser Arbeit sehr hilfreich war. Vielleicht kann der ein oder andere Leser von der hießigen Darstellung profitieren. Ebenfalls

²Mehrfach angemahnt wurde der etwas altertümliche Schreibstil — hier jedoch leisten Duden und Fremdwörterbuch wertvolle Dienste.

in den Anhang verbannt wurden große „Bildexperimente“, da solche Bildtafeln den Lesefluss sonst unnötig gestört hätten.

Danksagung

Bleibt abschließend den mittelbar und unmittelbar Beteiligten an dieser Arbeit Dank zu sagen. Mag das hier vorliegende Resultat nicht fehlerfrei und mit Sicherheit verbesserbar sein, so muss in der Retrospektive doch gesagt werden, dass die während der Erstellung vergangene Zeit in meinen Augen als produktiv und lehrreich anzusehen ist. In diesem Sinne möchte ich mich bei allen MitarbeiterInnen des Lehrstuhls Zenger für die angenehme und fruchtbare Arbeitsatmosphäre bedanken.

Besonders herauszustellen sind hiervon (in alphabetischer Reihenfolge der Nachnamen) Dr. Frank Günther, für das enthusiastische Heranführen an das Themengebiet und die freundschaftliche Unterstützung bei der Einarbeitung in die komplexe Materie, sowie Dr. Miriam Mehl für das unermüdliche Korrektur lesen, der Beachtung vieler von mir immer wieder sträflich vernachlässigten Nuancen der Materie und die immer wieder dringendst benötigte Nachhilfe in vielen mathematischen und fachlichen Bereichen. Desweiteren sei auch Tobias Neckel gedankt dafür, dass er unbeirrbar, auf meine Arbeit aufbauend diese überhaupt erst praxisrelevant gemacht hat und dabei auch an dem prototypischen Charakter vieler Teile während der Entwicklung nicht verzweifelt ist.

Abschließend möchte ich mich bei Prof. Dr. Dr. hc Christoph Zenger bedanken. Dabei geht es mir nicht darum, seine fachliche Unterstützung herauszuheben — seine Handschrift ist hier unverkennbar, seine Leistungen nicht anzuzweifeln. Auch seine offene und engagierte Art sowie sein bestechender Humor werden oft und gerne zitiert. Stattdessen liegt es mir besonders am Herzen, seine persönliche, herzliche Betreuung und sein persönliches Engagement, das zum Gelingen dieser Arbeit beigetragen hat, dankend herauszuheben. Dies ist mit Sicherheit an Hochschulen nicht die Regel.

Inhaltsverzeichnis

1 Die Navier–Stokes Gleichungen	1
1.1 Modellbildung	1
1.2 Die Navier–Stokes Gleichungen	2
1.2.1 Interpretation der Terme	5
1.2.2 Mathematische Klassifizierung	7
1.2.3 Rand- und Anfangswerte	8
1.2.4 Interessante Aspekte	8
1.3 Typische Testszenerien	9
2 Herkömmliche Lösungsverfahren für die Navier–Stokes Gleichungen	11
2.1 Diskretisierungsvarianten	11
2.2 Behandlung der Zeitableitung	12
2.2.1 Die θ -Methoden	13
2.2.2 Analyse der θ -Verfahren	15
2.3 Finite–Element–Diskretisierung der räumlichen Ableitungen der Impulsgleichung .	15
2.4 Einbringen der Kontinuitätsgleichung	17
2.4.1 Divergenzfreie Geschwindigkeitsansatzräume	17
2.4.2 Gemischte Ansätze	18
2.4.3 Approximation der Nebenbedingung über die Druckpoissongleichung	25
3 Lösungsalgorithmus	29
3.1 Algorithmus	30
3.1.1 Die $(\theta = 0)$ -Methode	30
3.1.2 Zeitschrittweitensteuerung	30
3.2 Finite–Elemente Formulierung der Einzelschritte	31
3.2.1 Präparationsschritt	32
3.2.2 Druckpoissongleichung	33
3.2.3 Der Cauchy–Schritt	34
3.3 Physikalische Interpretation des Algorithmus	35
3.4 Diskrete Funktionenräume	36
3.4.1 Kontinuitätserhaltende Ansatzfunktionen für die Geschwindigkeit	36
3.4.2 Konstruktionsprinzipien für den Druckraum	42
4 Diskrete Operatoren	47
4.1 Präparationsschritt	49
4.1.1 Diffusionsterm	50
4.1.2 Konvektionsterm	50
4.1.3 Massenmatrix	51
4.2 Druckpoissongleichung	51
4.2.1 Poissongleichung auf den [Blanke 04]–Elementen	54
4.2.2 Poissongleichung auf den augmentierten Elementen	57

5	Grundlagen zu Relaxationsverfahren	63
5.1	Iterative Verfahren	63
5.1.1	Ausprägungen der Iterationsvorschrift	64
5.1.2	Residuenbasierte Formalisierung	65
5.1.3	Relaxationsfaktoren	66
5.2	Operatorschreibweise	66
5.3	Implizite Gleichungssysteme	67
5.4	Abbruchkriterien und Effizienzmessung	68
5.4.1	Abbruchbedingung	68
5.4.2	Konvergenzrate	69
6	Gebietsdiskretisierung, Datenlokalisierung und algorithmische Realisierung	71
6.1	Gittercharakteristika	71
6.2	Spacetime-Diskretisierungen	72
6.3	Funktionsräume auf Spacetimes	76
6.3.1	Hängende Punkte	77
6.3.2	Zellbasierte Freiheitsgrade	77
6.3.3	Resolutionserhöhung der Funktionsdarstellung	79
6.3.4	Algorithmenimplementierung	82
6.4	Spacetime-Traversierung mittels raumfüllender Kurven	83
6.4.1	Die Peano-Kurve	84
6.4.2	Das Stackprinzip	85
6.4.3	Cache-optimale Algorithmen	86
7	Mehrgitteralgorithmen	89
7.1	Multiskalenbetrachtung	90
7.2	Bewertung der Mehrgittereigenschaften für instationäre Fluidsimulation	91
7.3	Konstruktion eines additiven Mehrgitterverfahrens für die Druckpoissongleichung	92
7.3.1	Formalisierung einer Mehrgittermethodik auf Spacetimes	93
7.3.2	Restriktionsoperator auf dem Residuum	93
7.3.3	Einbringen des Lokalitätsprinzips	94
7.3.4	Verschänken der Mehrgitterzyklen	96
7.4	Numerische Experimente	97
8	Fazit	103
8.1	Rekapitulation der Ergebnisse	103
8.2	Einordnung der Erkenntnisse	104
8.3	Ausblick	104
A	Assemblierungsmatrizen	107
A.1	Darstellung des nichtlinearen Terms	107
A.2	Bilineare Ansatzfunktionen	108
A.3	Kontinuitätserhaltende Ansatzfunktionen	110
B	Matrizen für Multilevelverfahren	113

C Maple Skripte	117
C.1 Die Elementmatrizen	117
C.1.1 Definition der Ansatzfunktionen:	117
C.1.2 Berechnung der Assemblierungsmatrizen:	118
C.2 Die Multileveloperatoren	119
D A Cache-aware Multigrid Navier-Stokes Solver	121
D.1 Abstract	121
D.2 Introduction	121
D.3 Space-Trees, Cell-oriented Operator Evaluation and a Cache-aware Hierarchical Domain Traversal	122
D.4 Shape Functions Preserving Continuity	124
D.5 The Pressure Equation	124
D.5.1 The Fine Grid Relaxation Scheme	126
D.5.2 The Coarse Grid Correction	126
D.6 Results and Conclusion	127
E Fallstudie Driven-Cavity	131

Inhaltsverzeichnis

1 Die Navier–Stokes Gleichungen

Diese Arbeit beschäftigt sich mit der numerischen Simulation von *Fluiden*, also beispielsweise Gas oder einer beliebigen Flüssigkeit. Dazu wird neben einer gewissen Grundbildung in der Mathematik und Informatik ein Fundament an Wissen um Forschungsinhalte, Bedeutung und Bezeichner der Anwendungsdomäne benötigt. Dieses soll in dem hier vorliegenden einführenden Abschnitt elaboriert werden. Dabei handelt es sich nicht um eine umfassende physikalische Herleitung oder mathematische Einführung — das Kapitel bezieht seine Daseinsberechtigung eher aus der Notwendigkeit einer gewissen Abgeschlossenheit der Gesamtarbeit.

Nachdem in einem ersten Teil das physikalische Modell und die Differentialgleichung skizziert wird, bietet der Folgeteil eine kurze mathematische Beleuchtung der Aufgabenstellung. Hierzu gehört die Diskussion zu spezifizierender Rand- und Anfangsbedingungen, sowie eine Klassifikation der Gleichungen. Infolge dieser Darstellung werden einige interessante Aspekte herausgestellt und es wird aufgezeigt, welchen hiervon in dieser Arbeit besonderes Augenmerk galt. Dieser mathematische Abschnitt wird dann abgerundet durch eine kurze Beschreibung typischer Testszenerien für die numerische Strömungssimulation, den *Computational Fluid Dynamics* (CFD).

Der physikalische Teil wurde schwerpunktmäßig aus [Griebel 95] extrahiert, die mathematische Klassifizierung stark von [Braess 02] beeinflusst. Bei der Interpretation der Gleichungsterme und der Auflistung der mathematisch interessanten Aspekte der Gleichungen handelt es sich hingegen um eine rein subjektive. Ist man an einer fundierten Herleitung der Gleichungen interessiert, so sei auf [Emans 03, Griebel 94, Neckel 05] verwiesen. Ergo kann der vorbelastete Leser dieses einführende Kapitel gut und gerne überspringen, sei jedoch zuvor noch explizit auf die Seiten 3 und 4 hingewiesen, auf denen die in dieser Arbeit verwendete Notation standardisiert wird.

1.1 Modellbildung

Der Simulation von Fluiden liegt, wie bei allen partiellen Differentialgleichungen, immer ein konkretes physikalisches Modell zugrunde. Dabei wird — wie bei Modellen generell — durch Abstraktion und Detailreduktion die Anzahl der Eingangsparameter auf ein übersichtliches Maß reduziert. Dem hier verwendeten Modell zur Beschreibung von Fluiden — den Navier–Stokes Gleichungen — liegen vier solche Einflussparameter zugrunde:

- **Flächenkräfte** Hierzu gehören beispielsweise Gegenstände, die durch den Raum „wandern“, aber auch feste Wände. Legt man hierauf den Arbeitsschwerpunkt, so tangiert man das Fachgebiet der Fluid–Struktur–Wechselwirkungen [Emans 03, Neckel 05].
- **Volumenkräfte** Bekanntester Einflussfaktor in der Simulation ist die Gravitation. Die Motivation zur Nomenklatur ist offensichtlich: Innere Kräfte wirken bei einer teilchenorientierten Sichtweise per Definition auf jedem Fluidpartikel. In dieser Arbeit werden innere Kräfte vernachlässigt.
- **Reibungskräfte** Werden im CFD Umfeld auch mit dem Begriff *Viskosität* umschrieben.
- **Trägheit** Die Trägheit ist ein bekanntes und wichtiges Phänomen in der Physik. Im CFD–Umfeld tritt sie über Transportterme in Erscheinung.

1 Die Navier–Stokes Gleichungen

Die Zähigkeit einer Flüssigkeit legt eine Gewichtung zwischen Reibungskräften und Trägheitsmomenten fest. Standardbeispiel ist der Vergleich von Honig und Luft, die ein Rohr umströmen: Während der Honig sich langsam um das Rohr windet, umstreift Luft selbiges nur und es bilden sich die aus dem Straßenverkehr bei LKWs bekannten Wirbel hinter dem Objekt. Grund hierfür ist der geringe Einfluss der Reibung. Bildlich gesprochen transportiert Reibung die Haftungskräfte, die das Fluid an der Wand halten, ins innere desselben. Man spricht in diesem Zusammenhang nach [Griebel 94] von innerer Reibung und bezeichnet Fluide mit hoher innerer Reibung als *viskose* Strömungen.

Die charakteristische Zähigkeit des Modells wird in den diese Arbeit begleitenden Gleichungen über die *Reynoldszahl* Re ausgedrückt. Ist diese klein, so wird die Reibung relativ stark gewichtet und man gelangt wieder zu einem „Honigmodell“. Die Reynoldszahl ist damit ein modellabhängiger Eingangsparameter in die Fluide beschreibenden Navier–Stokes Gleichungen. Wird für manche Untersuchungsgegenstände die Reibung irrelevant, so führt die Neglektion des viskosen Terms zu den sogenannten Euler–Gleichungen. Vernachlässigt man dagegen die Trägheit, weil die Reibung das dominierende Element ist, so erhält man die Stokes–Gleichungen.

Das Auftreten von Turbulenzen hängt signifikant neben der Viskosität auch von der Strömungsgeschwindigkeit ab. Ist diese einem relativ gemächlich hinfließendem Fluss ähnlich, so spricht man von einer *laminaren* Strömung. Das oben beschriebene LKW Beispiel wird als *turbulentes* Fluidverhalten bezeichnet.

Die letzte wichtige zu nennende Vereinfachung des grundlegenden physikalischen Modells ist die Annahme der Inkompressibilität des Fluids. Die Dichteinvarianz des Fluids zu postulieren, ist in der numerischen Simulation gang und gäbe. Damit kann in ein beliebiges Kontrollvolumen nur soviel Flüssigkeit eintreten, wie aus diesem herausfließt. In der mathematischen Behandlung wird dieses Verhalten durch den Fluiddruck sichergestellt. Dieser stellt sich zu jedem Zeitpunkt so ein, dass bei übermäßigem Fluideintritt diesem entgegengewirkt wird. Alternativ „saugt“ der Druck bei Bedarf auch Flüssigkeit¹ an.

1.2 Die Navier–Stokes Gleichungen

Formalisiert wird das Verhalten einer instationären, inkompressiblen, viskosen, laminaren Strömung nun über die Navier–Stokes Gleichungen²:

$$\begin{aligned} \partial_t u + (u \nabla) u - \frac{1}{Re} \Delta u + \nabla p &= 0 & u : \mathbb{R}^d &\mapsto \mathbb{R}^d \\ \operatorname{div} u &= 0 & p : \mathbb{R}^d &\mapsto \mathbb{R} \end{aligned}$$

Ersterer Term wird dabei als *Impulsgleichung* bezeichnet und beschreibt das Verhalten des Fluids in der Zeit. Insbesondere geht hier das bereits thematisierte Verhältnis von Reibungs– zur Transporteinfluss ein. Den zweiten Term nennt man *Kontinuitätsgleichung*. Dieser stellt die Inkompressibilität des Fluids sicher.

Man unterscheidet zwei fundamentale Ansätze diese Gleichungen aus den Naturgesetzen herzuleiten: Die räumlich motivierte Betrachtungsweise nach *Euler* und die materiell Betrachtungsweise nach *Lagrange*. Beide sind unter anderem in [Emans 03, Griebel 95, Neckel 05] zu finden. [Blanke 04] wählt einen deduktiven Ansatz, in dem die zugrundeliegenden Erhaltungssätze der Physik aus den Navier–Stokes Gleichungen abgeleitet werden und damit die Konsistenz derselben mit den physikalischen Grundgesetzen verifiziert wird.

¹Im Folgenden wird der Begriff der Flüssigkeit dem des Fluids gleichwertig benutzt.

²In dieser Arbeit wird durchgehend die sogenannte dimensionslose Form der Navier–Stokes Gleichungen verwendet, in die die Gesamtszenarioskalierung nicht eingeht. Für Probleme mit einer Validierungsmöglichkeit gegen Experimente sind alle Eingangsgrößen passend zu skalieren [Neckel 05].

Exkurs: Mathematische Notation

Die Semantik einer mathematischer Syntax ist leider nicht standardisiert. Ist eine konsistente Verwendung innerhalb einer Arbeit unumgänglich, so empfiehlt es sich zusätzlich, die verwendete Semantik explizit zu definieren. Dies soll dieser Exkurs leisten. Dabei konkurrieren in der angewandten Mathematik die innerhalb des so genannten Ricci-Kalküls verwendete Notationskonvention mittels Indices (Einsteinsche Summenkonvention)[Bronstein 01] und eine operatororientierte Schreibweise. In dieser Arbeit erhielt Zweitere den Vorzug, da ansonsten mit den in der Numerik unverzichtbaren Indices die Formeln an Übersicht einbüßen würden.

Als Demonstrationsobjekte gelten im Folgenden die Abbildungen

$$\begin{aligned} a &: \mathbb{R}^d \mapsto \mathbb{R}, \\ b &: \mathbb{R} \mapsto \mathbb{R}^d \text{ und} \\ c &: \mathbb{R}^d \mapsto \mathbb{R}^d, \end{aligned}$$

wobei der Ableitungsoperator $\frac{\partial}{\partial x}$ meist abkürzend als ∂_x notiert wird. Es gelte also

$$\nabla a := \begin{pmatrix} \partial_{x_1} a \\ \partial_{x_2} a \\ \dots \\ \partial_{x_d} a \end{pmatrix} = \begin{pmatrix} \partial_{x_1} \\ \partial_{x_2} \\ \dots \\ \partial_{x_d} \end{pmatrix} a, \quad \partial_x b := \begin{pmatrix} \partial_x b_1 \\ \partial_x b_2 \\ \dots \\ \partial_x b_d \end{pmatrix},$$

$$\mathbf{div} \, c := \left(\sum_i \partial_{x_i} c_i \right) \quad \text{und} \quad \Delta a := \left(\sum_i \frac{\partial^2}{\partial x_i^2} a \right) = \mathbf{div} \, \nabla a,$$

$$\Delta c := \begin{pmatrix} \Delta c_1 \\ \Delta c_2 \\ \dots \\ \Delta c_d \end{pmatrix} \quad \text{sowie} \quad (c \nabla) c := \begin{pmatrix} c \nabla c_1 \\ c \nabla c_2 \\ \dots \\ c \nabla c_d \end{pmatrix}.$$

Dabei sei ∇ der Gradient, \mathbf{div} die Divergenz und Δ der Laplace–Operator. Das Produkt zweier vektorwertiger Funktionen sei durch das euklidische Skalarprodukt definiert. Man beachte besonders die Schreibweise für den Gradientenoperator als Produkt eines (Operator-)Vektors mit einer Funktion. Aufgrund des Zusammenhangs

$$\mathbf{div} = \nabla^T = (\partial_{x_1} \ \partial_{x_2} \ \dots)$$

betont folglich die Notation

$$\Delta = \nabla^T \nabla$$

die Adjungiertheit der Operatoren Divergenz und Gradient. Der Laplace-Operator ergibt sich in dieser Notation als inneres Skalarprodukt $\Delta = \nabla^T \nabla$.

Exkurs: Notation

Der Fluch der Indices

Die Darstellung numerischer Sachverhalte leidet bisweilen unter der Vielzahl der verwendeten Indices. Daher soll in diesem Exkurs die Semantik der verschiedenartigen Indices definiert werden. Dabei wurde versucht, eine möglichst intuitive Notation zu verwenden. So werden Dimensionsindizes respektive die einzelnen Komponenten generell tiefgestellt notiert. Es gilt also

$$x \in \mathbb{R}^2 \Rightarrow x = (x_1 \quad x_2)^T.$$

Ein zweiter oft verwendeter Indextyp identifiziert die Elemente einer Folge und kommt gerade bei Iterationsformulierungen ständig zum Einsatz. Folgenindizes werden hochgestellt notiert, im Gegensatz zu Exponenten jedoch von Klammern umschlossen. Somit bezeichnet $(x^{(n)})_{n \in \mathbb{N}_0}$ eine Folge und

$$x^{(0)} \quad \text{sowie} \quad x^{(*)} := \lim_{n \rightarrow \infty} x^{(n)}$$

bezeichnen bei Iterationsformalisierungen die Startlösung respektive den Limes. Der Stern ist der BNF-Notation [Broy 98, Marcotty 86] entliehen.

Kernelement vieler numerischer Methoden ist die Überführung eines unendlich dimensional kontinuierlichen Problems auf eine endlich dimensionale diskrete Darstellung. Im Falle der Galerkin–Projektion wird die Funktion $u : \mathbb{R}^a \mapsto \mathbb{R}^b$ durch eine Linearkombination von k Funktionen $\varphi(i) : \mathbb{R}^a \mapsto \mathbb{R}^b \quad i \in \{1 \dots k\}$ ersetzt:

$$u \mapsto \sum_{i=1 \dots k} u(i) \varphi(i)$$

Hier sind also die Ansatzfunktionen $\varphi(i)$ Stützstellen im \mathbb{R}^a zugeordnet. Die Lösung ist also ein k -Tupel $(u(1) \dots u(k))^T \in \mathbb{R}^k$. Somit werden mit dieser Notation unendliche Folgen von endlichen Tupeln abgegrenzt — erstere werden hochgestellt, zweitere als Argument notiert.

Symbole

In dieser Arbeit bezeichnen kalligraphische Buchstaben wie \mathcal{A} und \mathcal{T} Operatoren, Matrizen als endlicher Sonderfall werden durch normale Großbuchstaben beschrieben. Funktionen- und Vektorräume werden als Großbuchstaben mit Doppelstrich ($\mathbb{R}, \mathbb{K}, \mathbb{U}, \mathbb{P}$) notiert. Es wurde versucht, die Raum- und Elementbezeichner stets passend zueinander zu wählen, also z.B.

$$p \in \mathbb{P} \quad \text{und} \quad u \in \mathbb{U}.$$

Zur Bezeichnung von Ansatzfunktionen kommen auch die griechischen Symbole ϕ und φ zum Einsatz, während andere griechische Buchstaben wie τ, ω und θ problemabhängige Parameter bezeichnen, die geeignet zu wählen sind.

Erzeugendensysteme

Grundlage dieser Arbeit ist nicht selten eine Multiskalensicht des betrachteten Problems — man kennt diese Technik aus jeder Fourier–Analyse. Daher werden Werte oftmals sowohl hierarchisch als auch dehierarchisiert benötigt. Ist der dehierarchisierte Wert mit p bezeichnet, so notiert man den zur hierarchisierten Darstellung gehörenden Tupel mit \hat{p} .

1.2.1 Interpretation der Terme

Es macht daher nur begrenzt Sinn, einen oder gar beide Ansätze hier erneut zu präsentieren. Stattdessen soll kurz der Einfluss der Einzeltermine qualitativ untersucht werden. Dadurch ist eine Zuordnung auf die Einflussfaktoren des zugrundeliegenden physikalischen Modells möglich.

Der Laplace–Operator

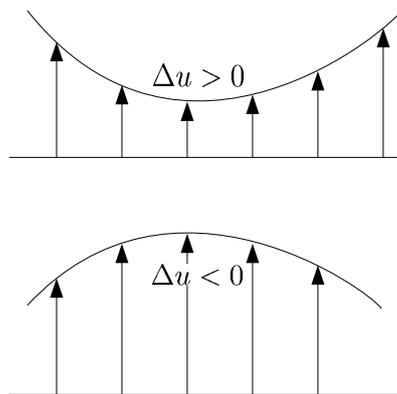


Abbildung 1.1: Auswirkung des Laplace–Operators. Man beachte die Beschleunigungs– (oben) respektive die Bremswirkung (unten).

Der Term über dem Laplace–Operator modelliert die Reibungskräfte: Betrachtet man einige Geschwindigkeitsverläufe, wie sie in Abbildung 1.1 dargestellt sind, so wird dies offensichtlich: Wird in der Umgebung eines Punktes „langsamer geströmt“ ($\Delta u < 0$), so wird auch der betrachtete Punkt, das betrachtete Partikel, gebremst. Die Umkehrargumentation erfolgt analog über das Mitreißen eines Partikels.

Der Laplace–Operator wirkt nach Seite 3 komponentenweise, es gilt also

$$\Delta u = \begin{pmatrix} \Delta u_1 \\ \Delta u_2 \end{pmatrix}.$$

Beschränkt man sich auf die Stokes–Gleichungen — vernachlässigt man also den nichtlinearen Term — so ergibt sich aufgrund der Definition des vektoriiellen Laplace–Operators eine Entkopplung der Komponenten des zu lösenden Gleichungssystems. Es ist einleuchtend, dass für eine numerische Diskretisierung diese Entkopplung nachzuvollziehen ist. Insbesondere darf die u_1 –Komponente die Zeitentwicklung der u_2 –Komponente nicht beeinflussen. Den Laplace–Operator–Term bezeichnet man in der Literatur als *Diffusionsterm*.

Der nichtlineare Term

Zur Interpretation des nichtlinearen Terms löse man sich von der gegebenen Formulierung, betrachte beide Eingangsgrößen als unterschiedliche und gelangt so zum allgemeinen Advektionsterm

$$(u \nabla) \phi = \begin{pmatrix} u \nabla \phi_1 \\ u \nabla \phi_2 \end{pmatrix} = (u, \nabla \phi_1) + (u, \nabla \phi_2).$$

1 Die Navier–Stokes Gleichungen

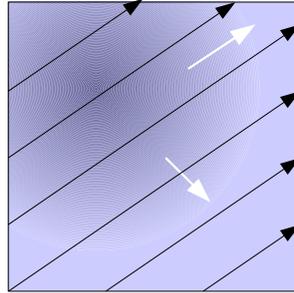


Abbildung 1.2: Transport einer Kenngröße (dunkles Gebiet) durch ein Transportmedium (schwarze Pfeile). Nur die Strömung senkrecht zur Kenngrößennormalen (weiße Pfeile) resultiert in einer Verschiebung derselben, eine Tangentialströmung hat keine Auswirkungen.

Diesen kann man schön anhand einer Stoffkonzentration motivieren: Strömt das transportierende Medium entlang der Konzentrationskante, so ändert sich die Konzentration nicht, strömt es gegen die Kante, so wird die Kante in Flussrichtung verschoben.

Nun kann man die Konzentration im Gedankenexperiment durch die Beschleunigung der Fluidpartikel ersetzen: Strömt das Fluid senkrecht zu einer Beschleunigungsfront, so wird diese nicht verschoben. Ein senkrecht aufeinandertreffen verstärkt jedoch die Flussgeschwindigkeit oder bremst diese, je nach Flussrichtung, ab. Den nichtlinearen Term bezeichnet man in der Literatur mit *Konvektionsterm*.

Der Druckgradient

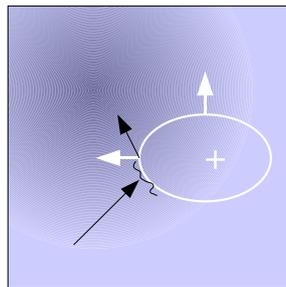


Abbildung 1.3: Der Druckgradient presst das Fluid aus einem Gebiet. Dadurch wird die Inkompressibilität sichergestellt.

Der Druckgradient ist mit Sicherheit einer der einfachsten zu interpretierenden Terme: Erhöht man den Druck in einem Punkt, so wird die Geschwindigkeit von diesem Punkt weggedrängt. Abbildung 1.3 liefert eine schematische Darstellung dieses Sachverhaltes.

Tatsächlich muss man sich nun fragen, wie der Druck im Kontext der Navier–Stokes Gleichungen zu interpretieren ist. Offensichtlich beachten weder Konvektions- noch Diffusionsterm die Bedin-

gung der Inkompressibilität. Als Folge stellt sich der Druck zu jedem Zeitpunkt sofort so ein, dass die Inkompressibilität in der Zeit gewahrt bleibt. In einer rein mathematischen Interpretation fungiert der Druck also als *Lagrange-Parameter*.

Der Divergenzoperator

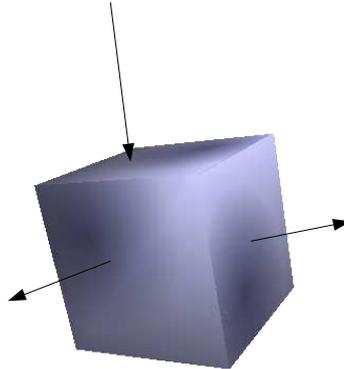


Abbildung 1.4: Zeitfortschaltung durch den Operator \mathcal{T}_τ .

Der Divergenzoperator formalisiert — wie bereits erwähnt — in Gestalt der Kontinuitätsgleichung die Inkompressibilität. Herleiten kann man sich diesen Term einfach über eine Hyperwürfelkontrollvolumenformulierung wie in Abbildung 1.4. Dabei identifiziert man den Einfluss über die Würfelhyperflächen mit dem Ausfluss und führt dann einen Grenzwertübergang auf den Punkt im Raum durch, indem man die Würfelkantenlänge gegen 0 gehen lässt. Dieser Grenzwertübergang resultiert in dem Divergenzoperator.

1.2.2 Mathematische Klassifizierung

Definition 1 Eine Differentialgleichung über u mit dem linearen Hauptoperator

$$-\sum_{i,k} a_{i,k} u \quad a : \Omega \mapsto \mathbb{R}$$

heißt

- **elliptisch** in einem Punkt x , falls die Matrix $(a_{i,k})$ dort positiv definit ist.
- **hyperbolisch** in einem Punkt x , falls die Matrix $(a_{i,k})$ dort einen negativen und ansonsten ausschließlich positive Eigenwerte hat.
- **parabolisch** in einem Punkt x , falls die Matrix $(a_{i,k})$ dort positiv semidefinit, aber nicht positiv definit ist.

Ist eine Differentialgleichung auf allen Punkten $x \in \Omega$ einem Typus zuzuordnen, so nennt man die Differentialgleichung selber elliptisch, hyperbolisch oder parabolisch.

1 Die Navier–Stokes Gleichungen

Nach der Hauptoperatordefinition für den Typus einer Differentialgleichung muss man die Navier–Stokes Gleichungen in die stationäre und die transiente Phase zerteilen: In der transienten Phase ist die Zeitableitung ungleich 0 und das Problem damit parabolisch. Ist die stationäre Phase erreicht, so bilden die Navier–Stokes Gleichungen eine Hybridklasse: Ist die Gleichung nach wie vor elliptisch, so dominiert je nach Gewichtung des Diffusionsteils über die Reynoldszahl der elliptische oder der nichtlineare Operator.

1.2.3 Rand– und Anfangswerte

Im Fall der instationären Navier–Stokes Gleichungen sind nach der Theorie Anfangs– und Randwerte zu spezifizieren. Dabei sind zwei Besonderheiten herauszustellen:

So müssen sowohl Anfangs– als auch Randbedingungen der Inkompressibilität genügen. Insbesondere muss also die Anfangsbedingung divergenzfrei sein. Genügen die Bedingungen der Kontinuitätsgleichung nicht, so handelt es sich um ein unphysikalisch gestelltes Problem. Besonders schwierig wird das Finden konkreter Bedingungen für nicht abgeschlossene Systeme. Hier sei im Kontext der hier entwickelten Algorithmen auf [Neckel 05] verwiesen.

Desweiteren klassifiziert man einige Randbedingungen für die Geschwindigkeit nach

- *no-slip*. Diese Bedingung entspricht der sogenannten Haftbedingung. Hierbei wird das Fluid in unmittelbarer Nähe zu einem Gebietsrand als ruhend angenommen, es haftet praktisch an ihm. Der Rand ist dabei undurchdringlich und man kann die Bedingung als homogenen Dirichlet–Rand formalisieren. Diese Randbedingung wird in dieser Arbeit ausschließlich benutzt.
- *free-slip*. Hierbei wird angenommen, dass das Fluid den Rand nicht durchdringen, jedoch frei an ihm entlang gleiten kann. Insbesondere treten keine Reibungs–, „Verluste“ auf. Diese Bedingung wird auch als Rutschbedingung bezeichnet und über das Nullsetzen der Normalenableitung der Geschwindigkeit erzeugt. Diese Randbedingung ist für diese Arbeit irrelevant.
- *Ein– / Ausströmbedingung*. Hierbei ändern sich die Geschwindigkeitskomponenten senkrecht zur Wand nicht, was man durch Festschreiben des Normalenanteils der Geschwindigkeit erreicht. Diese Randbedingung ist für diese Arbeit irrelevant, man konsultiere bei Interesse [Neckel 05].

Eine genauere Untersuchung der benutzten Randbedingung sowie die Diskussion etwaiger Randbedingungen für den Druck soll auf später verschoben werden.

1.2.4 Interessante Aspekte

Mit Sicherheit existiert eine Vielzahl an interessanten mathematischen Aspekten im Zusammenhang mit den Navier–Stokes Gleichungen. Drei nicht repräsentative seien hier gesondert genannt:

- Die Behandlung parabolischer Probleme ist nicht–trivial. Insbesondere existiert keine so fundierte Theorie wie für elliptische Differentialgleichungen. In dieser Arbeit wird daher ab Seite 12 über einen Finite–Differenzen Ansatz das parabolische Problem auf eine Sequenz elliptischer Probleme zurückgeführt.
- Die Nebenbedingung erlaubt keine unmittelbare Anwendung der Standardtheorie für Finite–Elemente zu einem festen Zeitpunkt oder im stationären Fall. Die FEM für elliptische Probleme führt die Lösung immer auf ein Energieminimierungsproblem zurück. Aufgrund der Nebenbedingung ist das Energieminimum der Impulsgleichung jedoch keine Lösung — ansonsten würde das Fluid als unendlich stark kompressibel angenommen. Tatsächlich erweist sich die

Beachtung der Inkompressibilität als springender Punkt in jedem numerischen Lösungsalgorithmus.

- Die Navier–Stokes Gleichungen beinhalten einen nichtlinearen Term. Insbesondere führt also ein Minimierungsansatz aufgrund der Nichteindeutigkeit nicht zwangsläufig zu einem globalen Energieminimum und damit zur Lösung. Der Behandlung dieser Problematik wird in dieser Arbeit kein Raum eingeräumt, stattdessen sei auf die Literatur verwiesen.

1.3 Typische Testszenarien

Diese Arbeit hat zweifellos theoretischen Charakter. Dennoch müssen die entwickelten Ansätze an einem konkreten Beispiel validiert oder verworfen werden. Dabei entsteht neben der Grundkomplexität der Materie weitere Problematik durch die Notwendigkeit der Wahl geeigneter Randbedingungen. Dass die Wahl derselben keineswegs trivial ist, zeigt [Neckel 05] sowohl theoretisch als auch praktisch.

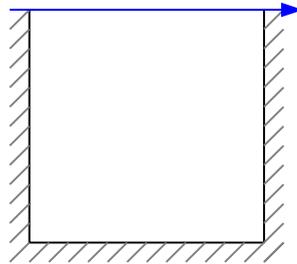


Abbildung 1.5: Schematische Darstellung der Driven–Cavity.

In selbiger Arbeit wird auch eine ganze Reihe von Szenarien nachgestellt, so dass sich der hier vorliegende Text auf ein einziges Szenario beschränkt, das relativ simpel gehalten ist: Es ist das Szenario der *Driven–Cavity*. Hierbei handelt es sich in diesem Projekt um ein quadratisches Gebiet, ohne Ein- und damit Ausströmung. Die obere Kante dieses Gebietes bewege sich nun mit konstanter Geschwindigkeit von links nach rechts.

Bei den resultierenden Randbedingungen handelt es sich nun ausschließlich um undurchlässige Ränder mit *no-slip* Bedingung. Dabei beginne man mit einer ruhenden Startlösung. Abbildung 1.5 gibt den Versuchsaufbau schematisch wieder, in Anhang E ist eine Vielzahl an Illustrationen zum Verhalten der Lösung in der Zeit zu verschiedenen Reynoldszahlen gegeben.

Neben der Driven–Cavity existieren noch einige Alternativszenarien, denen sich ein Strömungslöser stellen muss. Die wohl bekanntesten Szenarien sind die Stufenströmung, der freie Kanalfuss und die Kármánsche Wirbelstraße. Diese Szenarien werden in [Neckel 05] mit der hier entwickelten Methodik durchgespielt, in dieser Arbeit jedoch nicht weiter behandelt. Lediglich an einigen Stellen werden die in [Neckel 05] gewonnenen Ergebnisse als Beleg für hier getätigte Aussagen herangezogen.

Stichwortsammlung

Diffusionsterm, Reibung; Konvektionsterm, Trägheit; Druck, Lagrange–Parameter, Divergenzfreiheit, Inkompressibilität; parabolische, elliptische Differentialgleichung; Nebenbedin-

1 Die Navier-Stokes Gleichungen

gung, Nichtlinearität; Randbedingung, no-slip, physikalische Korrektheit; Driven-Cavity, Stufenströmung, Kármánsche Wirbelstraße

2 Herkömmliche Lösungsverfahren für die Navier–Stokes Gleichungen

Die Navier–Stokes Gleichungen für inkompressible Fluide in klassischer Form liefern noch keinen Hinweis auf ein probates Lösungsverfahren, da zu einem Zeitpunkt t mit gegebener Fluidgeschwindigkeit die Verteilung der Unbekannten „ungünstig“ ist:

$$\begin{aligned} \partial_t u + (u \nabla) u - \frac{1}{Re} \Delta u + \nabla p &= 0 && d + 1 \text{ Unbekannte } u \text{ und } \nabla p, \text{ zeitabhängig und} \\ \operatorname{div} u &= 0 && d \text{ Unbekannte, nicht zeitabhängig.} \end{aligned}$$

Gesucht ist eine geeignete Abbildung dieser beiden physikalischen Forderungen auf ein modifiziertes System partieller Differentialgleichungen, bei dem Zeit- und Raumbedingungen besser separiert sind, und aus dem sich somit mittels des Handwerkszeugs der Numerik Gleichungssysteme und approximative Lösungen berechnen lassen. Die dargestellten Methoden können als Axiome eines Kalküls zur Inferenz einer Vielzahl von Lösungsalgorithmen aufgefasst werden.

2.1 Diskretisierungsvarianten

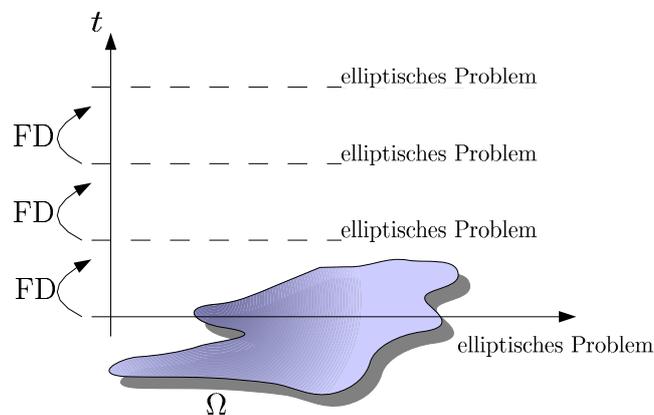


Abbildung 2.1: Die vertikale Methode als *fractional-step method*.

Bei der Behandlung parabolischer Probleme — zu denen die Navier–Stokes Impulsgleichungen gehören — haben sich zwei große Verfahrensklassen herauskristallisiert [Knabner 00]: Methoden, die auf einer *Volldiskretisierung* beruhen, stehen Methoden gegenüber, denen eine *Semidiskretisierung* zugrunde liegt. Im Falle der Volldiskretisierung werden in der FE–Technik Zeit–Raum–

Elemente verwendet, um das Problem in einer Variationsformulierung in der Raumzeit zu lösen. Auf diese Technik soll hier nicht weiter eingegangen werden¹.

Bei einer Semidiskretisierung werden die Diskretisierung der räumlichen und zeitlichen Ableitungen strikt getrennt behandelt. Generell unterscheidet man hierbei die vertikale von der horizontalen Linienmethode, wobei man der horizontalen Achse den Raum zuordnet und der vertikalen Achse die Zeit². Je nachdem, welche Unbekannte man zuerst diskretisiert, fällt das Verfahren in die eine oder andere Klasse. In dieser Arbeit wird eine vertikale Semidiskretisierung verwendet:

Die Raumdiskretisierung erfolgt dabei mittels Finiter Elemente, zuvor jedoch wird die Zeit mittels eines Finite–Differenzen–Verfahrens fortgeschaltet. Abbildung 2.1 visualisiert solch ein Verfahren: Das Problem wird durch die Zeitdiskretisierung auf eine elliptische Differentialgleichung pro Zeitschritt reduziert. Die resultierenden Gleichungen werden auch als semidiskrete Modelle bezeichnet.

Bezüglich der Diskretisierungsreihenfolge halte man fest: Zunächst wird ein „Zeitmodell“ festgelegt, anschließend ergibt sich die zu lösende (stationäre) Differentialgleichung für einen Zeitschritt. In diesem Kapitel wird demzufolge zunächst die Behandlung der Zeit thematisiert. Dabei sei kein Vollständigkeitsanspruch postuliert, stattdessen sollen lediglich die Techniken eingeführt werden, die später im Umfeld des hier vorgestellten neu entwickelten Algorithmus verwendet werden können. Weiterführende Techniken finden sich unter anderem in [Donea 03].

Der zweite Abschnitt beschäftigt sich mit der Lösung des elliptischen Problems pro Zeitschritt mittels der Finite–Element–Methode. Dabei wird die Impulsgleichung mittels einer Galerkin–Formulierung in eine schwache Form gebracht. Durch die Existenz der Nebenbedingung — der Kontinuitätsgleichung — sind hierbei theoretische Untersuchungen und Resultate für elliptische Probleme nicht unmittelbar übertragbar. Verschiedene Ansätze zur Lösung der Problematik werden kurz angerissen. Weiterführende Aussagen sind unter anderem [Braess 02, Donea 03, Ferziger 99, Fletcher 97, Girault 86, Gresho 98] zu entnehmen.

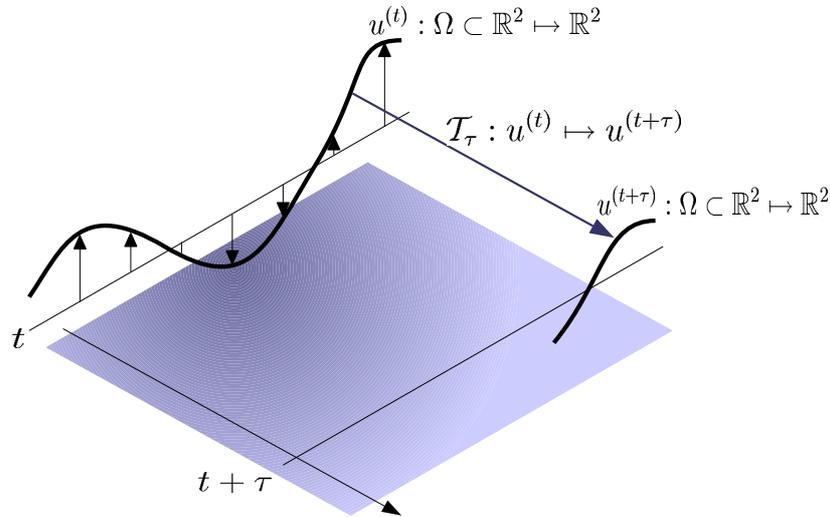
Alle vorhandenen Ansätze leiden unter einigen Einschränkungen, was in dieser Arbeit als Motivation zur Entwicklung eines neuen Verfahrens dient. Dieses Verfahren wird im Anschluss an dieses Kapitel aus den hier präsentierten Ansätzen entwickelt. Somit kann bei entsprechenden Vorkenntnissen dieses einführende Kapitel übersprungen werden — es stellt lediglich einen Streifzug durch vorhandenes Wissen dar.

2.2 Behandlung der Zeitableitung

Es existiert eine Vielzahl von Ansätzen, die Zeitfortschaltung zu realisieren — im Wesentlichen deckt sich das Wissen mit dem der Numerik gewöhnlicher Differentialgleichungen [Aulbach 04, Deuffhard 02]. Im Folgenden sei solch eine Realisierung der diskreten Zeitfortschaltung über einen endomorphen Zeitoperator identifiziert, wobei die Zeit in Intervalle der Länge τ eingeteilt sei. τ ist dabei generisch, also nicht unbedingt äquidistant, zu wählen. Dieser Endomorphismus liefere zu einer Geschwindigkeit zu einem Zeitpunkt t die Geschwindigkeit zum Zeitpunkt $t + \tau$.

¹In der Tat sind Volldiskretisierungen für praxisnahe Aufgaben nicht relevant — die aus den impliziten Gleichungssystemen resultierenden Gleichungen sind meist schlicht und einfach nicht zu lösen. Ein Beispiel sei [Hartmann 04]: Hier wird ein cache–optimales Verfahren analog zu [Günther 04, Pögl 04] für beliebige Dimensionen entwickelt. Trotz echter Mehrgitterkonvergenz zeigt sich, dass bereits vier Dimensionen nicht mehr vernünftig handhabbar sind. So benötigt man für die Approximation der Poisson–Gleichung auf einem 27^4 –Gitter (etwa 460000 Freiheitsgrade) bis zu einer Genauigkeit von 10^{-5} auf dem Referenzrechner etwa fünf Minuten. Bereits auf einem 81^4 Gitter hat man jedoch 41 Millionen Freiheitsgrade und auf dem gleichen Rechner eine Terminierungszeit von etwa sieben Stunden. Eine solche Feinheitgradzahl ist jedoch für zeitabhängige Probleme noch nicht ausreichend [Gresho 98, Neckel 05]. Abhilfe könnten hier eventuell dünne Gitter [Bungartz 98] verschaffen, dies ist jedoch noch Gegenstand aktueller Forschung.

²Die Variante mit zugrundeliegender äquidistanter Zeitdiskretisierung bezeichnet man auch als Rothe–Methode.

Abbildung 2.2: Zeitfortschaltung durch den Operator \mathcal{T}_τ .

$$\begin{aligned}\mathcal{T}_\tau : \mathbb{U} &\mapsto \mathbb{U} \\ \mathcal{T}_\tau(u^{(t)}) &= u^{(t+\tau)}\end{aligned}$$

Abbildung 2.2 illustriert die Wirkungsweise solch eines Operators. Die hier verwendete Formulierung ist sehr allgemein, in dieser Arbeit jedoch wird die Zeit ausschließlich linear fortgeschaltet. Es gilt folglich

$$\begin{aligned}\partial_t u &\approx \frac{u^{(t+\tau)} - u^{(t)}}{\tau} =: d_t u \\ \Rightarrow \mathcal{T}_\tau(u^{(t)}) &= u^{(t)} + \tau d_t u,\end{aligned}$$

wobei $\partial_t u$ respektive $d_t u$ über die Impulsgleichung gegeben ist³. Es verbleibt die Frage zu klären, zu welchem „Zeitpunkt“ $\partial_t u$ auszuwerten ist, auf welches u also die in der Impulsgleichung auftretenden Operatoren angewandt werden müssen. Die Varianten, bei denen nur Linearkombinationen der Operatorauswertungen zu den Zeitpunkten t und $t + \tau$ verwendet werden, bilden die Familie der θ -Methoden, die im folgenden Abschnitt beschrieben wird:

2.2.1 Die θ -Methoden

In die Familie der θ -Methoden fallen alle Methoden, die Operatoren auf der Lösungsfunktion über einer Linearkombination aus aktueller und zu bestimmender Funktion $u^{(t)}$ und $u^{(t+\tau)}$ auswerten.

³Im Folgenden sei $d_t u$ der Wert der Zeitableitung, also bereits der Zeitdiskretisierung entnommen.

2 Herkömmliche Lösungsverfahren für die Navier–Stokes Gleichungen

Definition 2 Sei $\theta \in [0, 1]$. Der Operator $\mathcal{T}_{\tau, \theta}$ falle in die Familie der θ -Methoden mit

$$\mathcal{T}_{\tau, \theta}(u^{(t)}) := u^{(t)} + \tau \underbrace{\left((1 - \theta)d_t u^{(t)} + \theta d_t u^{(t+\tau)} \right)}_{\cong \partial_t u}$$

Im Folgenden diene der Operator $\mathcal{F} : (\Omega \mapsto \mathbb{R}^2) \mapsto (\Omega \mapsto \mathbb{R}^2)$ als Sammelplatz für den Diffusions- und Konvektionsanteil, enthalte jedoch keine Druckanteile. Diese Definition sowie die Notation selbst ist analog zu [Chorin 68] gewählt. Damit gilt

$$\begin{aligned} d_t u^{(t)} &= -(u^{(t)} \nabla) u^{(t)} + \frac{1}{Re} \Delta u^{(t)} - \nabla p^{(t)} \\ &=: -\mathcal{F}(u^{(t)}) - \nabla p^{(t)} \end{aligned}$$

Ein klassisches Beispiel für ein Element der θ -Familie liefert das Forward–Euler–Verfahren wie es aus der Finite–Differenzen Methode bekannt ist [Braess 02] und die rechte Seite — hier also \mathcal{F} und ∇p — ausschließlich in $t + 0 = t$ auswertet:

Korollar 1 Das Forward–Euler–Verfahren ist Element der Familie der θ -Methoden mit $\theta = 0$.

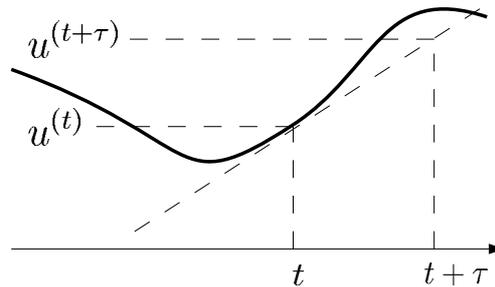


Abbildung 2.3: Die Forward-Euler-Methode.

Weitere bekannte Verfahren lassen sich direkt aus der Definition der θ -Methode ableiten. So gilt folgender Zusammenhang:

- Backward–Euler–Verfahren: $\theta = 1$
- Galerkin–Verfahren: $\theta = \frac{2}{3}$
- Crank–Nicolson–Schema: $\theta = \frac{1}{2}$

In der hier vorliegenden Arbeit wird wie in [Griebel 95] ausschließlich das explizite Euler–Verfahren zur Verwendung kommen, es gilt also $\theta = 0$. Somit erübrigt sich eine Annotation des Druckes mit einem Zeitindex, da in jedem Zeitschritt nur der aktuelle Druck von Interesse ist.

2.2.2 Analyse der θ -Verfahren

Mit Hilfe der θ -Methoden führt man ein System instationärer Gleichungen über in eine unendliche Sequenz stationärer Probleme. Diese Vereinfachung geht mit einem entscheidenden Nachteil einher⁴: Das Verfahren verfügt über einen numerischen Ereigniskegel endlicher Ausbreitungsgeschwindigkeit und genügt daher nicht automatisch der CFL-Bedingung [Braess 02]. Diese soll hier nicht breitgetreten werden, wird aber im Zusammenhang mit dem hier entwickelten Algorithmus ab Seite 29 nochmals in den Mittelpunkt des Interesses treten. Weitere Stabilitätsanalysen seien [Knabner 00] zu entnehmen, aus Mathematik-Ingenieur-Sicht bietet sich das Buch [Ganzha 96] an.

Trotz dieser Nachteile verwenden viele Verfahren eine explizite Zeitfortschaltung: Die Minimumsuche über den nichtlinearen Term der Navier-Stokes Gleichungen vereinfacht sich in diesem Fall zu einer einfachen Auswertung desselben.

Für den hier entwickelten Ansatz wird desweiteren nach der Vorstellung des neuartigen Verfahrens zum Lösen der Fluid-Gleichungen auffallen, dass das dort zu lösende System aus drei Differentialgleichungen durch $\theta = 0$ entkoppelt wird und infolge dessen sequentiell zu lösen ist. Die Herleitung der Differentialgleichungen ist jedoch prinzipiell θ -Wahl invariant.

2.3 Finite-Element-Diskretisierung der räumlichen Ableitungen der Impulsgleichung

Will man die Impulsgleichung mittels der Finite-Element-Methode lösen, so muss selbige durch Tests gegen geeignete Funktionen in die schwache Formulierung gebracht werden. Eine Einführung zu dieser als Galerkin-Formulierung bezeichneten Methodik findet sich unter anderem in [Braess 02, Knabner 00].

Um das Eingangs versprochene Kalkül zu bieten, wird die Zeitfortschaltung nun über den Operator $\mathcal{T}_{\tau,\theta}$ ausgedrückt. Damit erhält man nach vorgenommener Zeitdiskretisierung

$$\begin{aligned} u^{(t+\tau)} - u^{(t)} &= \mathcal{T}_{\tau,\theta}(u^{(t)}) - u^{(t+\tau)} \\ &= \tau \left((1-\theta)d_t u^{(t)} + \theta d_t u^{(t+\tau)} \right) \\ \Leftrightarrow \partial_t u &\approx \frac{1}{\tau} \left(u^{(t+\tau)} - u^{(t)} \right) = \left((1-\theta)(-\mathcal{F}(u^{(t)}) - \nabla p^{(t)}) + \theta(-\mathcal{F}(u^{(t+\tau)}) - \nabla p^{(t+\tau)}) \right) \end{aligned}$$

und damit die schwache Formulierung der Impulsgleichung

$$\left(\frac{1}{\tau} \left(\mathcal{T}_{\tau,\theta}(u^{(t)}) - u^{(t)} \right) + \mathcal{F}(u^{(t)}) + \nabla p^{(t)}, \varphi \right) = 0 \quad \forall \varphi \in V.$$

Die schwache Formulierung wird nun durch die Approximation von $u^{(t)}$ und $u^{(t+\tau)}$ in einem endlich dimensionalen Ansatzraum U sowie die Wahl eines ebenfalls endlich dimensionalen Testraums V in ein endliches System von Gleichungen überführt. Identifiziert man dabei Ansatz- und Testraum, so spricht man auch von einem Ritz-Galerkin-Verfahren.

Entgegen eines ersten Eindrucks ist die Standardtheorie der Finiten Elemente hier nicht anwendbar, da jede Lösung auch der Divergenzfreiheit, also der Inkompressibilität genügen muss, die sich im Rahmen der θ -Methoden kontinuierlich als

$$\mathbf{div} \mathcal{T}_{\tau,\theta}(u^{(t)}) = 0$$

⁴Tatsächlich existieren eine Reihe weiterer Nachteile: Die theoretische Fundierung der Finite-Differenzen-Methode ist bei weitem nicht so ausgereift, wie die Kenntnisse um die Finite-Element-Methode. Desweiteren erlaubt die explizite Zeitschrittsteuerung keine zeitadaptiven Gitter.

Exkurs: Funktionenräume

Generelle Notation

Mit C^k seien die k -mal stetig differenzierbaren Funktionen bezeichnet. Insbesondere bezeichnet der $C^0 = C$ den Raum der stetigen Funktionen, während C^∞ alle unendlich oft differenzierbaren Funktionen beinhaltet. Ist ein Funktionenraum mit einer tiefgestellten 0 annotiert, so ist der Raum eingeschränkt auf alle Elemente mit kompaktem Träger. Insbesondere muss der Träger vollständig in der Domäne liegen, auf der der Funktionenraum spezifiziert wird.

Ist dem gesamten Funktionenraumbezeichner ein Exponent nachgestellt, so handelt es sich um eine Dimensionsaussage bezüglich des Bildraums. So bezeichne $C(\Omega)^2$ den Raum der stetigen vektorwertigen Funktionen, d.h. $u \in C(\Omega)^2 \Rightarrow u(x) \in \mathbb{R}^2$.

Besondere Bezeichner

Einen besonderen Unterraum von C^∞ bilden je P^n und Q^n . Diese bezeichnen die Polynome n -ten Grades respektive die zugehörigen Tensorprodukträume. In dieser Arbeit sind die Mengen P^0 , also die konstanten Funktionen, sowie Q^1 als Raum der n -linearen Funktionen von Interesse.

Hilberträume

Mit

$$(u, v)_{L^2} = \int_{\Omega} (u(x), v(x)) \, dx,$$

$$L^2 = \{u \mid \|u\|_{L^2} = \sqrt{(u, u)_{L^2}} < \infty\}.$$

ist ein Funktionenskalarprodukt definiert, dass mit $\|\cdot\|_{L^2}^2 = (\cdot, \cdot)_{L^2}$ den Raum L^2 als Raum der Lebesgue-integrierbaren Funktionen definiert. Darauf aufbauend seien mit $m \geq 0$ über die schwachen Ableitungen ∂^α die Hilberträume H^m definiert als

$$(u, v)_{H^m} := \sum_{|\alpha| \leq m} (\partial^\alpha u, \partial^\alpha v)_{L^2},$$

$$H^m = \{u \mid \|u\|_{H^m} = \sqrt{(u, u)_{H^m}} < \infty\}.$$

Man beachte den Zusammenhang $L^2 = H^0$. Dabei besitzt $u \in L^2$ die schwache Ableitung $v = \partial^\alpha u$, falls für $v \in L^2$

$$(\phi, v)_{L^2} = (-1)^{|\alpha|} (\partial^\alpha \phi, u)_{L^2} \quad \forall \phi \in C_0^\infty(\Omega)$$

gilt. α ist ein Multiindex.

Lipschitz–Gebiete

Ist in dieser Arbeit von (hinreichend glatten) Gebieten die Rede, so seien damit stets *Lipschitz–Gebiete* bezeichnet. Nach [Braess 02] heißt ein Gebiet $\Omega \in \mathbb{R}^d$ Lipschitz–Gebiet, falls zu jedem Randvektor $x \in \partial\Omega$ eine Umgebung existiert, die sich als Graph einer Lipschitz-stetigen Funktion darstellen lässt. Oftmals ist die Lipschitz–Eigenschaft durch die intuitiv anschaulichere Kegelbedingung zu ersetzen — man konsultiere auch hierfür [Braess 02, Knabner 00].

formalisiert. Dieses Faktum wurde in der Einleitung bereits thematisiert. Die Divergenzfreiheit muss dabei durch Wahl eines passenden Drucks sichergestellt werden. Im Folgenden seien drei Ansätze aufgezeigt, diese Forderung in eine schwache Formulierung zu integrieren. Alle drei Verfahren bringen signifikante Nachteile mit sich, was die Entwicklung eines neuartigen Verfahrens motivierte.

Der erste Ansatz schränkt den Suchraum des diskreten Minimierungsproblems auf den Raum der divergenzfreien Funktionen ein. Der zweite erzwingt die Nebenbedingung durch einen weiteren Test und erhält somit $(d + 1)$ gekoppelte Differentialgleichungen, die iterativ zu lösen sind⁵. Die Klasse solcher Ansätze wird in [Ferziger 99] als *SIMPLE-type algorithms* bezeichnet. Alternativ findet man auch den Begriff Druckkorrekturmethode, da die Divergenzfreiheit durch eine Druckvariation in einem iterativen Verfahren sichergestellt wird.

Bereits thematisiert wurde die physikalische Rolle des Drucks: Er stellt die Divergenzfreiheit der zukünftigen Lösungen sicher. Der dritte Ansatz verwendet genau diese Grundidee, um den Druck explizit zu berechnen. Zunächst wird hierbei die Lösung der Impulsgleichung in mehrere Schritte aufgeteilt. Der Berechnung des Diffusions- und Konvektionsanteils folgt die explizite Berechnung des Druckgradienten, um abschließend die Zeit in einem Cauchy-Schritt fortzuschalten. Diese Art der Problemlösung bezeichnet man auch als *fractional-step method* und wurde unabhängig voneinander in [Chorin 68] und [Temam 69] entwickelt. Eine englischsprachige Abhandlung von [Temam 69] findet sich in [Temam 01], alternativ kann man [Donea 03] oder [Gresho 98] heranziehen. Algorithmen, die der Klasse der *fractional-step* Methoden zuzuordnen sind, finden vor allem in der Berechnung langsam fließender Strömungen Anwendung.

2.4 Einbringen der Kontinuitätsgleichung

2.4.1 Divergenzfreie Geschwindigkeitsansatzräume

Mit der Wahl eines diskreten divergenzfreien Geschwindigkeitsraums bildet man die Forderung der Divergenzfreiheit der Lösung in den zugrundeliegenden Funktionenraum ab, lässt also nur Basisfunktionen zu, die per se selbst divergenzfrei sind. Dadurch kann in der schwachen Formulierung der Druckterm eliminiert werden, da sich der Gradient des Drucks mit der Greenschen Formel auf die (divergenzfreie) Testfunktion „abwälzen“ lässt:

$$(\nabla p, \varphi)_{L^2} = -(p, \operatorname{div} \varphi)_{L^2} = 0 \quad \forall \varphi \in V_0$$

Die Lösung dieses elliptischen Problems liegt bei homogenen Dirichlet-Bedingungen⁶ nun nach [Gresho 98] ebenfalls im Raum der divergenzfreien Elemente⁷. Existenz, Fehleraussagen und Eindeutigkeit ergeben sich daher in diesem Fall aus den Sätzen von Lax–Milgram und dem Céa–Lemma.

Die ausschließliche Arbeit mit divergenzfreien Ansatzräumen bringt jedoch meist einige Probleme mit sich: So kann in vielen Fällen die in [Ciarlet 78] geforderte Eigenschaft der lokalen Träger nicht erfüllt werden oder die Konformität der diskreten Ansatzräume ist nicht gegeben. Im Fall der Nichtkonformität ist der Approximationsraum nicht Subraum des Lösungsraums der schwachen Formulierung und jede Fehlerabschätzung muss um einen Konsistenzfehler erweitert werden. Ein Beispiel für divergenzfreie Elemente findet sich in [Braess 02], [Gresho 98] führt die Historie der divergenzfreien Elemente genauer aus.

⁵Mit d sei im Folgenden stets die Dimension bezeichnet. Da es sich bei der Geschwindigkeit um ein Vektorfeld handelt, ist hier Dimension des Bildbereiches gleich der Dimension des Gebietes, d.h. mit $x \in \Omega, u(x) \in \mathbb{U}(\Omega)$ gilt $\dim \Omega = \dim \mathbb{U}(\Omega)$.

⁶Die Behandlung inhomogener Dirichlet-Bedingungen wird dem interessierten Leser auf Seite 20 nahegebracht.

⁷Man beachte, dass in dieser Arbeit der nichtlineare Term stets auch in \mathbb{U} approximiert wird. Eine Evaluierung des dadurch entstehenden Fehlers kann und soll hier nicht geboten werden.

2 Herkömmliche Lösungsverfahren für die Navier–Stokes Gleichungen

Obwohl die Lösung nach Vorgabe in $V_{\text{div } u=0}$ liegen muss, macht es somit zunächst Sinn, den Suchraum um die nicht-divergenzfreien Elemente zu erweitern. Der Limes eines jeden Iterationsverfahrens wird dann wieder im divergenzfreien Unterraum liegen. Tatsächlich wird in unserem erweiterten Ansatz diese Augmentierung zunächst vorgenommen, anschließend jedoch durch die sogenannte *Druckpoissongleichung* eine Einschränkung auf den probaten Unterraum vorgenommen.

2.4.2 Gemischte Ansätze

Der größten Beliebtheit auf dem Gebiet der *Computational Fluid Dynamics* (CFD) erfreuen sich zweifellos die gemischten Ansätze. Auch so bekannte Verfahren wie SIMPLE kombiniert mit der FEM lassen sich dieser Klasse von Verfahren zuordnen. Hierbei wird die Divergenzfreiheit nicht explizit in den Funktionenraum abgebildet, sondern wird durch einen weiteren Test der Kontinuitätsgleichung über $u^{t+\tau}$ erzwungen.

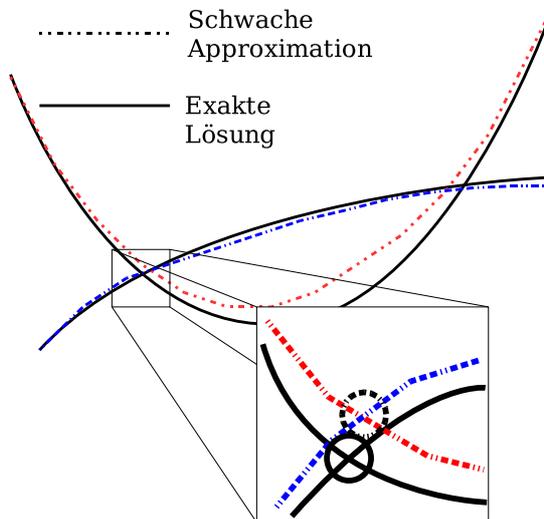


Abbildung 2.4: Die gemischten Methoden approximieren beide Gleichungen. Daher liegt die Lösung an einem unphysikalischen Ort, entspricht also i.A. unendlicher Energie.

Gelöst wird also zusätzlich zur schwachen Impulsgleichung noch

$$\left(\mathbf{div} \mathcal{T}_{\tau,\theta}(u^{(t)}), \phi \right)_{L^2} = 0.$$

Die meisten Verfahren verzichten auf eine iterative Lösung des Gesamtsystems und begnügen sich stattdessen mit einer festgelegten Iterationsreihenfolge. So liegt z.B. der Chorin–Temam–Methode [Temam 01] eine Zweiteilung zugrunde: Die Impulsgleichung wird zunächst mit einem Druckgradienten $\nabla p = 0$ gelöst und so eine vorläufige Geschwindigkeit errechnet. In einem zweiten Schritt korrigiert man das obige reduzierte Gleichungssystem mit der vorläufige Geschwindigkeit als Ausgangspunkt, wodurch sich die Gleichungen drastisch vereinfachen.

Derartige Verfahren werden auch als Druckkorrekturmethode bezeichnet, da sie über den Druck die Divergenzfreiheit „korrigieren“. Da beide Differentialgleichungen gekoppelt betrachtet werden,

gehört der Ansatz in die Klasse der gekoppelten Verfahren. Vergleiche gekoppelter Verfahren mit nicht gekoppelten Verfahren, wie sie in Abschnitt 2.4.3 angerissen werden, finden sich unter anderem in [Sivaloganathan 88, Sivaloganathan 91].

Wahl der Ansatzräume

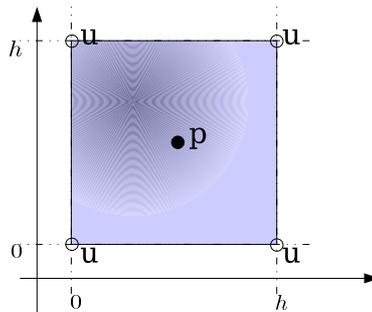


Abbildung 2.5: Das beliebte Q^1P^0 -Element: Hier werden bilineare Geschwindigkeitsansatzfunktionen auf den Gitterpunkten in Kombination mit zellweise konstantem Druck verwendet.

Direkt aus obiger Gleichung lassen sich die Bedingungen

$$\begin{aligned} u &\in \mathbb{U} \subseteq H^1(\Omega)^d \quad \text{und} \\ p &\in \mathbb{P} \subseteq H^0(\Omega) = L^2(\Omega) \end{aligned}$$

an die Glattheit der Lösung für die Geschwindigkeit u und den Druck p ableiten. Nach Definition des Ritz-Galerkin-Verfahrens verwendet man

$$\begin{aligned} \varphi &\in \mathbb{U} \quad \text{und} \\ \phi &\in \mathbb{P}. \end{aligned}$$

Nur durch obige Wahl von ϕ wird die Adjungiertheit respektive Transponiertheit des Divergenz- und Gradientenoperators beibehalten, wie es auf den Seiten 3 und 37 gefordert wird, weil sich aufgrund der lokalen Träger der Ansatz-/Testfunktionen durch partielles Ableiten der Gradient des Drucks auf die Testfunktion „abwälzen“ lässt:

$$\int_{\Omega} (\nabla p, \varphi) \, dx = - \int_{\Omega} p \, \mathbf{div} \, \varphi \, dx$$

Konvergenztheorie

Die Konvergenztheorie zu solch einer Variationsformulierung eines Sattelpunktproblems entspricht zunächst formal der Standardproblematik, betrachtet man die Lösung $(u, p) \in H^1(\Omega)^d \times H^0(\Omega)$. Dennoch lassen sich Standardresultate wie der Existenzsatz von Lax-Milgram nicht direkt auf diese Problemstellung übertragen, da mit den stetigen Bilinearformen

$$\begin{aligned} a : \mathbb{U} \times \mathbb{U} &\mapsto \mathbb{R}, & b : \mathbb{P} \times \mathbb{U} &\mapsto \mathbb{R}, \\ a(u, \varphi) + b(p, \varphi) &= 0 \\ b^*(u, \phi) &= 0 \end{aligned}$$

Exkurs: Einbringen von Randbedingungen

Dirichlet Randbedingungen

Seien zu einer elliptischen Differentialgleichung $u : \Omega \mapsto \mathbb{R}$ Dirichlet–Randwerte gegeben. Wir betrachten also beispielsweise

$$\begin{aligned} -\Delta u(x) &= 0 & \forall x \in \Omega, \\ u|_{\partial\Omega} &= g. \end{aligned}$$

Ist dabei $g \neq 0$, so spricht man von inhomogenen Randbedingungen. Offensichtlich verschwindet bei inhomogenen Rändern das Randintegral aus der Greenschen Formel in der schwachen Formulierung nicht. Dann kann man durch eine geeignete additive Aufteilung des Problems, das inhomogene Dirichlet–Problem auf ein homogenes Dirichlet–Problem mit modifizierter rechter Seite zurückführen. Man wähle

$$\begin{aligned} u &:= u_D + u_0, \\ u_D|_{\partial\Omega} &= g \quad \text{und} \\ u_0|_{\partial\Omega} &= 0. \end{aligned}$$

Man beschränkt sich jetzt bei der schwachen Formulierung wiederum auf „homogene“ Testfunktionen, also Funktionen, die auf dem Rand verschwinden und lässt somit beim Übergang in den diskreten Funktionenraum beide Randintegrale verschwinden. Der verbleibende Term $\int_{\Omega} (u_D, \varphi) dx$ ist von der approximierten Lösung der Differentialgleichung unabhängig und wird auf die rechte Seite gebracht.

Neumannsche Randwertaufgaben

Die Behandlung Neumannscher Randwertaufgaben gestaltet sich etwas subtiler. Sei also im Folgenden

$$\begin{aligned} -\Delta u(x) &= 0 & \forall x \in \Omega, \\ (u, n)|_{\partial\Omega} &= g \end{aligned}$$

Gegenstand der Betrachtung. Dann verbleibt auch bei der Formulierung mit auf dem Rand verschwindenden Testfunktionen der Term $\int_{\partial\Omega} (\nabla u, n) \varphi dx$ auf der linken Seite der Gleichung. Nach dem Spursatz [Braess 02] gibt die schwache Lösung nun die klassische Lösung wieder, falls

- die klassische Randwertlösung existiert, also $g \in C(\Omega)$ und
- $g \in L^2(\partial\Omega)$ gilt.

Dabei ist $u \in H^1(\Omega)$ und die klassische Lösung in $C^2(\Omega) \cap C^1(\overline{\Omega})$.

die schwache Formulierung der Impulsgleichung die beiden verschiedenen Räume \mathbb{U} und \mathbb{P} umfasst⁸. Abhilfe schafft hier eine Verallgemeinerung des Satzes von Lax–Milgram, die als inf–sup– oder Babuška–Brezzi–Bedingung bekannt ist:

Theorem 1 *Seien \mathbb{U} und \mathbb{P} Hilberträume, $a : \mathbb{U} \times \mathbb{U} \mapsto \mathbb{R}$ und $b : \mathbb{P} \times \mathbb{U} \mapsto \mathbb{R}$ stetige Bilinearform mit a \mathbb{U} -elliptisch und b aus dem Dualraum von \mathbb{U} ($b \in \mathbb{U}^*$). Dann hat das Problem*

$$\text{finde } p \in \mathbb{P} : a(u, \varphi) + b(p, \varphi) = 0 \quad \forall \varphi \in \mathbb{U}$$

genau eine Lösung genau dann wenn

$$\inf_{0 \neq p \in \mathbb{P}} \sup_{0 \neq \varphi \in \mathbb{U}} \frac{b(p, \varphi)}{\|p\|_{\mathbb{P}} \|\varphi\|_{\mathbb{U}}} \geq \gamma > 0$$

und

$$\forall 0 \neq \varphi \in \mathbb{U} : \sup_{p \in \mathbb{P}} b(p, \varphi) > 0$$

gelten.

Für einen Beweis dieses Satzes sei auf [Braess 02] und [Girault 86] verwiesen. Man beachte für die Eindeutigkeit des Raumes, dass es sich bei der \mathbb{P} -Norm um eine Halbnorm handelt, die lediglich die Ableitung miteinbezieht, aber translationsinvariant ist.

Bevor darauf eingegangen werden soll, welche Auswirkungen die Missachtung der inf–sup–Bedingung auf numerische Lösungen hat, wird kurz eine alternative Sichtweise, die eher der linearen Algebra entstammt, angerissen:

Operatormatrixformulierung

Nach [Donea 03] kann der Raumanteil, sprich das Sattelpunktproblem, der (Navier–) Stokes Gleichungen⁹ auch als Matrix–Funktionen–Produkt formalisiert werden:

$$\begin{pmatrix} D & 0 & & \\ 0 & D & & M^T \\ & & \dots & \\ & M & & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \dots \\ p \end{pmatrix} = \dots$$

Man beachte, dass hierbei bereits die im diskreten nachzuvollziehende Adjungiertheit von Gradient– und Divergenzoperator eingeflossen ist. Dieses System ist nach den Regeln der linearen Algebra nun eindeutig lösbar, falls die Operatormatrix vollen Rang hat. Aufgrund der Diagonalstruktur des Konvektions–/Diffusionsteils ist dies äquivalent zur Forderung, dass der Kern des Gradientenoperators die Dimension 0 aufweist, also

$$\dim \ker M^T = 0$$

gilt. Dabei sei wiederum auf die Halbnormeigenschaft des Druckraums hingewiesen: Ist für geschlossene Szenarien keine Einflussbedingung gegeben, so ist der Druck nur bis auf eine Konstante bestimmt¹⁰. Offensichtlich gilt folgender Zusammenhang:

⁸ b^* sei der zu b adjungierte Operator.

⁹Die Erweiterung um den nichtlinearen Teil ist trivial.

¹⁰Man erinnere sich an die Energienorm zu elliptischen Differentialgleichungen: Auch hier gehen nur die Ableitungen in die Normauswertung ein, alle konstanten Funktionen werden jedoch auf das Nullelement in \mathbb{R} abgebildet. Existieren Dirichlet–Ränder, so bringt man diese in den normierten Funktionenraum ein (siehe Seite 20), Hat man ausschließlich Neumann–Ränder gegeben, so normiert man über das L^2 -Skalarprodukt nach Belieben [Braess 02].

Korrolar 2 Gilt für ein gemischtes Problem die inf–sup–Bedingung, so hat das System in der Operatormatrixformulierung vollen Rang.

Es bleibt für das hier behandelte Problem folgendes festzustellen: Genüge ein Ansatzraum der inf–sup–Bedingung, so ist die Differentialgleichung analog zu bekannten elliptischen Problemen

- eindeutig lösbar, falls ein Dirichlet–Rand gegeben ist und
- bis auf eine Konstante eindeutig bestimmt, falls ausschließlich Neumann–Ränder gegeben sind.

Genügt ein System jedoch nicht der inf–sup–Bedingung, so existiert ein Druck $p_{unphysikalisch}$, so dass für alle Testfunktionen der Term $b(p_{unphysikalisch}, \varphi)$ verschwindet. Dabei handelt es sich aber nicht nur um die bereits angesprochenen konstanten Funktionen, die bei Fehlen von Dirichlet–Bedingungen aufgrund der Halbnormeigenschaft zu erwarten sind, sondern um nichtkonstante Druckentwicklungen. Ist folglich p_0 die (korrekte) Lösung, so ist in diesem Fall auch $p_0 + \alpha p_{unphysikalisch} \quad \forall \alpha \in \mathbb{R}$ eine Lösung. Das wäre tödlich für ein Iterationsverfahren, da hier die Konvergenz nicht sichergestellt sein kann, und natürlich unphysikalisch [Trottenberg 00], da es in der Natur nur einen korrekten Druck geben kann.

Spurious Modes

Für die meisten „nicht–inf–sup–Elemente“ kann zeigen, dass solche Druckverteilungen $p_{unphysikalisch}$ auf dem Rand im Allgemeinen nicht verschwindet. Offensichtlich impliziert also die Existenz eines Dirichlet–Randes für den Druck auch die Eindeutigkeit des desselben und damit die Stabilität der Methodik — auch für Ansatzräume, die nicht der inf–sup–Bedingung genügen. Es liegt also nahe, nach Ansätzen zu suchen, bei Fehlen eines echten Dirichlet–Randes über einen künstlichen Rand die Lösung zu stabilisieren. Doch auch hier können in der numerischen Lösung unphysikalische Instabilitäten auftreten, die Iterationsverfahren nicht wegglatzen können — solche Lösungen werden auch als *spurious modes* bezeichnet. Als Erklärung hierfür diene eine Konstruktionsvariante für solche Modi:

Die Grundidee ist hierbei, zunächst unter Vernachlässigung der Randbedingungen die unphysikalischen Druckverteilungen $p_{unphysikalisch}$ zu konstruieren¹¹ und anschließend durch entsprechende Skalierung daraus die *spurious modes* $p_{spurious}$ abzuleiten. Sei als Beispiel ein solches Druckmuster $p_{unphysikalisch}$ für die Driven–Cavity auf dem Einheitshyperwürfel gefunden, so kann man mittels der Skalierung

$$p_{spurious}(x) = p_{unphysikalisch}(x) \prod_{k=1}^d \sin(\pi x_k)$$

spurious modes für alle Dirichlet–Ränder konstruieren. Diese verschwinden jetzt auch auf dem Rand und genügen damit auch dem Dirichlet–Rand. Eine ausführlichere Diskussion dieser Konstruktionsmethodik für unphysikalisches Druckverhalten auf Basis einer Eigenwertbetrachtung findet sich beispielsweise [Zeeuw 97] ab Seite 92 — diese Modi werden dort als *near chess–board grid–function* bezeichnet.

Abhilfe schafft also meist nur die Verwendung zusätzlicher Qualitätsmetriken oder unregelmäßiger Gitter — letzteres wird sich gleich anhand der Konstruktion der Modi über Symmetrieachsen ergeben. Abschließend sei Professor Peter Hansbo vom Department of Applied Mechanics der Chalmers University of Technology in Göteborg zitiert, der am 25. Juni 1999 in der sci.math.num-analysis folgendes von sich gegeben hat:

¹¹Anschaulich betrachtet man also zunächst ein unendlich ausgedehntes Rechengebiet Ω .

”Spurious” is just a fancy FE word for ”unstable”. You have ”spurious” modes when under-integrating, you have ”spurious” oscillations in the Galerkin solution of convection-diffusion problems, and you have ”spurious” pressure modes in mixed FE-solutions of incompressible fluid and elasticity problems. Since it is unthinkable that FEM solutions are unstable (grin), then, if they are wrong, they have to be ”spurious”.

Schachbrettmuster

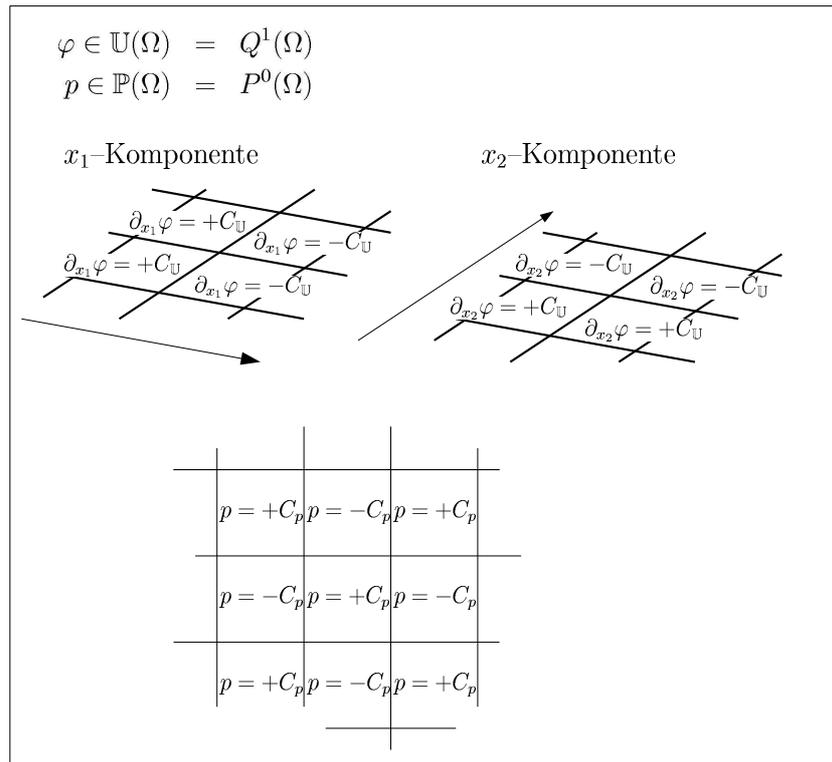


Abbildung 2.6: Konstruktion der *spurious modes* zum Q^1P^0 -Element.

Fasst man obige Überlegungen zusammen, so ergibt sich direkt, dass das beliebige Q^1P^0 Element instabil sein muss, weil bei einer schachbrettartigen Druckanordnung $p = +C$ respektive $p = -C$ selbiger im Test verschwindet. Die Konstruktionsprinzipien, also eine induktive Darstellung, sei im Folgenden gegeben¹²:

Man betrachte die Gradientenverteilung eines Geschwindigkeitshutes, wie sie in Abbildung 2.6 wiedergegeben ist. Offensichtlich verschwindet der x_1 Anteil, falls der Druck eine horizontale Rotationsachse besitzt, die in der Symmetrieachse der Ansatzfunktion verläuft. Analog verschwindet der x_2 Anteil bei einer passenden vertikalen Achse.

Man wähle nun (zunächst wieder unter völliger Vernachlässigung der Randbedingungen) auf einem regelmäßigen Gitter ein sogenanntes *Schachbrettmuster* (*checkerboard pattern*), das genau

¹²Bevorzugt man deduktives Argumentieren, also schlicht und einfach Nachrechnen, so obliegt dies dem geeigneten Leser.

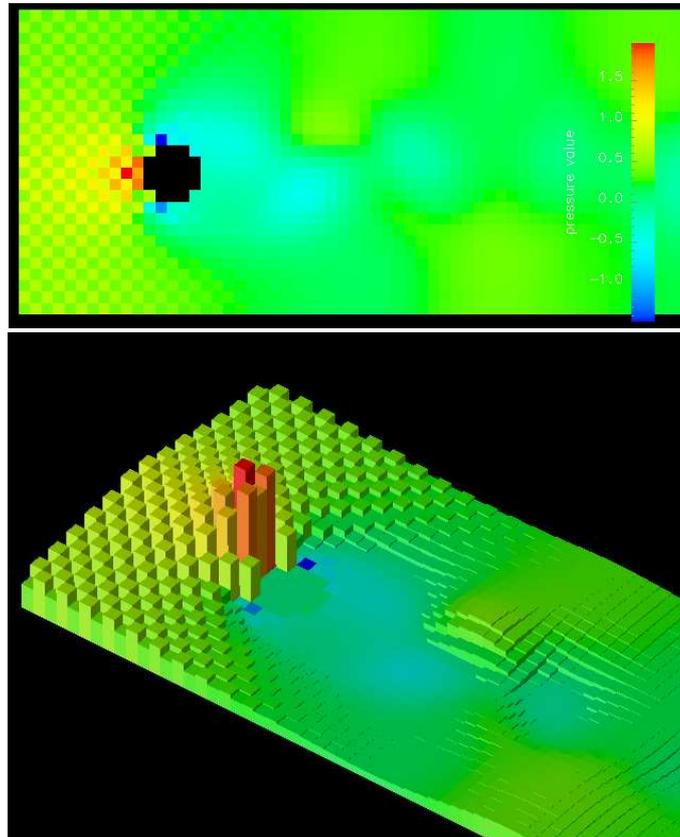


Abbildung 2.7: Checkerboard-Druckverteilung, wie sie im Programm für einen Kanal mit Hindernissen bei Verwendung des Q^1P^0 -Elements auftreten kann. Hierbei sind an der Ausflusskante Dirichlet-Randbedingungen für die Geschwindigkeiten gegeben. Das Fluid strömt im oberen Bild horizontal von links nach rechts.

solche Achsen aufweist. Mit obigen Wissen um die spurious modes lassen sich also durch Skalierung über Ω problematische Druckverteilungen konstruieren. Wie [Neckel 05] zeigt, ließen sich bei Verwendung instabiler Elemente solche Muster auch mit der hier entwickelten Software beobachten. Ein Beispiel solcher Muster gibt Abbildung 2.7.

Evaluierung

Gemischte Verfahren als Vertreter gekoppelter Methoden werden heute in einem sehr breiten Spektrum angeboten und gelten als sehr zuverlässig [Vanka 85]. Als nachteilig erweist sich bei derartigen Verfahren, dass die Divergenzfreiheit nur anhand eines zweiten Tests sichergestellt wird. [Verstappen 01] weist zurecht darauf hin, dass es sehr schwierig ist, Operatoren zu entwerfen, die den physikalischen Erhaltungssätzen genügen. Daher ist die gelieferte Lösung stets „etwas“ unphysikalisch.

Nach [Gresho 98] wird desweiteren die oben ausgeführte inf-sup-Stabilitätsbedingung in der Praxis oftmals ignoriert, wobei man mögliche Instabilitäten apriori durch Erfahrungswerte auszuschließen versucht. Gerade das Q^1P^0 -Element wird in diesem Zusammenhang oftmals verwendet.

Dennoch ist die Situation, eine Theorie zu haben, sie aber aus Praktikabilitätsgründen nicht zu verwenden, unbefriedigend. Es bleibt die Professor Gilbert Strang vom MIT zugeschriebene¹³ Aussage zu zitieren:

Mixed interpolation brings mixed blessings.

2.4.3 Approximation der Nebenbedingung über die Druckpoissongleichung

Die kontinuierliche Forderung $\mathbf{div} u = 0$ gilt zu jedem Zeitpunkt, insbesondere also auch zu den Zeitpunkten t und $t + \tau$. Wie bereits bekannt ist, muss $\mathbf{div} u^{(t)} = 0$ gelten, da ansonsten das Geschwindigkeitsfeld für den aktuellen Zeitschritt unphysikalisch ist. Der zweite Teil der obigen Forderung lässt sich als

$$\mathbf{div} u^{(t+\tau)} = \mathbf{div} \mathcal{T}_{\tau,\theta}(u^{(t)}) = 0$$

formalisieren.

Unter entsprechenden Glattheitsannahmen leitet sich somit für Zeitdiskretisierungen der θ -Methode die Druckpoissongleichung her:

$$\begin{aligned} \mathbf{div} u^{(t+\tau)} &= \mathbf{div} \mathcal{T}_{\tau,\theta}(u^{(t)}) \\ &= \mathbf{div} u^{(t)} + \tau \mathbf{div} \left((1 - \theta) d_t u^{(t)} + \theta d_t u^{(t+\tau)} \right) \\ &= \mathbf{div} u^{(t)} + \tau \mathbf{div} \left((1 - \theta) \left(-\mathcal{F}(u^{(t)}) - \nabla p^{(t)} \right) + \theta \left(-\mathcal{F}(u^{(t+\tau)}) - \nabla p^{(t+\tau)} \right) \right) \\ &= 0 \end{aligned}$$

Nimmt man analog zur Geschwindigkeit den Druckgradienten als linear in $[t, t + \tau]$ an, so gilt

$$(1 - \theta) \nabla p^{(t)} + \theta \nabla p^{(t+\tau)} = \nabla p^{(t+\theta\tau)}.$$

Man erhält nun mit $\mathbf{div} \nabla p = \Delta p$ tatsächlich eine Poisson-Gleichung für den Druck. Die rechte Seite ist dabei ausschließlich durch Konvektions- und Diffusionsterm bestimmt. Da nur ein Druck(gradient) in der Formel auftritt, sei dieser nicht mit einer Zeit annotiert:

$$\nabla p^{(t+\theta\tau)} = \Delta p = \frac{1}{\tau} \mathbf{div} u^{(t)} - (1 - \theta) \mathbf{div} \mathcal{F}(u^{(t)}) - \theta \mathbf{div} \mathcal{F}(u^{(t+\tau)})$$

Hierbei fordert man im Kontinuierlichen also $u \in C^3(\Omega)^d, p \in C^2(\Omega)$. Man beachte ferner, dass in beiden Formen der Druck bei Fehlen eines Zu- und Abflusses (keine Dirichlet-Bedingungen), insbesondere also bei der Driven-Cavity, bis auf eine Konstante bestimmt ist [Neckel 05]. Im Folgenden sei daher immer von Funktionenräumen $C^2(\Omega) \setminus P^0$ die Rede. Derartige Methoden werden auch als Chorinsche Projektionsmethoden [Chorin 68] bezeichnet. Chorin jedoch hat die Druckpoissongleichung in Zusammenhang mit einem räumlichen Finite Differenzen Verfahren und $\theta = 0$, also einer expliziten Euler-Zeitdiskretisierung, eingeführt

Termvereinfachungen

Der $\mathbf{div} u^{(t)}$ Term kann nach Voraussetzung vernachlässigt werden. Ebenso lässt sich der Term $\mathbf{div} \Delta u$ für jeden Zeitpunkt unter dieser Voraussetzung und hinreichender Glattheit durch Vertau-

¹³Man konsultiere hierzu [Gresho 98], Seite 469.

2 Herkömmliche Lösungsverfahren für die Navier–Stokes Gleichungen

schen der Ableitungsoperatoren wegdiskutieren:

$$\begin{aligned}
 \mathbf{div} \Delta u &= \mathbf{div} \begin{pmatrix} \Delta u_1 \\ \Delta u_2 \end{pmatrix} \\
 &= \partial_{x_1} \Delta u_1 + \partial_{x_2} \Delta u_2 \\
 &= \partial_{x_1} \partial_{x_1}^2 u_1 + \partial_{x_1} \partial_{x_2}^2 u_1 + \partial_{x_2} \partial_{x_1}^2 u_2 + \partial_{x_2} \partial_{x_2}^2 u_2 \\
 &= \partial_{x_1}^2 (\partial_{x_1} u_1 + \partial_{x_2} u_2) + \partial_{x_2}^2 (\partial_{x_1} u_1 + \partial_{x_2} u_2) \\
 &= 0
 \end{aligned}$$

Analog kann man den Konvektionsterm etwas vereinfachen. Beide Vereinfachungen werden später nicht berücksichtigt werden, um eventuelle numerische Fehler, also Abweichungen von der Divergenzfreiheit, automatisch in späteren Zeitschritten korrigieren zu können und eine Akkumulation der Fehler zu verhindern.

Evaluierung

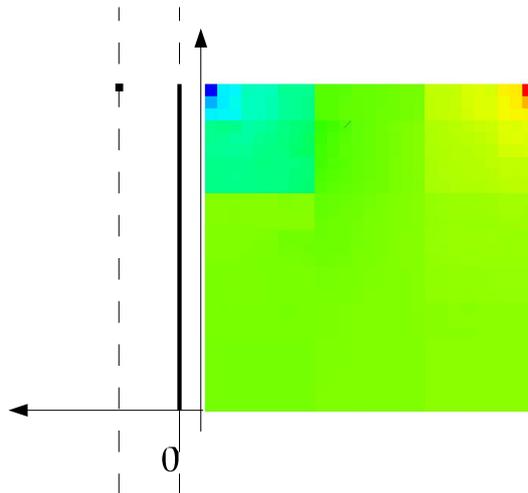


Abbildung 2.8: Im Driven–Cavity Szenario treten bereits an zwei Stellen an der oberen Kante im Drucknormalenanteil (Halb–) Singularitäten auf. Eine Lösung der Druckpoissongleichung über eine schwache Formulierung ist daher theoretisch nach dem Spursatz nicht untermauert.

Lösungen über die Druckpoissongleichung scheinen die Methode der Wahl zu sein, da der Druck explizit berechnet werden kann, sind erst einmal Konvektions- und Diffusionsteil bekannt. Auf den ersten Blick scheint dabei eine räumliche Approximation über die Finite–Element–Methode erfolgversprechend zu sein, da es sich um ein elliptisches Problem handelt. Zu dieser Problemklasse ist die Theorie wohlbekannt, die höhere Glattheitsforderung ist relativ einfach zu erfüllen.

Damit bringe man die Druckpoissongleichung in eine schwache Form und löse diese. Als tatsächlich problematisch an diesem Ansatz erweist sich jedoch die Definition geeigneter schwacher Randbedingungen. Die exakten Randbedingungen werden später diskutiert und finden sich ausführlicher in [Neckel 05], der sich intensiv mit Fluid–Struktur–Wechselwirkungen auseinandersetzt. Hier wird das Problem auf Neumannsche Randwertaufgaben zurückgeführt:

$$\frac{\partial}{\partial n} p = - \frac{\partial}{\partial n} \mathcal{F}(u)$$

Der Übergang in die schwache Formulierung erfordert nun nach dem Spursatz (siehe Seite 20) mindestens $p \in C(\overline{\Omega})$. Diese Forderung ist jedoch im Allgemeinen nur schwer zu erfüllen, da bekanntlich in der numerischen Strömungssimulation vielfach im Druck (Halb-) Singularitäten auftreten. Abbildung 2.8 zeigt dies anhand des Driven-Cavity Szenarios. Damit bleibt der theoretische Zugang über die schwache Druckpoissongleichung für viele praktische Probleme verwehrt.

Stichwortsammlung

Volldiskretisierung, Semidiskretisierung; fractional-step-method; Zeitoperator, θ -Methoden, Forward-Euler; divergenzfreie Räume; gemischte Verfahren, inf-sup-Bedingung, Operatormatrixformulierung, Existenz- und Eindeutigkeitsaussage, Schachbrettmuster, spurious modes; Druckpoissongleichung; Adjungiertheit der Operatoren ∇ und $\operatorname{div} = \nabla^T$

2 *Herkömmliche Lösungsverfahren für die Navier-Stokes Gleichungen*

3 Lösungsalgorithmus

Das vorangegangene Kapitel hat einige Ansätze zur Lösung der Navier–Stokes Gleichungen präsentiert. Neben der Darstellung der Ansätze war vor allem die Diskussion der Unzulänglichkeiten derselben von Interesse. Die Verwendung divergenzfreier Elemente gestaltet sich — wie dargestellt wurde — im Allgemeinen als sehr schwierig [Gresho 98]. Algorithmen über gemischte Ansätze bringen unphysikalische Lösungen mit sich und erfordern eine erweiterte Theorie. Für die Druckpoissonsgleichung existiert in vielen Fällen sogar gar keine (Existenz–) Theorie.

In diesem Kapitel soll ein Algorithmus vorgestellt werden, der versucht, die dargestellten Klippen zu umschiffen. Vorausgesetzt sei bei der Einführung dieser neuen Methode, dass das Gebiet bereits durch ein Gitter diskretisiert ist. Der Begriff der Diskretisierung als Zerlegung des Rechengebietes in eine endlich Zahl schnittfreier geometrischer Elemente wird ab Seite 71 genauer behandelt.

Grundlage der neuen Methode ist folgender Hilfssatz, für den [Blanke 04] einen konstruktiven Beweis führt:

Lemma 1 *Sei das Rechengebiet Ω diskretisiert als eine Menge von Mengen Ω_h , wobei alle Elemente $V \in \Omega_h$ Quadrate seien. Dann existieren Finite Elemente zu einem Ansatzraum \mathbb{U} mit*

$$\mathbf{div} u = 0 \text{ fast überall} \Leftrightarrow \int_V \mathbf{div} u \, dx = 0 \quad \forall V \in \Omega_h, u \in \mathbb{U}$$

Dieses Existenzlemma wird sich später im Rahmen der Konstruktion der Elemente ergeben. Im Folgenden bezeichne \mathbb{U} den zugehörige Ansatzraum und es ergibt sich sofort:

Theorem 2 *Sei $u^{(t+\tau)} \in \mathbb{U}$. Die punktweise Divergenzfreiheit von $u^{(t+\tau)}$ läßt sich auf ein endliches Gleichungssystem mit $|\Omega_h|$ Elementen zurückführen.*

Aussagen über Eindeutigkeit und Lösbarkeit dieses Gleichungssystems werden sich im Laufe dieses Kapitels ergeben. Dabei zeigt sich, dass sich analog zur Chorinschen Projektionsmethode die Divergenzfreiheit durch eine Variation des Drucks einstellt: Nachdem man sich auf den (noch zu definierenden) Subraum \mathbb{U} zurückgezogen hat, kann man den Druck über die Druckpoissonsgleichung als Funktion von $u^{(t)}$ und $u^{(t+\tau)}$ darstellen. Statt dabei jedoch die Druckpoissonsgleichung kontinuierlich herzuleiten, argumentiere man aufgrund der im Hilfssatz postulierten auf dem Unterraum \mathbb{U} geltenden Äquivalenz über die „schwache“, d.h. die Integralformulierung¹, und erhält mit der Forderung

$$\int_V \mathbf{div} \nabla p \, dx = \int_V \frac{1}{\tau} \mathbf{div} u^{(t)} - (1 - \theta) \mathbf{div} \mathcal{F}(u^{(t)}) - \theta \mathbf{div} \mathcal{F}(u^{(t+\tau)}) \, dx \quad \forall V \in \Omega_h$$

eine Formel zur Berechnung des Druckgradienten².

¹Tatsächlich wurde der Begriff „schwach“ gewählt, weil die kontinuierliche Forderung durch ein Integral abgelöst wurde. Die Divergenzfreiheit wird jedoch nach wie vor fast überall exakt erfüllt. Man beachte, dass die neue Formulierung auch als Finite–Volumen–Methode oder als Ritz–Verfahren mit dem Testraum der charakteristischen Funktionen der geometrischen Elemente aufgefasst werden kann.

²Natürlich kann hier wieder $\mathbf{div} u^{(t)}$ wegdiskutiert werden. Verbleibt der Term jedoch in der Formulierung, so können Rechenfehler der vorangegangenen Schritte über den Druckausgleich eliminiert werden.

Algorithmus 1 Generischer θ -Algorithmus

-
- 1: Initialisierung
 - 2: **while** Maximalzeit noch nicht erreicht **do**
 - 3: Berechne Zeitschrittweite τ
 - 4: Berechne Divergenz-/Konvektionsterm $\mathcal{F}(u^{(t)})$ mit der FEM (Präparationsschritt)
 - 5: **while** Abbruchkriterium der Druckiteration noch nicht erreicht **do**
 - 6: Berechne vorläufiges $u^{(t+\tau)}$ mit der FEM (Cauchy-Schritt):

$$u^{(t+\tau)} = \mathcal{T}_{\tau,\theta}(u^{(t)})$$

- 7: Berechne Divergenz-/Konvektionsterm $\mathcal{F}(u^{(t+\tau)})$ mit der FEM (Präparationsschritt)
- 8: Iterationsschritte auf Druckpoissongleichung

$$\int_V \mathbf{div} \nabla p \, dx = \int_V \frac{1}{\tau} \mathbf{div} u^{(t)} - (1 - \theta) \mathbf{div} \mathcal{F}(u^{(t)}) - \theta \mathbf{div} \mathcal{F}(u^{(t+\tau)}) \, dx \quad \forall V \in \Omega_h$$

- 9: **end while**
- 10: Schalte Zeit fort:

$$t \mapsto t + \tau$$

- 11: **end while**
-

3.1 Algorithmus

Hat man eine passende Gebietsdiskretisierung vorgenommen und verwendet die kontinuieritätserhaltenden Finiten Elemente, so ergibt sich Algorithmus 1.

3.1.1 Die ($\theta = 0$)-Methode

Verwendet man eine ($\theta = 0$)-Methode, so sind die vorläufige Berechnung des zukünftigen Geschwindigkeitsverlauf und des Divergenz-/Konvektionsterms innerhalb der Druckiteration hinfällig und die Methode vereinfacht sich zu Algorithmus 2, also zu einer einfachen Druckkorrekturmethode. Der zeitgenerische Begriff *Präparationsschritt* bezieht sich dann ausschließlich auf die Geschwindigkeit zum Zeitpunkt t .

Nachdem der Algorithmus definiert worden ist, widmet sich der Rest des Kapitels verschiedenen handwerklichen Aufgaben. So wird zunächst die schwache Formulierung der Impulsgleichung vorgenommen und die Glattheit der Lösungsfunktionen diskutiert. Anschließend wird aufbauend auf der Konstruktion der Geschwindigkeitselemente für \mathbb{U} nach [Blanke 04] erörtert, welche Druckräume zu welchen Geschwindigkeitsräumen „passen“. Grundlegend hierfür ist die Wahl der linearen Zeitfortschaltung sowie die schon mehrfach angesprochene Adjungiertheit der Operatoren. Da die Nebenbedingung exakt erfüllt wird, braucht die Existenz und Eindeutigkeit der diskreten Lösung nicht gezeigt werden, da sie sich direkt aus der FEM Standardtheorie [Braess 02] ergibt.

3.1.2 Zeitschrittweitensteuerung

Als problematischer erweist es sich, eine Stabilitätsbedingung an die Zeitschrittweite zu definieren: Hier existiert mit der Courant-Friedrichs-Levi-Bedingung (CFL) eine obere Grenze an die Zeitschrittweite, die von der Granularität der Raumdiskretisierung abhängt. Sie besagt, dass der kontinuierliche Ereigniskegel der Simulation stets im numerischen Ereigniskegel enthalten sein muss, ein Ereignis (z.B. ein Geschwindigkeits-Peak, aber auch ein Fehler) sich also im Kontinuierlichen nicht

Algorithmus 2 Simulation mit der $(\theta = 0)$ -Methode

- 1: Initialisierung
- 2: **while** Maximalzeit noch nicht erreicht **do**
- 3: Berechne Zeitschrittweite τ
- 4: Berechne Divergenz-/Konvektionsterm $\mathcal{F}(u^{(t)})$ mit der FEM (Präparationsschritt)
- 5: **while** Abbruchkriterium der Druckiteration noch nicht erreicht **do**
- 6: Iterationsschritte auf Druckpoissongleichung

$$\int_V \mathbf{div} \nabla p \, dx = \int_V \frac{1}{\tau} \mathbf{div} \, u^{(t)} - \mathbf{div} \mathcal{F}(u^{(t)}) \, dx \quad \forall V \in \Omega_h$$

- 7: **end while**
- 8: Berechne $u^{(t+\tau)}$ mit der FEM (Cauchy-Schritt):

$$u^{(t+\tau)} = \mathcal{T}_{\tau, \theta=0}(u^{(t)})$$

- 9: Schalte Zeit fort:

$$t \mapsto t + \tau$$

10: **end while**

schneller ausbreiten darf als es die Simulation simulieren kann. [Peyret 83, Roache 76] behandeln diesen Aspekt ausführlich.

Gleichzeitig sollte der Zeitschritt aus Effizienzgründen natürlich immer möglichst groß gewählt werden. Für die numerische Zeitschrittweite τ nennt [Griebel 95] die praktikablen Bedingungen

$$\begin{aligned} h_{min} &:= \min_{V \in \Omega_h} h_V \\ \frac{2\tau}{Re} &< \left(\frac{2}{h_{min}^2} \right)^{-1} = \frac{h_{min}^2}{2} \\ \Leftrightarrow \tau &< Re \frac{h_{min}^2}{4} \end{aligned}$$

und

$$\tau |u_1|_{max} \leq h_{min} \quad \wedge \quad \tau |u_2|_{max} \leq h_{min},$$

wobei h_V die Zellenbreite der quadratischen geometrischen Elemente bezeichnet. [Tome 94] schlägt nun vor, das Minimum über die beiden Bedingungen zu bilden und mit einem „Sicherheitsfaktor“ $[0 \dots 1]$ zu skalieren — das Resultat ist dann als aktuelle Zeitschrittweite zu verwenden. Man erhält dadurch ein nichtäquidistantes Zeitgitter, also eine *adaptive Zeitschrittweitensteuerung*³.

3.2 Finite-Elemente Formulierung der Einzelschritte

Im Folgenden werden alle Einzelschritte des Algorithmus in ihre passende schwache und schließlich Finite-Elemente Formulierung gebracht [Braess 02]. Der mit der Galerkin-Projektion verbundene Übergang zu endlich dimensionalen Funktionenräumen und damit zu Summendarstellungen, also zur Berechnung der Matrixeinträge, sei dabei noch nicht Gegenstand der Ausführungen — dieser

³Verwendet man jedoch ein nichtexplizites θ -Verfahren, so kann bei entsprechender θ -Wahl ein wesentlich größerer Zeitschritt verwendet werden.

Schritt wird erst im Kapitel 4 vollzogen. Für die Operatoren werden jedoch bereits Bezeichner eingeführt, die später als Matrixbezeichner dienen werden.

3.2.1 Präparationsschritt

Die schwache Formulierung des Präparationsschritts bezieht sich auf die Gleichung

$$\mathcal{F}(u) = (u\nabla)u - \frac{1}{Re}\Delta u,$$

die in der Finite-Elemente Formulierung zu

$$\begin{aligned} \int_{\Omega_h} (\mathcal{F}(u), \varphi) dx &= \int_{\Omega_h} \left(\left((u\nabla)u - \frac{1}{Re}\Delta u \right), \varphi \right) dx \quad \forall \varphi \in \mathbb{U} \\ &= \int_{\Omega_h} ((u\nabla)u, \varphi) dx + \frac{1}{Re} \int_{\partial\Omega_h} (\nabla u, n) \varphi dx \\ &\quad - \frac{1}{Re} \int_{\Omega_h} \begin{pmatrix} \nabla u_1 \\ \nabla u_2 \\ \dots \end{pmatrix} \begin{pmatrix} \nabla \varphi_1 \\ \nabla \varphi_2 \\ \dots \end{pmatrix} dx \\ &= ((u\nabla)u, \varphi)_{L^2} + \frac{1}{Re} \int_{\partial\Omega_h} (\nabla u, n) \varphi dx - \frac{1}{Re} \sum_i (\nabla u_i, \nabla \varphi_i)_{L^2} \\ \Leftrightarrow \Omega\mathcal{F} &= C(u)u - \frac{1}{Re}Du \end{aligned}$$

wird. Grundlage ist die wohlbekannte Tatsache, dass sich die zweite Ableitung im Laplace-Operator durch partielle Integration auf die Testfunktion „abgewälzen“ lässt⁴. Somit fordert man für den Geschwindigkeitsraum abermals (siehe 2.4.2)

$$\mathbb{U} \subseteq H^1(\Omega)^d,$$

\mathcal{F} muss aufgrund der Druckpoissongleichung im Raum $H^{1/2}(\Omega)^d$ enthalten sein, wird in dieser Arbeit jedoch auch im diskretisierten \mathbb{U} -Raum dargestellt⁵.

Geschwindigkeitsrandwerte

Bereits in der Themenhinführung wurden die Rand- und Startwerte für Navier-Stokes Probleme erörtert. Neben der Spezifikation der benötigten Werte (Startwerte und Ein- respektive Ausfluss) wurde dabei darauf verwiesen, dass Einfluss und Ausfluss der Divergenzfreiheit, also der Massenerhaltung genügen müssen. Andernfalls erhält man ein unphysikalisch gestelltes Problem. Bei solchen Dirichlet-Randwerten hat der Geschwindigkeitsraum \mathbb{U} auf dem Gebietsrand $\partial\Omega_h$ keine Freiheitsgrade — schließlich ist die Geschwindigkeitsentwicklung hier explizit vorgegeben⁶.

Die Behandlung von Randvorgaben wurde bereits auf Seite 20 dargestellt, wodurch man sich bei der Approximation der Geschwindigkeiten auf $\mathbb{U} \subseteq H_0^1(\Omega)^d$ zurückziehen kann. Da in dieser Arbeit ausschließlich ein abgeschlossenes Gebiet ohne Zufluss betrachtet wird, ergibt sich mit dem äußeren Normalenvektor n auf dem gesamten Rand

$$(u, n)|_{\partial\Omega} = 0 \quad \forall t \geq 0 \quad \Rightarrow \quad (\partial_t u, n)|_{\partial\Omega} = 0 \quad \forall t \geq 0.$$

⁴ n bezeichne hier den äußeren Normalenvektor auf dem Rand $\partial\Omega_h$ des Rechengebietes.

⁵Der dadurch entstehende zusätzliche Diskretisierungsfehler der Nichtlinearität wird hier nicht weiter untersucht.

⁶In dieser Arbeit werden nur abgeschlossene Systeme mit einer no-slip-Bedingung [Gresho 87] verwendet. Weiterführende Aussagen zur Randvektorenwahl im Kontext des hier verwendeten Ansatzes findet sich in [Neckel 05].

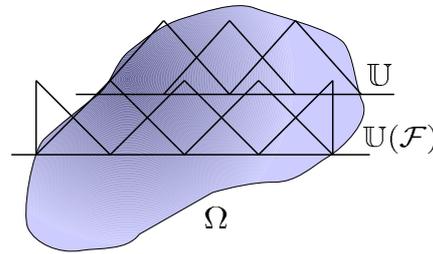


Abbildung 3.1: Der Raum $\mathbb{U}(\mathcal{F})$ hält im Gegensatz zu \mathbb{U} auch Freiheitsgrade auf dem Rand des Gebietes Ω .

Nach [Gresho 87, Gresho 98] ergeben sich mit der Inkompressibilitätsbedingung, die punktweise auch auf dem Rand gelten muss, automatisch die fehlenden passenden Randbedingungen. Insbesondere erweist sich die no-slip-Bedingung als über die Divergenzfreiheit physikalisch sinnvoll motivierte Randbedingung. Ausführliche Informationen zur Randbehandlung für komplexere Modelle sind [Gresho 98] zu entnehmen — im Kontext dieses Projekts bietet sich die kompaktere Darstellung in [Neckel 05] an.

Konvektions- und Diffusionsteil \mathcal{F} sind auf dem Rand direkt über u gegeben — schließlich repräsentiert auch das diskrete \mathcal{F} einen Operator. Später wird sich zeigen, dass dessen Wert mit dem Druckgradienten korrespondieren muss. Bezeichnet man mit $\mathbb{U}(\mathcal{F})$ den Ansatzraum für \mathcal{F} , so liegen folglich zunächst auch Freiheitsgrade auf dem Gebietsrand, also auf dem Abschluss des offenen Gebietes.

3.2.2 Druckpoissongleichung

Der Druckausgleich ist unter Verwendung der Kontrollvolumenäquivalenz und der Druckpoissongleichung über die Kontrollvolumina V formalisiert. Eine Beibehaltung dieser Darstellung ist nur bedingt sinnvoll, stattdessen wird sie mit dem Satz von Gauß auf eine Untermannigfaltigkeit projiziert. Dies ist möglich, falls Geschwindigkeit und Druck fast überall geschlossene Vektorfelder sind⁷:

$$\begin{aligned} \int_V \mathbf{div} u \, dx &= \int_{\partial V} (u, n) \, dx = Mu \quad \forall V \in \Omega_h \\ \int_V \mathbf{div} \nabla p \, dx &= \int_{\partial V} (\nabla p, n) \, dx = M \nabla p \quad \forall V \in \Omega_h \end{aligned}$$

⁷Tatsächlich ist diese Anforderung zu stellen in der Literatur nicht üblich: So argumentiert z.B. [Neckel 05] zur Herleitung des Divergenzoperators ausschließlich über die in [Verstappen 01] geforderte Operatoradjungiertheit und kann so auf diese erweiterte Glattheitsanforderungen in Form der Existenz des Gradienten auf dem Rand verzichten. Dies ist eine mathematische Interpretation des Drucks als Lagrange-Parameter. Als Folge kann er mit zellweise konstantem Druck arbeiten. In dieser Arbeit soll zunächst über die Glattheit im mathematischen Sinn argumentiert werden, die Adjungiertheit muss sich dann später aus der Druckwahl ableiten. Die getroffene Wahl eines Druckelements bringt eine Absicherung gegen die sogenannten Checkerboard-Instabilität automatisch mit sich, da über eine der inf-sup-Bedingung äquivalenten Eindeutigkeitsforderung selbige bereits ausgeschlossen wird, und ist daher in den Augen des Autors einer Argumentation ausschließlich über die Operatoradjungiertheit überlegen.

3 Lösungsalgorithmus

Es ergibt sich damit die Gleichung $M\nabla p = Mu^{(t+\tau)}$. Analog zur Diskussion expliziter Druckverfahren überträgt sich also die Glattheitsanforderung an die Geschwindigkeit auf den Druck und man erhält

$$p \in H^1(\Omega).$$

Will man den Satz von Gauß anwenden, erhält man insbesondere die Definiertheitsforderung von $\nabla p|_{\partial V}$ fast überall. Diese Bedingung ist freilich aufgrund der Integraldarstellung über V nicht nötig, bietet sich aus verschiedenen später diskutierten Gründen jedoch an. Darüber hinaus vereinfacht sich die Behandlung des Randes — existiert der Druckgradient hier nicht, muss ausschließlich über schwache Randbedingungen argumentiert werden und man erhält die gleichen Probleme, wie bei den expliziten Druckverfahren. Wie im Exkurs zu adjungierten Operatoren ausgeführt, kann der Gradientenoperator nun auf den Divergenzoperator zurückgeführt werden und es ergibt sich die diskrete Druckpoissongleichung

$$-M\Omega^{-1}M^T p = Mu^{t+\tau},$$

wobei Ω die Massenmatrix zum diskreten Geschwindigkeitsraum \mathbb{U} ist.

Randwerte

Die Wahl der Randbedingungen ergibt sich wiederum über die Tatsache, dass der Zufluss in Normalenrichtung konstant 0 ist. Man erhält die Neumann-Bedingungen

$$(\partial_t u, n)|_{\partial V} = 0 \quad \Rightarrow \quad (\nabla p, n) = (\mathcal{F}, n)|_{\partial V}$$

für das Szenario der Driven-Cavity.

Bei abgeschlossenen Szenarien ist damit aufgrund fehlender Dirichlet-Ränder der Druck nur bis auf eine Konstante definiert, da ausschließlich die Druckgradienten in die Rechnungen eingehen. Man nimmt daher durch Normierung des Drucks eine Suchraumrestriktion auf $\mathbb{P} \setminus P^0(\Omega)$ vor. Für offene Szenarien ändert sich dies. Da hier ausschließlich Driven-Cavity-Szenarien von Interesse sind, soll diesem Detail keine weitere Beachtung gewidmet werden und es sei abermals auf [Gresho 87, Gresho 98, Neckel 05] verwiesen. Damit hat auch der Druck auf dem Rand Tangentialfreiheitsgrade im Gradient.

3.2.3 Der Cauchy-Schritt

Der Cauchy-Schritt kann ebenfalls in einer schwachen Formulierung vollzogen werden. Dann gilt für den Cauchy-Schritt in schwacher Formulierung

$$\Omega d_t u + C(u)u - \frac{1}{Re} Du - \Omega \Omega^{-1} M^T p = 0,$$

wobei hier die Adjungiertheit der Operatoren bereits eingeflossen ist. Je nach θ -Wahl sind die Vektoren u und p dabei als Linearkombination aus aktuellem und zukünftigen Wert auszuwerten. Ein Sonderfall ergibt sich abermals für ein explizites Zeitschema, wie man direkt Algorithmus 2 entnehmen kann:

Da $\mathcal{F}(u^{(t)})$ bereits explizit vorliegt, macht die schwache Auswertung nur begrenzt Sinn, wenn ∇p fast überall bekannt ist, und man kann die Geschwindigkeit bei einem expliziten Zeitschrittverfahren punktweise fortschalten. Im Programm verwende man also

$$d_t u = -\mathcal{F}(u) - \nabla p$$

Exkurs: Diskretisierungsprinzipien — Symmetrie elliptischer Operatoren

Die FEM für elliptische Probleme mit homogenen Dirichlet-Rand liefert eine Approximation der Lösung zu dem Problem

$$\begin{aligned} - \sum_{l,k=1}^n \partial_l \partial_k u &= b \quad \forall x \in \Omega \\ u|_{\partial\Omega} &= 0 \end{aligned}$$

mit $u \in C^2(\Omega) \cap C^0(\bar{\Omega})$. Aufgrund der Glattheit der Lösung gilt bekanntlich

$$\partial_k \partial_l u = \partial_l \partial_k u.$$

Bringt man nun die Gleichung in eine schwache Formulierung, so kann man mittels der Green'schen Formel die Glattheitsanforderungen reduzieren, erhält somit also

$$\begin{aligned} \int_{\Omega} - \sum_{l,k=1}^n (\partial_l \partial_k u, \varphi) \, dx &= \int_{\Omega} \sum_{l,k=1}^n (\partial_k u, \partial_l \varphi) \, dx \quad \forall \varphi \\ &= \int_{\Omega} \sum_{l,k=1}^n (\partial_l u, \partial_k \varphi) \, dx \quad \forall \varphi. \end{aligned}$$

Die Symmetrie der Steifigkeitsmatrix für den diskreten Fall ist also das Konterfei der Glattheit respektive der Vertauschbarkeit der Ableitungen im Kontinuierlichen.

in den Gitterpunkten. Dieser Zusammenhang wird mit der Diskussion der Druck- und Geschwindigkeitsräume offensichtlich werden. Dabei handelt es sich jedoch um einen angenehmen Zufall — im Allgemeinen ist die konsistente, permanente Verwendung einer schwachen Formulierung aus Konsistenzgründen hier vorzuziehen. In diesem Fall leitet sich aus der Impulsgleichung die reduzierte Glattheitsanforderung $p \in H^0(\Omega)$ nach der FEM ab, falls man den Gradientenoperator auf die Testfunktion abwälzt.

3.3 Physikalische Interpretation des Algorithmus

Mehrfach erwähnt wurde bisher die exakte Erfüllung der Divergenzfreiheit. Es stellt sich folglich die Frage, wie diese Tatsache physikalisch bezüglich des FE Minimierungsprinzips nach Ritz zu interpretieren ist. Betrachtet man die Abbildung 2.4, so zeigt sich, dass man sich in der Lösungsm minimierung ausschließlich auf dem Unterraum bewegt, der durch die Kontinuitätsgleichung definiert ist. Man beschränkt sich aber im Gegensatz zu der in 2.4.1 beschriebenen Methodik nicht auf den Unterraum der divergenzfreien Funktionen, sondern auf einen Unterraum, in dem kontinuierliche und diskrete Divergenzfreiheit fast überall äquivalent sind — stellt also über die Druckpoissonsgleichung sicher, dass die finale Lösung wieder im Unterraum der divergenzfreien Lösungen liegt.

Im Gegensatz zu den gemischten Verfahren erhält man damit eine bezüglich der Inkompressibilität fehlerfreie Lösung — es verbleibt ein Fehler durch die schwache Formulierung der Impulsgleichung. Für diesen ist jedoch mit dem Céa-Lemma eine genaue qualitative und quantitative

Abschätzung bekannt. Da es sich hierbei um ein (für hinreichend kleine Reynoldszahl) elliptisches Problem handelt, kann mit dem Aubin–Nitsche–Trick über das duale Problem sogar eine Fehlerabschätzung in der L^2 -Norm gegeben werden [Braess 02]:

Korollar 3 *Die Verwendung kontinuieritätserhaltender Elemente in Verbindung mit der exakten Erfüllung der diskreten Divergenzfreiheit erlaubt eine qualitative und quantitative Fehlerangabe über die Standard-FEM-Theorie.*

Eine weitere direkte Folge der kontinuieritätserhaltender Elemente ist die in [Verstappen 01] geforderte Erhaltung der Impuls- und Energiesätze. Der Beweis dieser Tatsache erfolgt über die Untersuchung der Gestalt der Konvektions- und Diffusionsmatrizen und wird in [Blanke 04] geführt. Hier soll nicht weiter darauf eingegangen werden.

3.4 Diskrete Funktionenräume

Grundlage der Finite-Elemente-Methode ist die Approximation der Lösung in einem endlich dimensionalen Funktionenraum. Diese Restriktion auf einen endlichen Subraum wird auch als Galerkin-Projektion bezeichnet. In diesem Abschnitt soll der in den vorangegangenen Abschnitten geforderte diskrete Geschwindigkeitsfunktionenraum samt Druckraum für die Navier–Stokes Gleichungen entwickelt werden. Zuvor seien nochmals die bereits beschriebenen Anforderungen zusammen mit den in [Ciarlet 78] definierten FEM Charakteristika an solche Elemente rekapituliert:

- Der Elementträger sei quadratisch und lokal,
- für die durch Translation und Skalierung des Referenzelements entstehende Basis mit zugehörigem Funktionenraum \mathbb{U}_h gelte $\mathbb{U}_h \subset H^1(\Omega)$ und
- die punktweise Divergenzfreiheit sei äquivalent zu der in Lemma 1 postulierten Kontrollvolumenformulierung.

Elemente, die der letzten Forderung genügen, seien in Folge als *kontinuitätserhaltende Finite Elemente* bezeichnet.

3.4.1 Kontinuitätserhaltende Ansatzfunktionen für die Geschwindigkeit

Die Existenz derartiger Funktionenräume ist keineswegs trivial [Gresho 98], da sich Stetigkeit und Divergenzfreiheit über das Kontrollvolumen nicht einfach zusammenbringen lassen. Dennoch existieren sie und eine derartige Basis wird in [Blanke 04] konstruktiv vorgestellt:

Die [Blanke 04]-Elemente

Um die inhaltliche Abgeschlossenheit dieser Arbeit zu wahren, seien die wichtigsten Aspekte hier nochmal kurz dargestellt. Zunächst wird ein Referenzelement für eine nodale Basis [Braess 02] entwickelt. Im Rahmen der d -linearen Ansatzräume wird dieses auch als Hutfunktion bezeichnet. Nun zieht man sich im d -linearen Fall auch bei der Konstruktion vektorwertiger Funktionenräume zunächst auf ein skalares Referenzelement $\hat{\phi} : \Omega \mapsto \mathbb{R}^1$ zurück und leitet daraus durch Nullergänzung d vektorwertige Referenzelemente her⁸:

⁸Hier sei d analog zur Gesamtarbeit gleich der Dimension des Bildraumes und der Dimension des Urbilds des Vektorfeldes.

Exkurs: Diskretisierungsprinzipien — Adungierte Operatoren

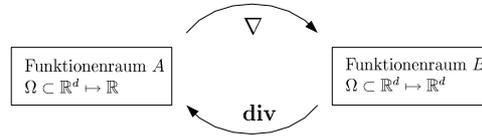


Abbildung 3.2: Gradient- und Divergenzoperator als Funktionenraumübergang. Im Kontinuierlichen gilt bekanntlich die Operatortransponiertheit von Gradient und Divergenz, wie sie auf Seite 3 bereits aufgezeigt worden ist. Diese Tatsache sei nun anhand der in obiger Abbildung definierten Funktionenräume diskutiert.

Der Poissonoperator

Im Diskreten sind die Elemente des Raumes A n -Tupel, wenn n die Zahl der Stützstellen ist. Die Elemente von B sind folglich n -Tupel über d -Tupel. Nach einer vollständigen Linearisierung der Elemente von B können die diskreten Operatoren für ∇ bzw. \mathbf{div} aufgrund ihrer Linearität als $(nd) \times n$ -Matrizen bzw. $n \times (nd)$ -Matrizen dargestellt werden. Für $a \in A$ respektive $b \in B$ erhält man $(\nabla a)_h \in B$ und $(\mathbf{div} b)_h \in A$ und somit im Fall der FEM als Äquivalent zur Dualität von Gradient und Divergenz im Kontinuierlichen

$$\begin{aligned} \Omega_B(\nabla a)_h &= -M^T a \quad \text{und} \\ \Omega_A(\mathbf{div} b)_h &= Mb, \end{aligned}$$

wobei Ω_A und Ω_B die Massenmatrizen in den jeweiligen Ansatzräumen bezeichnen. Hierbei ist für die schwache Formulierung des Gradienten der Divergenzoperator durch partielle Integration auf die Testfunktion abgewälzt worden. Dass die Divergenzdiskretisierung der Transponierten des Gradientenoperators entspricht, ist durch geeignete Wahl, d.h. durch zueinander passende Ansatzräume sicherzustellen: Falls die schwache Formulierung über M einer punktwisen Auswertung äquivalent ist, muss also insbesondere auch die Gradientenrechnung über $\Omega_B^{-1}M^T$ exakt sein. Damit gilt für die schwache Formulierung des Poissonoperators

$$\int_{\Omega} \mathbf{div} \nabla a \chi_V dx \quad \forall V \in \Omega_h \quad \Leftrightarrow \quad (\mathbf{div} \nabla a)_h = -\Omega_A^{-1} M \Omega_B^{-1} M^T a.$$

Da man an $(\mathbf{div} \nabla a)_h$ in nodaler Basis nicht interessiert ist, sondern nur die FEM-Formulierung benötigt, ergibt sich die diskrete Druckpoissongleichung über dem Operator $-M\Omega_B^{-1}M^T$.

Energieerhaltung

Identifiziert man nun B mit \mathbb{U} und kennt $(\nabla p) \in \mathbb{U}$ in der diskreten Darstellung, so ergibt sich

$$\begin{aligned} \nabla p &\xrightarrow{\text{schwache Formulierung}} \int_{\Omega} (\nabla p, \varphi) dx \quad \forall \varphi \in \mathbb{U} \\ &\Leftrightarrow \Omega_{\mathbb{U}}(\nabla p) = -\Omega_{\mathbb{U}}\Omega_{\mathbb{U}}^{-1}M^T p = -M^T p \end{aligned}$$

und die in [Verstappen 01] hergeleiteten Aussagen für die Energieerhaltung.

3 Lösungsalgorithmus

$$\phi_1 := \begin{pmatrix} \hat{\phi} \\ 0 \\ \dots \end{pmatrix} \quad \phi_2 := \begin{pmatrix} 0 \\ \hat{\phi} \\ \dots \end{pmatrix} \quad \dots$$

Dieser Ansatz führt hier nicht direkt zum Ziel, stattdessen sei bereits das Referenzelement eine vektorwertige Funktion $\hat{\phi} = (\hat{\phi}_1, \hat{\phi}_2)^T$ — hier mit Dimension zwei. Sei der Wert des Referenzelements im Mittelpunkt des Trägers für den x_1 -Geschwindigkeitswert $(1, 0)^T$ und verschwinde an den anderen Gitterpunkten für alle Komponenten. Grundidee für die Konstruktion der Form des Referenzelements ist nun, das geometrische Element⁹ in vier gleichschenklige Dreiecke zu partitionieren. Auf diesen Dreiecken wird nun linear interpoliert, wobei der Wert im inneren Punkt zunächst frei wählbar sei, also ein „virtueller“ Freiheitsgrad sei. Eine solche Anordnung der Freiheitsgrade ist in Abbildung 3.3 wiedergegeben.

Der Wert im Elementmittelpunkt ergibt sich nun eindeutig aus der Forderung der Äquivalenz kontinuierlicher und diskreter Divergenzfreiheit. Insbesondere nimmt die erste Komponente dort stets den Wert $\frac{1}{4}$ an, die zweite Komponente alterniert mit $\pm\frac{1}{4}$. Die Werte in der zweiten Komponente — und das ist der Unterschied zu Triviale Tensorproduktansatzfunktionen — verschwinden also nicht. Die dadurch erzeugte Kopplung der beiden Geschwindigkeiten ist wesentlich für die Erfüllung der Äquivalenzforderung¹⁰

Die Konstruktion der d Ansatzfunktionen für die nodale Basis erfolgt nun nicht durch Nullergänzung, sondern über eine Komponentenpermutation mit

$$\phi_1 := \begin{pmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \end{pmatrix} \quad \text{und} \quad \phi_2 := \begin{pmatrix} \hat{\phi}_2 \\ \hat{\phi}_1 \end{pmatrix}.$$

Theorem 3 Für die in [Blanke 04] entwickelten Elemente gilt die Äquivalenz in der Kontrollvolumenformulierung. Daher sind die Elemente als *kontinuitätserhaltend* zu bezeichnen.

Erweiterte kontinuieritätserhaltende Elemente

Die in [Blanke 04] entwickelten Elemente haben sich im Laufe dieser Arbeit durch die Lokalisierung der Geschwindigkeitsfreiheitsgrade in den Ecken der Elemente als anfällig für die bereits eingeführten *spurious modes* erwiesen. Um diesem Verhalten abzuwehren, bedient man sich in der numerischen

⁹Mit geometrisches Element sei stets das Kontrollvolumen V , also ein Gitterelement referenziert. Vier geometrische Elemente ergeben folglich den Träger eines Finiten Elements. Der Begriff Finites Element bezeichnet dagegen geometrisches Element samt den darauf platzierten Ansatzfunktionen.

¹⁰Ergänzend sei ein konstruktiver Ansatz präsentiert: Gelte die Kontrollvolumeneigenschaft über den vier Gitterelementen des Trägers eines geometrischen Elements. Dann ergeben sich aus der punktwisen Divergenzfreiheit für jeden „virtuellen“ Freiheitsgrad vier Bedingungen — eine aus jedem adjazenten Dreieck. In [Blanke 04] wird nun gezeigt, dass je zwei Bedingungen äquivalent sind, und man durch geeignete Wahl des „virtuellen“ Freiheitsgrades mit $\pm\frac{1}{4}$ der Huthöhe die geforderte punktwise Divergenzfreiheit sicherstellen kann. Für das in Abbildung 3.3 farbig hinterlegte geometrische Referenzelement der Seitenlänge h sei hier der Anteil der Ansatzfunktion gegeben, die mit kartesischen Koordinaten im Punkt $(0, 0)^T$ den Wert 1 annimmt:

$$\hat{\phi}_1(x_1, x_2) := \begin{cases} 1 - \frac{x_1}{h} - \frac{x_2}{2h} & \text{falls } x_1 \geq x_2 \wedge x_1 \leq h - x_2 \\ \frac{1}{2} - \frac{x_1}{2h} & \text{falls } x_1 \geq x_2 \wedge x_1 \geq h - x_2 \\ \frac{1}{2} - \frac{x_2}{2h} & \text{falls } x_1 \leq x_2 \wedge x_1 \geq h - x_2 \\ 1 - \frac{x_1}{2h} - \frac{x_2}{h} & \text{falls } x_1 \leq x_2 \wedge x_1 \leq h - x_2 \end{cases}$$

$$\hat{\phi}_2(x_1, x_2) := \begin{cases} \frac{x_2}{2h} & \text{falls } x_1 \geq x_2 \wedge x_1 \leq h - x_2 \\ \frac{1}{2} - \frac{x_1}{2h} & \text{falls } x_1 \geq x_2 \wedge x_1 \geq h - x_2 \\ \frac{1}{2} - \frac{x_2}{2h} & \text{falls } x_1 \leq x_2 \wedge x_1 \geq h - x_2 \\ \frac{x_1}{2h} & \text{falls } x_1 \leq x_2 \wedge x_1 \leq h - x_2 \end{cases}$$

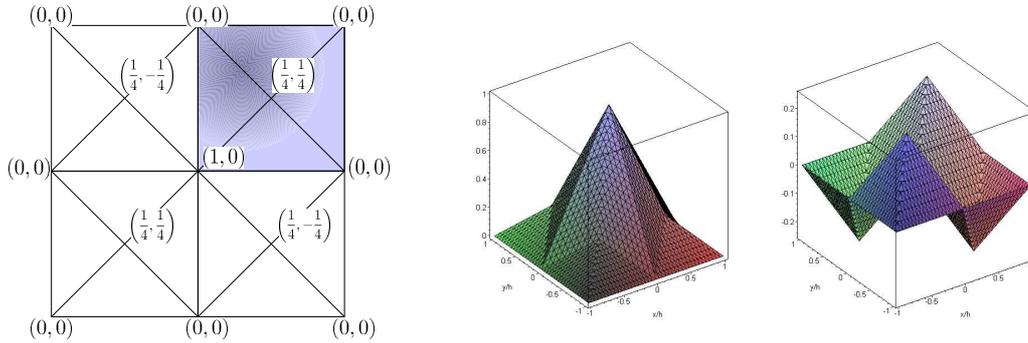


Abbildung 3.3: Das in [Blanke 04] entwickelte Referenzelement.

Strömungsmechanik¹¹ der Technik, (zusätzlich) u_1 -Freiheitsgrade mittig auf den Vertikalkanten und u_2 -Freiheitsgrade auf den Horizontalkanten der Zellen zu platzieren. In dieser Arbeit wurde dieser Ansatz für die Finite-Elemente-Methode nachvollzogen:

Der endgültige Geschwindigkeitsraum ergibt sich dann als Raumaddition des in [Blanke 04] eingeführten Raumes und der erweiterten Komponenten. Abbildung 3.4 definiert die Nomenklatur der zugehörigen Basisfunktionen. Resultat ist aufgrund der additiven Funktionenraumkonstruktion eine quasi-hierarchische Darstellung mit den in [Blanke 04] entwickelten Elementen und hierarchischen Normalenüberschüssen auf den Kanten. Man kann nachrechnen, dass diese Basis inhärent hierarchisch ist, also keine nodale Interpretation gefunden werden kann. \perp bezeichne dabei nach [Broy 98] undefinierte Überschüsse:

Für die Konstruktion des hierarchischen Überschusses nehme man in jeder Raumrichtung eine Bipartition des geometrischen Referenzelements vor. Dabei führe man — im Zweidimensionalen — wiederum einen erweiterten „virtuellen“ Freiheitsgrad in der Zellmitte ein. Der Wert dort sei als $u(M)$ bezeichnet. Auf den resultierenden vier Subvolumina platziere man wiederum die in [Blanke 04] eingeführten Elemente. Diese erweiterten Elemente seien in dieser Arbeit als *9-Punkte-Elemente* bezeichnet, während die in [Blanke 04] vorgestellten ab jetzt *5-Punkte-Elemente* heißen sollen. Diese Nomenklatur wird im Kontext des Druckausgleiches klar werden. Alle nicht angesprochenen weiteren Ansatzstellen seien 0 in beiden respektive in den nicht definierten Komponenten.

Theorem 4 *Definiert man für das 9-Punkte-Element den „virtuellen“ Freiheitsgrad des hierarchischen Überschusses mit*

$$\hat{u}(M) = \left(\frac{1}{2}(\hat{u}_1(W) + \hat{u}_1(O)), \frac{1}{2}(\hat{u}_2(N) + \hat{u}_2(S)) \right)^T,$$

so ist das Element kontinuieritätserhaltend. Dabei liege die in Abbildung 3.4 gewählte Anordnung der Freiheitsgrade zugrunde, wobei alle hierarchischen Überschüsse mit \hat{u} bezeichnet seien.

Der Beweis beschränkt sich auf einfaches Nachrechnen: Analog zu [Blanke 04] werden nun zunächst für die erweiterte Menge von Freiheitsgraden Interpolationsregeln für die Geschwindigkeiten hergeleitet, so dass gerade die postulierte Äquivalenz von kontinuierlicher und diskreter Divergenzfreiheit fast überall gilt. Dabei bezeichne u den tatsächlichen Wert, \hat{u} den hierarchischen

¹¹Man denke an [Emans 03, Griebel 95] und die Finite-Volumen-Methode.

3 Lösungsalgorithmus

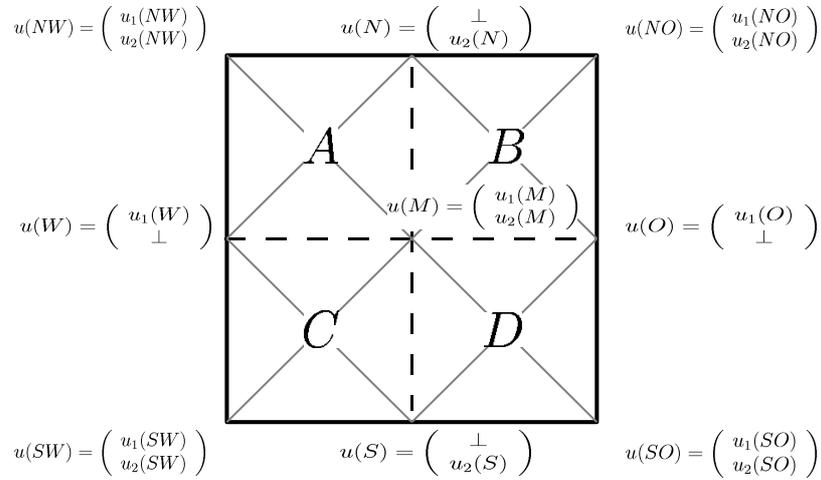


Abbildung 3.4: Die Augmentierung des in [Blanke 04] entwickelten Raumes um hierarchische Kantenfreiheitsgrade. Der Punkt $u(M)$ ist analog zur konstruktiven Herleitung der 5-Punkte-Elemente ein „virtueller“ hierarchischer Freiheitsgrad.

Überschuss, der durch Überschüsse auf den Elementkanten erzeugt wird. Damit hat die Funktion auf dem Mittelpunkt eines geometrischen Elements nach [Blanke 04] den Wert

$$\begin{aligned}
 \hat{u}_1(W) &= u_1(W) - \frac{1}{2}(u_1(NW) + u_1(SW)) \\
 \hat{u}_2(N) &= u_2(N) - \frac{1}{2}(u_2(NW) + u_2(NO)) \\
 &\dots \\
 u_1(M) &= \frac{1}{4}(u_1(NW) + u_1(NO) + u_1(SW) + u_1(SO)) \\
 &\quad + \frac{1}{4}(-u_2(NW) + u_2(NO) + u_2(SW) - u_2(SO)) \quad \text{siehe [Blanke 04]} \\
 &\quad + \hat{u}_1(M) \\
 &= \frac{1}{2}(u_1(O) + u_1(W)) + \frac{1}{4}(-u_2(NW) + u_2(NO) + u_2(SW) - u_2(SO)) \\
 u_2(M) &= \dots + \hat{u}_2(M) \\
 &= \frac{1}{2}(u_2(N) + u_2(S)) + \frac{1}{4}(u_1(SW) - u_1(SO) + u_1(NO) - u_1(NW)).
 \end{aligned}$$

Obige Herleitung ist historisch bedingt. Tatsächlich bietet erleichtert eine permanente Behandlung mittels dehierarchisierter Werte den Umgang mit den Elementen¹². Abermals führe man nun über den Gaußschen Integralsatz die Kontinuitätsbedingung auf den Rand des Kontrollvolumens zurück und erhält mit einer Bezeichnung gemäß den Himmelsrichtungen

¹²Man müsste zwischen einer einfachen dehierarchisierten Darstellung und der komplexeren, nodal orientierten Sichtweise in einer neuerlichen schriftlichen Darstellung sicher abwägen.

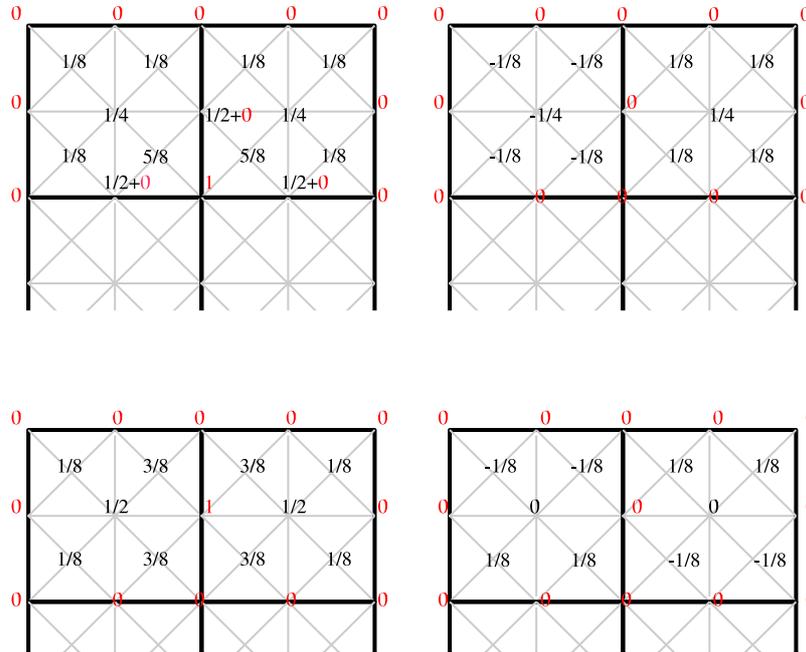


Abbildung 3.5: Teile des Referenzelements zur augmentierten [Blanke 04]–Basis. Oben eine Ansatzfunktion zur nodalen Basis, unten eine zum hierarchischen Überschuss. Die roten Werte bezeichnen die Höhe der Ansatzfunktionen respektive des Überschusses, die schwarzen Werte die Interpolante. Durch Spiegelung und passende Rotation erhält man die restlichen Formfunktionen auf dem Element. Auf den Dreiecken verwende man lineare Interpolation.

$$\begin{aligned}
 0 &= \int_V \mathbf{div} u \, dx = \int_{\partial V} (u, n) \, dx \\
 \Leftrightarrow 0 &= \frac{h}{4} (u_1(NO) + 2u_1(O) + u_1(SO) - u_1(NW) - 2u_1(W) - u_1(SW)) \\
 &\quad + \frac{h}{4} (u_2(NW) + 2u_2(N) + u_2(NO) - u_2(SW) - 2u_2(S) - u_2(SW)).
 \end{aligned}$$

Im Folgenden ziehe man sich auf die Subelemente A, B, C und D zurück und zeige, dass auf diesen die diskrete Kontinuitätsbedingung ebenfalls gilt. Da die Elemente auf den Subelementen direkt aus [Blanke 04] entnommen und damit kontinuieritätserhaltend sind, impliziert dies die punktweise Divergenzfreiheit und es verbleibt nichts mehr nachzurechnen.

- Element A :

$$\int_A \mathbf{div} u \, dx = \int_{\partial A} (u, n) \, dx = 0$$

3 Lösungsalgorithmus

$$\begin{aligned}
 &\Leftrightarrow \underbrace{(u_2(NW) + u_2(N))}_{\text{obere Kante}} + \underbrace{(u_1(N) + u_1(M))}_{\text{rechte Kante}} - \underbrace{(u_2(M) + u_2(W))}_{\text{untere Kante}} - \underbrace{(u_1(W) + u_1(NW))}_{\text{linke Kante}} \\
 &= u_2(NW) + u_2(N) + u_1(N) + u_1(M) - u_2(M) - u_2(W) - u_1(W) - u_1(NW) \\
 &= u_2(NW) + u_2(N) + \underbrace{u_1(N)}_{= \frac{u_1(NW) + u_1(NO)}{2}} + u_1(M) \\
 &\quad - u_2(M) - \underbrace{u_2(W)}_{= \frac{u_2(NW) + u_2(SW)}{2}} - u_1(W) - u_1(NW) \\
 &\dots \\
 &= 0
 \end{aligned}$$

- Element B :

$$\begin{aligned}
 &\int_B \mathbf{div} u \, dx = \int_{\partial B} (u, n) \, dx = 0 \\
 &\Leftrightarrow \underbrace{(u_2(N) + u_2(NO))}_{\text{obere Kante}} + \underbrace{(u_1(NO) + u_1(O))}_{\text{rechte Kante}} - \underbrace{(u_2(O) + u_2(M))}_{\text{untere Kante}} - \underbrace{(u_1(M) + u_1(N))}_{\text{linke Kante}} \\
 &\dots \\
 &= 0
 \end{aligned}$$

- Element C :

$$\begin{aligned}
 &\int_C \mathbf{div} u \, dx = \int_{\partial C} (u, n) \, dx = 0 \\
 &\Leftrightarrow \underbrace{(u_2(M) + u_2(W))}_{\text{obere Kante}} + \underbrace{(u_1(M) + u_1(S))}_{\text{rechte Kante}} - \underbrace{(u_2(SW) + u_2(S))}_{\text{untere Kante}} - \underbrace{(u_1(SW) + u_1(W))}_{\text{linke Kante}} \\
 &\dots \\
 &= 0
 \end{aligned}$$

- Element D :

$$\begin{aligned}
 &\int_D \mathbf{div} u \, dx = \int_{\partial D} (u, n) \, dx = 0 \\
 &\Leftrightarrow \underbrace{(u_2(M) + u_2(O))}_{\text{obere Kante}} + \underbrace{(u_1(O) + u_1(SO))}_{\text{rechte Kante}} - \underbrace{(u_2(SO) + u_2(S))}_{\text{untere Kante}} - \underbrace{(u_1(S) + u_1(M))}_{\text{linke Kante}} \\
 &\dots \\
 &= 0
 \end{aligned}$$

3.4.2 Konstruktionsprinzipien für den Druckraum

Analog zur Konstruktion der Geschwindigkeitselemente seien nun zuerst Anforderungen an die Druckelemente rekapituliert respektive explizit herausgestellt:

- Der Druck genüge den bereits postulierten Glattheitsanforderungen, also insbesondere $p \in H^1(\Omega)$.
- Desweiteren fordere man analog zu den gemischten Verfahren, dass sich auf regulären Gittern keine *spurious modes* konstruieren lassen. Für die gemischten Methoden wurde diese Eigenschaft über die inf-sup-Bedingung gefordert, hier ist sie explizit zu nennen.

- In [Verstappen 01] wird mehrfach die Transponiertheit der Operatoren ins Feld geführt. Diese ist durch passende Wahl der Ansatzräume sicherzustellen und
- abschließend soll der Druckraum durch einen Tensorproduktansatz konstruiert werden können. Dies ermöglicht die einfache Übertragung des Algorithmus in den \mathbb{R}^3 .

Die Transponiertheit der diskreten Operatoren \mathbf{div} und ∇ impliziert automatisch die Energieerhaltung [Verstappen 01], wobei man nach den Ausführungen auf Seite 37 den Druckgradientenraum in \mathbb{U} darstellt. In dem hier entwickeltem Algorithmus wird die Divergenzfreiheit der Geschwindigkeit durch den Druck(gradienten) sichergestellt. Dabei wurde mit Hilfe kontinuieritätserhaltender Elemente die kontinuierliche Divergenzfreiheit auf eine zellweise Formulierung gebracht. Man erhielt $|\Omega_h|$ Bedingungen, die es zu erfüllen gilt. Aufgrund der Eindeutigkeit des Drucks ergibt sich also, dass für diesen im diskreten Fall global die Anzahl der Freiheitsgrade gleich der Anzahl der geometrischen Elemente sein muss, also für jedes Element ein Druckfreiheitsgrad existieren muss. Es gilt somit

$$\#\text{Freiheitsgrade } p = |\Omega_h|.$$

In [Weinzierl 05] wird dies bereits als ausreichende Motivierung einer zellbasierten Druckdiskretisierung bezeichnet. Man kann diese Tatsache jedoch noch genauer herleiten.

Glattheitsdiskussion

Zuvor seien noch einige weiterführende Anmerkungen gemacht: Die Forderung $p \in H^1(\Omega)$ kann getrost vernachlässigt werden, falls man ausschließlich über die diskreten Operatoren und normale Relaxationsverfahren argumentiert. Für Galerkin–Mehrgitterverfahren ist dies nicht ausreichend, weil sich Interpolations– und Restriktionsvorschriften ohne diese Forderung nicht ableiten lassen. Da die Divergenzfreiheit exakt erfüllt wird, wäre analog eine exakte Gradientendarstellung wünschenswert, also $\nabla p \in \mathbb{U} \quad \forall p \in \mathbb{P}$. Obwohl solch ein Funktionenraum wohl existiert, ist es uns nicht gelungen, einen derartigen praktikablen Raum zu konstruieren.

Verblieben sind zwei Ansätze: Bei Verwendung eines zellweise konstanten Drucks verwende man die Transponierte des Divergenzoperators M aus [Blanke 04], um den „Druckgradienten“ zu berechnen, wenngleich selbiger in einer Funktionenrauminterpretation nicht existiert. Man erhält einen 5–Punkte–Stern für den Druck. In diesem Fall ist der Druck rein im mathematischen Sinn als Lagrange–Parameter zu interpretieren. Die Entwicklung eines Mehrgitterverfahrens mit dieser Diskretisierung wurde nicht weiter verfolgt.

Der Fokus dieser Arbeit liegt auf der Verwendung versetzter (siehe unten) bilinearer Elemente. Hierbei ergibt sich der punktweise Gradient auf den Gitterlinien genau als Transponierte des Divergenzoperators, wenn man den 5–Sterne–Elemente–Raum nach 3.4.1 augmentiert. Dadurch erhält man einen 9–Punkte–Stern über den Druckfreiheitsgraden. Da es Ziel dieser Arbeit ist, eine praktikable und effiziente Implementierung zu diskutieren, soll es bei dieser Darstellung auch belassen werden. Eine Interpretation dieses Ansatzes im Finite–Element Sinn steht also noch aus.

Versetzte Gitter

Nach den Sobolovschen Einbettungssätzen impliziert $p \in H^1(\Omega) \subset C(\Omega)$ die Stetigkeit des Drucks¹³. Für Rechtecksgitter bietet sich die Arbeit mit bilinearen Elementen an. In diesem Zusammenhang existieren verschiedene Möglichkeiten, über die Platzierung des Druckfreiheitsgrades zu entscheiden. Sitzt der Druck wie die Geschwindigkeit auf den Gitterpunkten, so spricht man in diesem Zusammenhang von einem *collocated grid*. Platziert man den Druck in der Zellmitte, so wird das

¹³Hierzu konsultiere man eine Einführung in die Numerik elliptischer Probleme wie [Braess 02], oder gleich eine Abhandlung aus der Funktionalanalysis [Werner 02].

3 Lösungsalgorithmus

Schema als *partially staggered* bezeichnet. Als (*fully*) *staggered* bezeichnet man Gitter, mit in der Zelle platziertem Druckwert und auf den Kanten platzierten Normalenanteil der Geschwindigkeiten. Diese spielen im Kontext von Finiten Volumen Methoden eine große Rolle, im Kontext der FEM sind sie aufgrund der vorhandenen Unstetigkeiten ein Beispiel für nichtkonforme Elemente und sollen hier keine Rolle spielen [Blanke 04]. Abbildung 3.6 zeigt die drei angerissenen Gittertypen, [Ferziger 99] diskutiert die Qualitätseigenschaften der Alternativen ausführlich. Im Folgenden wird mit versetztem Gitter respektive *staggered grid* immer das druckversetzte Gitter, also das *partially staggered grid* bezeichnet.

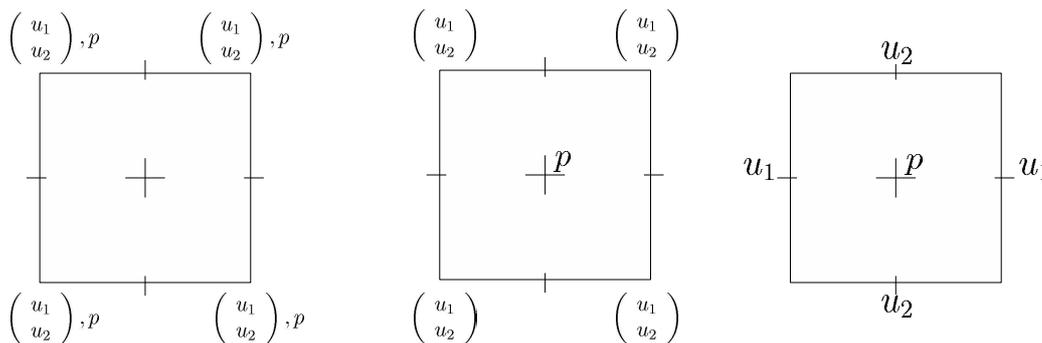


Abbildung 3.6: *Collocated, partially staggered* und (*fully*) *staggered grids*.

Interessant ist nun der in [Verstappen 01] gezeigte Zusammenhang zwischen *collocated grids* und bezüglich des Polynomgrads minimalen Druckelementen: Aufgrund der Symmetrieeigenschaften der Basisfunktionen des Geschwindigkeitsraumes entspricht die Diskretisierung des Druckgradienten einem *centered-scheme*. Der Wert des Druckgradienten wird also ausschließlich durch den Druckwert der Nachbarn definiert. Damit erhält man analog zu den gemischten Verfahren ein einmal reduzibles Druckgleichungssystem, kann also *spurious modes* konstruieren. Dies erfolgt analog zu dem in 2.4.2 erbrachten Nachweis der Instabilität des Q^1P^0 -Elements. Damit entspricht ein derartiges Gitter nicht den hier postulierten Anforderungen. Ein versetztes Gitter schafft hier Abhilfe, zudem spricht die zellweise Definition des Divergenzoperators, der eine Abbildung des Geschwindigkeitsraumes in den Druckraum darstellt, gegen eine Zuordnung des Druckes zu den Zellecken.

Als praktischer „Nebeneffekt“ einer versetzten Anordnung der Freiheitsgrade im stetigen Kontext erweist sich, dass der Gradient auf den Gitterkanten fast überall exakt berechenbar ist. Der Divergenzoperator kann also analog zum Divergenzoperator für die Geschwindigkeit mit dem Satz von Gauß auf die Untermannigfaltigkeiten zurückgeführt werden und man erhält schließlich den diskreten Gradientenoperator

$$\nabla_h = \Omega^{-1}M^T$$

mit der Massenmatrix Ω des Geschwindigkeitsraumes und dem diskreten Divergenzoperator M .

Stichwortsammlung

Kontinuitätserhaltende Elemente, Kontrollvolumenformulierung; Algorithmus, Präparations-schritt; Suchraumrestriktion, schwache Impulsformulierung, Divergenz-/Konvektionsterm

Exkurs: Diskretisierungsprinzipien — Symmetrie der Skalarprodukte

In [Blanke 04, Emans 03, Verstappen 01] wird die Energieerhaltung im Diskreten im Raum auf Anforderungen an die Diskretisierungsmatrizen abgebildet. Grundlage hierfür ist der physikalische Erhaltungssatz für die Energie \mathbb{E}

$$\partial_t \mathbb{E} = \partial_t \rho \int |u|^2 dx \leq 0.$$

Eine Ersetzung des \leq durch die Gleichheit entspricht dem Ignorieren der Reibung. Wegen der hinreichenden Glattheit der Lösung ist das Vertauschen von Ableitung und Integration erlaubt und man erhält für die Energieerhaltung eine Bedingung über das L^2 -Skalarprodukt:

$$\partial_t |u|^2 = (\partial_t u, u)_{L^2} + (u, \partial_t u)_{L^2} \leq 0$$

Im Kontinuierlichen verwende man nun die Symmetrie des Skalarprodukts und es reicht, sich auf $(\partial_t u, u)$ oder $(u, \partial_t u)$ zu beschränken. Im Diskreten ist diese Vereinfachung zu einschränkend, stattdessen sind die Aussagen über die Matrixsummen

$$D + D^T, \quad C(u) + C^T(u) \quad \text{und} \quad p^T M u + u^T M^T p$$

zu machen:

$$\begin{aligned} (\partial_t u, u)_{L^2} + (u, \partial_t u)_{L^2} &\xrightarrow{\text{Diskretisierung}} (d_t u, \Omega u) + (u, \Omega d_t u) \\ &= (d_t u)^T \Omega u + u^T \Omega (d_t u) \\ &= u^T \Omega^T (d_t u) + u^T \Omega (d_t u) \end{aligned}$$

In diese Formel setze man nun die semidiskrete Formulierung für $\Omega(\partial_t u)$ ein und erhält die in [Verstappen 01] postulierten (Un-)Gleichungen. Man beachte, dass für die Matrix D durchaus eine Symmetrieforderung besteht. Diese motiviert sich aber aus der Glattheit der Lösung und der Vertauschbarkeit der Ableitungen (siehe Seite 35), nicht aus der Symmetrie des Skalarproduktes.

Somit ergibt sich abschließend für die Energiebilanz des diskreten Schemas mit der Semidiskretisierung

$$\Omega(d_t u) + C(u)u + Du - M^T p = 0$$

für die Impulsgleichung der folgende Zusammenhang:

$$\partial_t (u^T \Omega u) = -u^T (C(u) + C^T(u)) - u^T (D + D^T) u + \underbrace{p^T M u}_{=0 \text{ über Nebenbedingung}} + (M^T p)^T u$$

Auch die Impulsargumentation muss so erfolgen. Tatsächlich ist obige Zeile abermals eine Motivation für die Transponiertheit von Gradient- und Divergenzoperator: Da der Druck keine Energie „produziert“ muss der entsprechende Term in der diskreten Energieformulierung verschwinden. Dies erreicht man nur über eine passende Transponiertheit.

3 Lösungsalgorithmus

\mathcal{F} , Druckpoissongleichung; Cauchy-Schritt; Reduktion der nichtlinearen Terme auf Auswertung; Versetzte Gitter, Druckstabilität, inf-sup-Bedingung; physikalische Erhaltungssätze; 5-Punkte-Element, augmentiertes 9-Punkte-Element

4 Diskrete Operatoren

Kapitel 3 hat einen Algorithmus mit exakter Erfüllung der Kontinuitätsgleichung zur Lösung der Navier–Stokes Gleichungen definiert. Die Gebietsdiskretisierung wurde dabei noch nicht explizit vorgenommen, jedoch wurde bereits festgelegt, dass der Träger der Geschwindigkeitsfunktionen eine Quadratkomposition von vier quadratischen geometrischen Elementen ist. Desweiteren wurden die beteiligten, kontinuieritätserhaltenden Ansatzfunktionen bereits präsentiert.

In der FEM und der hier entwickelten Kontrollvolumenformulierung wird nun die Lösung einer Differentialgleichung in einer schwachen Formulierung über einer Linearkombination der Ansatzfunktionen errechnet. Die Einträge der daraus resultierenden Operatormatrizen werden also durch die Integrale auf den beteiligten Ansatzfunktionen errechnet. Da Integrale über den Träger ihres Integranden zerfallen, sind die beteiligten Matrizen also bereits vollständig über die Angabe der Matrizenbeiträge auf einem Referenzelement definiert: Die Kontribution eines jeden geometrischen Elements des Trägers einer Ansatzfunktion kann separat integriert werden. Man erhält so elementweise definierte Differenzensterne und der Gesamtoperator ergibt sich durch Akkumulation der Elementanteile in jedem Gitterpunkt. Dieser Ansatz entspricht der Technik der *elementweisen Assemblierung* [Braess 02]. In dieser Arbeit ist die Elementkontribution zu einer Diskretisierungsmatrix A als A_{local} notiert.

Wie die Lösung solcher implizit gegebenen linearen Gleichungssysteme im Kontext dieses Projekts zu bewerkstellern ist, wird später analog zu [Günther 04, Pögl 04] nochmals aufgezeigt. Dieses Kapitel definiert die beteiligten Operatoren, wobei auch Aspekte wie Mehrgitterverfahren oder Relaxationsvarianten noch außen vor bleiben.

Natürlich macht es wenig Sinn, die beteiligten lokalen Matrizen händisch zu berechnen. Stattdessen drängt sich geradezu die Verwendung einer Computeralgebra auf, wie es ab Anhang A vorexerziert wird. Im Anhang finden sich auch die verwendeten lokalen Assemblierungsmatrizen zu Massen-, Diffusions- und Konvektionsoperatoren. Dabei kommt nicht der augmentierte Geschwindigkeitsraum zur Verwendung, stattdessen ziehe man sich auf die in [Blanke 04] definierten Funktionenräume zurück. Der durch die Subraumrestriktion in der Impulsgleichung entstehende zusätzliche Fehler sei hier nicht Gegenstand der Untersuchung, während die Erweiterung der gegenwärtigen Implementierung um die augmentierten Elemente eine reine CAS-Programmieraufgabe und damit auch keine konzeptionelle Herausforderung darstellt. Eine „korrekte“ Behandlung der Impulsgleichung mit augmentierten Geschwindigkeitselementen ist Gegenstand zukünftiger Arbeit, die hier nicht mehr geleistet werden konnte¹. Verwendet werden die augmentierten Elemente dagegen bei der Diskretisierung der Kontinuitätsgleichung, also des Divergenzoperators.

Diese Arbeit konzentriert sich auf die Entwicklung eines effizienten Verfahrens für die Druckpoissongleichung. Daher fokussiert sich dieses Kapitel auch auf diese Gleichung und rekapituliert andere Ergebnisse nur, sofern es für das Verständnis unerlässlich ist. Aufgrund der komplexen Differentialoperatoren werden jedoch die schwachen Formen für alle beteiligten Terme explizit hergeleitet. Nur so bleibt die Möglichkeit erhalten, Eigenschaften und Erzeugung der Diskretisierung im Anhang nachzuvollziehen. Somit erhielt eine durchgehende Beleuchtung von der Konzeption des Programms bis zur Umsetzung des Algorithmus den Vorzug vor einer exklusiven kompakten

¹Die hierarchischen Überschüsse auf den Kanten seien daher auf 0 gesetzt — der Funktionswert auf den Kantenmittelpunkten ergibt sich folglich aufgrund der linearen Interpolation auf den Elementkanten ausschließlich durch Mittelung.

Exkurs: Nummerierungen — Teil I

In dieser Arbeit werden verschiedene Nummerierungen der Freiheitsgrade auf einer Zelle verwendet: Die *lexikographische Nummerierung*, die die Freiheitsgrade in Schreibrichtung aufzählt, die *Kurvennummerierung*, die *kartesische Nummerierung*, der ein kartesisches Koordinatensystem im linken unteren Punkt zugrunde liegt, und schließlich eine Nummerierung *entgegen dem Uhrzeigersinn*. Alternativ kommt auch eine Bezeichnung mittels der Himmelsrichtung zur Verwendung. Abbildung 4.1 zeigt die diese Alternativen — in der gerade definierten lexikographischen Anordnung.

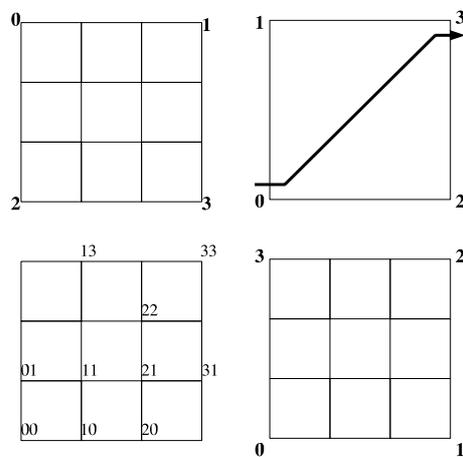


Abbildung 4.1: Die verschiedenen verwendeten Nummerierungen.

Lexikographische Nummerierung

Die lexikographische Nummerierung wird zur Darstellung der Diskretisierungsmatrizen innerhalb der Implementierung verwendet. In mancher Literatur [Gresho 98] kommt hierzu auch die Nummerierung entgegen dem Uhrzeigersinn zur Verwendung, welche hier der Vollständigkeit halber aufgeführt ist.

Kartesische Nummerierung

Die kartesische Nummerierung eignet sich zur Formulierung von Restriktion und Interpolation. Dabei sitzen dann die Grobgitteransatzfunktionen auf den Punkten (00), (30), (03) und (33). Bei hierarchischen Erzeugendensystemen auf Zweigitter liegen damit hier je zwei Freiheitsgrade, während auf den Feingitterknoten (ab) $a, b \in \{1, 2\}$ sowohl für nodale als auch hierarchische Erzeugendensysteme je ein Freiheitsgrad liegt.

Exkurs: Nummerierungen — Teil II

Kurvennummerierung

Die Nummerierung entlang der Peano-Kurve bewährt sich bei Formalisierung von Punktfärbung, Cache-Verwaltung und Durchlaufrichtungen. Dabei erhält nach [Günther 04] im zweidimensionalen Fall der Eintrittspunkt der Peano-Kurve stets den Index 0, dem folgt der linke Freiheitsgrad. Dem zur Kurve rechtsliegenden Punkt weist man den Index 2 zu, um dem Austrittspunkt der Zelle mit 3 zu bezeichnen. Man beachte, dass infolge eines Zellenwechsels dabei der Index 3 stets zum Index 0 wird. Der ursprüngliche Punkt 0 wird in der nächsten Zelle nicht abermals benötigt. In mathematischen Herleitungen spielt diese Darstellung keine Rolle. Eine dreidimensionale Variante definiert [Pögl 04], [Hartmann 04] entwickelt ein dimensionsgenerisches Nummerierungskonzept.

Darstellung der Druckpoissongleichung unter Verwendung einer Vielzahl weiterer Ergebnisse aus einschlägiger Literatur. Dem ausschließlich an der Druckgleichung interessierten Leser sei dennoch ein Sprung zu Abschnitt 4.2 empfohlen. Ist man jedoch an der elementweisen Herleitung der Operatoren interessiert, so sei darauf hingewiesen, dass bei der elementweisen Darstellung stets die lexikographische Nummerierung² zugrunde liegt.

4.1 Präparationsschritt

Zunächst sei die Zeitdiskretisierung außer Acht gelassen: Im Präparationsschritt berechnet man $\mathcal{F}(u) = (u\nabla)u - \frac{1}{Re}\Delta u$, indem man die Gleichung in die schwache Formulierung

$$\int_{\Omega} \mathcal{F}(u^{(t)})\varphi \, dx = \int_{\Omega} \left((u\nabla)u - \frac{1}{Re}\Delta u \right) \varphi \, dx$$

nach 3.2 bringt. Wie bei Ritz-Verfahren üblich, fungiert hierbei der selbe Geschwindigkeitsraum \mathbb{U} als Test- und Ansatzfunktionenraum. In die Druckpoissongleichung geht \mathcal{F} als rechte Seite ein, im Cauchy-Schritt wird \mathcal{F} dann ein zweites Mal benutzt, wobei nochmals erwähnt sei, dass \mathcal{F} in \mathbb{U} approximiert wird. Die drei diskreten Matrizen werden als Massenmatrix, Konvektions- und Diffusionsmatrix (von links nach rechts) bezeichnet und analog zu [Blanke 04] als Ω , $C(u)$ und D notiert.

Die Operatoren wurden in dieser Arbeit nicht erneut berechnet, da sie — wie bereits erwähnt — [Blanke 04] oder alternativ dem Anhang zu entnehmen sind. Neu ist in diesem Kontext lediglich die konsequente Verwendung einer Computeralgebra als ein CASE-Werkzeug (*Computer Aided Software Engineering*) [Chibisov 02, Chibisov 03, Ganzha 01/2].

²Eine Definition aller benutzten Nummerierungsarten findet sich auf Seite 48.

4.1.1 Diffusionsterm

Der Diffusionsterm ist unspektakulär und entspricht der vektoriellen Poisson-Gleichung. Hierbei seien $\varphi(k)$ die vektoriellen Ansatzfunktionen³:

$$\begin{aligned}
 \int_{\Omega} \Delta u \varphi(j) dx &= \int_{\Omega} \begin{pmatrix} \Delta u_1 \\ \Delta u_2 \end{pmatrix} \begin{pmatrix} \varphi_1(j) \\ \varphi_2(j) \end{pmatrix} dx \quad \forall j \\
 &= - \int_{\Omega} \nabla u_1 \nabla \varphi_1(j) + \nabla u_2 \nabla \varphi_2(j) dx \\
 &= - \sum_i u(i) \int_{\Omega} \nabla \varphi_1(i) \nabla \varphi_1(j) + \nabla \varphi_2(i) \nabla \varphi_2(j) dx \\
 &= - \sum_i u(i) \int_{\Omega} \partial_{x_1} \varphi_1(i) \partial_{x_1} \varphi_1(j) + \partial_{x_2} \varphi_1(i) \partial_{x_2} \varphi_1(j) \\
 &\quad + \partial_{x_1} \varphi_2(i) \partial_{x_1} \varphi_2(j) + \partial_{x_2} \varphi_2(i) \partial_{x_2} \varphi_2(j) dx \\
 &=: Du
 \end{aligned}$$

Man betrachte nun gleich die in der Implementierung stets kritische Skalierung der diskreten Operatoren: Die lokalen Assemblierungsmatrizen zur Poissongleichung sind im zweidimensionalen Fall mit 1 zu skalieren, da das Trägerintegral für reguläre Gitter einen Faktor h^2 liefert, der durch das Quadrat des Faktors h^{-1} des Gradienten eliminiert wird.

Deutlich zu sehen ist die auf Seite 35 schon thematisierte Symmetrie der Matrix. Nicht offensichtlich ist hingegen, dass sich für die Kontinuitätserhaltenden Elemente der 5-Punkte-Stern aus der Finite-Differenzen-Methode in jeder Komponente ergibt und x_1 und x_2 -Komponente vollständig entkoppelt sind [Blanke 04]. Dies ist aufgrund der Kopplung der Komponenten für Kontinuitätserhaltende Elemente überraschend, entspricht aber den in 1.2.1 Eigenschaften der Stokes Gleichung.

4.1.2 Konvektionsterm

$$\begin{aligned}
 \int_{\Omega} (u \nabla) u \varphi(j) dx &= \int_{\Omega} \begin{pmatrix} u \nabla u_1 \\ u \nabla u_2 \end{pmatrix} \begin{pmatrix} \varphi_1(j) \\ \varphi_2(j) \end{pmatrix} dx \quad \forall j \\
 &= \int_{\Omega} (u, \nabla u_1) \varphi_1(j) + (u, \nabla u_2) \varphi_2(j) dx \\
 &= \sum_{l,k} u(l) u(k) \int_{\Omega} (\varphi(l), \nabla \varphi_1(k)) \varphi_1(j) + (\varphi(l), \nabla \varphi_2(k)) \varphi_2(j) dx \\
 &= \sum_{l,k} u(l) u(k) \int_{\Omega} \varphi_1(l) \partial_{x_1} \varphi_1(k) \varphi_1(j) + \varphi_2(l) \partial_{x_2} \varphi_1(k) \varphi_1(j) \\
 &\quad + \varphi_1(l) \partial_{x_1} \varphi_2(k) \varphi_2(j) + \varphi_2(l) \partial_{x_2} \varphi_2(k) \varphi_2(j) \\
 &=: C(u)u
 \end{aligned}$$

Die lokale Assemblierungsmatrix des Konvektionsterms erhält eine Skalierung mit h im zweidimensionalen Fall. Dies resultiert aus dem Faktor h^2 des Trägerintegrals, der mit h^{-1} aus dem Gradienten verrechnet wird. Nach [Blanke 04] erhält man mit den Kontinuitätserhaltenden Ansatzfunktionen eine antisymmetrische Matrix mit Kopplung der u_1 - und u_2 -Komponente. Dies wiederum sichert nach [Verstappen 01] die Erhaltung der diskreten Energie zu.

³Man beachte, dass Vektorskalarprodukte je nach Lesbarkeit manchmal als einfache Produkte, oftmals jedoch als (\cdot, \cdot) notiert werden.

Tatsächlich erfolgt die Realisierung des nichtlinearen Terms im Programm auch über Matrizen: Dazu halte man sich vor Augen, dass obige lokale Assemblierungsvorschrift lediglich acht Bilinearformen über den Eingabezweipeln $u(1), u(2), u(3)$ und $u(4)$ definiert. Diese werden im Anhang A dargestellt.

4.1.3 Massenmatrix

$$\begin{aligned} \int_{\Omega} u \varphi(j) \, dx &= \sum_i u(i) \int_{\Omega} \varphi(i) \varphi(j) \, dx \\ &= \sum_i u(i) \int_{\Omega} \varphi_1(i) \varphi_1(j) + \varphi_2(i) \varphi_2(j) \, dx \\ &=: \Omega u \end{aligned}$$

Im Projekt kommt anstatt dieser Massenmatrix eine *lumped mass matrix* mit

$$\Omega \mapsto^{\text{mass lumping}} \Omega = h^2 id$$

zum Einsatz. Diese ist mit $\Omega^{-1} = h^{-2} id$ einfach zu invertieren. Dadurch kann der diskrete Wert \mathcal{F} in einer „Iteration“ berechnet werden⁴. Für explizite θ -Verfahren ist dies zu tun naheliegend, da die Präparationsphase komplett vor der eigentlichen Druckiteration abzuschließen ist⁵.

4.2 Druckpoissonsgleichung

Die Druckpoissonsgleichung formalisiert sich als

$$\begin{aligned} \int_V \mathbf{div} \nabla p \, dx &= \int_V \frac{1}{\tau} \mathbf{div} u^{(t)} - (1 - \theta) \mathbf{div} \mathcal{F}(u^{(t)}) - \theta \mathbf{div} \mathcal{F}(u^{(t+\tau)}) \, dx \quad \forall V \in \Omega_h \\ &\Downarrow \\ -M\Omega^{-1}M^T p &= \frac{1}{\tau} Mu - M\mathcal{F}(u), \end{aligned}$$

wobei die Vektoren u wiederum θ -Wahl abhängig als Linearkombination aus aktuellem und zukünftigem Wert gewählt werden müssen. In der nachfolgenden Abhandlung wird statt dem obigen diskreten System mit einem negierten gearbeitet, da die Arbeit mit negativer linken Seite ungewohnt ist — man denke nur an die negativen Diagonalelementeinträge. Zwei Bemerkungen seien hierbei noch angebracht:

- In 4.1 wird das Gleichungssystem

$$\Omega \mathcal{F}(u) = -Du + C(u)u$$

schrittweise dargestellt. Damit liegt $\mathcal{F}(u)$ bereits korrekt skaliert, also in \mathbb{U} , auf den Elementecken vor.

⁴An dieser Stelle möchte ich meinen Mentor Prof. Christoph Zenger zitieren:

Weniger als eine Iteration ist schwierig — das leuchtet mir ein.

⁵Verwendet man implizite Zeitverfahren, beinhaltet jeder Iterationszyklus eine Aktualisierung des diskreten Wertes \mathcal{F} zu $t + \tau$. In diesem Kontext kann es Sinn machen, die Invertierung implizit iterativ vornehmen zu lassen und damit auch auf unregelmäßigen Gittern eine höhere Konvergenzordnung zu erhalten. Dabei erscheint eine Konvexkombination der hießigen Massenmatrix mit einem modifizierten *variational mass lumping*-Ansatz — also ein *template mass lumping* — am erfolgversprechendsten zu sein [Felippa 04, White 04]. Dies muss jedoch erst in zukünftigen Arbeiten evaluiert werden.

Exkurs: Mass Lumping — Teil I

Treten in der schwachen Formulierung einer Differentialgleichung Terme der Bauart

$$\sum_i \dots \int_{\Omega} \varphi(i)\varphi(j) dx \quad \forall j$$

auf, so bezeichnet man die korrespondierende Matrix als *Massenmatrix*, die hier konsequent mit Ω bezeichnet werden soll. Berechnet man selbige im Kontext eines klassischen Ritz-Verfahrens, so spricht man von der sogenannten *direct stiffness method*. Diese kann elementweise analog zur Assemblierung der Steifigkeitsmatrix implementiert werden. Im Umfeld zeitabhängiger Verfahren muss nun diese Massenmatrix stets implizit oder explizit invertiert werden. Dabei stellt sich heraus, dass selbige ohne Einschränkung der Allgemeinheit meist relativ dicht besetzt ist, was dazu führt, dass man in der Anwendung die Massenmatrix durch eine leichter zu invertierende Matrix approximiert. Idealerweise handelt es sich dabei um eine Diagonalmatrix. Der Vorgang der Approximation wird als *mass lumping* bezeichnet.

Mass-Lumping

Man unterscheidet nach [Felippa 04] drei verschiedene Ansätze, die Massenmatrix zu berechnen:

- Beim *variational mass lumping* wird die Massenmatrix durch einen Variationsansatz errechnet. Behält man als Testraum den Ansatzraum bei, erhält man die *direct stiffness method* und damit die sogenannte *consistent mass matrix*. Durch eine Variation des Ansatz- und Testraums erreicht man unter Umständen eine Vereinfachung der Matrix. Einen derartigen Ansatz mit Lagrange-Polynomen findet man in [White 04].
- Entkoppelt man die Freiheitsgrade vollständig anhand des „Raumes um sie herum“, so spricht man vom *direct mass lumping*. Dies kann man als *variational mass lumping* auffassen, wobei zu jedem Freiheitsgrad eine charakteristische Funktion definiert wird, die ein Kontrollvolumen spezifiziert. Dieser Ansatz entspricht auch den Illustrationen in [Blanke 04].
- Bildet man zu guter Letzt eine Linearkombination

$$\Omega = \sum_{i=1 \dots k} \mu_i \Omega_i$$

über verschiedenen approximierten Massenmatrizen Ω_i , so spricht man vom *template mass lumping*. Dieser Ansatz hat sich für Akkuranzuntersuchungen bewährt. Alle *lumped mass matrices* sollten nach [Felippa 04] folgende Eigenschaften aufweisen:

- Symmetrie, d.h. $\Omega = \Omega^T$,
- Wiedergabe der Elementsymmetrien (insbesondere also z.B. $(\Omega)_{ii} = (\Omega)_{jj}$ bei uniformen Gittern),
- Erhaltung der Gesamtmasse,
- positive Definitheit.

Exkurs: Mass Lumping — Teil II

Row-Summing

In [Gresho 98] wird vorgeschlagen, die Massenmatrix als Diagonalmatrix anzusetzen und die Einträge über die Zeilensumme der globalen Massenmatrix (*row summing*) zu errechnen. Dies entspricht der *direct stiffness method*. Aufgrund der geforderten Gesamtmasseerhaltung muss das Resultat eine h^d -skalierte Einheitsmatrix sein. Allerdings ist der Begriff *row summing* leicht irreführend, da er sich auf die globale Massenmatrix bezieht — es können sich allerdings Einträge der lokalen Massenmatrizen aufheben. Arbeitet man ausschließlich mit lokalen Assemblierungsmatrizen bietet es sich daher an, über alle Matrixeinträge zu summieren und anschließend durch die Zeilenzahl mal Spaltenzahl zu dividieren.

Diese Technik wird in dieser Arbeit verwendet. Eine Konsistenzuntersuchung derselben über die Taylorreihenentwicklung liefert [Blanke 04] — sie liegt bei regulären Gittern nach wie vor bei $O(h^2)$. [Fletcher 84] bietet eine noch ausführlichere Evaluierung verschiedener Techniken.

- Für explizite Zeitverfahren vereinfacht sich die Druckpoissongleichung zu

$$\begin{aligned} M\Omega^{-1}M^T p &= -\frac{1}{\tau}Mu^{(t)} + M\mathcal{F}(u^{(t)}) \\ &= -\frac{1}{\tau}Mu^{(t)} - M\Omega^{-1}Du^{(t)} + M\Omega^{-1}C(u^{(t)})u^{(t)}. \end{aligned}$$

Der hier vorliegende Algorithmus arbeitet jedoch mit explizit vorliegendem $\mathcal{F}(u^{(t)})$.

Für das lineare Gleichungssystem bleibt damit abschließend noch die Matrix M respektive M^T zu definieren:

Zuerst wird der zu [Blanke 04] gehörende Divergenzoperator rekapituliert. Wie der Glattheitsdiskussion zu entnehmen ist, korrespondiert dieser zu einem stückweisen konstanten Druck — somit kann der Ableitungsoperator nicht kontinuierlich interpretiert werden und ergibt sich ausschließlich aus der Geschwindigkeitsdiskretisierung über die negierte Transponiertheitsforderung für diskrete Divergenz und Gradienten. Im Gegensatz zu den vorangegangenen Operatoren sollen ab nun die globalen Operatoren, d.h. die Diskretisierungssterne, angegeben werden. Für die 5-Punkte-Diskretisierung wird man analog zu den Q^1P^0 -Elementen direkt die *checkerboard*-Instabilität auslesen können, obgleich das hier nur peripher von Interesse ist.

Desweiteren ergibt sich aus den Sternen der Bezeichner 5-Punkte-Element respektive 9-Punkte-Element. Grundlegend für das Kapitel zu Gleichungslöser ist später darüber hinaus der Wert des Diagonalelements. Dieser Wert lässt sich bequem aus den globalen Sternen auslesen, daher werden hier beide Darstellungsweisen — globale und elementweise Sterne — angegeben.

Der letzte Abschnitt beschäftigt sich mit den augmentierten Geschwindigkeitselementen und der Herleitung des zu ihnen passenden Divergenzoperators. Bereits angesprochen wurde, dass hier, zu diesen Elementen passend, bilineare versetzte Druckelemente gewählt werden, weil diese einen passenden kontinuierlichen Gradienten aufweisen und hinreichend glatt sind. Der Operator $\Omega^{-1}M^T$ gibt für versetzte bilineare Druckelemente den exakten Ableitungswert auf den Gitterlinien wieder.

Der Druck ist *staggered* — versetzt — lokalisiert. Dadurch erstrecken sich die Sterne für den gesamten Druckpoissonoperator je über neun geometrische Elemente. In der Herleitung der Matrizen wird das Gleichungssystem jedoch auf Druckgradienten definiert, die analog zu den Geschwindigkeitsfreiheitsgraden auf den Gitterpunkten liegen. Dieser „Trick“ wird sich später als

4 Diskrete Operatoren

vorteilhaft bezüglich des Glätters erweisen, aber es auch ermöglichen, das Programm in das von [Günther 04, Pögl 04] entworfene Framework zu integrieren.

4.2.1 Poissongleichung auf den [Blanke 04]–Elementen

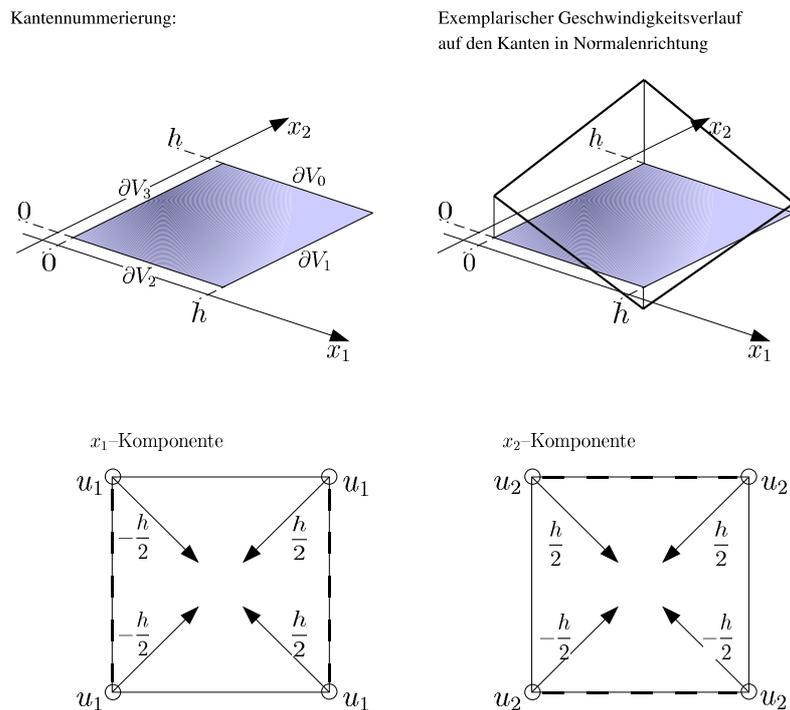


Abbildung 4.2: Der Divergenzoperator auf den [Blanke 04]–Elementen. Die gestrichelt markierten Kanten der Elemente sind Indikatoren für einen linearen Wertverlauf.

Für die Definition des Divergenzoperators ziehe man, wie dies bereits in vorangegangenen Abschnitten der Fall war, den Gaußschen Integralsatz zu Rate. Anschließend verwende man die Linearität des Normalenanteils des Eingabedatums auf den Elementkanten und erhält

$$0 = \int_V \mathbf{div} u \, dx \quad \forall V \in \Omega_h \quad \Leftrightarrow \quad 0 = Mu$$

$$\Leftrightarrow \quad M_{local} u = 0 \quad \forall V \in \Omega_h$$

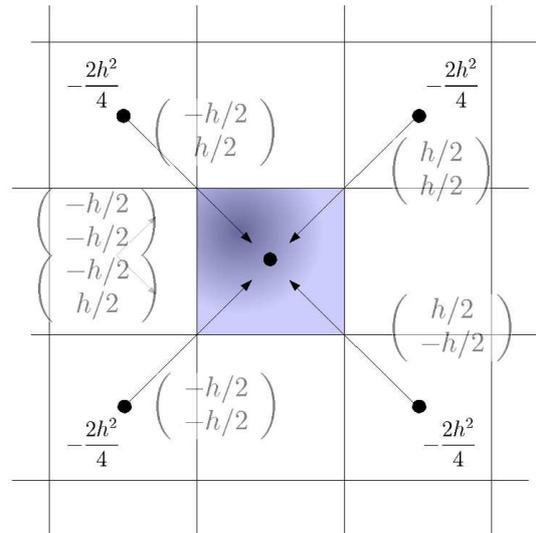


Abbildung 4.3: Der 5–Punkte–Druckstern zu den in [Blanke 04] entworfenen Ansatzfunktionen. Die hier zu beobachtende Auslöschung der vertikalen und horizontalen Nachbar-druckbeiträge führt zu der bekannten *checkerboard* Instabilität.

$$\begin{aligned}
 M_{local} \begin{pmatrix} u_1(0) \\ u_1(1) \\ u_1(2) \\ u_1(3) \\ u_2(0) \\ u_2(1) \\ u_2(2) \\ u_2(3) \end{pmatrix} &= \int_{\partial V} (u, n) \, dx \\
 &= \int_{\partial V_0} u_2 \, dx + \int_{\partial V_1} u_1 \, dx - \int_{\partial V_2} u_2 \, dx - \int_{\partial V_3} u_1 \, dx \\
 &= h \frac{1}{2} (u_2(0) + u_2(1)) + h \frac{1}{2} (u_1(1) + u_1(3)) \\
 &\quad - h \frac{1}{2} (u_2(2) + u_2(3)) - h \frac{1}{2} (u_1(0) + u_1(2)) \, dx \\
 &= \frac{h}{2} (-1 \ 1 \ -1 \ 1 \ 1 \ 1 \ -1 \ -1) \begin{pmatrix} u_1(0) \\ u_1(1) \\ u_1(2) \\ u_1(3) \\ u_2(0) \\ u_2(1) \\ u_2(2) \\ u_2(3) \end{pmatrix}.
 \end{aligned}$$

4 Diskrete Operatoren

Um die Übersicht wahren zu können definiere man die folgende Notationskonvention:

$$M_{local} = \frac{h}{2} (-1 \ 1 \ -1 \ 1 \ 1 \ 1 \ -1 \ -1) =: \frac{h}{2} \left[\begin{pmatrix} -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 \end{pmatrix} \right]$$

Drei Beobachtungen prägen nun alle weiteren Überlegungen zur algorithmischen Realisierung des Gesamtoperators $M\Omega^{-1}M^T$ in der Druckpoissongleichung:

- Der Operator M lässt sich zellweise auswerten. Man kann also alle Zellen traversieren und auf jeder Zelle unmittelbar die Divergenz durch Anwendung von M_{local} errechnen.
- Der transponierte Divergenzoperator M^T wirft Daten aus dem Zellzentrum auf die Eckpunkte. Aufgrund der Bedeutung der Matrix in der Impulsgleichung bezeichne man diese Daten als *Druckgradient*, obgleich es für zellweise konstanten Druck keine korrespondierende Größe im Druckfunktionenraum, also keine kontinuierliche Interpretation gibt.
- Die Auswertung von M^T ist nichtlokal. Um das Bilddatum auf einem Gitterpunkt errechnen zu können, muss man folglich alle vier adjazenten Zellen „anfassen“ und über die Ergebnisse der vier M_{local}^T -Anwendungen summieren.

Letztere Beobachtung führt zu folgender Aussage:

Definition 3 *Kann ein Eintrag des Ergebnisses der Anwendung eines diskreten Operators errechnet werden, indem man ausschließlich eine Zelle beachtet, so erfülle dieser Operator das **Lokalitätsprinzip**. Müssen für einen Bildeintrag Teilresultate akkumuliert werden, so erfüllt der Operator das **Lokalitätsprinzip** nicht.*

In der algorithmischen Realisierung wird später niemals der Gesamtoperator $M\Omega^{-1}M^T$ in einem Schritt berechnet. Stattdessen splittet man ihn in M und $M\Omega^{-1}$ auf und verwende Hilfsvariablen für den „Druckgradienten“, wodurch eine elementweise Auswertung möglich wird. Diese Technik wird später bei der Diskussion der Klassifizierung des Algorithmus nochmals explizit herausgestellt.

Korrolar 4 *Der Divergenzoperator M genügt dem Lokalitätsprinzip, der Operator M^T genügt ihm nicht.*

Spricht man in Begriffen der linearen Algebra, so bedeutet das Lokalitätsprinzip, dass über jede Zelle des diskreten Gebietes Ω_h bereits eine komplette Zeile des Gleichungssystems definiert ist. Im Falle der in vielen Büchern [Braess 02, Günther 04, Pögl 04] behandelten Poisson-Gleichung mit Freiheitsgraden auf den Gitterpunkten leistet jedes geometrische Element einen Beitrag zu 2^d Gleichungszeilen. Entsprechend entsteht eine komplette Zeile des zugrundeliegenden Gleichungssystems durch Akkumulation der Beiträge von 2^d beteiligten Zellen. Die elementweisen Operatoren genügen dem Lokalitätsprinzip also nicht. Setzt man die Begriffe Lokalitätsprinzip und Assemblierung in Korrelation, so fällt auf, dass für Operatoren, die dem Lokalitätsprinzip genügen, elementweise und nodale Assemblierung identisch sind.

Zurück zur Druckpoissongleichung — einer Poissongleichung auf einem versetzten Gitter. Für die im nachfolgenden Kapitel behandelten Gleichungslöser ist das globale Gleichungssystem, wie sie eine explizite Assemblierung liefert, irrelevant. Von Bedeutung ist jedoch sehr wohl der Wert des Diagonalelements zu $M\Omega^{-1}M^T$. Dieser lässt sich direkt aus Abbildung 4.3 auslesen und beträgt für stückweise konstanten Druck

$$diag(M\Omega^{-1}M^T) = \underbrace{\frac{1}{h^2}}_{\text{aus der Massenmatrix}} \cdot \underbrace{4}_{\text{Zahl der Eckpunkte}} \cdot \underbrace{2}_{x_1 \text{ und } x_2 \text{ Beitrag}} \cdot \left(\frac{h}{2}\right)^2 = 2$$

Dieser Rechnung liegt eine interessante Beobachtung zugrunde: Die Beträge eines jeden vertikalen und horizontalen Nachbarn nivellieren sich über die x_1 und x_2 Komponente. Dadurch ergibt sich eine Entkopplung der Druckwerte: Eine Druckwertänderung affektiert nur die Diagonalnachbarn — der Divergenzwert auf den direkten Nachbarn wird nicht beeinflusst. Dies bedeutet, dass das globale Gleichungssystem M genau einmal reduzibel ist, also in zwei Systeme zerfällt. Dies ist wiederum eine Ausprägung des Nichterfüllens der inf-sup-Bedingung und kann sich durch das Auftreten von *checkerboard*-Mustern äußern, wie sie in Abbildung 2.7 auf Seite 24 dargestellt sind⁶.

4.2.2 Poissongleichung auf den augmentierten Elementen

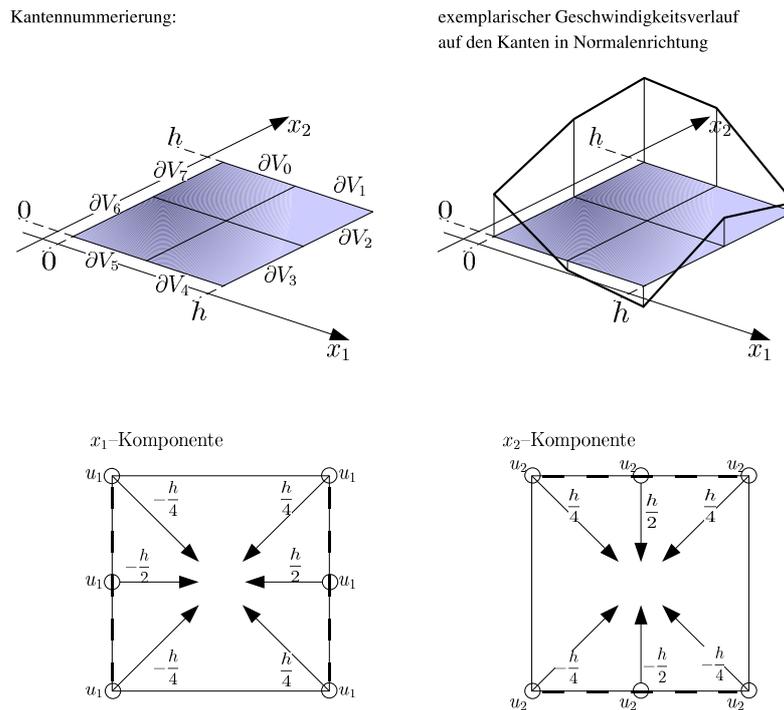


Abbildung 4.4: Der Divergenzoperator auf den augmentierten Geschwindigkeitselementen.

Divergenzoperator

Nun seien die versetzten bilinearen Druckelemente Gegenstand der Untersuchung. Aus der Diskussion der Q^1P^0 -Elemente in 4.2.1 übernehme man den Satz von Gauß und die Matrixnotationskonvention. Die implementierungsnahe lexikographische Nummerierung wurde aus Lesbarkeitsgründen durch eine Notation über die Himmelsrichtungen ersetzt. Die Kantenummerierung sei mit Abbildung 4.4 gegeben.

⁶Ein weiteres Detail mag für den Leser mit einer gewissen Affinität zur linearen Algebra von Interesse sein: In 7.8 wird das Konvergenzverhalten eines Relaxationsverfahrens auf den Q^1P^0 -Elementen den bilinearen Druckelementen gegenübergestellt. Deutlich ist zu erkennen, dass man sich die erhöhte Druckstabilität mit steigenden Iterationszahlen erkauft. Dies ist darauf zurückzuführen, dass dem einfacheren Drucklayout zwei Gleichungssysteme mit lediglich $\frac{\pi}{2}$ Unbekannten zugrundeliegen.

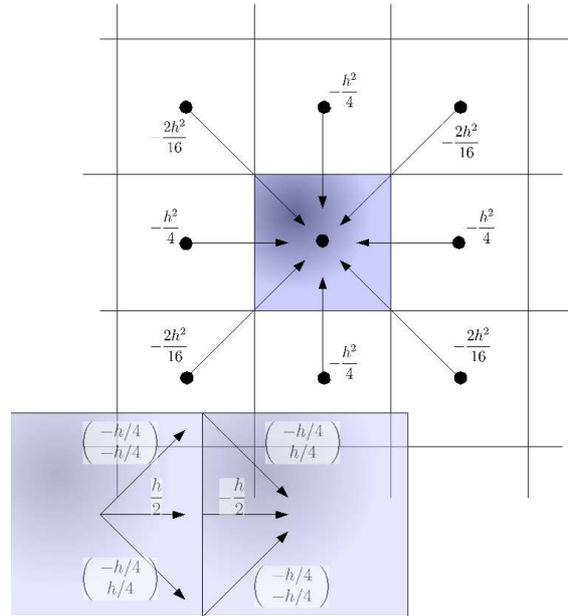


Abbildung 4.5: Der 9-Punkte-Druckstern, der zu den in dieser Arbeit neu entworfenen Ansatzfunktionen passt. Im Gegensatz zum 5-Punkte-Stern tritt hier keine Entkopplung von Zellen mit gemeinsamer Kante auf. Der Stern ist also in dieser Hinsicht stabil. Desweiteren beobachtet man, dass die vertikalen / horizontalen Nachbarn keine Werte aufgrund von Auslöschung über die Gitterpunkte keine Werte, also ausschließlich über die Kanten Information transportieren.

$$\begin{aligned}
 \int_V \mathbf{div} u \, dx &= \int_{\partial V} (u, n) \, dx \\
 &= \sum_{i \in \{0, \dots, 7\}} \int_{\partial V_i} (u, n) \, dx \\
 &= \sum_{i \in \{0, 1\}} \int_{\partial V_i} u_2 \, dx - \sum_{i \in \{4, 5\}} \int_{\partial V_i} u_2 \, dx \\
 &+ \sum_{i \in \{2, 3\}} \int_{\partial V_i} u_1 \, dx - \sum_{i \in \{6, 7\}} \int_{\partial V_i} u_1 \, dx
 \end{aligned}$$

Aufgrund der Linearität des Druckgradienten auf den Randabschnitten können die Integrale auf den Untermannigfaltigkeiten exakt ausgewertet werden:

$$\begin{aligned}
\int_{\partial V_0} u_2 \, dx &= \int_{\partial V_0} u_2(NW) + \frac{x}{h/2} (u_2(N) - u_2(NO)) \, dx \\
&= u_2(NW)x + \frac{x^2}{h} (u_2(N) - u_2(NW)) \Big|_{x=0}^{h/2} \\
&= \frac{h}{4} (u_2(0) + u_2(N)) \\
\int_{\partial V_1} u_2 \, dx &= \int_{\partial V_1} u_2(N) + \left(\frac{x}{h/2} - 1 \right) (u_2(NO) - u_2(N)) \, dx \\
&= (2u_2(N) - u_2(NO))x + \frac{x^2}{h} (u_2(NO) - u_2(N)) \Big|_{x=h/2}^h \\
&= \frac{h}{4} (u_2(N) + u_2(NO)) \\
\int_{\partial V_2} u_1 \, dx &= \frac{h}{4} (u_1(NO) + u_1(O)) \\
\int_{\partial V_3} u_1 \, dx &= \frac{h}{4} (u_1(O) + u_1(SO))
\end{aligned}$$

$$\begin{aligned}
\int_{\partial V_4} u_2 \, dx &= \frac{h}{4} (u_2(SO) + u_2(S)) \\
\int_{\partial V_5} u_2 \, dx &= \frac{h}{4} (u_2(SW) + u_2(S)) \\
\int_{\partial V_6} u_1 \, dx &= \frac{h}{4} (u_1(SW) + u_1(W)) \\
\int_{\partial V_7} u_1 \, dx &= \frac{h}{4} (u_1(NW) + u_1(W))
\end{aligned}$$

Der Vollständigkeit halber sei der Operator noch als Matrix angegeben. Hierzu fehlt jedoch bisher die Definition einer passenden Nummerierung, weshalb die Nummerierung beginnend mit dem linken oberen Eck im Uhrzeigersinn erfolgen soll:

$$M_{local} = \frac{h}{4} \left[\begin{pmatrix} -1 & 0 & 1 & 2 & 1 & 0 & -1 & -2 \\ 1 & 2 & 1 & 0 & -1 & -2 & -1 & 0 \end{pmatrix} \right]$$

Durch die Variation des Ansatzraumes erhalten blieb, dass sich die Beträge der horizontalen und vertikalen Zellnachbarn über die Zellecken nivellieren — ihr Informationstransport erfolgt ausschließlich über die Kanten. Dies ist am deutlichsten in Abbildung 4.5 zu entnehmen, wo dieser Sachverhalt exemplarisch explizit für horizontale Nachbarn aufgezeigt wird. Aussagen über die Lokalität des Operators bleiben auch hier gültig. Das später für den Glätter auf der Poissongleichung benötigte Diagonalelement errechnet sich nun jedoch als

$$diag(M\Omega^{-1}M^T) = \frac{1}{h^2} \left(4\frac{2h^2}{16} + 4\frac{h^2}{4} \right) = \frac{6}{4}$$

und ist auf Randzellen entsprechend der reduzierten Nachbarzahl ebenfalls zu vermindern.

4 Diskrete Operatoren

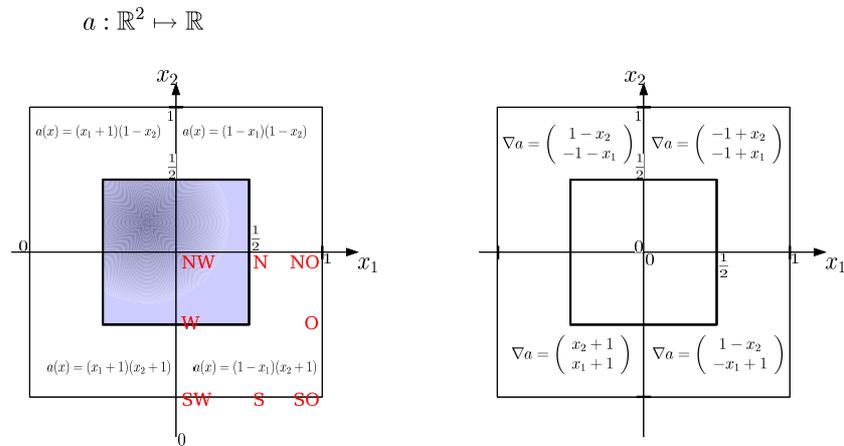


Abbildung 4.6: Nomenklatur der Eingangsgrößen des Divergenzoperators nach Himmelsrichtungen. Der Träger $]-\frac{1}{2}, \frac{1}{2}[\times]-\frac{1}{2}, \frac{1}{2}[$ einer versetzten Pagodenfunktion auf einem „Gitter“ mit Einheitslänge ist farbig markiert.

Bilineare Druckelemente

Zu guter Letzt löse man sich für einen kurzen Exkurs der geforderten Dualität der Operatoren, um eine Interpretation des Gradienten als kontinuierliche Größe eines bilinearen Funktionenraums herzuleiten. Dazu betrachte man eine versetzte Pagodenfunktion auf einem geometrischen Element der Kantenlänge h . Diese definiere die Funktion $a : \mathbb{R}^2 \mapsto \mathbb{R}$ mit der Huthöhe δa . Abbildung 4.6 zeigt solche eine Funktion und gibt auch die vektorwertige Gradientenfunktion an. Man rechne mit der Kontrollvolumenformulierung nach⁷. Dabei seien die Eingangsdaten x_1 und x_2 bereits bezüglich der Kantenlänge h skaliert.

$$\begin{aligned} \nabla a(NW) &= \frac{\delta a}{h} \begin{pmatrix} 1 - x_2 \\ -1 - x_1 \end{pmatrix} (x_1 = -0.5, x_2 = 0.5) = \begin{pmatrix} +\frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} \frac{\delta a}{h} \\ \nabla a(N) &= \begin{pmatrix} \perp \\ -1 \end{pmatrix} \frac{\delta a}{h} \\ \nabla a(NO) &= \frac{\delta a}{h} \begin{pmatrix} -1 + x_2 \\ -1 + x_1 \end{pmatrix} (x_1 = 0.5, x_2 = 0.5) = \begin{pmatrix} -\frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} \frac{\delta a}{h} \\ \nabla a(O) &= \begin{pmatrix} -1 \\ \perp \end{pmatrix} \frac{\delta a}{h} \\ \nabla a(SO) &= \begin{pmatrix} -\frac{1}{2} \\ +\frac{1}{2} \end{pmatrix} \frac{\delta a}{h} \\ \nabla a(S) &= \begin{pmatrix} \perp \\ 1 \end{pmatrix} \frac{\delta a}{h} \end{aligned}$$

⁷⊥ bezeichne — wie in der Informatik üblich — undefiniert.

$$\begin{aligned}\nabla a(SW) &= \begin{pmatrix} +\frac{1}{2} \\ +\frac{1}{2} \end{pmatrix} \frac{\delta a}{h} \\ \nabla a(W) &= \begin{pmatrix} 1 \\ \perp \end{pmatrix} \frac{\delta a}{h}\end{aligned}$$

Man stellt fest, dass der exakte Druckgradient für bilineare Ansatzfunktionen genau dem über die Transponierte definierten Wert, skaliert mit $\frac{1}{2}$, entspricht. Es gilt also für bilineare Druckfunktionen $\nabla = 2\Omega^{-1}M^T$, wobei die Ursache für den Faktor $\frac{1}{2}$ in der Integration über die je halbe Kantenlänge zu sehen ist. Der Operator $\Omega^{-1}M^T$ entspricht also bis auf einen konstanten Faktor tatsächlich der exakten Gradientenberechnung für versetzte, stückweise bilineare Druckfunktionen.

Stichwortsammlung

elementweise Assemblierung, lokale Assemblierungsmatrizen; Massenmatrix, *mass lumping*; Nummerierungsalternativen; *checkerboard pattern*;

4 Diskrete Operatoren

5 Grundlagen zu Relaxationsverfahren

Im bisherigen Verlauf dieser Arbeit wurde die Fluidsimulation sukzessive auf ein zu lösendes lineares diskretes System zurückgeführt. Bevor nun über die Angabe einer Raumdiskretisierung der Übergang zum konkreten Programm vollzogen wird, sollen einige Grundlagen zu Lösern für lineare Gleichungssysteme rekapituliert werden.

Es soll dabei kein Überblick über Lösungsansätze generell gegeben werden — hier konsultiere man beispielsweise [Braess 02]. Stattdessen werden gezielt die Aspekte linearer Gleichungslöser herausgegriffen, denen im Rahmen der Implementierung Relevanz beizumessen ist. Dieser Fundus an Aspekten überschreitet dabei das für [Günther 04, Pögl 04] benötigte Wissen. Obwohl die dort entwickelte Methodik lediglich weiterentwickelt worden ist, ist es sinnvoll, bestimmte Grundlagen trotz der Redundanz abermals darzustellen, um eine in sich abgeschlossene Darstellung liefern zu können.

Ausgehend von dieser Zielrichtung werden zunächst kompakt einige Grundlagen niedergeschrieben. Dazu erfolgt zunächst eine algorithmische Darstellung iterativer Löser samt der für die Interpretation der Messergebnisse wichtigen Konvergenzeigenschaften. Diesen, hauptsächlich [Braess 02, Knabner 00] entnommenen, Aspekten wird anschließend eine Operatorinterpretation nachgestellt. Diese orientiert sich an [Griebel 94, Schöberl 99] und wird gegen Ende dieser Arbeit bei der Klassifizierung des Mehrgitterverfahrens von Nutzen sein.

Der letzte Abschnitt dieses Kapitels ebnet dann den Übergang zur Implementierung und damit zum nachfolgenden Kapitel, indem dem Konzept der expliziten Assemblierung die implizite Gleichungsdefinition, die hier verwendet wird, gegenübergestellt wird. Kommt man aus der linearen Algebra, so argumentiert man stets auf den Matrizen zu einem Problem. In dem hier entwickelten Programm werden jedoch zu keiner Zeit globale Matrizen aufgestellt, stattdessen werden die Ergebnisse aller Operatorauswertungen bei jeder Geometrietraversierung durch Akkumulation von Anteilen aller beteiligten Zellen berechnet. Dadurch erspart man sich die Beschäftigung mit der Organisation und Speicherung der im FEM Umfeld typischerweise auftretenden dünnbesetzten, aber enorm großen Matrizen, sowie die explizite Berechnung der globalen Operatoren in adaptiven Gittern.

5.1 Iterative Verfahren

In dieser Arbeit kommen ausschließlich stationäre Iterationsverfahren zur Verwendung. Stationär bedeutet in diesem Kontext, dass die Iterationsvorschrift glättungsschrittinvariant ist. Selbige werden infolge anhand des Modellproblems

$$Ax = b$$

präsentiert, wobei A stets eine positiv definite quadratische Matrix sei, wie sie bei jeder Diskretisierung eines elliptischen Problems auftritt. Aus einer Vielzahl von Gründen sind die bei partiellen Differentialgleichungen auftretenden Matrizen nicht einfach explizit invertierbar. Anstatt der expliziten Invertierung verwendet man daher einen iterativen Gleichungslöser. Hierzu spalte man die Matrix A additiv in

$$A = A_l - A_r$$

5 Grundlagen zu Relaxationsverfahren

auf und identifiziert die Lösung des Gleichungssystems mit dem Fixpunkt

$$A_l x = A_r x + b.$$

Dies führt direkt zur Iterationsvorschrift

$$x^{(n+1)} = A_l^{-1} (A_r x^{(n)} + b).$$

Offensichtlich ist es also sinnvoll, A_l leicht invertierbar zu wählen.

5.1.1 Ausprägungen der Iterationsvorschrift

Definition 4 Für Iterationsverfahren der Bauart

$$x^{(n+1)} = A_l^{-1} (A_r x^{(n)} + b) \quad \forall n \in \mathbb{N}$$

definiere man folgende Namen:

- $A_l = \lambda \text{id}$ Richardson-Verfahren
- $A_l = \text{diag}(A)$ Jakobi-Verfahren (*diag* „extrahiert“ die Diagonaleinträge)
- $A_l = L(A)$ Gauss-Seidel-Verfahren (L bezeichne die obere Diagonalmatrix)

Man aktualisiere nun die Komponenten des Iterationsvektors x sukzessive, bis ein bestimmtes Abbruchkriterium erreicht ist. In jedem Schritt wird dabei im Falle der Konvergenz der Fehler reduziert. Die oben genannten Verfahren bezeichnet man auch als *Glätter*.

Notiert man die Iterationsvorschrift explizit, so unterscheiden sich Jakobi- und Gauss-Seidel-Verfahren in der Aktualität des Eingangsdatums. Dies ist die Begründung für die besseren Konvergenzraten des Gauss-Seidel-Verfahrens. Die Konvergenzordnung ist und bleibt jedoch linear, also 1:

Definition 5 Sei $e^{(k)}$ $k \in \mathbb{N}$ der Fehler der Lösung im k -ten Iterationsschritt. Ein Iterationsverfahren konvergiert, falls

$$\|e^{(k+1)}\| \leq \rho \|e^{(k)}\|^q \quad \rho < 1 \wedge q \geq 1.$$

Hierbei bezeichne ρ die **Konvergenzrate**, q die **Konvergenzordnung**.

Theorem 5 Ein Relaxationsverfahren konvergiert genau dann wenn $\|A_l^{-1} A_r\| < 1$.

Der Standardbeweis der linearen Konvergenz der obigen Verfahren kann in wenigen Zeilen erbracht werden und sei hier aufgrund der späteren Diskussion „optimaler“ Konvergenzraten wiedergegeben. Hierbei setze man schlicht und einfach die Fixpunkteigenschaft der Lösung ein:

$$\begin{aligned} \|e^{(k+1)}\| &= \|x^{(k+1)} - x^{(*)}\| \\ &= \|A_l^{-1} A_r x^{(k)} + A_l^{-1} b - x^{(*)}\| \\ &= \|A_l^{-1} A_r x^{(k)} + A_l^{-1} b - A_l^{-1} A_r x^{(*)} + A_l^{-1} b\| \\ &= \|A_l^{-1} A_r (x^{(k)} - x^{(*)})\| \\ &\leq \|A_l^{-1} A_r\| \|x^{(k)} - x^{(*)}\| = \|A_l^{-1} A_r\| \|e^{(k)}\| \\ &= \|A_l^{-1} A_r\|^k \|e^{(0)}\| \end{aligned}$$

Mit der Definition des Spektralradius $\rho(B)$ über die Eigenwerte λ einer Matrix B als

$$\lambda_{max} = \rho(B) = \lim_{k \rightarrow \infty} \|B^k\|^{\frac{1}{k}}$$

kann man nun nach [Braess 02, Trottenberg 00] zeigen, dass obige Iterationsverfahren von der minimalen Gitterweite h abhängigen Konvergenzraten $\rho(A_l^{-1}A_r)$ aufweisen.

Theorem 6 *Für Diskretisierungsmatrizen zu elliptischen Problemen gilt*

$$\rho(A_l^{-1}A_r) \in 1 - O(h^2)$$

Löst man also ein Gebiet immer feiner auf, so verschlechtert sich das Konvergenzverhalten quadratisch mit der Gitterweite.

5.1.2 Residuenbasierte Formalisierung

Ein bereits in [Günther 04, Pögl 04] stetig auftretender Begriff ist der des Residuums, des Bildraumfehlers:

Definition 6 *Sei $e^{(k)}$ der Fehler der approximativen Lösung eines linearen Gleichungssystems im k -ten Schritt. Dann definiert*

$$\begin{aligned} b = A(x^k + e^k) &= Ax^{(k)} + \underbrace{Ae^{(k)}}_{=: res^{(k)}} \\ \Leftrightarrow res^{(k)} &= Ae^k = b - Ax^{(k)} \end{aligned}$$

den Begriff **Residuum** im k -ten Schritt.

Mit dieser Definition lassen sich die eingangs definierten Iterationsverfahren nun in eine residuenbasierte Formulierung bringen:

$$\begin{aligned} x^{(k+1)} &= A_l^{-1} (A_r x^{(k)} + b) \\ &= A_l^{-1} (-Ax^{(k)} + b + A_l x^{(k)}) \\ &= x^{(k)} + A_l^{-1} res^{(k+1)} \end{aligned}$$

Da das Residuum jederzeit berechnet werden kann, liefert dies ebenfalls direkt eine Aktualisierungsvorschrift für die Unbekannten. Man beachte im späteren Verlauf, dass die Einträge der Matrix A über Integrale definiert sind, die über die geometrischen Elemente zerfallen — also „zerfällt“ auch A und damit das Residuum über den geometrischen Elementen. Die elementweise Assemblierung nutzt genau diese Eigenschaft und akkumuliert das Residuum.

Im Unterschied zum Jakobi-Verfahren wird beim Gauß-Seidel-Verfahren lediglich das Residuum nach jeder Komponentenmodifikation sofort aktualisiert. Damit kann man die Multiplikation mit A_l^{-1} sowohl für Jakobi- als auch für Gauß-Seidel-Verfahren als Multiplikation der Diagonalmatrixinversen interpretieren, falls jede Lösungsveränderung in der Gauß-Seidel Variante die sofortige Aktualisierung des Residuums nach sich zieht. Man darf die Iteration also nicht mehr vektoriell auffassen, sondern muss das Residuum generisch interpretieren. Im Folgenden sei deshalb A_l immer als D für Diagonalmatrix dargestellt, unabhängig vom tatsächlich zur Verwendung kommenden Verfahren.

5.1.3 Relaxationsfaktoren

Oftmals kombiniert man ein Iterationsverfahren additiv mit der Trivialiteration

$$x^{(k+1)} = id x^{(k)}$$

und erhält

$$\begin{aligned} x^{(k+1)} &= \omega D^{-1} res^{(k)} + \omega x^{(k)} + (1 - \omega)x^{(k)} \\ &= \omega D^{-1} res^{(k)} + x^{(k)}. \end{aligned}$$

Man kann zeigen [Braess 02], dass ein solches Verfahren für $\omega \in]0, 2[$ nach wie vor konvergiert. Für $\omega < 1$ spricht man von Unter-, für $\omega > 1$ von Überrelaxation. In manchen Fällen lässt sich durch dieses „simple“ Vorgehen eine signifikante Konvergenzverbesserung erreichen.

Theorem 7 Für ein optimal relaxiertes Iterationsverfahren gilt

$$\rho(\omega A_l^{-1} A_r) \in 1 - O(h).$$

Die Konvergenzordnung lässt sich also um eine Potenz erhöhen. Der Beweis dazu ist nachzulesen in [Trottenberg 00]. Den optimalen Relaxationsparameter zu finden, ist jedoch im Allgemeinen nicht mit vertretbarem Aufwand möglich. Einer näheren Erörterung dieser Problematik widmet sich unter anderem [Stoer 90], nach [Griebel 95] hat sich ein Wert $\omega \approx 1.7$ bewährt. Diesen Wert konnte [Neckel 05] nicht bestätigen und verwendet stattdessen empirisch motiviert $\omega \approx 1.2$. Schließlich sei noch eine neue Konvergenzklasse definiert, die am Ende dieser Arbeit von entscheidender Bedeutung sein wird:

Definition 7 Konvergiert ein Iterationsverfahren linear mit h -invarianter Konvergenzrate, so weist selbiges **Mehrgitterkonvergenz** auf¹.

5.2 Operatorschreibweise

Man nehme nun eine etwas abstraktere Sichtweise der zuvor beschriebenen Verfahren ein und definiere das Update der k -ten Unbekannten über den Operator $(1 - S_k)^2$. Eine solche Sichtweise nehmen [Griebel 94, Schöberl 99] ein und können nun Jacobi- und Gauß-Seidel-Verfahren in einer Operatornotation angeben:

$$\begin{aligned} \text{Jacobi-Glätter:} \quad x^{(k+1)} &= (1 - S_N - S_{N-1} - \dots - S_2 - S_1) x^{(k)} \\ &= 1 - \sum_{i=1}^N S_i x^{(k)} \\ \text{Gauß-Seidel-Glätter:} \quad x^{(k+1)} &= (1 - S_N)(1 - S_{N-1}) \dots (1 - S_2)(1 - S_1) x^{(k)} \\ &= \prod_{i=1}^N (1 - S_i) x^{(k)} \end{aligned}$$

Beim Jacobi-Verfahren verdeutlicht diese Schreibweise noch einmal die Ausführung der lokalen Korrektur ohne Berücksichtigung bereits erfolgter Aktualisierungen anderer Komponenten des Lösungsvektors x . Bei Gauß-Seidel dagegen werden die Korrekturen jeweils sofort auf den aktuellsten Lösungsvektor aufgeschlagen, bevor der nächste Operator angewandt wird. Ausgehend von dieser Notation motivieren sich die Begriffe *multiplikatives* respektive *additives Verfahren*.

¹Wohlweislich weist jedoch [Trottenberg 00] explizit darauf hin, dass die tatsächliche Effizienz eines Verfahrens durch das Konvergenzverhalten in Relation zum zu erbringenden Aufwand pro Iteration zu beurteilen ist. Dieser Aufwand ist jedoch im hier vorgestellten Umfeld eine pro Freiheitsgrad konstante Größe [Günther 04], kann bei (mathematischen) Effizienzbetrachtungen also außer Acht gelassen werden.

² S stehe für *Smother*

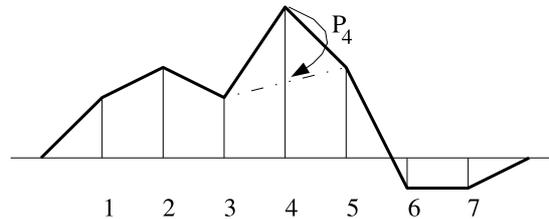


Abbildung 5.1: Wirkung des stationären Glättungsoperators auf dem Fehlerplot.

5.3 Implizite Gleichungssysteme

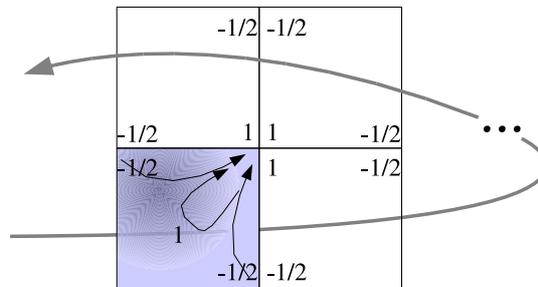


Abbildung 5.2: Elementweise Auswertung der Diskretisierungssterne mit implizit assemblierten Gleichungssystemen am Beispiel des 5-Punkte-Sterns für den Laplace-Operator.

In dieser Arbeit kommen permanent Operatoren vor, die dem Lokalisierungsprinzip nicht aufweisen — dies ist politisch sehr korrekt ausgedrückt: In der Tag genügt ausschließlich der Operator M dieser Charakterisierung. Für ihn kann das Residuum also direkt ausgewertet werden, falls der „Druckgradient“³ auf den Elementecken vorliegt. Im Folgenden betrachte man nodale Operatorauswertung, bei denen alle adjazenten geometrischen Elemente einen Beitrag leisten:

Zur Auswertung des Residuums in Verbindung mit einem solchen Operator platziere man in jedem Freiheitsgrad eine „Hilfsvariable“ res , die zu Beginn auf Null gesetzt sei. Für jede Zelle, die man betritt, wertet man nun den Residuumsanteil aus, indem man die elementweise Assemblierungsmatrix anwendet. Das Ergebnis addiert man auf die Hilfswerte in den vier Zellecken auf. Hat man bezüglich eines Freiheitsgrades alle adjazenten Zellen besucht und lokal ausgewertet, so liegt in res das Residuum fertig berechnet vor und man kann den Wert des Freiheitsgrades nach der Iterationsvorschrift aktualisieren. Anschließend ist die Akkumulationsvariable res für den korrigierten Wert null zu setzen.

Dabei machen sich [Günther 04, Pögl 04] zunutze, dass das Diagonalelement sich über die Zahl der adjazenten inneren Zellen eindeutig berechnen lässt. Dies ist für allgemeine elliptische Gleichungen

³Im Folgenden sei dieser Begriff uneingeschränkt benutzt, wenngleich nicht in allen Fällen eine Funktionenrauminterpretation existiert.

chungssysteme der Bauart

$$\sum_{k,l} \partial_k a(x) \partial_l u(x)$$

über der Unbekannten u nicht möglich, wenn a eine nicht-konstante Funktion ist. Hier muss analog zum Residuum auch das Diagonalelement in den Gitterpunkten akkumuliert werden. Dieses Prinzip ist offensichtlich, führt im Rahmen einer Mehrgittermethodik später jedoch zu Problemen, wie [Pentenrieder 05] zu entnehmen ist.

Als Folge dieser Methodik erspart man sich das Aufstellen der Assemblierungsmatrix, das Gleichungssystem ist also implizit in der Geometrie gegeben. Es verbleibt die Einordnung in die Relaxationsklassen zu klären, welche leicht zu klären ist. Da die Aktualisierung der Unbekannten vorgenommen wird, sobald alle vier adjazenten Elemente abgearbeitet sind, geht der aktualisierte Wert in dieser Iteration nicht mehr in die Akkumulation eines anderen Freiheitsgrads ein. Infolge dessen handelt es sich um ein additives Verfahren mit den Konvergenzeigenschaften des relaxierten Jakobi-Glätters.

Hat man einen Operator vorliegen, der dem Lokalisierungsprinzip genügt, so lässt sich auf diesem nach Definition ein multiplikativer Glätter implementieren — schließlich können der Operator und damit das Residuum in einem Schritt ausgewertet werden und damit die Korrektur sofort erfolgen.

5.4 Abbruchkriterien und Effizienzmessung

Wie eingangs motiviert, sind direkte Löser für Gleichungssysteme aus Diskretisierungen partieller Differentialgleichungen in der Praxis irrelevant. Damit einhergehend stellt sich die Frage nach einem passenden Abbruchkriterium für ein jedes Iterationsverfahren. Problematisch ist dabei abermals, dass man den Fehler $e^{(k)}$ im k -ten Schritt nicht kennt. Stattdessen versucht man, über das Residuum respektive die Korrektur zu argumentieren, und daraus Rückschlüsse auf den aktuellen Fehler zu ziehen. Ein analoges Problem erhält man für die Bestimmung der Konvergenzrate.

Ist ein Abbruchkriterium spezifiziert, so ist die tatsächliche Iterationszahl abhängig von der Qualität der Startlösung. Wir beginnen mit einer Nullstartlösung für alle beteiligten Unbekannten, verwenden nach jeder Zeitfortschaltung dann jedoch die Endapproximation des Vorgängerschritts.

5.4.1 Abbruchbedingung

Mit der Residuumsdefinition ergibt sich aus dem zu lösenden linearen Gleichungssystem $Ax = b$ und der Definition des kleinsten Eigenwerts λ_{min} von A

$$\begin{aligned} \|e^{(k)}\| &\leq \|A^{-1}\| \|res^{(k)}\| \\ &\leq \frac{1}{\lambda_{min}(A)} \|res^{(k)}\|. \end{aligned}$$

Wie [Günther 04] zu entnehmen ist, korreliert dabei auf nodalen Basen der kleinste Eigenwert direkt mit h^2 und man erhält den qualitativen Zusammenhang

$$\|e^{(k)}\| \leq \frac{1}{C \cdot h^2} \|res^{(k)}\| \quad C \in \mathbb{R}_+.$$

Definiert man nun eine Abbruchbedingung über das Residuum mit

$$\|res^{(k)}\| \leq \epsilon,$$

so ist für den Vergleich verschiedener Gitter in die ϵ -Wahl die minimale Gitterweite mit dem Faktor h^2 einzubeziehen, wie [Günther 04] sehr schön demonstriert. Als passende Norm wähle man nach [Griebel 95, Trottenberg 00] die L^2 - oder die Maximumnorm.

Dabei weist Ersterer zu Recht darauf hin, dass bei Auftreten großer Lösungswerte obige „scharfe“ Abbruchbedingung problematisch ist. Er schlägt für Lösungen mit Singularitäten eine relative Toleranzgrenze vor, bei der die Abbruchgrenze mit der Norm des Startdatums skaliert wird. Bekanntermaßen treten in der numerischen Strömungssimulation im Druck oft (Halb-) Singularitäten auf. Betrachtet man nun Abbildung 7.10, so fällt auf, dass der Maximaldruck bei Verwendung eines Mehrgitterverfahrens sich bereits nach wenigen Iterationen einstellt, für ein Relaxationsverfahren aber nur langsam gegen den Zielwert konvergiert. Dies ist bei der Konstruktion des Abbruchkriteriums mit in die Überlegungen einzubeziehen.

5.4.2 Konvergenzrate

Analog zum Abbruchkriterium stellt sich auch bei der Analyse der Konvergenzrate das Problem, dass der Fehler nicht bekannt ist. Zur Analyse des Konvergenzverhaltens ist also ein geeigneter Schätzer zu konstruieren. Dabei sei hier auf [Trottenberg 00] verwiesen⁴, der darstellt, dass

$$q_1(k) = \frac{\|res^{(k)}\|}{\|res^{(k-1)}\|}$$

oder das geometrische Mittel

$$\begin{aligned} q_2(k) &= (q_1(m) q_1(m-1) \dots q_1(0))^{\frac{1}{k}} \\ &= \left(\frac{\|res^{(k)}\|}{\|res^{(0)}\|} \right)^{\frac{1}{k}} \end{aligned}$$

geeignete Formeln sind.

Stichwortsammlung

residuenbasierte Darstellung, lokale Assemblierungsmatrixkontributionen; Konvergenzrate, Konvergenzordnung, Relaxationsverfahren, Mehrgitterkonvergenz; Operatorschreibweise, additive Verfahren, multiplikative Verfahren; explizite Assemblierung, implizite Gleichungssysteme; Abbruchkriterien; Konvergenzrate

⁴Tatsächlich dürfte nach Definition der Konvergenzrate auf Seite 64 die Konstruktion der Schätzer keinen Leser mit auch nur rudimentären Statistikkenntnissen mehr vor große Probleme stellen.

5 Grundlagen zu Relaxationsverfahren

6 Gebietsdiskretisierung, Datenlokalisierung und algorithmische Realisierung

Gegenstand dieses Kapitels ist die Geometriediskretisierung, die Konstruktion zugehöriger Traversierungsalgorithmen und die Umsetzung der Finite-Elemente-Methode samt zugehörigem Glätter auf solchen Gittern. Dazu muss zunächst geklärt werden, wie die in 3.4 definierten Funktionenräume auf dem Gitter konstruiert sind. Als zentrales Element dieses Kapitels fungiert hier, dass das Prinzip der Intervallschachtelung in der Mathematik dem Prinzip der Bi- respektive Tripartition in der Informatik äquivalent ist. Daher lassen sich mathematische Prinzipien, die auf Grenzwertübergängen von Intervallschachtelung basieren, auf natürliche Art und Weise in der Informatik mit Rekursionsalgorithmen darstellen.

Dabei bezeichnet Bipartition / -sektion die Zerlegung einer Menge in zwei disjunkte Teilmengen. Die Bisektion sei hier stets als nicht-entartet angenommen, folglich sind beide Teilmengen stets nicht leer. Liegt Tripartition vor, wird jede Menge in drei Teilgebiete zerlegt. Die Intervallschachtelung ist spätestens seit dem Nachweis der Archimedischen Eigenschaft der reellen Zahlen \mathbb{R} im Werkzeugkasten eines jeden Mathematikers zu finden.

6.1 Gittercharakteristika

Da Rechner nur endliche Strukturen darstellen können, muss ein Gebiet, auf dem eine partielle Differentialgleichung zu „lösen“ ist, zunächst diskretisiert werden.

Definition 8 *Unter einer Diskretisierung versteht man die Abbildung eines Gebietes auf ein stückweise polynomiell berandetes Gebiet, das anschließend in eine endliche Zahl disjunkter Teilgebiete zerlegt wird. Diese Teilgebiete heißen (geometrische) Elemente und seien wiederum polynomiell berandet.*

[Liseikin 99] gibt dem Interessierten einen Überblick über Algorithmen zur Gebietsdiskretisierung. In [Weinzierl 04] werden die folgenden fünf Aspekte für die Beurteilung einer Diskretisierung identifiziert, die in [Pögl 04] aufgestellten Aspekte bilden hiervon eine Untermenge.

- Adaptivität / Modifizierbarkeit,
- Heterogenität,
- Hierarchie,
- Algorithmenineffizienz und
- Speichereffizienz.

Die Adaptivität gewährleistet die Möglichkeit, interessante Gebiete fein auflösen zu können. Löst man nämlich das Gesamtgebiet äquidistant feiner auf, so ist man mit einer Gitterelementexplosion konfrontiert. Mit Modifizierbarkeit wird die dynamische Adaptivität bezeichnet. Heterogene Gitter gewährleisten, dass die Form der Elemente zur Problematik und insbesondere randabhängig passend gewählt werden kann.

Mit Hierarchie wird die Inklusionsbeziehung zwischen Mikro- und Macroelementen bezeichnet. Hierarchische Gitter entstehen bei den meisten Verfeinerungsalgorithmen, da hierbei das ursprüngliche Gitter invariant ist, jedoch neue Unterteilungen eingeführt werden. Eine Gitterverwaltung sollte die Hierarchieinformation explizit vorhalten, da hierdurch die Arbeit mit hierarchischen Erzeugendensystemen [Griebel 94] erleichtert wird.

Abschließend bleibt festzustellen, dass in Bezug auf Effizienz der Algorithmen Mehrgitteralgorithmen die Methode der Wahl zur Lösung elliptischer partieller Differentialgleichungen sind. Durch einen Multiskalenansatz werden dabei Fehler aller Frequenzbereiche in Iterationsverfahren effizient reduziert¹.

Dabei arbeiten die meisten Realisierungen von Algorithmen auf hierarchischen Daten mit Bäumen. Wählt man nun eine zeigerbasierte Implementierung eines Baumkonzepts, so ist der dadurch entstehende Speicherbedarf, der über die Nutzdaten hinaus geht, als Overhead zu sehen. Desweiteren ist die Traversierung derartiger Datenstrukturen ohne Einschränkung der Allgemeinheit bezüglich des Speicherortes hochgradig nichtlokal. Auf aktuellen Rechnerarchitekturen bringt diese Nichtlokalität eine hohe Zahl an sogenannten *Cache-Misses* [Günther 04, Langlotz 04/1] mit sich. Dies ist insbesondere aufgrund der Tatsache ärgerlich, dass sich der Speicherzugriff bei heutigen Rechnern immer als Flaschenhals erweist. Die Reduktion der praktischen Leistungsfähigkeit eines Rechners über die Speicherzugriffszeiten wird sich dabei nach dem Mooreschen Gesetz innerhalb der nächsten Jahre noch spürbar negativer auswirken, da das Wachstum der Prozessorleistungsfähigkeit die Reduktion der Buszugriffszeiten übertrifft. Eine Besserung dieses Umstandes ist nicht in Sicht und so muss der Speichereffizienz bei der Konstruktion praxisrelevanter Algorithmen ein zunehmend breiteres Interesse eingeräumt werden. Nach der Präsentation der Bewertungskriterien wird nun zunächst die in dieser Arbeit verwendete Diskretisierung vorgestellt. Dabei wird versucht, den aufgestellten Anforderungen gerecht zu werden. Anschließend werden realisierungsrelevante Aspekte bei der Arbeit mit derartigen Diskretisierungen näher beleuchtet.

6.2 Spacetree-Diskretisierungen

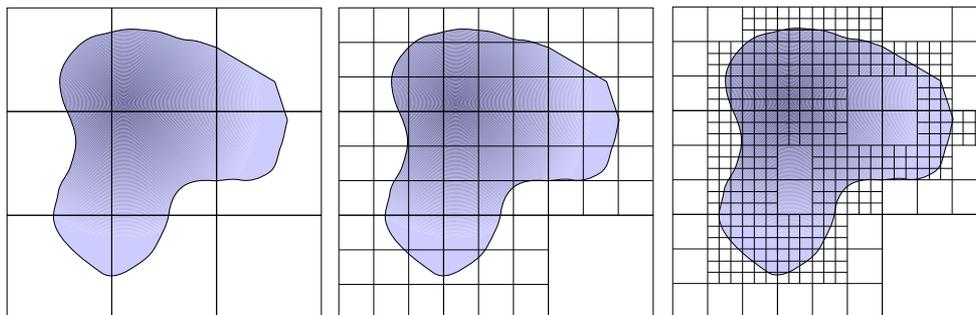


Abbildung 6.1: Diskretisierung eines Beispielgebietes mit einem trisektionsbasierten Spacetree der Tiefe eins (links), zwei (Mitte) oder drei (rechts).

Das eingangs erwähnte Prinzip der n -Partitionierung einer Menge lässt sich auf d -dimensionale Probleme einfach dimensionsrekursiv erweitern, indem man eine Teilung in jeder Raumrichtung

¹Tatsächlich ist die Effizienz bezüglich der Iterationszahlen jedoch in Relation mit der zu verrichtenden Arbeit zu setzen [Trottenberg 00], um Aussagen über die Praxisrelevanz und die Algorithmeneffizienz machen zu können

vornimmt. Verwendet man nun ausschließlich eine äquidistante Partitionierung und bettet die Rechendomäne zuvor in den Standardhyperwürfel ein, so entstehen n^d Submengen, die wiederum Hyperwürfel sind. Der Partitionierungsvorgang lässt sich nun rekursiv wiederholen. Da die Submengen disjunkt sind, liegt dem Vorgang eine Baumstruktur zugrunde, wobei jeder Nicht-Blattknoten genau n^d Kinder hat. Die Höhe des Baumes wird dabei durch die Anzahl der Gebietszerteilungsschritte bestimmt.

Dieses Konzept ist die Basis der Diskretisierung eines Gebietes mit *Spacetrees*. Spacetrees erfreuen sich in vielen Anwendungsgebieten großer Beliebtheit. Gründe hierfür und Eigenschaften dieses Typs von Datenstrukturen finden sich z.B. in [Frank 00, Hartmann 04, Mahler 03]. In der hiesigen Arbeit wird das Gebiet mittels Spacetrees diskretisiert, die auf Dreiteilung in jeder Raumrichtung basieren. Grund ist, dass Spacetrees zum Einen fast alle oben geforderten Eigenschaften aufweisen. Zum Anderen erlauben sie eine sogenannte *cache-optimale* Traversierung, die später in diesem Kapitel Gegenstand der Diskussion sein soll.

Abbildung 6.1 zeigt, wie ein Rechengebiet sukzessive mittels des Spacetrees-Ansatzes diskretisiert wird. Offensichtlich erlaubt das Konzept eine beliebig feine Auflösung des Gebiets, wobei die Gitterauflösung mit einer gegebenen Minimallänge zu der Maximalhöhe des Baumes korrespondiert. Dabei wird in diesem Beispiel insbesondere der Rand fein aufgelöst, während in den Außengebieten und in einigen Innenbereiche nicht weiter verfeinert wird. Gründe für die Verfeinerung eines Gitters lassen sich unterteilen in zwei Gruppen:

Klassifiziert man eine Submenge als homogen, d.h. entweder im Gebiet oder außerhalb des Gebietes liegend, so muss je nach Randverlauf eine hinreichend feine Auflösung gewählt werden, um charakteristische Gebietsmerkmale darstellen zu können. Somit handelt es sich um ein geometrisches Verfeinerungsargument zu gegebener Minimalkantenlänge h . Eine Alternative respektive Ergänzung zur Randverfeinerung liefert [Wagner 05]. Daneben existiert die große Klasse der numerischen Gründe für eine weitere (adaptive) Verfeinerung. Hierbei spielen die Reduktion von Randfehlern und die Auflösung von numerischen interessanten Teilgebieten mit starken Lösungsänderungen die Hauptrollen. Eine umfassende Abhandlung der Verfeinerungstechniken im Rahmen des hier vorgestellten Ansatzes findet sich in [Krahnke 04].

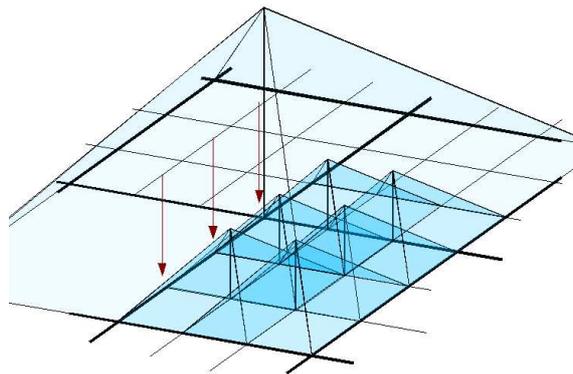


Abbildung 6.2: Einige Ansatzfunktionen auf einem Teilgebiet mit hängenden Feingitterknoten — einige davon sind durch Pfeile markiert.

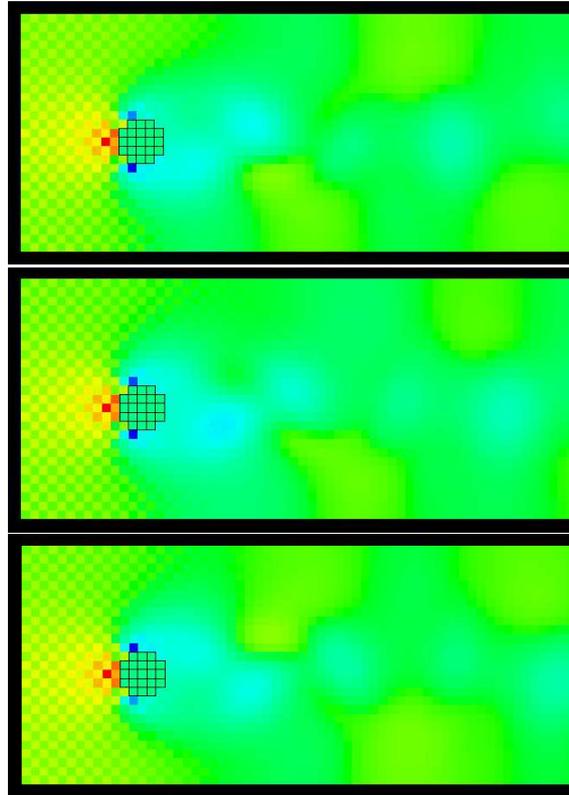


Abbildung 6.3: Wirbelwanderung bei der Kármánschen Wirbelstraße. Die Bewegung der Wirbel korrespondiert mit wandernden Unterdruckgebieten. Das Beispiel ist [Neckel 05] entnommen.

Evaluierung

Offensichtlich genügt eine Spacetreediskretisierung per se der eingangs geforderten Modifizierbarkeit und Möglichkeit, adaptive Gitter darstellen zu können. Desweiteren ist die Bereitstellung der Hierarchieinformation ebenfalls implizit sichergestellt. [Frank 00] zeigt, dass sich aufgrund der Simplität und rekursiven Definition des Algorithmus viele wichtige Operationen auf Spacetrees effizient in Laufzeit und Speicherbedarf implementieren lassen. In [Günther 04, Pögl 04] wurde darüber hinaus eine neuartige Linearisierung entwickelt, die hinsichtlich dieser beiden Punkte noch weiteres Verbesserungspotential ausnutzt. Diese Linearisierung kommt in dieser Arbeit zur Verwendung und wird weiter unten beschrieben.

Oftmals wird die Hyperwürfelzerlegung des Gebiets als großer Nachteil derartiger Zerlegungsansätze bezeichnet. Dabei zielt die Kritik auf zwei Stoßrichtungen ab: Die Auflösung des Randes und die Nichtbeachtung problemspezifischer Lösungseigenschaften. So beachtet eine äquidistante Zerteilung in allen Raumrichtungen achsenabhängige Eigenschaften der Lösung natürlicherweise nicht. Gerade bei der Simulation von Fluiden treten aber oft flußrichtungs- und randabhängige Lösungseigenschaften auf, die bei in der Praxis verwendeten Diskretisierungen oftmals apriori berücksichtigt werden: So wählt man beispielsweise beim Kanalfuß den Diskretisierungsabstand senkrecht zum Rand um Größenordnungen kleiner als in Tangentialrichtung. Abhilfe bei der Verwendung von Spacetrees schafft hier zum Teil jedoch eine dimensionsrekursive Betrachtung, wie

sie [Hartmann 04] vorstellt, mit anschließender problemabhängiger Subdivisionsordnung. Zudem kann man stets argumentieren, dass solche Phänomene meist nur kleine Gebiete, insbesondere Untermannigfaltigkeiten, betreffen und die betroffene Zellzahl daher in der Größenordnung reduziert ist.

Kann mit adaptiven Gittern die Zahl der Zellen zum Erreichen einer gewissen Genauigkeit nach [Frank 00, Krahnke 04, Pögl 04] um eine Potenz (von $O(h^{-d})$ auf $O(h^{-d+1})$) reduziert werden, so tritt auf dem Rand des Gitters bei homogenen Gittern ein Negativeffekt in den Vordergrund: Hier ist mit Standardverfahren für elliptische Probleme nur eine Approximationsordnung von $O(h)$ statt $O(h^2)$ zu erreichen. Da es sich bei hinreichend glattem Rand um eine Untermannigfaltigkeit handelt, ist diesem Problem nur durch Reduktion von h auf \sqrt{h} auf dem Rand beizukommen. Die dadurch erhöhte Zahl an benötigten Zellen von $O(h^{-d+1})$ auf $O(h^{2(-d+1)})$ fällt jedoch erst ab der dritten Dimension bei komplexer Berandung ins Gewicht. Alternative Optimierungsansätze zur Erhöhung der Approximationsordnung finden sich in [Frank 00, Meier 00, Wagner 05].

Abschließend sei nochmals explizit auf die Arbeit [Krahnke 04] verwiesen, in der das Konzept von [Günther 04, Pögl 04] um posteriori-Adaptivität erweitert worden ist. Da das hier vorgestellte Programm auf [Günther 04] basiert, ist ein Einsatz der dynamischen Verfeinerung damit auch im Kontext der numerischen Strömungsmechanik ohne Probleme möglich. Interessant erscheinen dabei zwei Charakteristika der hier behandelten Differentialgleichungen:

Zum einen treten bei Strömungen im Druck nahezu ständig (Halb-) Singularitäten auf. Nach [Trottenberg 00] führt dies zu einer sogenannten *Pollution* und reduziert die Approximationsgüte in der Singularität von $O(h^2)$ auf $O(h^{2/3})$. In diesem Fall scheint die Verwendung einer adaptiven Randverfeinerung a [Krahnke 04] sehr sinnvoll. Der zweite interessante Aspekt ist die in dieser Arbeit thematisierte Behandlung instationärer Probleme, die in vielen Fällen² über keine stationäre, sondern ausschließlich über eine nicht-transiente, quasistationäre Entwicklung verfügen. In diesem Kontext macht es Sinn, stets nur die Bereiche fein zu diskretisieren, die eine hohe Druckvariation aufweisen. Da diese Gebiete — meist sind es Unterdruckgebiete, die einen Wirbel „tragen“ — wandern, kann nur eine posteriori, zeitabhängige Adaptivität hier wirklichen Effizienzgewinn bringen. Ein Beispiel für wandernde Wirbel ist in Abbildung 6.3 zu sehen, bei der Betrachtung transienter Phasen des Driven-Cavity Szenarios werden weitere derartige Phänomene auftreten.

Terminologie

Eine substantielle Behandlung von Spacetree-Datenstrukturen kann und soll diese Arbeit nicht leisten — hier sei vor allem auf [Frank 00, Günther 04] verwiesen. Eingeführt werden jetzt jedoch noch einige Bezeichner, die im späteren Verlauf der Arbeit benutzt werden. So unterscheidet man je nach Homogenität der Zelle die *inneren Zellen* von den *äußeren Zellen*. Zellen, durch die ein Rand verläuft, die also heterogen bezüglich ihrer „Füllung“ sind, bezeichne man als *Randzellen*³. Die Eckknoten äußerer Zellen tragen folglich keine Freiheitsgrade, Randknoten sind nach Dirichlet- und Neumann-Rändern zu klassifizieren. Wie bereits dargestellt, verfügen Neumann-Ränder über eine reduzierte Freiheitsgradcardinalität.

Die Blatthyperwürfel eines Spacetrees bilden die (*geometrischen*) *Elemente* der Diskretisierung und dienen später als die bereits eingeführten Kontrollvolumen. Alternativ werden sie auch als (*Mikro-*) *Zellen* bezeichnet. Würfel, die weiter verfeinert werden, bezeichne man dazu korrespondierend als *Macrozellen*. Im Geometriebaum selber haben folglich die zu Mikrozellen gehörenden Knoten keine Kinder mehr, während Macrozellenknoten im Zweidimensionalen neun Kinder aufweisen.

Die Gittertiefe der Baumknoten korrespondiert wie bereits eingeführt mit dem sogenannten *Gitterlevel*. Jeder Gitterdurchlauf korrespondiert mit einer Anordnung der Baumknoten und kann bei

²Man betrachte die in [Neckel 05] ausgiebig behandelte Kármánsche Wirbelstraße.

³Die Nomenklatur sollte intuitiv klar sein.

Beachtung des Gitteraufbaus als *Traversierung* dargestellt werden. Dabei ist kein Sprung zwischen verschiedenen Baumästen gestattet. Das Gitter mit maximaler Verfeinerungsstufe wird auch als *Feingitter* bezeichnet.

Besondere Beachtung ist stets den *hängenden Knoten* zu widmen. Diese verfügen auf ihrem Gitterlevel über weniger als 2^d adjazente geometrische Elemente. Abbildung 6.2 zeigt solche Punkte. Hängenden Punkten muss man bei der Platzierung von Freiheitsgraden stets besondere Aufmerksamkeit zukommen lassen, um die Stetigkeit der diskreten Lösung sicherzustellen.

6.3 Funktionenräume auf Spacetrees

Die Konstruktion von Funktionenräumen auf Spacetrees erfolgt nach Definition der Ansatzfunktionen ab Kapitel 3.4 sowohl für die Druck- als auch Geschwindigkeitsräume über hierarchische Erzeugendensysteme⁴. Dieser Vorgang ist in [Günther 04, Pögl 04] ausführlich beschrieben und wird in nahezu allen darauf aufbauenden Arbeiten wieder aufgegriffen. Gerade [Krahnke 04] widmet sich der Thematik nochmals ausführlich. Alle Arbeiten können in der Tradition von [Griebel 94] aufgefasst werden.

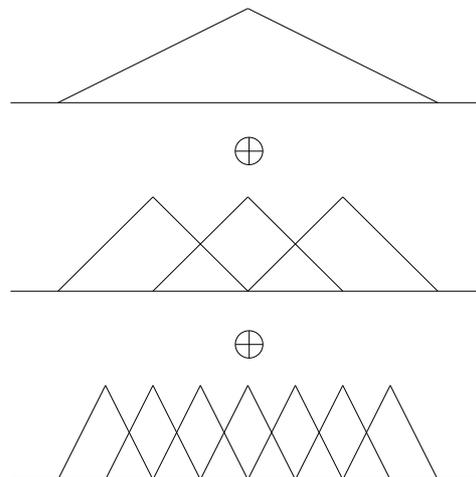


Abbildung 6.4: Das obligatorische [Griebel 94] entstammende Bild zur Einführung hierarchischer Erzeugendensysteme im eindimensionalen Fall.

Grundlage hierarchischer Erzeugendensysteme auf Spacetrees ist es, eine nodale Basis auf jedem Gitterlevel anzusetzen. Lineare Gleichungen auf dem resultierenden Raum sind hochgradig indefinit, da die Ansatzfunktionen eben gerade keine Basis mehr bilden, sondern nur noch ein Erzeugendensystem. Der Übergang in eine nodale Darstellung erfolgt durch die top-down oder bottom-up Traversierung des Baumes, wobei die Funktionswerte in einem Punkt aufsummiert werden. Dieser Vorgang wird auch als *Dehierarchisierung* bezeichnet.

Ist der Funktionsverlauf über diskrete Stützstellen eines Levels gegeben und will man den Funktionswert auf einem tieferen Level erhalten, so *interpoliert* man den Wert auf die feinere Basis. Das Gegenstück ist die *Restriktion*, wie sie im Kapitel zu Mehrgitteralgorithmen thematisiert wird.

⁴Für den Geschwindigkeitsraum macht dies nur bei $(\theta \neq 0)$ -Wahl Sinn, da man sich ansonsten bei der Geschwindigkeitsfortschaltung vollständig auf das Feingitter beschränken kann.

In diesem Abschnitt ist wichtig, dass durch Interpolation und eine top-down Traversierung der dehierarchisierte Funktionswert berechnet werden kann. Identifiziert man den Wert einer Levelansatzfunktion mit dem zum Stützpunkt gehörenden Knoten im Spacetree, so ist diese Traversierung einem pre-order Baumdurchlauf äquivalent⁵.

Statt bereits bekannte Resultate aus [Günther 04, Hartmann 04, Herder 05, Krahnke 04] oder [Langlotz 04/1, Pögl 04, Wagner 05] und vieler anderer bezüglich der hier zur Verwendung kommenden Methodik erneut zu präsentieren, soll das Augenmerk des Lesers auf einige im Kontext dieser Arbeit interessante Aspekte des Gitters gelenkt werden:

- Zunächst sollen hängende und unechte Dirichlet-Knoten der Geschwindigkeit behandelt werden. Hängende Knoten sind stets ein interessantes Fallbeispiel — auch bezüglich der algorithmischen Behandlung — und treten gerade bei hierarchischen Erzeugendensystemen verglichen mit anderen diskreten Funktionenräumen gehäuft auf.
- Dabei ist gerade die Platzierung von Druckfreiheitsgraden auf Randzellen eines Gitterlevels interessant zu betrachten.
- Bei zellbasierten Randzellen unterscheiden sich Rand- und „hängende“ Zellen im hier vorliegenden Algorithmus und schließlich
- weist der Übergang von einer Grobleveldarstellung in eine nodale Darstellung zellbasierter Freiheitsgrade bezüglich des Lokalisierungsprinzips wiederum Parallelitäten zum Divergenzoperator auf.

6.3.1 Hängende Punkte

Definition 9 *Ein hängender Punkt ist ein Gitterpunkt mit weniger als 2^d adjazenten geometrischen Elementen im gleichen Level.*

Hängende Punkte wurden bereits in Abbildung 6.2 dargestellt. Dort wurde allerdings noch nicht die Definition hierarchischer Erzeugendensysteme verwendet. Da bei derartigen Systemen auf allen Leveln nodale Basen angesetzt werden, können nun auf einem Gitterpunkt sowohl echte als auch hängende Freiheitsgrade sitzen. Abbildung 6.5 ist [Pögl 04] entnommen und präsentiert diese Eigenschaft anschaulich. Dabei zeigt sich analoge Problematik für sogenannte unechte Dirichlet-Punkte, also Grobgitter-Randknoten.

Auf hängenden Punkten eines Levels werden keine Formfunktionen angesetzt. Damit ergibt sich aufgrund der Baumeigenschaft der Diskretisierung der dortige Funktionswert ausschließlich durch die Interpolation höher liegender Gitterlevel. Insbesondere sitzen dort wegen der oftmals zitierten Adjungiertheit der Operatoren keine Druckgradienten.

6.3.2 Zellbasierte Freiheitsgrade

Daraus direkt leitet sich infolge ab, dass echte Randzellen und adaptiv entstandene, unechte Randzellen unterschiedlich behandelt werden müssen. Auf echten Randzellen ist der Druckgradient auf dem Rand gegeben, auf durch Adaptivität entstandenen „hängenden“ Randzellen eines Levels wird vorgeschlagen, keine Druckfreiheitsgrade zu platzieren. Ansonsten würde der Druckgradient auf dem Rand der benachbarten Grobzelle einen Knick aufweisen. Es stellt sich daher die Frage, wie die Auswertung der Druckpoissongleichung auf „hängenden“ Zellen zu interpretieren ist:

Offensichtlich ist die exakte Divergenzfreiheit nicht zu erfüllen, da keine Druckvariation vorgeommen werden darf. Somit kann also in solch eine Levelrandzelle mehr Flüssigkeit eintreten, als

⁵Die Ordnung auf den Kindknoten wird in diesem Kapitel ab 6.4 bestimmt.

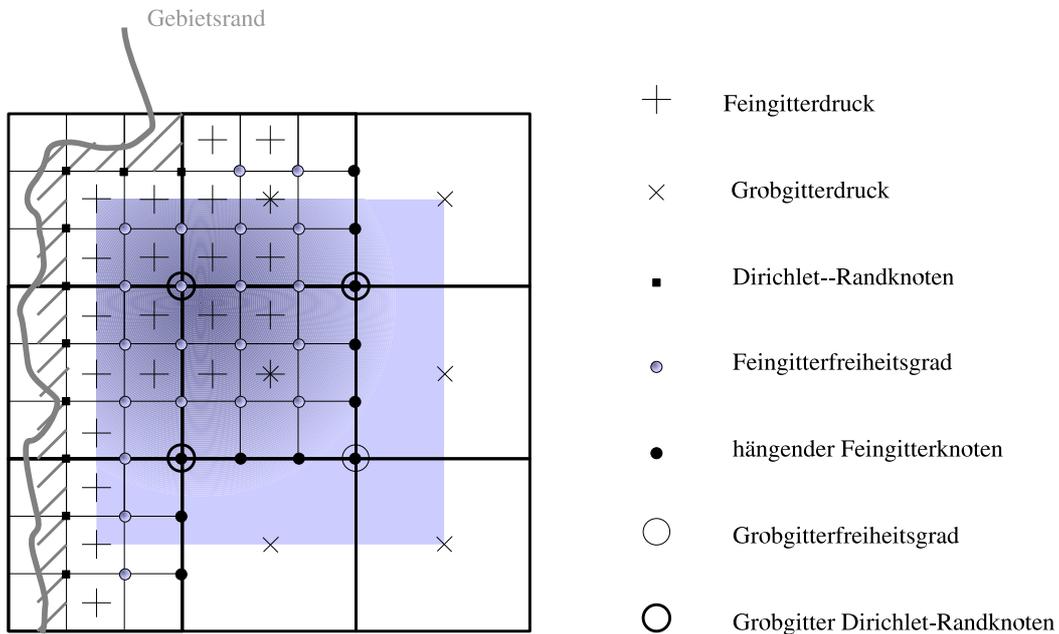


Abbildung 6.5: Dieses Bild entstammt ursprünglich [Pögl 04] und klassifiziert die Gitterelemente in einer Darstellung in einem hierarchischen Erzeugendensystem. Hinzugefügt wurde die Platzierung der zellbasierten Freiheitsgrade. Die Farbfläche repräsentiert exemplarisch den Träger einer Grobgitterdruckfunktion gemäß 6.3.2.

durch den Druck wieder herausgepresst wird — das Fluid wird kompressibel interpretiert. Diesem Missstand ist mit nodalen Basen nicht beizukommen. Insbesondere wird also das zugrundeliegende Druckgleichungssystem unlösbar. Im Rahmen der Mehrgittermethodik wird jedoch für jede verfeinerte Grobgitterzelle über das Residuum ihrer Kinder summiert. Damit wird der Fehler, der auf Adaptivrandzellen entsteht, ein Level nach oben durchgereicht.

In diesem Kontext kann es Sinn machen, auch weitere Constraints auf dem zugrundeliegenden Spacetree zu fordern, um die Qualität der Lösung sicherzustellen. Man beachte in diesem Zusammenhang abermals Abbildung 6.2: Verschiebt man den Rand um eine Feingitterweite nach rechts, so entstehen Randzellen, auf denen der Druck nach obigem Platzierungsvorschlag null sein muss. Dies ist physikalisch nicht zu motivieren, daher sollte über einen Baumconstraint eine Mindestfeinzellenranddicke zugesichert werden.

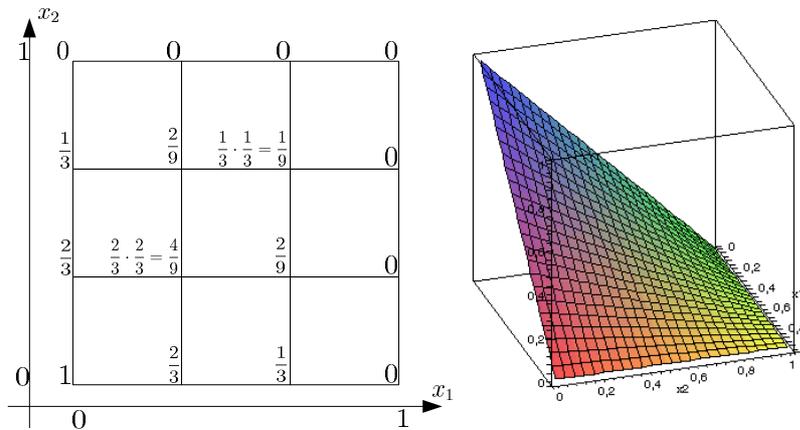


Abbildung 6.6: Die Interpolation bilinearer nodaler Ansatzfunktionen.

6.3.3 Resolutionserhöhung der Funktionsdarstellung

Interpolation nodaler Freiheitsgrade

Bereits eingeführt wurde der Begriff der Interpolation⁶. Hierbei wird die Funktionsdarstellung auf eine feinere Basis transportiert, bildlich gesprochen also die Auflösung des Funktionenraums erhöht. In [Günther 04] wird dies anhand der dort verwendeten Pagodenfunktionen demonstriert. Die Interpolation für bilineare nodale Elemente gestaltet sich aufgrund der Tensorproduktkonstruktionsvorschrift als besonders einfach. Die folgende Vorschrift gibt die Interpolation auf dem Standardelement $(\hat{x}_1, \hat{x}_2)^T \in]0, h[^2$ bezüglich der normierten Ansatzfunktion im Punkt $(0, 0)^T$ wieder:

$$\begin{aligned}
 \hat{u}(\hat{x}_1 = 0, \hat{x}_2 = 0) &= 1 \\
 \hat{u}(\hat{x}_1, h) &= u(h, \hat{x}_2) = 0 \\
 \hat{u}\left(\frac{h}{3}, 0\right) &= u\left(0, \frac{1}{3}\right) = \frac{2}{3} \\
 \hat{u}\left(\frac{h}{3}, \frac{h}{3}\right) &= \frac{2}{3} \cdot \frac{2}{3} = \frac{4}{9} \\
 \hat{u}\left(\frac{2h}{3}, \frac{h}{3}\right) &= \hat{u}\left(\frac{h}{3}, \frac{2h}{3}\right) = \frac{1}{3} \cdot \frac{2}{3} = \frac{2}{9} \\
 \hat{u}\left(\frac{2h}{3}, \frac{2h}{3}\right) &= \frac{1}{9}
 \end{aligned}$$

Analog zu bilinearen Elementen kann auch die Interpolation für komplexere Ansatzfunktionen aufgesplittet werden auf die beteiligten Grobgitterfunktionen. Allerdings bietet es sich abermals

⁶Da in diesem Kapitel ausschließlich die Konstruktion eines Relaxationsverfahrens vorliegt, mag der Übergang auf ein hierarchisches Erzeugendensystem befremdlich wirken. Tatsächlich wird dieses seine Stärken erst bei der Implementierung des Mehrgitteralgorithmus ausspielen, hier sind ja noch alle hierarchischen Werte, die nicht dem Blattlevel zuzuordnen sind, auf null gesetzt. Es macht dennoch Sinn, die Interpolation gleich im Rahmen der Gebietstraversion und der Funktionenraumdiskussion abzuhandeln, da damit die Argumentation über Ansatzfunktionen nicht über die Arbeit verstreut ist.

an, für die in [Blanke 04] vorgestellten Funktionen, oder für gar noch komplexere Gebilde, auf die Hilfe einer Computeralgebra zurückzugreifen und diese die beteiligten Interpolationsmatrizen generieren zu lassen.

Anhang B listet die Matrizen für die Druckinterpolation auf, Anhang C sind die benutzten Maple-Skripte zu entnehmen. Interpolationsskripte für die Interpolation der (augmentierten) kontinuierlichkeitserhaltenden Elemente existieren nicht, da in begleitend zu dieser Arbeit lediglich $(\theta = 0)$ -Methoden implementiert wurden. Bei diesen können \mathcal{F} und $u^{(t+\tau)}$ auf dem Feingitterlevel explizit berechnet werden.

Interpolation zellbasierter Freiheitsgrade

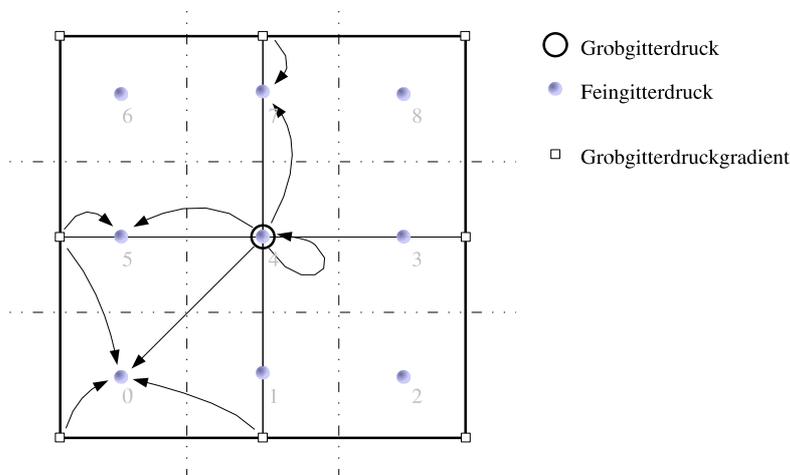


Abbildung 6.7: Illustration der Interpolation eines bilinearen versetzten Drucks durch den Operator P_p . Die Interpolation ist nur möglich, weil der Gradient des Drucks explizit auf den Gitterpunkten zwischengespeichert ist.

Die Interpolation für bilineare Druckelemente kann man direkt aus der zuvor diskutierten nodalen Interpolation übernehmen, indem man eine Translation um eine halbe Feingitterweite auf dem Interpolationsschema durchführt. Offensichtlich würde man nun aber in einer Grob-Druckzelle den Druckwert aller acht benachbarten Grob-Druckelemente benötigen. Diese Nichtlokalität ist mit dem später präsentierten Traversierungskonzept nicht vereinbar und verletzt das Lokalitätsprinzip.

Nimmt man jedoch wiederum an, dass beim Baumabstieg der Druckgradient auf der Grob-Druckzelle bereits vorliegt, so kann die Interpolation ohne Probleme vollzogen werden. Dies bedeutet, dass man sowohl den Druck als auch die Druckgradienten auf die neun Subzellen einer verfeinerten Zelle interpolieren kann, ohne die Nachbardruckwerte auf dem Grob-Druckgitter zu kennen. In einem Baumabstieg kann folglich sowohl für den Druck als auch für den Druckgradienten die Dehierarchisierung vorgenommen werden. Gelangt man auf Blattlevel an, so liegt der echte Druckwert, respektive der echte dehierarchisierte Druckgradientenwert in nodaler Darstellung vor.

Die Abbildungen 6.7 und 6.8 illustrieren diese Tatsache, der Operator P_p bezeichne die Druckprolongation, P_∇ die Gradienteninterpolation. Seien nun die Zellen wie in Abbildung 6.7 durchnummeriert, für die Gradientenfreiheitsgrade gelte die Nummerierung nach Himmelsrichtungen. Dann errechnet sich der interpolierte Druck zum Grob-Druck p als

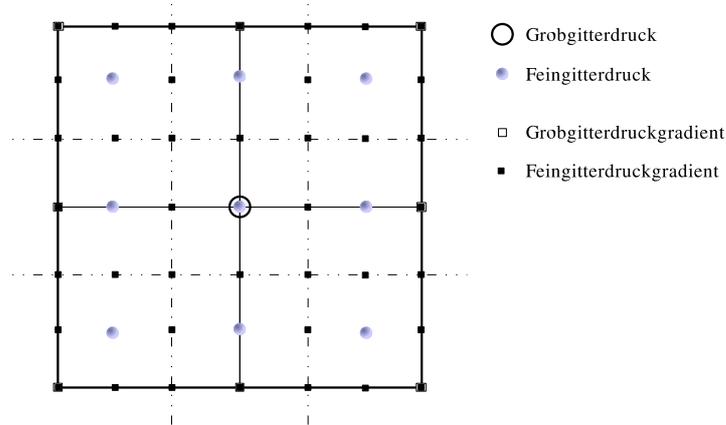


Abbildung 6.8: Die Anordnung der Fein- und Grobgitterdruckgradientenfreiheitsgrade.

$$\begin{aligned}
 p_{ip}(4) &= p \quad \text{trivial,} \\
 p_{ip}(1) &= p - \frac{2}{3} \cdot \frac{h}{2} \partial_{x_2} p(S), \\
 p_{ip}(3) &= p + \frac{h}{3} \partial_{x_1} p(O), \\
 p_{ip}(5) &= p - \frac{h}{3} \partial_{x_1} p(W) \quad \text{und} \\
 p_{ip}(7) &= p + \frac{h}{3} \partial_{x_2} p(N).
 \end{aligned}$$

Für die Diagonalaußenzellen ergibt sich ein geringfügig komplexeres Interpolationsmuster, wobei die Darstellung nicht eindeutig ist, da sich der Druck auf jeder Zelle auf zwei „Wegen“ berechnen lässt. Der Druckgradient ist ein reiner Hilfswert, der die lokale Auswertung von Funktionen ermöglicht. Insbesondere ist er damit vollständig redundant und könnte jederzeit über die Druckwerte rekonstruiert werden. Die Konsistenz der Druckgradienten mit der globalen Druckverteilung ist folglich durch korrekte Operatoren zuzusichern.

$$\begin{aligned}
 p_{ip}(0) &= p_{ip}(1) - \frac{2}{3} \frac{h}{2} \left(\frac{2}{3} \partial_{x_1} p(SW) + \frac{1}{3} \partial_{x_1} p(6) \right) \\
 &= p_{ip}(5) - \frac{h}{3} \left(\frac{2}{3} \partial_{x_2} p(SW) + \frac{1}{3} \partial_{x_2} p(S) \right)
 \end{aligned}$$

Explizit notiert ergeben sich bei stärkerer Gewichtung der x_1 -Beiträge folgende Vorschriften:

$$\begin{aligned}
p_{ip}(0) &= p - \frac{h}{3}\partial_{x_1}p(W) - \frac{2h}{9}\partial_{x_2}p(SW) - \frac{h}{9}\partial_{x_2}p(S) \\
p_{ip}(2) &= p + \frac{h}{3}\partial_{x_1}p(O) - \frac{2h}{9}\partial_{x_2}p(SO) - \frac{h}{9}\partial_{x_2}p(S) \\
p_{ip}(6) &= p - \frac{h}{3}\partial_{x_1}p(W) + \frac{2h}{9}\partial_{x_2}p(NW) + \frac{h}{9}\partial_{x_2}p(N) \\
p_{ip}(8) &= p + \frac{h}{3}\partial_{x_1}p(O) - \frac{2h}{9}\partial_{x_2}p(NO) + \frac{h}{9}\partial_{x_2}p(N)
\end{aligned}$$

Die Gradienteninterpolation ist aufgrund der höheren Zahl der ein- und ausgehenden Daten wesentlich komplexer. Hier soll auf den Anhang verwiesen werden. Problematisch erweist sich bei der Algorithmenkonstruktion eine Auflösungsreduktion, also die Restriktion, da auch hier das eigentliche Restriktionsschema analog zu Abbildung 6.6 Nachbargrobzellen affektieren würde. Es ist also wieder festzuhalten, dass der zugehörige Operator dem Lokalisierungsprinzip nicht entspricht: Insbesondere werden bei einer Modifikation des Grobgitterdruckwertes auch Druckwerte benachbarter Spacetreer-Äste beeinflusst. Dies wird zentrale Problematik in der Mehrgitterimplementierung sein.

6.3.4 Algorithmenimplementierung

Abschließend sei festgehalten, welche Daten im Gitter gehalten werden müssen: Auf den Gitterpunkten sind zunächst die in [Günther 04, Pögl 04] angesetzten Tupel um ein Duplikat für jede Raumdimension zu erweitern. Bereits erwähnt wurde, dass die gesamte dort entwickelte Systematik für $(\theta = 0)$ -Wahl hinfällig ist, weil sich dort eine hierarchische Geschwindigkeitsfortschaltung erübrigt. Auf jeden Fall benötigt wird jedoch ein Variablenduplikat für den Zeitpunkt $t + \tau$.

Über das Lokalisierungsprinzip motiviert wurde bereits, dass es Sinn macht, den Druckgradienten in den Gitterpunkten zu speichern. Es ist also dem dort abgelegten Feld ein weiterer d -Tupel für den hierarchischen Gradienten hinzuzufügen. [Günther 04, Pögl 04] halten zu jeder Unbekannten je einen hierarchischen Wert, den interpolierten Wert, der beim Baumabstieg überschrieben wird, und einen Residuumsakkumulationswert. Letzterer entfällt im Fall des Gradienten natürlich. Analog müssen auf den Untermannigfaltigkeiten (Kanten) der geometrischen Elemente Gradientenfreiheitsgrade platziert werden.

An zellbasierter Information ist zunächst einmal der hierarchische Überschuss des Drucks festzuhalten. Auf ein Datum für die interpolierten Werte und das Zellresiduum kann für die Implementierung eines Glätters verzichtet werden, schließlich kann man das Residuum sofort auswerten. Dabei halte man sich erneut vor Augen, dass der dehierarchisierte Druckgradient bei Erreichen des Blattlevels bereits vorliegt und damit der diskrete Druck-Laplace-Operator umgeschrieben werden kann:

$$M\Omega^{-1}M^T p = MP \underbrace{\nabla \hat{p}}_{\text{lag bei Abstieg bereits fertig in hierarchischer Darstellung auf den Levelgitterpunkten}}$$

Hat man nun das Feingitterresiduum berechnet, so kann man den Feingitterdruck gleich korrigieren und den Gradienten auf den Eckpunkten aktualisieren. Nun öffnen sich zwei implementierungsabhängige Optionen, wobei angenommen wird, dass ausschließlich beim ersten Abstieg auf einen Knoten der interpolierte Gradientenwert dort analog zu [Günther 04] berechnet wird: Ändert man nach einer Druckkorrektur nur den hierarchischen Überschuss, so bekommen die Nachbarzellen bei der Residuumsauswertung nichts davon mit. Man hat also einen Jakobi-Glätter implementiert. Aktualisiert man korrespondierend zum hierarchischen Gradientenwert auch den interpolierten

Gradientenwert, so wird die Korrektur gleich an die Nachbarzellen weitergegeben und man erhält einen Gauß-Seidel-Glätter.

Schreibt man den Glätter nieder, so zeigt sich, dass die eigentliche Operatorauswertung verschoben erfolgt, da Korrekturen immer gleich auf die Gradienten aufgeschlagen werden. Der Gesamtalgorithmus sei abschließend nochmals komplett dargestellt. Alle benötigten Eingabedaten sollen dabei in der Implementierung bereits vorliegen.

- Ist eine Zelle verfeinert, so interpoliere sowohl Druck als Druckgradienten und steige rekursiv in der Geometrie ab. Ansonsten
- wird das Residuum auf der Zelle über

$$res_V = -M_{local} \left(\frac{1}{\tau} u^{(t)} + \mathcal{F}(u^{(t)}) \right) + M_{local} \nabla p$$

berechnet.

- Man korrigiert nun den hierarchischen Druckanteil mit

$$p \mapsto p + \omega \text{diag}^{-1} res_V,$$

wobei das Diagonalelement entsprechend der Zahl der inneren Nachbarzellen zu wählen ist, und

- schlägt die Zellkontributionen sofort wieder auf die Kanten. Je nach Glättertypus sind dabei nur hierarchischer oder auch interpolierter Wert zu modifizieren:

$$\nabla p \mapsto \nabla p + \Omega^{-1} M_{local}^T$$

6.4 Spacetime-Traversierung mittels raumfüllender Kurven

Der Glätter und die zugrundeliegende Datenstruktur sind spezifiziert, einer Umsetzung ist also nach geeigneter Wahl eines Gitterdurchlaufes nichts mehr entgegen zu setzen. Gesucht ist nun noch eine Spacetime-Traversierung, also eine komplette Linearisierung aller Fein- und Grobgridzellen. Dabei soll mit der größten Zelle begonnen und auch wieder geendet werden, um die Interpolation umsetzen zu können. Diese Definition auf der Baumstruktur impliziert, dass insbesondere keine Sprünge im Baum durchgeführt werden dürfen, was offensichtlich mit dem viel zitierten Lokalitätsprinzip korrespondiert, und definiert damit eine Ordnung auf den Kindzellen einer jeden Grobzelle. Ist solch eine Ordnung gegeben, lässt sich der Baumdurchlauf eben auch als pre-order Traversierung auffassen.

Spricht man aus mathematischer Sicht, so korrespondiert die Baumkonstruktion mit einer Partitionierung des $]0, 1[^d$, die Linearisierung jedoch mit $]0, 1[$, wobei die Linearisierung des Hyperwürfels die Existenz einer surjektiven Abbildung $C :]0, 1[\rightarrow]0, 1[^d$ voraussetzt. Dabei fordert das „nicht springen dürfen“ im Baum die Stetigkeit dieser Abbildung, weil die Gebiete der Intervallschachtelung disjunkt sind. Dies scheint auf den ersten Blick ein Widerspruch zu sein, jedoch hat Cantor 1878 nachgewiesen, dass tatsächlich beide Mengen gleiche Kardinalität haben. Einen konstruktiven Beweis hierbei kann man über die Angabe einer *raumfüllenden Kurve* liefern:

Definition 10 Sei $C :]0, 1[\rightarrow]0, 1[^d$ stetig. Dann heißt das Bild von f **Kurve** mit dem Anfangspunkt $f(0)$ und dem Endpunkt $f(1)$.

Definition 11 Sei f eine Kurve mit $\text{img}(f) \subset \mathbb{R}^d$. Hat $\text{img}(f)$ positiven Lebesgue-Inhalt, so bezeichne man f als **raumfüllende Kurve**.

Der Übergang von der mathematischen Sichtweise auf die diskrete Gebietsdarstellung ist nicht offensichtlich, wird sich aber durch die Angabe einer raumfüllenden Kurve ergeben. Dabei sei bemerkt, dass eine Vielzahl solcher Kurven existieren. Die Wahl in diesem Projekt fiel dabei auf die Peano-Kurve [Sagan 94], da diese bestimmte bilddimensionsinvariante Eigenschaften [Günther 04, Hartmann 04, Pögl 04] Eigenschaften aufweist, und motiviert die Verwendung der Tripartition anstelle der in der Informatik gebräuchlicheren Bipartition.

6.4.1 Die Peano-Kurve

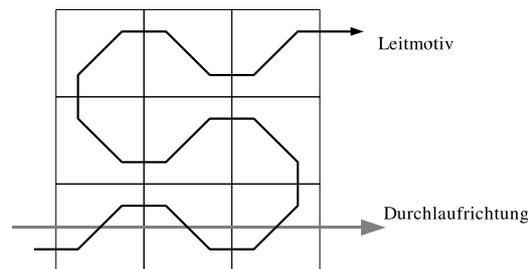


Abbildung 6.9: Das Leitmotiv der Peano-Kurve.

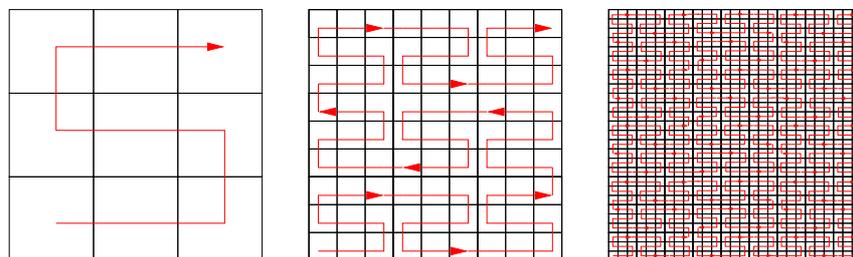


Abbildung 6.10: Konstruktion des approximierenden Polygons zur Peano-Kurve über das Leitmotiv. Das Bild ist [Günther 04] entnommen.

Eine wesentlich ausführlichere und auch verständlichere Darstellung raumfüllender Kurven ist [Sagan 94, Günther 04, Hartmann 04, Pögl 04] vorbehalten. Hier sei die Peano-Kurve für zweidimensionale Hyperwürfel konstruktiv dargestellt:

Die Durchlaufrichtung durch eine Zelle sei wie in Abbildung 6.9 festgelegt. Man nehme nun eine Tripartition des Quadrates in jeder Raumrichtung vor und passe das sogenannte *Leitmotiv* ein. Diesen Vorgang kann man rekursiv wiederholen. Das entstehende Polygon bezeichnet man auch als *approximierendes Polygon* und kann aufgrund der rekursiven Definition direkt als Traversierungsprinzip für einen Tripartition-basierenden Spacetree aufgefasst werden. Die raumfüllende Kurve ergibt sich im Grenzwertübergang.

Die zugrundeliegende Linearisierung des Baumes kann sowohl durch eine Funktion auf einer geeigneten Zahlbasis — dem Ternärsystem — wie es in [Günther 04, Sagan 94] vorexerziert wird,

als auch als Grammatik [Pögl 04] formalisiert werden. Auf letztere sei insbesondere für Implementierungsinteressierte verwiesen, da über eine Grammatik der gesamte Traversierungsalgorithmus samt Daten als Kellerautomat spezifiziert werden kann. Diese Einsicht wurde konsequent in [Hartmann 04] verwendet, während [Langlotz 04/1, Herder 05] die Lokalitätseigenschaft des approximierenden Polygons, die sich aus der Stetigkeit der Peano-Kurve ergibt, gewinnbringend in der Parallelisierung nutzen.

6.4.2 Das Stackprinzip

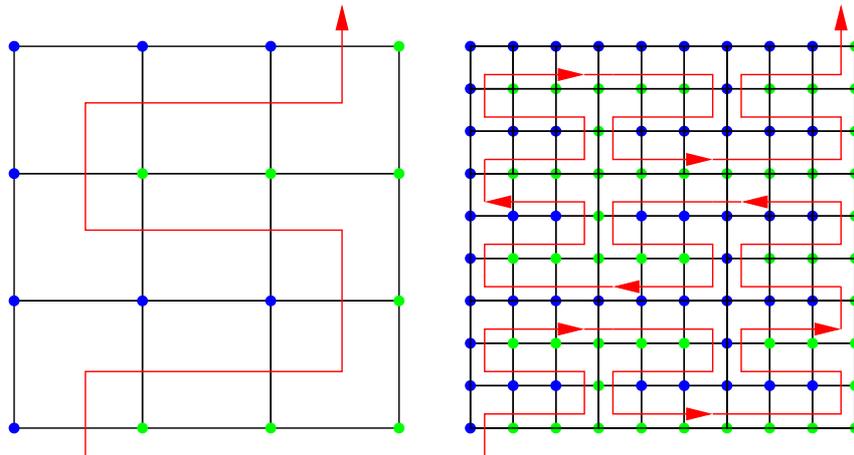


Abbildung 6.11: Die Färbung der Knoten und die damit einhergehende Stackzuweisung. Auch dieses Bild ist [Günther 04] entnommen.

Die Lokalität der Peano-Kurve impliziert, dass die Zahl der Interpolationen und Glättungsschritte minimal zu realisieren ist. Eine weitere positive Eigenschaft der Verwendung raumfüllender Kurven offenbart sich, betrachtet man zunächst einmal reguläre Gitter. Ein solches ist in Abbildung 6.10 ja gegeben. Die folgende Diskussion bezieht sich dabei ausschließlich auf nodale Daten.

Offensichtlich zerfallen die Gitterpunkte über zwei Klassen: Die Hälfte befindet sich links von der Traversierungskurve, die andere Hälfte rechts davon. Desweiteren kann man eine Ordnung \sqsubseteq_1 definieren. Es gelte $a \sqsubseteq_1 b$ genau dann, wenn a das erste mal vor b verwendet wird. Analog definiere $a \sqsubseteq_2 b$, dass a beim zweiten Bedarf vor b benutzt wird.

Theorem 8 *Es gilt*

$$a \sqsubseteq_1 b \Leftrightarrow b \sqsubseteq_2 a.$$

Die Knoten zerfallen mit dieser Definition nicht nur in zwei Klassen, beide Relationen definieren zusätzlich eine totale Ordnung auf allen Elementen einer Klasse. Obige Darstellung entspricht bekanntlich genau den Signaturinvarianten einer algebraischen Stackspezifikation. Es liegt also nahe, alle Freiheitsgrade auf Stacks zu verwalten. Die Knotenklassen wurden dabei in [Günther 04, Pögl 04] mit einer Färbungen assoziiert und die verwendeten Stacks lassen sich nun dieser Färbungen zuweisen. Auf Seite 48 wurde ein Nummerierungskonzept relativ zur Peano-Kurve definiert. Aufgrund der Klasseneigenschaft der Freiheitsgrade ist nun offensichtlich, warum dieses Konzept die Spezifikation von Algorithmen erleichtert. Einzelheiten und genauere Details sind [Günther 04] zu entnehmen.

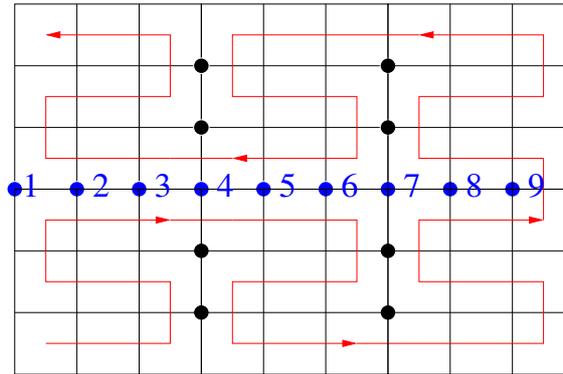


Abbildung 6.12: Motivation der Stackidentifikation nach [Günther 04].

Ein Fallbeispiel anhand der ebenfalls [Günther 04] entnommenen Abbildung 6.12 verdeutlicht diese Überlegungen: Man betrachte Punkt 3. Nach der Evaluation des Sterns auf der ersten adjazenten Zelle wird die berechnete Korrektur in 3 gespeichert. Man schreitet nach rechts fort. In der nächsten Zelle wird neben dem Datum auf 3 auch das Datum auf 4 zur Auswertung des Sterns herangezogen. Während 4 auch in der nächsten Zelle noch benötigt wird, kann nun 3 auf dem Stack landen. Nach der Auswertung der nächsten Zelle landet dann auch 4 auf eben diesem Stack. Beim Besuch der dritten zu 4 adjazenten Zelle wird nun 4 vom Stack genommen. Schlussendlich liegt bei der Auswertung der dritten und vierten zu 3 adjazenten Zelle 3 oben auf dem Stack.

Das Prinzip der stackbasierten Nodalfreiheitsverwaltung wurde hier nur für reguläre Gitter ange-rissen. [Günther 04, Pögl 04] haben dieses Konzept durch Einführung weiterer Stacks für hierarchi-sche Erzeugendensysteme erweitert. [Hartmann 04, Krahnke 04] liefern eine fundierte Darstellung⁷. Dabei ist wichtig zu erwähnen, dass die Zahl der Stacks Gittertiefen-invariant ist und nur von der Dimension des Raums abhängt. Die wichtige Eigenschaft bezüglich moderner Computerarchitek-turen ist jedoch, dass Stacks per se Cache-optimal sind.

6.4.3 Cache-optimale Algorithmen

Die Erklärung ist leicht zu liefern: Moderne Computerarchitekturen versuchen die Problematik der Speicherlatenz zu entschärfen, indem im Prinzip ein kleiner schneller Zwischenspeicher zwischen Hauptspeicher und CPU-Register geschaltet wird [Engesser 93]. Die Konsistenz der Daten sichert dabei auf Einprozessorsystemen eine Zugriffsschichtung zu.

Wird von der CPU ein Datum angefordert (das noch nicht im Cache liegt), so wird zunächst ein Duplikat dieses Datums in den Cache geladen und anschließend an die CPU durchgereicht. Bei neuerlichem Zugriff auf selbiges Datum muss damit die Übertragung von Speicher in Zwischenspeicher nicht erneut erfolgen. Interessant ist nun, dass nicht nur ein Datum im Cache liegt, sondern zu einem Datum auch immer gleich eine Menge an Daten geladen wird. Typischerweise handelt es sich dabei um einen Ausschnitt aus dem linearisierten Hauptspeicher. Solch ein Datenpaket im Cache-Speicher wird dabei als *Cache-Line* bezeichnet.

Werden die Daten in den Speicher zurückgeschrieben, so läuft dieser Schreibvorgang ebenfalls über den Cache. Der eigentliche Hauptspeicher ist also vor der CPU verborgen und kann nur

⁷Durch die dort entwickelte systematische dimensionsgenerische Methodik kann auch leicht ein Stackmanagement für kantenbasierte Freiheitsgrade entwickelt werden. Bei Entstehung des diese Arbeit begleitenden Programmes waren diese Erkenntnisse jedoch noch nicht so weitgehend ausgereift, weshalb die Kantenfreiheitsgrade jeweils ihrem südlichen respektive westlichem Nachbarknoten zugeordnet worden sind.

über die Cache-Schicht modifiziert werden. Wird jede Änderung im Cache unmittelbar in den Hauptspeicher übernommen, so bezeichnet man die Schreibstrategie als *write-through*, andernfalls als *write-back*. Dadurch wird die Datenkonsistenz entweder durch eine zur Schichtung passende Inklusionsinvariante oder ein Transaktionskonzept sichergestellt.

Muss bei einem Lesevorgang eine Cache-Line nachgeladen werden, so spricht man von einem *Cache-Miss*. Diesen kann man weiter klassifizieren nach

- *Cold-Miss* (Das Datum wurde zuvor noch nie benötigt),
- Datum wurde schon einmal geladen, dann aber aus Platzgründen verdrängt oder
- *invalid* Datum wurde im Hauptspeicher direkt (z.B. von einem Peripheriegerät) modifiziert, das Duplikat im Cache war also nicht mehr gültig.

Die ersten beiden Fälle treten bei Standardcomputern am weitaus häufigsten auf, lassen sich aber auf ein Minimum beschränken, falls man versucht, innerhalb des Speichers immer möglichst lokal zu arbeiten. Zeigerbasierte Baumstrukturen leisten genau dies nicht: Hier wird wild im Speicher umhergesprungen. Liegt dagegen eine vollständige Linearisierung der Daten vor und arbeitet man ausschließlich mit Stacks, wie das in diesem Projekt der Fall ist, so bewegt man sich im Speicher sehr lokal. [Günther 04, Pögl 04, Hartmann 04] haben gezeigt, dass sich infolge dessen die Zahl der *Cache Misses* nahe dem Minimum bewegt. Diese Erkenntnis wurde unter anderem in [Günther 05, Langlotz 04/2, Weinzierl 05] bereits publiziert. In [Neckel 05] wurde nachgerechnet, dass auch im Fall des Strömungslösers die Zahl *L2-Cache-Misses*⁸

Stichwortsammlung

Gittercharakteristika; Cache-Miss, Cold-Miss, Speicherlokalität; Spacetree, (posteriori) Adaptivität, Randverfeinerung; Macrozelle, Microzelle, Zelltypen; Hängende Punkte, Hängende Zellen, Druckinterpolation, Geschwindigkeitsinterpolation; pre-order Traversierung, raumfüllende Kurve, Peano-Kurve, approximierendes Polygon; Stackprinzip, Färbung; Datenlokalität, Cache-optimale Algorithmen

⁸Das Konzept des Zwischenschaltens von Speicher kann natürlich beliebig oft angewandt werden. Die Zwischenspeicher werden in diesem Fall durchnummeriert und als Level X Cache bezeichnet. Der *L2-Cache* ist auf heutigen Computern stets vertreten und verknüpft Zentraleinheit und Speicher, ist also der für unsere Zwecke relevante. Eine ausführlichere Darstellung findet sich in [Günther 04].

7 Mehrgitteralgorithmen

In den vorangegangenen Kapiteln wurden bereits alle Ingredienzien eines Finite-Element Programms präsentiert. Im Rahmen dieser Arbeit wurden selbige in die zu [Günther 04] gehörende Implementierung für die $(\theta = 0)$ -Methode integriert. Eine Erweiterung zu allgemeinen θ -Methoden ist dabei trivial, da die zusätzlich benötigte Funktionalität in Form eines nodalen Glätters bereits in [Günther 04, Pögl 04] implementiert war.

Erfreulicherweise sind die hier erarbeiteten theoretischen Konzepte bereits im Rahmen der Arbeit [Neckel 05] anhand praxisrelevanter Szenarien wie der Kármánschen Wirbelstraße oder der Stufenströmung erprobt worden. Dies hat zur Folge, dass der numerische Evaluationsteil in dieser Arbeit nahezu gänzlich entfallen kann, gleichzeitig aber dennoch eine Validierung der Rechenergebnisse gegen [Griebel 95, Gresho 98] erfolgt ist. Stellt man die in [Neckel 05] präsentierten Szenarien nach, so fällt auf, dass hier bereits bei groben Gittern die Berechnungszeiten astronomische Höhen erreichen. Als Lichtblick bezüglich der hohen Rechenzeiten erschien während der konzeptionellen Phasen das Einbringen der Mehrgittermethodik, die in diesem Kapitel an die Bedürfnisse des vorgestellten Algorithmus angepasst werden soll. Im späteren Arbeitsverlauf trat der Aspekt der Iterationsreduktion dann in den Hintergrund¹, andere Eigenschaften der Mehrgittermethodik erwiesen sich jedoch als elementar.

Um dies zu erarbeiten, präsentiert der einführende Abschnitt nochmals die grundlegenden Aspekte der Mehrgittertechnik, um anschließend ihre Bedeutung für das im Rahmen dieser Arbeit entwickelte Programm ausführlich zu evaluieren. Der erste Teil ist Gleichungstypus-unabhängig, der zweite Teil fokussiert sich auf die Bedeutung für die Druckgleichung. Dafür sind drei Gründe relevant: Zum Einen wurde die Implementierung eines Mehrgitters für die Geschwindigkeiten nicht umgesetzt, da ausschließlich ein $(\theta = 0)$ -Algorithmus umgesetzt wurde. Sinn und Zweck dieser Einschränkung war es, sich auf wesentliche neue Aufgaben konzentrieren zu können, insbesondere also auf die Implementierung von Glättern für versetzte Freiheitsgradanordnung. Die Umsetzung eines Mehrgitters — und das ist der zweite Grund — für nodale Größen wurde bereits in [Günther 04, Pögl 04] vorgestellt. Dabei haben [Günther 04, Pögl 04] ein additives Mehrgitter mit additivem Feinglätter² umgesetzt. Hier waren also keine neuen Aspekte zu beleuchten. Schlussendlich hat man das Glück, im Kontext der Druckgleichung für θ -Methoden eine Poissongleichung, also ein elliptisches Problem, vorliegen zu haben. Hierfür sind theoretische Aussagen und ausreichend Praxiserfahrung vorhanden, so dass man sich nicht auf dünnes theoretisches Eis begibt.

Bei der Implementierung des Mehrgitters für versetzte Gitteranordnung fällt nun auf, dass — wie bereits kurz angerissen — die Restriktion analog zur Umsetzung des diskreten Divergenzoperators $\Omega^{-1}M^T$ dem Lokalisitätsprinzip nicht entspricht, also nicht direkt vorgenommen werden kann. Es war folglich nach einer Möglichkeit zu suchen, diese Klippe der Nichtlokalität zu umschiffen. Resultat ist ein additives Mehrgitterverfahren mit multiplikativem Glätter, das mit verschränkten Restriktionszyklen arbeitet. Die Konstruktion des Verfahrens wird in diesem Kapitel vorgestellt. Der Feinglätter wurde dabei bereits auf den vorangegangenen Seiten erschöpfend beschrieben, muss

¹Tatsächlich ist wohl nur durch zeitadaptive Gitter ein deutlicher Simulationsgeschwindigkeitsvorteil zu erarbeiten. Dabei sei für Semidiskretisierungen auf [Griebel 95] oder die Thematisierung in dieser Arbeit verwiesen. Im Falle der Evaluierung einer Volldiskretisierung bietet [Hartmann 04] mit einem 4D-Code in Verbindung mit [Krahnke 04] interessante Ansätze.

²Hier sei für die Bezeichnerdefinition auf die Arbeiten [Griebel 94] und [Schöberl 99], aber auch auf Kapitel 5.2 in dieser Arbeit verwiesen.

also nicht mehr thematisiert werden. Abschließend soll dann anhand einiger einfacher Testbeispiele anhand des Driven-Cavity-Szenarios gezeigt werden, dass das Verfahren wirklich effizient im Mehrgittersinn arbeitet.

7.1 Multiskalenbetrachtung

Betrachtet man nun stationäre Iterationsverfahren, wie sie in Kapitel 5 dargestellt wurden, für eine Poissongleichung, so fällt auf, dass starke Oszillationen im Fehler sehr schnell gedämpft werden — hieraus rührt Bezeichner *Glätter*. Dies soll mit Abbildung 7.1 verdeutlicht werden und ergibt sich direkt, wenn man sich für ein eindimensionales Laplace-Problem einen Jacobi-Glätter explizit komponentenweise notiert. [Trottenberg 00] nennt diese Eigenschaft *Smoothing Principle*:

Definition 12 Smoothing Principle *Many classical iterative methods (Gauß-Seidel etc.) if appropriately applied to discrete elliptic problems have a strong smoothing effect on the error of any approximation.*

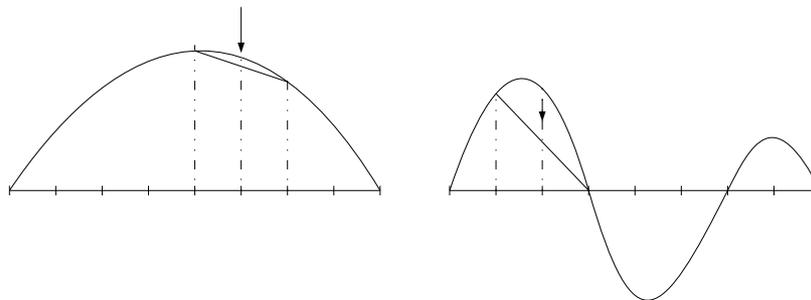


Abbildung 7.1: Das Glättungsverhalten eines stationären Iterationsverfahrens: Hochfrequente Fehler werden wesentlich schneller gedämpft, als glatte Fehlerkomponenten. Die Abbildung ist [Braess 02] entnommen.

Eine genauere analytische Aussage liefert eine Multiskalenanalyse des Laplace-Operators: Man bringe den Fehler dabei in seine Fourier-Darstellung und analysiere so die Eigenfunktionen und die zugehörigen Eigenwerte [Braess 02] der Iterationsmatrix. Trägt man zu den Fehlerfrequenzen die Dämpfung — gegeben durch den entsprechenden Eigenwert der Iterationsmatrix — an, so ergibt sich ein Bild qualitativ ähnlich zu 7.2. Offensichtlich werden je nach Wahl des Relaxationsparameters bestimmte Frequenzbereiche gut, andere weniger schnell reduziert.

Diese Erkenntnis ist Grundlage des klassischen Mehrgitterverfahrens: Man glätte die hohen Oszillationsfrequenzen des Fehlers auf dem Feingitter durch Wahl eines geeigneten Relaxationsparameters³ und transportiere den Fehler respektive das Residuum anschließend auf ein gröberes Gitter. Dort erscheinen die ursprünglich niederfrequenten Fehler nun abermals hochfrequent und man kann iterativ wieder den Glätter anwenden. Abermals sei [Trottenberg 00] bemüht, der diesen Ansatz als *coarse grid principle* bezeichnet:

³Nach der Mehrgittertheorie liefert ein Faktor $\frac{1}{3} \leq \omega \leq \frac{2}{3}$ die besten Ergebnisse. In [Günther 04, Pögl 04] wurde denn auch der Faktor $\frac{2}{3}$ empirisch als optimal ermittelt. Bei der Berechnung instationärer Probleme in dieser Arbeit erwies sich jedoch die Wahl der Untergrenze für den Relaxationsparameter als besser, weil sich hier in der Zeit keine Frequenzen hochschaukeln können. Diesen Vorteil erkaufte man sich mit etwas schlechteren Konvergenzraten im ersten Zeitschritt.

Definition 13 Coarse grid principle *A smooth error term is well approximated on a coarse grid. A coarse grid procedure is substantially less expensive (substantiell fewer grid points) than a fine grid procedure.*

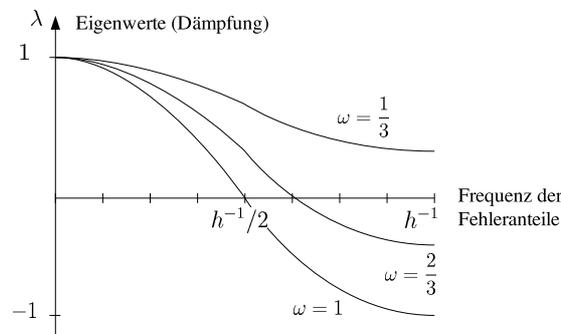


Abbildung 7.2: Qualitative Darstellung der Glättung der Fehlerfrequenzen zu verschiedenen Relaxationsparametern relativ zur Gitterweite.

Die Darstellung des Grobresiduums erfolgt also in einer größeren Basis, als sie das Feingitter definiert. Hier offenbart sich der Zusammenhang zu den hierarchischen Erzeugendensystemen: Diese liefern genau verschiedene Basisauflösungen zu einem Gitter. Sind nun im Kontext eines Mehrgitterverfahrens Grob- und Feingitter dem selben Typus zuzuordnen (also beide beispielsweise bilinear), so spricht man von einem *Galerkin-Mehrgitterverfahren*. Die Restriktion und Interpolation sind so durch die Form der Ansatzfunktionen bereits eindeutig gegeben. Dieser Ansatz wird in dieser Arbeit analog zu [Günther 04, Hartmann 04, Krahnke 04, Pögl 04] verfolgt.

Die Klassifizierung des Mehrgitterverfahrens erfolgt dabei analog zu Abschnitt 5.2 über die Komposition der Korrektur der Lösungskomponenten. Die Operatorschreibweise offenbart den Typus des Verfahrens sofort: Geht das Ergebnis der Feingitterglättung in das Grobgitterverfahren ein, so handelt es sich um ein multiplikatives Mehrgitterverfahren, wird die Glättung parallel auf beiden Leveln vorgenommen, so handelt es sich um ein additives Verfahren. [Schöberl 99] weist in diesem Kontext explizit darauf hin, dass sich unterschiedliche Verfahren für Grob- und Feingitterglätter durchaus kombinieren lassen. In [Günther 04, Pögl 04] wurde nach dieser Terminologie ein additives Mehrgitterverfahren mit einem additiven Feinglätter umgesetzt. Dort findet sich auch der empirische Nachweis, dass die Konvergenzraten der Mehrgitterkonvergenzdefinition auf Seite 66 entsprechen.

7.2 Bewertung der Mehrgittereigenschaften für instationäre Fluidsimulation

Im Rahmen dieser Arbeit wurden vier Vorteile der Anwendung eines Mehrgitterverfahrens auf der Druckgleichung identifiziert. Diese sollen nun samt einer Bewertung bezüglich praxisrelevanter Szenarien a [Griebel 95, Neckel 05] aufgelistet werden:

- Der Definition der Mehrgitterkonvergenz zeigt auf, dass sich durch Einführen einer solchen Technik signifikant bessere Rechenzeiten erreichen lassen⁴. Dies ist jedoch für die Lösung

⁴Abermals sei hier der Hinweis auf [Trottenberg 00] gestattet, dass die Reduktion der Iterationszahl immer in Relation zum zu erbringenden Aufwand zu sehen ist.

instationärer Probleme nur bedingt relevant. Zwar hat sich während der Entstehung von [Neckel 05] gezeigt, dass die Lösung der Druckpoissonsgleichung im ersten Zeitschritt sehr zeitaufwendig ist, und somit ist es natürlich wünschenswert diese Zeit drastisch zu reduzieren. Für die Folgezeit verwende man jedoch als Startiterierte der Druckpoissonsgleichung stets die Lösung des Vorschriffs. Bei kleiner Zeitschrittweite ändert sich nun die rechte Seite der Druckgleichung nur marginal und man erreicht in wenigen Nichtmehrgitterschritten eine gute Lösung. Summa summarum lässt sich sagen, dass die Zeit für die Lösung der Druckgleichung im ersten Zeitschritt für eine lange Simulationsdauer nur zu einem kleinen Teil den Gesamtaufwand beeinträchtigt. Eine andere Eigenschaft der Mehrgittermethodik affektiert die Lösung der instationären Probleme jedoch sehr wohl:

- Nach Herleitung der Methodik wird die Lösung sehr viel glatter, als bei ausschließlicher Verwendung herkömmlicher Relaxationsverfahren⁵. Diese positive Eigenschaft bleibt nur bei der Verwendung von Mehrgitterverfahren auch in der Zeit erhalten.
- Bei der Evaluierung des hier entwickelten Verfahrens hat sich gezeigt, dass sich im Rahmen der Driven-Cavity der Maximaldruck bei Verwendung des hier verwendeten Mehrgitterverfahrens relativ schnell einstellt, während die Konvergenzraten für Relaxationsverfahren hier sehr mäßig sind. Dies ist im Rückblick bei Umsetzung der Erkenntnisse auf 5.4.1 von Interesse. Da von (Halb-)Singularitäten im Druck bei nahezu allen praxisrelevanten Singularitäten auszugehen ist, handelt es sich bei dieser Eigenschaft durchaus um ein gewichtiges Argument für die Verwendung einer Mehrgittersystematik.
- Bei der Diskussion der adaptiven Gitter hat sich gezeigt, dass adaptive Gitter ohne hierarchische Erzeugendensysteme und einer darauf basierenden Mehrgittersystematik im Druckraum physikalisch nur bedingt Sinn machen.

Offensichtlich ist die Effizienz eines Mehrgitterverfahrens im Rahmen der instationären Fluidsimulation mittels einer Halbdiskretisierung nicht mehr der entscheidende Grund. Stattdessen treten qualitative Argumente in den Vordergrund. Diese waren auch die treibende Kraft bei der Entwicklung des hier vorgestellten Verfahrens. Gerade für die Arbeit mit komplex berandeten Gebieten sei jedoch noch ein Hinweis auf [Wagner 05] gestattet, der versucht, das Versagen des Mehrgitteransatzes dort durch eine intelligente Randapproximation abzufangen.

7.3 Konstruktion eines additiven Mehrgitterverfahrens für die Druckpoissonsgleichung

Verwendet man ein Galerkin-Verfahren, so wird auf dem Grobgitter die gleiche Art von Ansatzfunktionen verwendet, wie auf dem Feingitter. Ein direktes Resultat daraus ist nach [Günther 04, Trottenberg 00], dass sich der Restriktionsoperator direkt als transponierter Interpolationsoperator ergibt. Mit den in 6.3.3 eingeführten Operatoren und der Nomenklatur von Seite 3 ergibt sich

$$p = P_p \hat{p}, \quad \hat{p} = P_p^T p \quad \text{und} \quad \nabla p = P_\nabla \nabla \hat{p}, \quad \nabla \hat{p} = P_\nabla^T \nabla p.$$

Das Druckresiduum ist an den gleichen Plätzen wie die Druckfreiheitsgrade lokalisiert, entspricht also folglich einem *staggered layout*. Somit muss es über den Operator P_p^T restringiert werden.

⁵[Braess 02] empfiehlt zur Beurteilung dieser Tatsache die L^2 -Norm durch die H^1 -Norm zu ersetzen.

7.3.1 Formalisierung einer Mehrgittermethodik auf Spacetrees

Man führe rhs als Abkürzung der rechten Seite zur diskreten Druckpoissongleichung ein. Eine Mehrgittermethodik zum Feingitterproblem

$$M\Omega^{-1}M^T p = rhs$$

ist damit nach [Griebel 94] dem Lösen des linearen Gleichungssystems

$$P_p^T M\Omega^{-1}M^T P_p \hat{p} = P_p^T rhs$$

mit einem Relaxationsverfahren äquivalent. Beschränkt man sich auf ein Zweigitterproblem, so entspricht das neben der Feingitterglättung dem Lösen der Gleichung

$$M\Omega^{-1}M^T e_{3h}^{(k)} = res_{3h}^{(k,0)},$$

wobei $res_{3h}^{(k,0)}$ das restringierte Residuum zu einer Approximation $p^{(k)}$ ist. Hat man $e_{3h}^{(k)}$ gelöst oder angenähert, so wird anschließend $e_{3h}^{(k)}$ wieder auf das Feingitter interpoliert und auf die Ausgangslösung aufaddiert. Diese allgemeine Definition macht noch keine Aussagen über die Art und Weise, wie $e_{3h}^{(k)}$ zu errechnen ist. Analog zum Feingitterglätter kann man hier auch ein Iterationsverfahren mit

$$\begin{aligned} res_{3h}^{(k,i)} &:= res_{3h}^{(k,0)} - M\Omega^{-1}M^T e_{3h}^{(k,i)} \\ e_{3h}^{(k,i+1)} &= e_{3h}^{(k,i)} + \omega_{3h} diag^{-1} res_{3h}^{(k,i)} \end{aligned}$$

spezifizieren. Im vorliegenden Programm wird dabei je ein Glättungsschritt auf Fein- und Grobgitter pro top-down-depth-first Baumtraversierung durchgeführt — insbesondere wird also für eine Zweigittermethodik der Fehler $e_{3h}^{(k,0)} = 0$ angesetzt.

Man kehre zurück zur Formalisierung über ein Erzeugendensystem: Nach [Griebel 94, Schöberl 99] kann man eine beliebige Kombination additiver und multiplikativer Glätter wählen. Während [Günther 04, Pögl 04] hier permanent ein additives Verfahren vorschlagen, wird in dieser Arbeit zwischen den Leveln additiv, auf den Leveln multiplikativ vorgegangen, da die Restriktion des Residuums und die Korrektur des Feingitterwertes parallel erfolgen.

Sei der Operator S der Feingitterglätter, S_{3h} in der Tradition von [Schöberl 99] der Grobgitterglätter, so entspricht das Vorgehen der Formalisierung

$$\hat{p}^{(k+1)} = (1 - S_{3h} - S)\hat{p}^{(k)}.$$

Bei der Diskussion der Baumtraversierung wurde bereits erwähnt, dass neben dem Druck auch der Druckgradient beim Baumabstieg interpoliert wird. Dies setzt voraus, dass nach einer Grobdruckänderung auch der entsprechende Grobdruckgradient geändert wird. Die entsprechende Aktualisierungsvorschrift liefert auch hier aufgrund des Galerkin-Ansatzes die analoge Anwendung von $\Omega^{-1}M^T$ auf das grobe Gitter.

7.3.2 Restriktionsoperator auf dem Residuum

Abbildung 7.3 zeigt den Transport des Residuums für eindimensionale Probleme, Abbildung 7.4 gibt die konkrete Restriktion für zweidimensionale Elemente wieder. Beide Abbildungen offenbaren auch schon das essentielle Problem bei der Implementierung eines Mehrgitterverfahrens auf

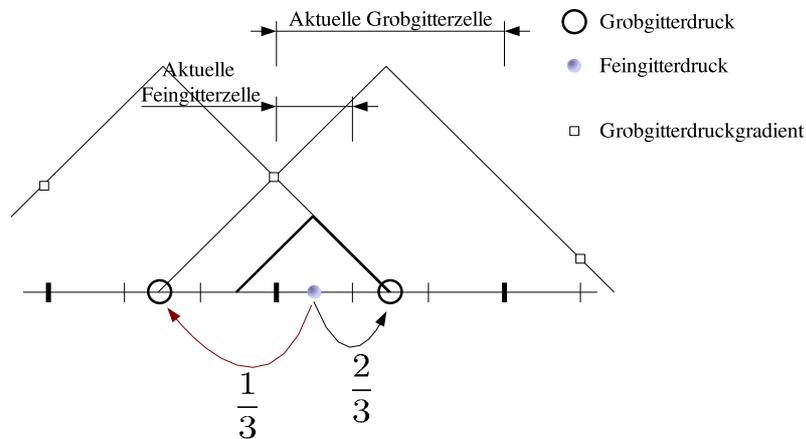


Abbildung 7.3: Übergang des Residuums auf eine Grobgitterdarstellung (Restriktion). Bereits in diesem eindimensionalen Fall lässt sich die Nichtlokalität der Restriktion erkennen (roter Pfeil). Desweiteren erkennt man, dass ein Feingitterresiduum beide Gradienten der Nachbargrobzelle beeinflusst.

dem Druck: Während sich die Interpolation des Drucks und Druckgradienten, die Auswertung des Divergenzoperators sowie die Berechnung der Gradienten der so berechneten Druckkorrektur — letztere über Akkumulation der einzelnen Zellbeiträge — problemlos in einem einzigen Baumdurchlauf realisieren lassen, kann die zur Grobgitterkorrektur benötigte Restriktion des Residuums nicht in einem Baumaufstieg durchgeführt werden, da im Operator P_p^T Feingitterzellen nicht nur den Wert ihrer Väter verändern, sondern auch zu benachbarten Grobgitterresiduen beitragen. Die Restriktion erfüllt das Lokalisitätsprinzip auf der Grobzelle nicht.

Dabei bezeichnet das Lokalisitätsprinzip der Grobzelle, dass nach einem Abarbeiten aller Kindzellen die dort errechneten Ergebnisse komplett verrechnet werden können. Da Kindzellen einer Macrozelle jedoch auch den Druck in benachbarten Nachbarzellen beeinflussen, erfüllt der Restriktionsoperator genau diese Eigenschaft nicht.

Während der Implementierung des diese Arbeit begleitenden Programmes wurden einige Ansätze zur Lösung dieses Problems angedacht: Da man beim Baumabstieg nach 6.3.4 bereits die Gradienten des Drucks interpolieren kann, war eine Idee, analog auch den Residuumsgradienten zu restringieren. Es zeigt sich aber bereits in Abbildung 7.3, dass der Ausbreitungskegel der Restriktion auf dem Gradienten sogar noch größer als der des Druckresiduums ist. Eine Modifikation der dort markierten Zelle beeinflusst nach Basisdarstellung auch die benachbarte Grobgitterzelle: Eine Änderung deren Druckwertes wiederum beeinflusst auch die Gradienten auf der äußeren Grobgitterkante. Viele Gedankenexperimente haben letztendlich immer wieder auf diese Grundproblematik geführt, und der Ansatz, über die Residuumsgradienten zu arbeiten, wurde letztendlich ad acta gelegt.

7.3.3 Einbringen des Lokalisitätsprinzips

Nach dieser induktiven Hinführung an die Problematik sei die implementierte Lösung nun zunächst für einen Zweigitteralgorithmus dargestellt. Anschließend wird versucht, den Algorithmus für mehr

7.3 Konstruktion eines additiven Mehrgitterverfahrens für die Druckpoissongleichung

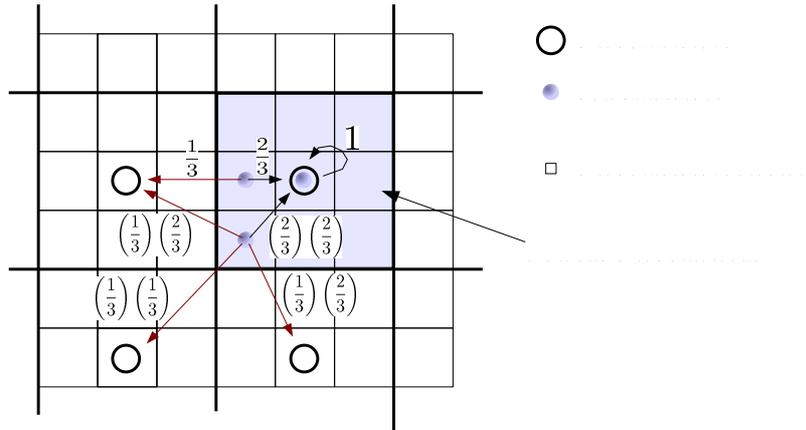


Abbildung 7.4: Restriktionsoperator P_p^T im zweidimensionalen Fall für drei exemplarisch gewählte Feingitterresiduen. Die roten Einflüsse führen zum Verletzen des Lokalisitätsprinzips der Grobzelle. Aufgrund der Symmetrie und des Tensorproduktansatzes ist der komplette Restriktionsfaktor mit dieser Zeichnung spezifiziert.

als zwei Level zu interpretieren. Ein numerischer Nachweis der Mehrgitterkonvergenz für das Driven-Cavity Szenario rundet das Kapitel ab.

Die Gradienten auf den Elementkanten und -ecken sind Hilfsgrößen, die aufgrund des Lokalisitätsprinzips eingeführt werden mussten⁶. Da auch die Restriktion mit diesem Problem zu kämpfen hat, liegt es nahe, wieder nach Wegen zu suchen, Information über die Hyperwürfelkanten und -ecken zu transportieren. Man betrachte nun die Restriktion auf den drei verschiedenen Kindzellklassen, die auch in Abbildung 7.4 dargestellt werden: Die Restriktion des Residuums auf der Mittelzelle wäre direkt im Baumaufstieg möglich, weil dieses Feinresiduum nicht auf Grobitternachbarn wirkt. Der direkte horizontale respektive vertikale Nachbar der Mittelkindzelle transportiert seinen Residuumswert an die Grobitternachbarn über je eine Kante, die Diagonalnachbarn über zwei Kanten und einen Eckpunkt.

In dem hier präsentierten Ansatz lokalisiere man nun in der Grobitterzellenmitte einen Hilfswert res_{3h} und auf jeden Hyperwürfelement, auf dem ein Gradient sitzt ebenfalls eine Hilfsvariable res_{3h} — diese Anordnung läuft völlig analog zur Platzierung der (Hilfs-)Druckgradienten also an Zellkanten und -ecken, zusätzlich jedoch existiert noch ein Wert in der Zellmitte. Terminiert der rekursive Abstieg in eine Kindzelle, steigt man also gerade aus selbiger auf, so addiere man das dort errechnete Feingitterresiduum gemäß der Gewichtung in Abbildung 7.5 auf die beteiligten Hilfswerte. Man kann nachrechnen, dass beim nächsten Abstieg in eine verfeinerte Zelle die Summe über alle neun Hilfswerte genau das restringierte Residuum ergibt. Man benötigt also zwei Gittertraversionen: Bei der ersten wird in jedem Levelaufstieg das Residuum auf die Hilfsvariablen gelegt, beim zweiten Abstieg ergibt sich das Grobresiduum, indem über die Hilfswerte addiert wird.

Theorem 9 *Verwendet man das in Abbildung 7.5 definierte Schema über Hilfsvariablen anstatt einer Restriktion nach Aufstieg aus jeder Kindzelle, so ist die Summe über alle Hilfsvariablen einer Zelle beim nächsten Abstieg genau das in 7.4 spezifizierte Grobitterresiduum auf der Grobitterzelle nach der Galerkin-Restriktion über P_p^T .*

⁶Man rufe sich in Erinnerung, dass auch der Gesamtoperator $M\Omega^{-1}M^T$ ein dem Prinzip nicht genügt hat.

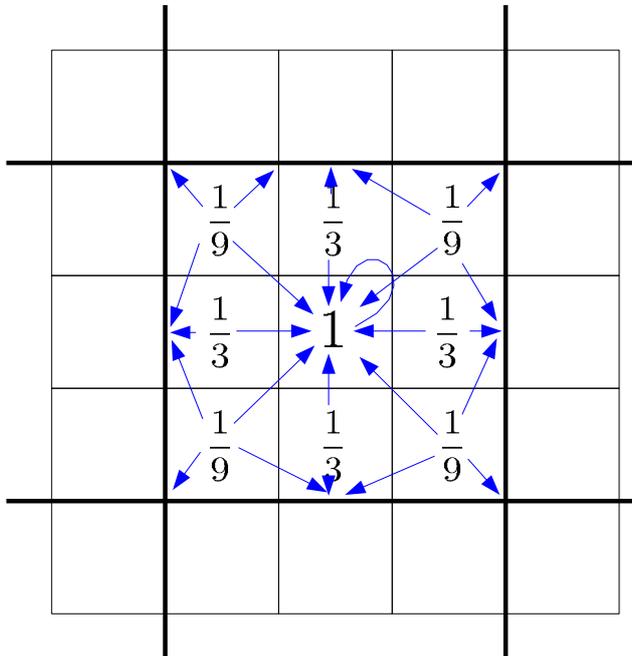


Abbildung 7.5: Transport des Feinresiduums auf Hilfsvariablen im Baumaufstieg, über die dann beim nächsten Baumabstieg summiert wird, um das Residuum zu erhalten.

Damit ist nun eine Zweigittermethodik gefunden, die nach Definition dem Lokalisierungsprinzip genügt.

Korollar 5 Die definierte Zweigittermethodik genügt dem Lokalisierungsprinzip.

7.3.4 Verschränken der Mehrgitterzyklen

Nach der hier verwendeten Terminologie muss das entwickelte Zweigitterverfahren als additiv bezeichnet werden. Will man nun mit dem hier entwickelten Speicherschema einen kompletten Mehrgitterzyklus implementieren, so muss das Gebiet mehrfach traversiert werden, da die Residuumsinformation in jedem Schritt nur einen einzigen Level nach oben propagiert wird. Abbildung 7.6 illustriert dies. Infolge dieses Faktums ist die Residuumsinformation für einen Spacetime der Höhe k erst nach $k - 1$ Schritten auf das größte Level transferiert.

Aufgrund der zugrundeliegenden Tripartition ist die Komplexität des Algorithmus damit nicht mehr in der klassischen Mehrgitterkomplexitätsklasse $O(n)$. Stattdessen handelt man sich, ersetzt man den Gitterdurchlauf durch das in 7.6 präsentierte Schema, eine Logarithmuskalierung ein.

Korollar 6 Bildet man den klassischen Mehrgitterzyklus mit dem hier präsentierten Ansatz nach, so verschlechtert sich die Mehrgitterkomplexität zum Erreichen einer gegebenen Genauigkeit von $O(n)$ auf $O(n \log_{3^d} n)$.

Dies mag für theoretische Resultat von Interesse sein. In der Praxis ist es naheliegend, immer auf jedem Level den multiplikativen Glätter anzusetzen. Arbeiten nun die Glätter $S_{3^i h}$ $i \in \mathbb{N}$ auf den einzelnen Gitterleveln, so kann man die Gesamtiteration folgendermaßen formalisieren:

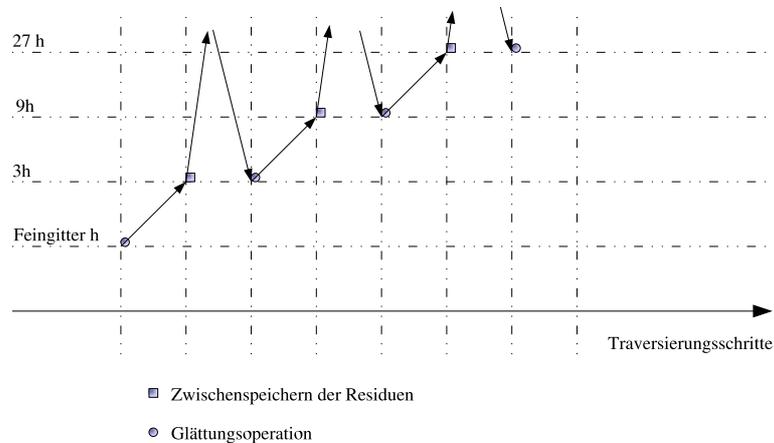


Abbildung 7.6: Umsetzung eines klassischen Mehrgitterzyklus mithilfe des Zweigitteralgorithmus, der dem Lokalitätsprinzip genügt.

$$\begin{aligned}
 \text{Eine Traversierung:} \quad S &= id - S_h \\
 \text{Zwei Traversierungen:} \quad S &= (id - S_{3h} - S_h(id - S_h)) \\
 \text{Drei Traversierungen:} \quad S &= (id - S_{9h} - S_{3h}(id - S_{3h} - S_h(id - S_h)) \\
 &\quad - S_h(id - S_h)(id - S_h)) \\
 &\dots
 \end{aligned}$$

Da der Glätter auf dem größten Level stets auf dem alten Residuum arbeitet, ist der Mehrgitteralgorithmus als verschränktes additives Verfahren zu bezeichnen. Abbildung 7.7 versucht, den unübersichtlichen Algorithmusverlauf anschaulich wiederzugeben. Es ist dabei nicht auszuschließen, dass zu solch einem Algorithmus keine befriedigende Konvergenztheorie existiert. Numerische Experimente können nichts desto trotz ausgeführt werden und offerieren vielversprechende Konvergenzraten:

7.4 Numerische Experimente

Da diese Arbeit theoretisch gehalten ist, sollen nur einige exemplarische Versuche wiedergegeben werden, die zentrale Erkenntnisse und Behauptungen dieser Arbeit belegen:

- Zunächst werden die Konvergenzraten der 5-Punkte-Elemente den der 9-Punkte-Elemente gegenübergestellt. Abbildung 7.8 zeigt die bereits theoretisch hergeleitete Erkenntnis, dass ein SOR-Verfahren auf den 5-Punkte-Elementen das Druckresiduum schneller reduziert. Die bessere Konvergenzgeschwindigkeit erkaufte man sich dabei jedoch mit womöglich und tatsächlich auftretenden Schachbrettmustern. Man beachte hierzu Abbildung 2.7. Die Grafik gibt Endresiduum zu Anfangsresiduum wieder.

7 Mehrgitteralgorithmen

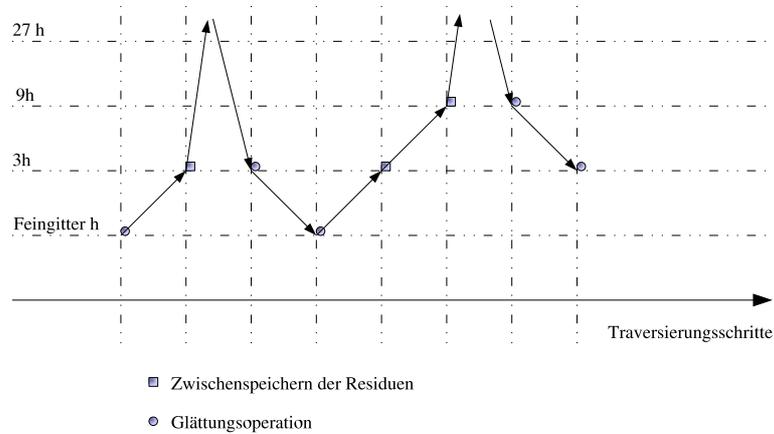


Abbildung 7.7: Ablaufschema des schlussendlich implementierten Mehrgitteralgorithmus.

- Bild 7.9 stellt die Residuenentwicklung im Initialschritt des Mehrgitteralgorithmus dem in [Neckel 05] reinen SOR-Verfahren gegenüber. Für das Mehrgitter kam $\omega = \frac{1}{3}$ zur Verwendung, für den reinen Glätter $\omega = 1.2$. Beide Faktoren wurden empirisch auf das reguläre Gitter optimiert.
- Abbildung 7.10 gibt zu gleichem Szenario die Entwicklung des Maximaldrucks wieder. Man beobachtet hier ein quasioptimales Verhalten des Mehrgitteralgorithmus. Dass der Maximaldruck mit kleinerer Minimalgitterweite ansteigt, ist auf die bessere Auflösung der Halbsingularitäten in den Gebietsecken zurückzuführen und belegt die Notwendigkeit einer dynamischen Adaptivität.
- Die folgenden Tabellen geben einige Glättungsverhalten für ($\theta = 0$ -Experimente wieder. Offensichtlich profitiert man von der Mehrgittermethodik ausschließlich zu Simulationsbeginn, beschränkt man seine Evaluierung auf die Iterationszahlen.
- Einige weitere Grafiken wurden in Anhang E verlegt: Die dort zu findenden Illustrationen zur Driven-Cavity dienen lediglich als Nachweis, dass sich Geschwindigkeit und Druck wie in [Griebel 95] entwickeln.

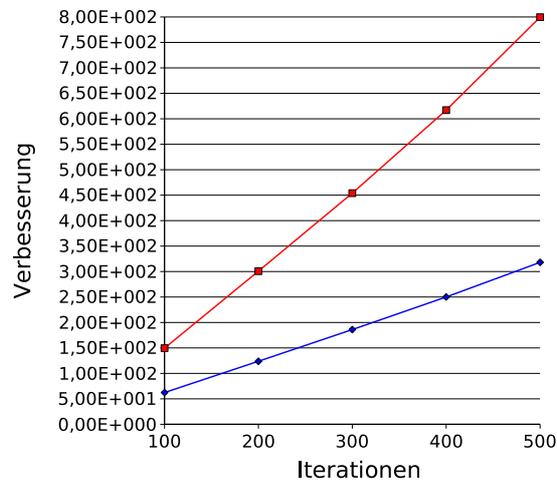


Abbildung 7.8: Druckresiduumsverbesserung für ein SOR-Verfahren mit den 5-Punkte-Elementen und den 9-Punkte-Elementen auf einem regulären 81^2 Gitter mit $\omega = 1.2$. Die bessere Konvergenzrate der 5-Punkte-Elemente ist auf das zugrundeliegende reduzierbare Gleichungssystem zurückzuführen.

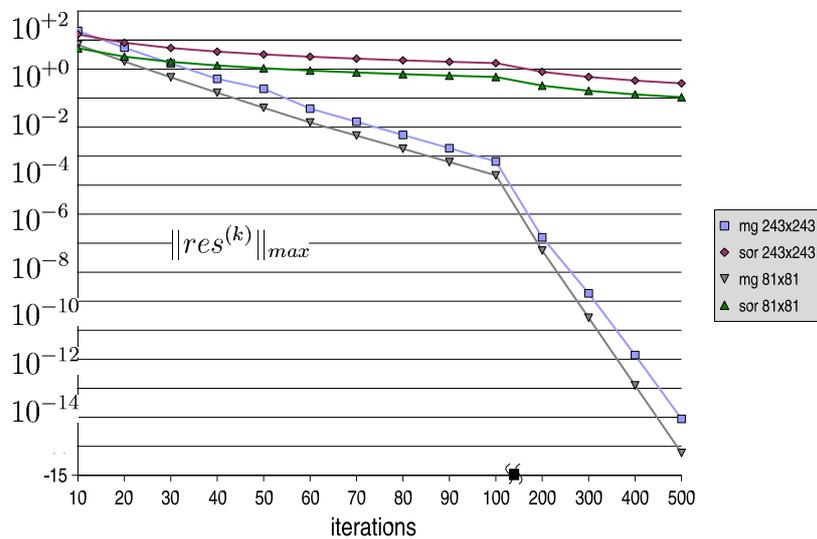


Abbildung 7.9: Konvergenzverhalten des implementierten Mehrgitteralgorithmus auf zwei verschiedenen regulären Gittern.

7 Mehrgitteralgorithmen

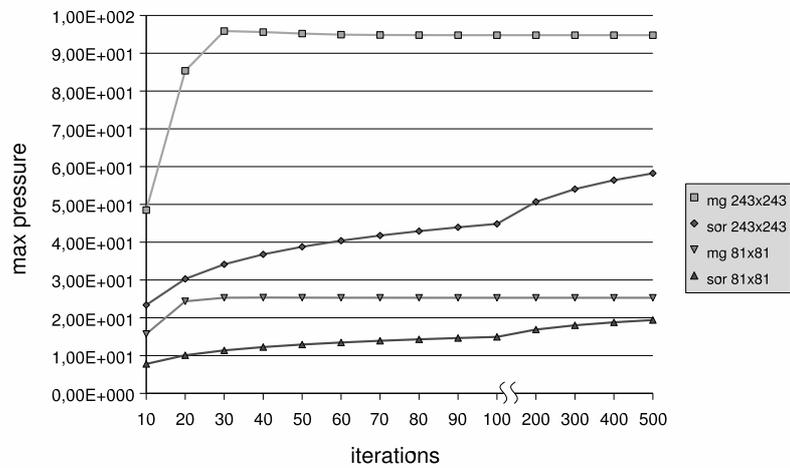


Abbildung 7.10: Entwicklung des Maximaldrucks bei SOR-Verfahren und Mehrgitterverfahren.

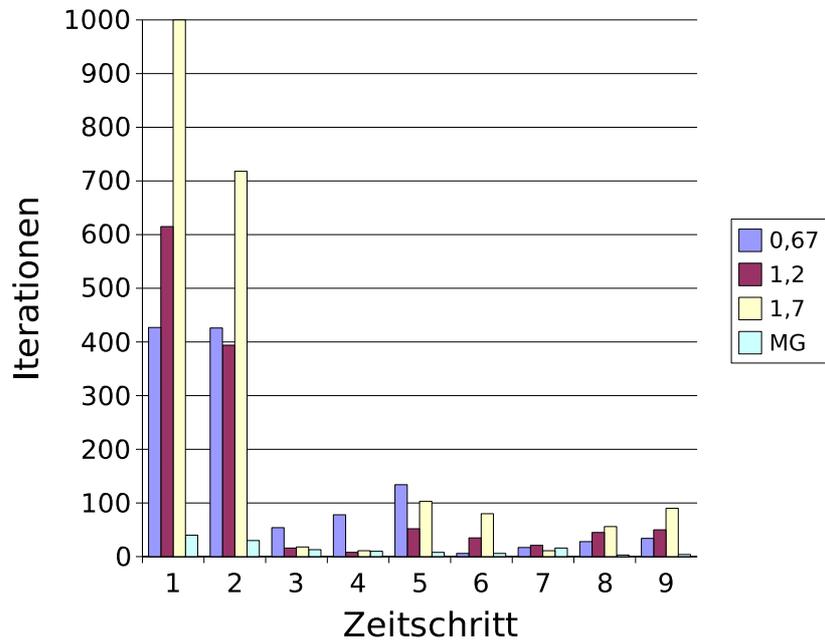


Abbildung 7.11: Benötigte Iterationszahlen. Es liegt ein reguläres 81^2 Gitter zur Reynoldszahl 10 mit einem absoluten Druckabbruchkriterium von $0.81 \cdot 10^{-3}$ zugrunde. Vorgegebene Maximaliterationenzahl war dabei 1000.

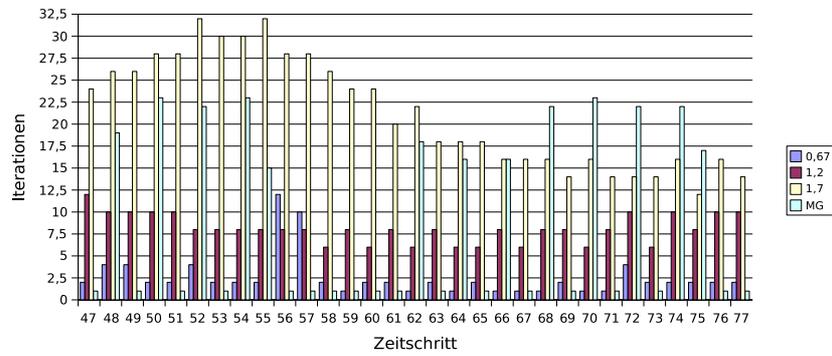


Abbildung 7.12: Weiterer Ausschnitt aus dem Verlauf des in Abbildung 7.11 beschriebenen Szenarios.

7 Mehrgitteralgorithmen

8 Fazit

Einem abschließenden Fazit sei die Rekapitulation der Ergebnisse vorangestellt. Erst anschließend können selbige kritisch beurteilt und eingeordnet werden. Diesen beiden Aspekten gilt nun das Hauptaugenmerk dieses Kapitels, das durch einen Ausblick auf interessante Gegenstände zukünftiger Forschung abgerundet werden soll.

8.1 Rekapitulation der Ergebnisse

In dieser Arbeit wurde eine neuentwickelte Basis aus [Blanke 04] erstmals in einen Algorithmus zur Lösung der Navier–Stokes Gleichungen integriert. Im zugrundeliegenden Funktionenraum ist die punktweise Divergenzfreiheit, also Inkompressibilität einer über Kontrollvolumen definierten diskreten Divergenzfreiheit äquivalent. Obwohl man sich also in Funktionenraum nicht auf die divergenzfreien Geschwindigkeitselemente zurückzieht, kann man durch eine weitere Testphase die Inkompressibilität exakt erfüllen. Dies ist verglichen mit gemischten Verfahren ein Vorteil, gleichzeitig engt man sich nicht so sehr ein, wie dies bei divergenzfreien Ansatzräumen der Fall ist. Ein physikalischer und ein mathematischer Vorteil gehen mit diesem Ansatz einher: So impliziert die punktweise Divergenzfreiheit nach [Blanke 04, Emans 03, Verstappen 01] die Energie- und Impulserhaltung im diskretisierten System. Desweiteren kann die Standard FEM Theorie mit ihren Konvergenz- und Existenzaussagen zur Anwendung kommen.

Die in [Blanke 04] entwickelten Elemente haben sich in ersten Versuchen als anfällig für die sogenannten *spurious modes* gezeigt. Aus dieser Motivation wurden sie zu den augmentierten Kontinuitäts-erhaltenden Geschwindigkeitselementen weiterentwickelt. Die resultierende Diskretisierung vereint Vorteile Finiter Volumen Schemata [Emans 03] mit der theoretischen Fundiertheit der FEM und war vor dieser Arbeit nicht bekannt.

Desweiteren wurde gezeigt, dass der zu den augmentierten Elementen passende Druckraum als echter Funktionenraum interpretiert werden kann. Resultierend hieraus konnte sowohl die Zeitfortschaltung explizit interpretiert (man erinnere sich, dass der Term \mathcal{F} explizit auf den Gitterknoten vorliegt), als auch eine Galerkin–Interpolation auf Erzeugendensystemen verwirklicht werden.

Den resultierenden Algorithmus in das von [Günther 04, Pögl 04] entwickelte Gesamtkonzept zu integrieren hat sich als nichttrivial herausgestellt und war erst zur die Aufspaltung und das versetzte Auswerten der Operatormatrizen überhaupt möglich. Doch durch diesen relativ komplexen Ansatz konnten zwei Vorteile des genannten Gesamtkonzepts auf das Programm zum Lösen der instationären Navier–Stokes Gleichungen hinübergerettet werden:

So ist das resultierende Programm cache–optimal. Insbesondere wurde die in [Günther 04] erzielte, de facto minimale L_2 –Cache–Hit–Rate erhalten. Damit lässt sich der Strömungslöser effizient auf Standardrechnern implementieren. Nicht ohne Erwähnung bleibe das in [Pögl 04] erarbeitete und in [Hartmann 04] konsequent standardisierte Vorgehen, Teile respektive komplette Freiheitsgradkeller in Dateien auszulagern. Macht dies im Dreidimensionalen nur bei extrem feinen Auflösungen oder sehr starken Singularitäten Sinn, so ist es bei der Lösung instationärer Probleme unabdingbar, nicht alle Daten im Speicher zu halten. Dies wurde in [Neckel 05] gezeigt und stimmt mit den in [Hartmann 04] gewonnenen Erkenntnissen überein.

Während das dieser Arbeit zugrundeliegende Programm [Günther 04] als Proof–of–Concept der Cache–Optimalität zu sehen ist, lag in der dreidimensionalen Implementierung [Pögl 04] des glei-

chen Ansatzes der Fokus auf großen Problemen. Hier wurde gezeigt, dass man sich aufgrund der streng deterministischen und lokalen Linearisierung der Geometrie auf zwei Bits pro Freiheitsgrad für die komplette Geometriebeschreibung zurückziehen kann. Zum Zeitpunkt der Fertigstellung dieser Arbeit war dem Autor kein anderer Ansatz bekannt, adaptive Gitter dermaßen effizient codieren zu können.

Das Aufsplitten der Operatoren zur Wahrung des Lokalitätsprinzips innerhalb der Spacetrees hat sich darüber hinaus auch aus numerischer Sicht bewährt: So war es möglich, erstmals statt den in [Günther 04, Pögl 04] verwendeten Jakobi–Glättern einen multiplikativen Feinglätter einzusetzen. Mit der erhöhten Speicher– und Algorithmenkomplexität versetzter Gitter geht also die Möglichkeit einher, die Konvergenzkonstante gegenüber bereits bekannter Ansätze zu verbessern.

Im Rahmen des hier entwickelten Lösers wurde ein Mehrgitterverfahren für den Druck entwickelt. Ein additives Mehrgitterverfahren für die Geschwindigkeit hätte man dagegen kanonisch aus [Günther 04, Pögl 04] ableiten können. Bis auf die Glattheit der Lösung in der Zeit konnten alle in 7.2 postulierten Vorteile eines Mehrgitterverfahrens für den Druck auch tatsächlich nachgewiesen werden. Gezeigt wurde in der Arbeit, dass sich ein Verfahren mit echter Mehrgitterkonvergenz nur mit einer ganzen Reihe von Hilfsüberlegungen in das aus [Günther 04, Pögl 04] entnommene Konzept integrieren lässt. Diese Überlegungen wurden in der vorliegenden Arbeit aufgezeigt.

8.2 Einordnung der Erkenntnisse

Es verbleibt letztendlich dem Leser, eine kritische Evaluierung der Erkenntnisse zu betreiben. Tatsache ist, dass in der hier vorliegenden Arbeit zwei verschiedene Traditionen erstmals integriert worden sind: Das Konzept der Traversierung von Spacetreediskretisierungen mittels raumfüllender Kurven und Stacks [Günther 04, Pögl 04] wurde mit dem der kontinuieritätserhaltenden Funktionenraumdiskretisierungen [Blanke 04, Emans 03, Verstappen 01] verheiratet. Resultat ist eine hocheffiziente Implementierung, die physikalischen Erhaltungssätzen und damit einer notwendigen Bedingung der numerischen Stabilität genügt.

Dabei haben sowohl Implementierung als auch die hier vorliegende Arbeit theoretischen respektive Demonstrationscharakter: Der Fokus lag bei beiden, neue Algorithmen und Methoden zu entwickeln und ihre Durchführbarkeit zu beweisen. Damit fehlt der für Strömungssimulationen unbedingt zu erbringende Nachweis der Praxistauglichkeit anhand von Fallstudien. Mit Sicherheit wurde hier in [Neckel 05] bereits ein zukünftiger Weg skizziert, als ausreichend dürfte das jedoch noch nicht gewertet werden können. Damit liefert diese Arbeit nicht die finale Integration vieler Erkenntnisse, sondern eher Grundlage für weitere praxisrelevante Anwendung.

Desweiteren ist dem entstandenen Programm allenfalls a [Günther 04, Pögl 04] Proof-of-Concept Charakter zuzugestehen. Viele Aspekte, die in der hier vorliegenden Arbeit angerissen oder behandelt wurden, konnten innerhalb des engen Zeitplans nicht verwirklicht werden. Man denke dabei nur an ($\theta \neq 0$)–Methoden oder die Umsetzung adaptiver Gitter.

8.3 Ausblick

Damit ist ein Umreißen zukünftiger potentieller Betätigungsfelder womöglich fruchtbarer, als eine Diskussion des Ist–Status. Die folgende Auflistung ist sowohl in Umfang als auch Reihenfolge rein subjektiver Natur, kann aber unter Umständen das Potential dieser Arbeit relativ gut umschreiben:

- Die Simulation zweidimensionaler Strömungen ist schon alleine aufgrund der besseren Visualisierungsmöglichkeiten mit Sicherheit zur Evaluierung neuer Techniken der Arbeit mit dreidimensionalen Problemen vorzuziehen. In der Praxis relevant sind jedoch gerade dreidi-

mensionale Fluidbeobachtungen. Daher ist eine Integration des entstandenen Programmes mit [Pögl 04] mit Sicherheit Arbeit wert.

- Bereits angesprochen wurde, dass die in [Krahnke 04] entwickelten Techniken im Umfeld der Strömungssimulation hervorragend integriert werden können, da sie de facto posteriori zeitadaptive Gitter erlauben. Da die in [Krahnke 04] entwickelten Ansätze in das Programm von [Pögl 04] bereits integriert worden sind, besteht die Komplexität dieser Aufgabe in der Detailmenge — eine fundamentale theoretische Hürde existiert nicht.
- Letztendlich hat man bei praxisrelevanten Aufgaben immer mit der Größe der zugrundeliegenden Programme zu kämpfen. Bereits die relativ einfachen Studien in [Neckel 05] verlangten tagelanges Rechnen. Abhilfe verspricht hier nur die konsequente Parallelisierung des Algorithmus. In [Langlotz 04/1] wurde das in [Pögl 04] entwickelte Programm auf Cluster gebracht, so dass auch hier eine Integration eher als eine Ingenieursherausforderung zu sehen ist. Gerade im Kontext der zeitadaptiven Gitter liefert dabei die auf [Langlotz 04/1] aufsetzende Arbeit [Herder 05] mit automatischer Lastverteilung interessante Kontributionen.
- Die Mehrgittereffizienz eines Algorithmus steht und fällt mit der intelligenten Behandlung des Gebietsrandes. Im Kontext der Spacetime-Diskretisierungen liefert hierzu [Wagner 05] interessante Erkenntnisse, die sich in den Strömungslöser integrieren lassen sollten. Gerade im Rahmen adaptiver Gitter können die dort entwickelten Randtechniken auch für die Behandlung von Zellen mit hängenden Knoten von Vorteil sei.
- Bisher sind ausschließlich ($\theta \neq 0$)–Methoden umgesetzt. Verallgemeinert man die Implementierung hinsichtlich impliziter Zeitverfahren, so kann man den Komplex des Lösens nichtlinearer Gleichungen einbringen. In [Hartmann 04] wurde der [Günther 04, Pögl 04] Ansatz dimensionsgenerisch abstrahiert und auch umgesetzt. Mit dem Lösen der Navier–Stokes Gleichungen im Vierdimensionalen eröffnet sich ein nahezu unbearbeitetes Forschungsgebiet.
- Unpraktikabel waren Volldiskretisierungen bisher vor allem aufgrund der unbeherrschbaren Freiheitsgradzahl. Ein substantieller Beitrag dieses Problem anzugehen, leistet mit Sicherheit die Dünngittermethode [Bungartz 98]. Diese in den Spacetime-Ansatz zu integrieren scheint naheliegend, ist aber bisher nicht geschehen.
- Die im Rahmen der Arbeit mit raumfüllenden Kurven entstandenen Programme leben mit und leiden unter der Integration einer Vielzahl komplexer Techniken. So werden Traversierung, implizite Gleichungssysteme, Mehrgittermethodik, additive Relaxationsverfahren und vieles mehr in einem Code integriert. Die daraus resultierende Komplexität auch nur ansatzweise in den Griff zu bekommen, erfordert auf Dauer eine ausgeklügelte Methodik, Abstraktionseinsatz und die Unterstützung entsprechender Werkzeuge. Ein Ansatz hierzu liefert die im Anhang vorgestellte Generierung weiter Teile des mathematischen Codes aus einer Computeralgebra heraus. In diesem Kontext kann es sinnvoll sein, in [Chibisov 02, Chibisov 03] entwickelte Verfahren bis zur Praxistauglichkeit zu erweitern.

Diese Liste ist beliebig erweiterbar, und man mag versucht sein, den Argentinier Ernesto Guevara zu zitieren:

Seien wir realistisch, versuchen wir das Unmögliche.

Jedoch wurde das Sammelsurium an Ansatzpunkten ja bereits mit einer stattlichen Zahl an Literaturverweisen versehen, so dass der Ausblick zwar eine Fülle von Aufgaben identifiziert, kein Aspekt aber als unlösbar anzusehen ist.

8 Fazit

A Assemblierungsmatrizen

In 3.2 wurde die Berechnung des Konvektions- und Diffusionsterms \mathcal{F} mittels einer schwachen Formulierung aufgezeigt. Die zugehörigen lokalen Assemblierungsmatrizen für Elemente, wie sie nach 4 zur Anwendung kommen, seien hier gegeben. Die manuelle Errechnung derselben erscheint spätestens beim nichtlinearen Konvektionsteil als Fleißarbeit und kann einem Computeralgebra-System — hier Maple — anvertraut werden. Dadurch ist auch die Möglichkeit gegeben, schnell die Eigenschaften neuer Ansatzfunktionen zu evaluieren. Die Skripte sind in C zu finden. Die Idee, ein Computeralgebra-System zum Rapid Prototyping wissenschaftlicher Software zu verwenden und die mathematische Modellierung mittels solcher Werkzeuge durchzuführen, findet sich ausführlicher in [Chibisov 02, Chibisov 03] und wird in [Ganzha 01/1, Ganzha 01/2] in die Richtung Computer Aided Software Engineering erweitert.

Die Möglichkeit zur Validierung der Maple-Ausgaben ergibt sich durch Validierung der Assemblierungsmatrizen für bilineare Elemente z.B. gegen die in [Gresho 98] angegebenen. Dabei verwendet [Gresho 98] jedoch eine Nummerierung entgegen dem Uhrzeigersinn, wie sie auf Seite 48 beschrieben ist. In dieser Arbeit kommt jedoch konsequent die lexikographische Nummerierung zur Anwendung, wobei erst die Komponenten in der ersten Dimensionsrichtung, dann die der zweiten Richtung gelistet werden. Wie in der gesamten Arbeit stellt D den Diffusionsterm dar, C den nichtlinearen Konvektionsterm und Ω schließlich die Massenmatrix¹. Alle Matrizen sind elementweise auf dem Referenzelement V gegeben:

$$\begin{aligned}\Omega_{local} &\in \mathbb{R}^{8 \times 8} \\ D_{local} &\in \mathbb{R}^{8 \times 8} \\ C_{local}(u) &\in \mathbb{R}^{8 \times 8}\end{aligned}$$

Zu jeder elementweisen Massenmatrix Ω_{local} ist die Massenmatrix $\Omega_{lumped,local}$ gegeben, die über *mass lumping* hergeleitet worden ist. Man konsultiere hierzu auch Seite 52.

A.1 Darstellung des nichtlinearen Terms

Mittels der schwachen Formulierung lässt sich der konvektive Teil auf ein Gleichungssystem der Form $C(u)u$ zurückführen, wobei die Matrix C selbst von u abhängt. In der hier vorliegenden Arbeit komme folgender Formalismus zum Einsatz, der die Zeilen der Matrix-Vektor-Multiplikation auf eine Bilinearform zurückführt.

$$C_{local}(u)u =: \begin{pmatrix} u^T C_1 \\ u^T C_2 \\ \dots \\ u^T C_8 \end{pmatrix} u =: u^T \begin{pmatrix} C_1 \\ C_2 \\ \dots \\ C_8 \end{pmatrix} u = u^T C_{local} u$$

¹Man beachte, dass Ω mit h^2 zu skalieren ist, während bei C ein h als Skalierung eingeht, D im Zweidimensionalen jedoch frei von h -Abhängigkeiten ist.

A Assemblierungsmatrizen

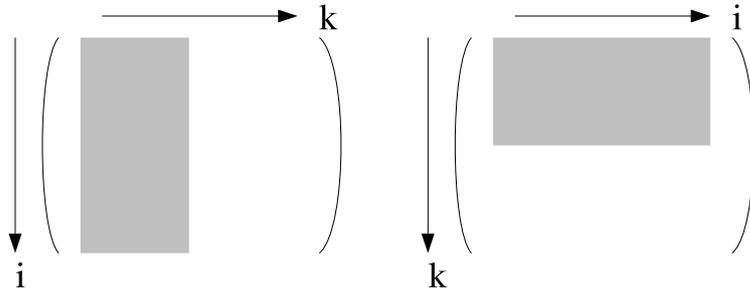


Abbildung A.1: Die konkurrierenden Ausrichtungen der Konvektionsmatrizen C_j . Die Besetztheitsstruktur ist für bilineare Elemente und $1 \leq j \leq 4$ spezifiziert.

$$\begin{aligned} C_j &\in \mathbb{R}^{8 \times 8} \quad \forall j \in \{1, \dots, 8\} \\ \Rightarrow C_{local} &\in \mathbb{R}^{8 \times 8 \times 8} = \mathbb{R}^{64 \times 8} \end{aligned}$$

Betrachtet man in 4.1.2 die Form des Konvektionsterms, so fällt auf, dass die Darstellung nicht eindeutig ist. Aufgrund der Eigenschaft

$$u^T C_j u = x = x^T = u^T C_j^T u$$

ist die Notation noch festzulegen, d.h. ob die Indices i, k jeweils Zeilen oder Spalten indizieren (siehe Abbildung A.1). In dieser Arbeit kommt konsequent die ik -Notation der linken Abbildung zur Verwendung.

A.2 Bilineare Ansatzfunktionen

Die folgenden lokalen Assemblierungsmatrizen entstammen der Arbeit mit bilinearen Geschwindigkeitselementen. Damit können sie — wie bereits erwähnt — gegen [Gresho 98] validiert werden. Praktisch angewandt werden sie u.a. in [Neckel 05]. Da vektorielle n-lineare Elemente durch einfachen Tensorproduktansatz skalarer n-linearer Elemente erzeugt werden, entkoppeln sich die Geschwindigkeitsanteile und die Matrizen lassen sich mit der Konvention

$$\begin{aligned} \Omega_{local} &=: \begin{bmatrix} \hat{\Omega} & 0 \\ 0 & \hat{\Omega} \end{bmatrix} & \Omega_{lumped,local} &= \left(\frac{1}{9} + \frac{1}{18} + \frac{1}{18} + \frac{1}{36} \right) id = \frac{1}{4} id, \\ D_{local} &=: \begin{bmatrix} \hat{D} & 0 \\ 0 & \hat{D} \end{bmatrix} & \text{und} & \\ C_{local} &=: \begin{bmatrix} \hat{C} & 0 \\ 0 & \hat{C} \end{bmatrix} \end{aligned}$$

kompakt darstellen:

$$\hat{D} = \begin{bmatrix} 2/3 & -1/6 & -1/6 & -1/3 \\ -1/6 & 2/3 & -1/3 & -1/6 \\ -1/6 & -1/3 & 2/3 & -1/6 \\ -1/3 & -1/6 & -1/6 & 2/3 \end{bmatrix}$$

$$\hat{\Omega} = \begin{bmatrix} 1/9 & 1/18 & 1/18 & 1/36 \\ 1/18 & 1/9 & 1/36 & 1/18 \\ 1/18 & 1/36 & 1/9 & 1/18 \\ 1/36 & 1/18 & 1/18 & 1/9 \end{bmatrix}$$

$$\hat{C} = \begin{bmatrix} -1/12 & 1/12 & -1/36 & 1/36 \\ -1/24 & 1/24 & -\frac{1}{72} & \frac{1}{72} \\ -1/36 & 1/36 & -1/36 & 1/36 \\ -\frac{1}{72} & \frac{1}{72} & -\frac{1}{72} & \frac{1}{72} \\ 1/12 & 1/36 & -1/12 & -1/36 \\ 1/36 & 1/36 & -1/36 & -1/36 \\ 1/24 & \frac{1}{72} & -1/24 & -\frac{1}{72} \\ \frac{1}{72} & \frac{1}{72} & -\frac{1}{72} & -\frac{1}{72} \\ -1/24 & 1/24 & -\frac{1}{72} & \frac{1}{72} \\ -1/12 & 1/12 & -1/36 & 1/36 \\ -\frac{1}{72} & \frac{1}{72} & -\frac{1}{72} & \frac{1}{72} \\ -1/36 & 1/36 & -1/36 & 1/36 \\ 1/36 & 1/36 & -1/36 & -1/36 \\ 1/36 & 1/12 & -1/36 & -1/12 \\ \frac{1}{72} & \frac{1}{72} & -\frac{1}{72} & -\frac{1}{72} \\ \frac{1}{72} & 1/24 & -\frac{1}{72} & -1/24 \\ -1/36 & 1/36 & -1/36 & 1/36 \\ -\frac{1}{72} & \frac{1}{72} & -\frac{1}{72} & \frac{1}{72} \\ -1/36 & 1/36 & -1/12 & 1/12 \\ -\frac{1}{72} & \frac{1}{72} & -1/24 & 1/24 \\ 1/24 & \frac{1}{72} & -1/24 & -\frac{1}{72} \\ \frac{1}{72} & \frac{1}{72} & -\frac{1}{72} & -\frac{1}{72} \\ 1/12 & 1/36 & -1/12 & -1/36 \\ 1/36 & 1/36 & -1/36 & -1/36 \\ -\frac{1}{72} & \frac{1}{72} & -\frac{1}{72} & \frac{1}{72} \\ -1/36 & 1/36 & -1/36 & 1/36 \\ -\frac{1}{72} & \frac{1}{72} & -1/24 & 1/24 \\ -1/36 & 1/36 & -1/12 & 1/12 \\ \frac{1}{72} & \frac{1}{72} & -\frac{1}{72} & -\frac{1}{72} \\ \frac{1}{72} & 1/24 & -\frac{1}{72} & -1/24 \\ 1/36 & 1/36 & -1/36 & -1/36 \\ 1/36 & 1/12 & -1/36 & -1/12 \end{bmatrix}$$

A.3 Kontinuitätserhaltende Ansatzfunktionen

Die folgenden Matrizen treten während der Arbeit mit den in [Blanke 04] entwickelten Elementen auf. Aufgrund der Kopplung des Vertikal- und Horizontalanteils sind die lokalen Assemblierungsmatrizen i.A. irreduzibel. Eine Ausnahme bildet der Diffusionsanteil. Man beachte die interessante Tatsache, dass sich hier als FE-Stern der standard 5-Punkte-Stern für den Laplace-Operator ergibt [Blanke 04]. Im Rahmen der kontinuieritätserhaltenden Ansatzfunktionen führe man die Notation

$$C_{local} =: \begin{bmatrix} \hat{C}_1 & \hat{C}_2 \\ \hat{C}_2 & \hat{C}_1 \end{bmatrix}$$

$$\Rightarrow \hat{C}_1, \hat{C}_2 \in \mathbb{R}^{32 \times 4}$$

zur kompakteren Darstellung ein. Abermals ergibt sich aufgrund der Spiegelungs- /Symmetrieeigenschaften der Ansatzfunktionen eine Blocksymmetrie der lokalen Assemblierungsmatrizen. Die zu erwartende Symmetrie der Massen- und Diffusionsmatrix ist zu beobachten. Allerdings sind die Diagonalblöcke aller Matrizen im Gegensatz zu den bilinearen Elementen eben keine Nullmatrizen.

$$\Omega_{local} = \begin{bmatrix} 1/8 & 1/24 & 1/24 & 1/24 & -1/24 & 0 & 0 & -1/24 \\ 1/24 & 1/8 & 1/24 & 1/24 & 0 & 1/24 & 1/24 & 0 \\ 1/24 & 1/24 & 1/8 & 1/24 & 0 & 1/24 & 1/24 & 0 \\ 1/24 & 1/24 & 1/24 & 1/8 & -1/24 & 0 & 0 & -1/24 \\ -1/24 & 0 & 0 & -1/24 & 1/8 & 1/24 & 1/24 & 1/24 \\ 0 & 1/24 & 1/24 & 0 & 1/24 & 1/8 & 1/24 & 1/24 \\ 0 & 1/24 & 1/24 & 0 & 1/24 & 1/24 & 1/8 & 1/24 \\ -1/24 & 0 & 0 & -1/24 & 1/24 & 1/24 & 1/24 & 1/8 \end{bmatrix}$$

$$\Omega_{lumped,local} = \frac{1}{4} id$$

$$D_{local} = \begin{bmatrix} 1 & -1/2 & -1/2 & 0 & 0 & 0 & 0 & 0 \\ -1/2 & 1 & 0 & -1/2 & 0 & 0 & 0 & 0 \\ -1/2 & 0 & 1 & -1/2 & 0 & 0 & 0 & 0 \\ 0 & -1/2 & -1/2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1/2 & -1/2 & 0 \\ 0 & 0 & 0 & 0 & -1/2 & 1 & 0 & -1/2 \\ 0 & 0 & 0 & 0 & -1/2 & 0 & 1 & -1/2 \\ 0 & 0 & 0 & 0 & 0 & -1/2 & -1/2 & 1 \end{bmatrix}$$

A.3 Kontinuitätserhaltende Ansatzfunktionen

$$\hat{C}_1 = \begin{bmatrix} -\frac{7}{72} & \frac{5}{72} & -1/36 & 1/24 \\ -1/36 & 1/18 & -1/36 & 0 \\ -\frac{1}{72} & 1/36 & -1/24 & 1/36 \\ -1/36 & \frac{1}{72} & 0 & \frac{1}{72} \\ \frac{7}{72} & 1/36 & -\frac{5}{72} & -1/24 \\ \frac{1}{72} & 1/24 & -1/36 & -1/36 \\ 1/36 & 1/36 & -1/18 & 0 \\ 1/36 & 0 & -\frac{1}{72} & -1/36 \\ -1/18 & 1/36 & 0 & 1/36 \\ -\frac{5}{72} & \frac{7}{72} & -1/24 & 1/36 \\ -\frac{1}{72} & 1/36 & -1/36 & 0 \\ -1/36 & \frac{1}{72} & -1/36 & 1/24 \\ 1/24 & \frac{1}{72} & -1/36 & -1/36 \\ 1/36 & \frac{7}{72} & -1/24 & -\frac{5}{72} \\ 0 & 1/36 & -1/36 & -\frac{1}{72} \\ 1/36 & 1/36 & 0 & -1/18 \\ -1/24 & 1/36 & -\frac{1}{72} & 1/36 \\ 0 & 1/36 & -1/36 & \frac{1}{72} \\ -1/36 & 1/24 & -\frac{7}{72} & \frac{5}{72} \\ -1/36 & 0 & -1/36 & 1/18 \\ 1/18 & 0 & -1/36 & -1/36 \\ \frac{1}{72} & 1/36 & -1/36 & 0 \\ \frac{5}{72} & 1/24 & -\frac{7}{72} & -1/36 \\ 1/36 & 1/36 & -\frac{1}{72} & -1/24 \\ -1/36 & 0 & -\frac{1}{72} & 1/36 \\ -1/36 & 1/24 & -1/36 & \frac{1}{72} \\ 0 & 1/36 & -1/18 & 1/36 \\ -1/24 & 1/36 & -\frac{5}{72} & \frac{7}{72} \\ \frac{1}{72} & \frac{1}{72} & 0 & -1/36 \\ 0 & 1/18 & -1/36 & -1/36 \\ 1/36 & 1/36 & -1/24 & -\frac{1}{72} \\ 1/24 & \frac{5}{72} & -1/36 & -\frac{7}{72} \end{bmatrix}$$

$$\hat{C}_2 = \begin{bmatrix} -\frac{1}{72} & 1/36 & 1/36 & -1/24 \\ \frac{1}{72} & -1/36 & \frac{1}{72} & 0 \\ -\frac{1}{72} & 0 & 1/36 & -1/36 \\ \frac{1}{72} & 0 & \frac{1}{72} & -\frac{1}{72} \\ \frac{1}{72} & -1/36 & -1/36 & 1/24 \\ \frac{1}{72} & -1/36 & 0 & 1/36 \\ -\frac{1}{72} & -\frac{1}{72} & 1/36 & 0 \\ -\frac{1}{72} & -\frac{1}{72} & 0 & 1/36 \\ -1/36 & \frac{1}{72} & 0 & \frac{1}{72} \\ 1/36 & -\frac{1}{72} & -1/24 & 1/36 \\ 0 & \frac{1}{72} & -1/36 & \frac{1}{72} \\ 0 & -\frac{1}{72} & -1/36 & 1/36 \\ 1/36 & -\frac{1}{72} & -1/36 & 0 \\ 1/36 & -\frac{1}{72} & -1/24 & 1/36 \\ \frac{1}{72} & \frac{1}{72} & -1/36 & 0 \\ \frac{1}{72} & \frac{1}{72} & 0 & -1/36 \\ -1/36 & 1/36 & \frac{1}{72} & 0 \\ -\frac{1}{72} & 1/36 & -\frac{1}{72} & 0 \\ -1/36 & 1/24 & \frac{1}{72} & -1/36 \\ -\frac{1}{72} & 0 & -\frac{1}{72} & 1/36 \\ 1/36 & 0 & -\frac{1}{72} & -\frac{1}{72} \\ 0 & 1/36 & -\frac{1}{72} & -\frac{1}{72} \\ -1/36 & 1/24 & \frac{1}{72} & -1/36 \\ 0 & 1/36 & \frac{1}{72} & -1/36 \\ 1/36 & -\frac{1}{72} & 0 & -\frac{1}{72} \\ 1/36 & -1/36 & 0 & \frac{1}{72} \\ 0 & -\frac{1}{72} & 1/36 & -\frac{1}{72} \\ 1/24 & -1/36 & -1/36 & \frac{1}{72} \\ -\frac{1}{72} & 0 & \frac{1}{72} & \frac{1}{72} \\ 0 & -1/36 & \frac{1}{72} & \frac{1}{72} \\ -1/36 & 0 & 1/36 & -\frac{1}{72} \\ -1/24 & 1/36 & 1/36 & -\frac{1}{72} \end{bmatrix}$$

A Assemblierungsmatrizen

B Matrizen für Multilevelverfahren

Der Interpolationsoperator P_p auf den augmentierten Elementen wurde im Hauptteil der Arbeit bereits präsentiert. Für den zugehörige Interpolationsoperator P_{∇} jedoch gebietet es sich schon alleine aufgrund dessen Dimension $P_{\nabla} \in \mathbb{R}^{98 \times 12}$ diesen aus dem Hauptteil herauszulösen. Aus Vollständigkeitsgründen sei er hier im Anhang platziert. Dabei wird er nach der Konvention

$$P_{\nabla} = \frac{1}{3} \cdot \begin{pmatrix} \hat{P}_1 \\ \hat{P}_2 \\ \hat{P}_3 \end{pmatrix}$$

aufgespaltet. Abermals kam das Computer Algebra System Maple zum Einsatz, um die Matrixeinträge zu generieren. Grundidee bei diesem generativen Ansatz ist es, über das geometrische Element ein „virtuelles“ Gitter der Gitterweite $\frac{h}{6}$ zu legen. Jedem Punkt dieses virtuellen Gitters werden zwei Freiheitsgrade zugeordnet, woraus die imposante Zeilenzahl der Matrix von 98 resultiert.

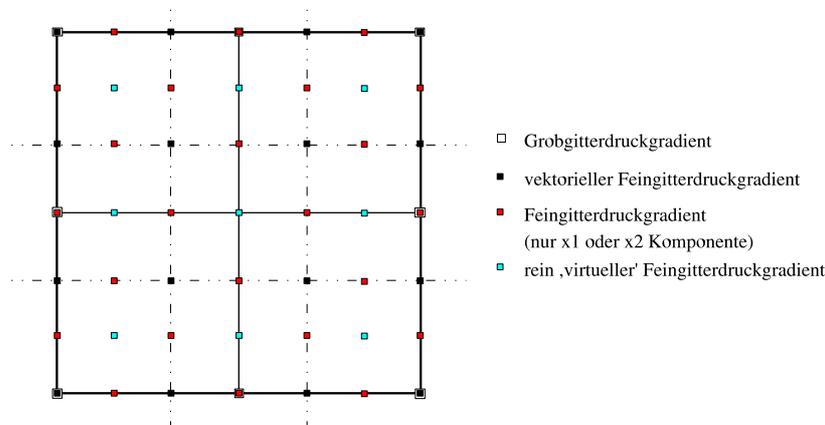


Abbildung B.1: Generierung der Interpolationsmatrix P_{∇} unter Verwendung eines virtuellen Feingitters.

Die Nummerierung erfolgt lexikographisch, wobei jeder Gradiententupel sofort linearisiert wird. Der erste Eintrag der Matrizen bezieht sich also auf die x_1 -Komponente des linken oberen Knotens, der zweite Eintrag auf die x_2 -Komponente. Abbildung B.1 zeigt den verwendeten Ansatz. Die resultierende Matrix ist dünnbesetzt, verwendet man die Konvention $\perp \mapsto 0$.

B Matrizen für Multilevelverfahren

C Maple Skripte

C.1 Die Elementmatrizen

```
> restart;
> with(LinearAlgebra):
> with(plots):
```

C.1.1 Definition der Ansatzfunktionen:

> Definiere eine (Element-)Ansatzfunktion, die 1 auf (0,0) ist, auf den anderen
> drei Punkten jedoch keine Werte annimmt. Dabei wird die Konstruktion der
> anderen sieben Ansatzfunktionen spaeter auf diese zurueckgefuehrt,
> indem entweder entsprechende Drehungen ausgefuehrt werden
> oder die Komponenten permutiert werden.

Bilineare Ansatzfunktionen:

```
> h:=1;
> phiX1:=(x1,x2)->piecewise(
> x1<=0 or x2<=0 or x1>=h or x2>=h, 0,
> (1-(x1/h))*(1-(x2/h))
> );
> phiX2:=(x1,x2)->0;
> K:=8;

> Nun werden ausgehend von der 'Basisansatzfunktion' alle anderen
> Ansatzfunktionen erstellt. Die Ansatzfunktionen 1 bis 4 dienen der
> ersten Komponente, 5 bis 8 behandeln die zweite
> Geschwindigkeitskomponente. Folglich sind 5 bis 8 durch Permutation
> der Komponenten auf 1 bis 4 zurueckzufuehren. Die Nummerierung
> der Freiheitsgrade entspricht der lexikographischen Ordnung.
> phi[1] := (x1,x2)-> < phiX1(x1,1-x2), phiX2(x1,1-x2) >;
> phi[2] := (x1,x2)-> < phiX1(1-x1,1-x2), phiX2(1-x1,1-x2) >;
> phi[3] := (x1,x2)-> < phiX1(x1,x2), phiX2(x1,x2) >;
> phi[4] := (x1,x2)-> < phiX1(1-x1,x2), phiX2(1-x1,x2) >;
> phi[5] := (x1,x2)-> < phiX2(x1,1-x2), phiX1(x1,1-x2) >;
> phi[6] := (x1,x2)-> < phiX2(1-x1,1-x2), phiX1(1-x1,1-x2) >;
> phi[7] := (x1,x2)-> < phiX2(x1,x2), phiX1(x1,x2) >;
> phi[8] := (x1,x2)-> < phiX2(1-x1,x2), phiX1(1-x1,x2) >;

> Diese Umnummerierung entspricht der Gresho-Sani-Notation und darf
> nicht verwendet werden, falls man die errechneten Matrizen in das Programm
> integriert. Sie dienen ausschliesslich der manuellen Verifikation.
> phi[1] := (x1,x2)-> < phiX1(x1,x2), phiX2(x1,x2) >;
> phi[2] := (x1,x2)-> < phiX1(1-x1,x2), phiX2(1-x1,x2) >;
> phi[3] := (x1,x2)-> < phiX1(1-x1,1-x2), phiX2(1-x1,1-x2) >;
> phi[4] := (x1,x2)-> < phiX1(x1,1-x2), phiX2(x1,1-x2) >;
> phi[5] := (x1,x2)-> < phiX2(x1,x2), phiX1(x1,x2) >;
> phi[6] := (x1,x2)-> < phiX2(1-x1,x2), phiX1(1-x1,x2) >;
> phi[7] := (x1,x2)-> < phiX2(1-x1,1-x2), phiX1(1-x1,1-x2) >;
> phi[8] := (x1,x2)-> < phiX2(x1,1-x2), phiX1(x1,1-x2) >;
```

Kontinuitaetserhaltende Ansatzfunktionen:

```

> h:=1:
> K:=8:
> phiX1:=(x1,x2)->piecewise(
> x1<=0 or x2<=0 or x1>=h or x2>=h, 0,
> x1>=x2 and x1<=h-x2, 1-1/h*x1-1/(2*h)*x2,
> x1>=x2 and x1>=h-x2, 0.5-1/(2*h)*x1,
> x1<=x2 and x1>=h-x2, 0.5-1/(2*h)*x2,
> x1<=x2 and x1<=h-x2, 1-1/(2*h)*x1-1/h*x2,
> -1
> ):
> phiX2:=(x1,x2)->piecewise(
> x1<=0 or x2<=0 or x1>=h or x2>=h, 0,
> x1>=x2 and x1<=h-x2, x2/(2*h),
> x1>=x2 and x1>=h-x2, 0.5-x1/(2*h),
> x1<=x2 and x1>=h-x2, 0.5-x2/(2*h),
> x1<=x2 and x1<=h-x2, x1/(2*h),
> -1
> ):
> Analog zu den bilinearen Ansatzfunktionen werden die
> neuen Ansatzfunktionen zusammengebastelt. Allerdings ist
> die Negation in den phiX2 Komponenten zu beachten!
> phi[1] := (x1,x2)-> < phiX1(x1,1-x2), -phiX2(x1,1-x2) >:
> phi[2] := (x1,x2)-> < phiX1(1-x1,1-x2), phiX2(1-x1,1-x2) >:
> phi[3] := (x1,x2)-> < phiX1(x1,x2), phiX2(x1,x2) >:
> phi[4] := (x1,x2)-> < phiX1(1-x1,x2), -phiX2(1-x1,x2) >:
> phi[5] := (x1,x2)-> < -phiX2(x1,1-x2), phiX1(x1,1-x2) >:
> phi[6] := (x1,x2)-> < phiX2(1-x1,1-x2), phiX1(1-x1,1-x2) >:
> phi[7] := (x1,x2)-> < phiX2(x1,x2), phiX1(x1,x2) >:
> phi[8] := (x1,x2)-> < -phiX2(1-x1,x2), phiX1(1-x1,x2) >:
>
> Gibt die Grundfunktionen aus:
> phiX1(x1,x2);
> phiX2(x1,x2);
>
> Plottet die Grundfunktionen:
> plot3d(phiX1(x1,x2),x1=-0.5..1.5,x2=-0.5..1.5,axes=boxed);
> plot3d(phiX2(x1,x2),x1=-0.5..1.5,x2=-0.5..1.5,axes=boxed);

```

C.1.2 Berechnung der Assemblierungsmatrizen:

```

> Berechnung der lokalen Assemblierungsmatrix. Die Berechnungsvorschrift
> gestaltet sich recht komplex, weil die Gebietsunterteilung manuell
> erfolgen muss. Andernfalls kann Maple das piecewise nicht auswerten.
> Wir teilen das Quadrat in vier Dreiecke und integrieren in je zwei Zeilen
> ueber diese Dreiecke. Dabei beginnen wir mit dem unten liegenden
> und gehen im Uhrzeigersinn vor.
>
> integrate := proc(T)
> int( int( T, x2=0..x1 ), x1=0..0.5) +
> int( int( T, x2=0..1-x1 ), x1=0.5..1) +
> int( int( T, x1=0..x2 ), x2=0..0.5) +
> int( int( T, x1=0..1-x2 ), x2=0.5..1) +
> int( int( T, x2=1-x1..1 ), x1=0..0.5) +
> int( int( T, x2=x1..1 ), x1=0.5..1) +
> int( int( T, x1=1-x2..1 ), x2=0..0.5) +
> int( int( T, x1=x2..1 ), x2=0.5..1);
> end proc:
>
> for j from 1 to K do
> for i from 1 to K do
> tempOmega := phi[i](x1,x2)[1]*phi[j](x1,x2)[1] + phi[i](x1,x2)[2]*phi[j](x1,x2)[2];
> matrixOmega[j,i] := integrate(tempOmega);
> end do;
> end do;

```

```

> for j from 1 to K do
> for i from 1 to K do
> tempD := diff(phi[i](x1,x2)[1],x1)*diff(phi[j](x1,x2)[1],x1)
> + diff(phi[i](x1,x2)[1],x2)*diff(phi[j](x1,x2)[1],x2)
> + diff(phi[i](x1,x2)[2],x1)*diff(phi[j](x1,x2)[2],x1)
> + diff(phi[i](x1,x2)[2],x2)*diff(phi[j](x1,x2)[2],x2);
> matrixD[j,i] := integrate(tempD);
> end do;
> end do;

> Die Konvektionsmatrix:
> for j from 1 to K do
> for l from 1 to K do
> for k from 1 to K do
> tempC := phi[l](x1,x2)[1] * diff(phi[k](x1,x2)[1],x1) * phi[j](x1,x2)[1]
> + phi[l](x1,x2)[2] * diff(phi[k](x1,x2)[1],x2) * phi[j](x1,x2)[1]
> + phi[l](x1,x2)[1] * diff(phi[k](x1,x2)[2],x1) * phi[j](x1,x2)[2]
> + phi[l](x1,x2)[2] * diff(phi[k](x1,x2)[2],x2) * phi[j](x1,x2)[2];
> matrixC[(j-1)*K+1,k] := integrate(tempC);
> end do;
> end do;
> end do;

> Ausgabe der Matrizen:
> Matrix(K,matrixOmega);
> Matrix(K,matrixD);
> Matrix(K*K,K,matrixC);

```

C.2 Die Multileveloperatoren

```

> restart:
> with(LinearAlgebra):

```

Berechnet die Eintraege fuer die Interpolationsmatrix. Dabei wird ein virtuelles Gitter ueber die Standardzelle mit einer Unterteilungsstufe gelegt, dessen Schrittweite gerade 1/6 ist. Jedem virtuellen Gitterpunkt werden zwei Freiheitsgrade (x1, x2-Richtung) assoziiert. Diese zu grosse Matrix muss im Programm natuerlich wiederreduziert werden.

```

> restart:
> numberOfRows := 7*7*2;
> numberOfCols := 12;
> for row from 1 to numberOfRows do
> for col from 1 to numberOfCols do
> interpolationMatrix[row,col] := 0.0;
> end do;
> end do;

```

C Maple Skripte

```

> Interpoliert den x1-Gradienten auf dem Element B, wenn man index10 auf den
> Gradientenindex der rechten Kante setzt und index11 auf den den x1-Gradienten
> auf der rechten oberen Ecke des Vaterelements.
> interpolateX1onB := proc( index10, index11, startIndex, directionX, directionY )
> global interpolationMatrix;
> interpolationMatrix[startIndex+directionX*2 + directionY*14*0, index11] := 1.0;
> interpolationMatrix[startIndex+directionX*6 + directionY*14*0, index11] := 1.0;
> interpolationMatrix[startIndex+directionX*2 + directionY*14*1, index11] := 2.0/3.0;
> interpolationMatrix[startIndex+directionX*2 + directionY*14*1, index10] := 1.0/3.0;
> interpolationMatrix[startIndex+directionX*6 + directionY*14*1, index11] := 2.0/3.0;
> interpolationMatrix[startIndex+directionX*6 + directionY*14*1, index10] := 1.0/3.0;
> interpolationMatrix[startIndex+directionX*2 + directionY*14*2, index11] := 1.0/3.0;
> interpolationMatrix[startIndex+directionX*2 + directionY*14*2, index10] := 2.0/3.0;
> interpolationMatrix[startIndex+directionX*6 + directionY*14*2, index11] := 1.0/3.0;
> interpolationMatrix[startIndex+directionX*6 + directionY*14*2, index10] := 2.0/3.0;
> interpolationMatrix[startIndex+directionX*2 + directionY*14*3, index10] := 1.0;
> interpolationMatrix[startIndex+directionX*6 + directionY*14*3, index10] := 1.0;
> end proc:
> interpolateX2onD := proc( index00, index10, startIndex, directionX, directionY )
> global interpolationMatrix;
> interpolationMatrix[startIndex+directionX*0 + directionY*14*1, index00] := 1.0;
> interpolationMatrix[startIndex+directionX*2 + directionY*14*1, index00] := 2.0/3.0;
> interpolationMatrix[startIndex+directionX*2 + directionY*14*1, index10] := 1.0/3.0;
> interpolationMatrix[startIndex+directionX*4 + directionY*14*1, index00] := 1.0/3.0;
> interpolationMatrix[startIndex+directionX*4 + directionY*14*1, index10] := 2.0/3.0;
> interpolationMatrix[startIndex+directionX*6 + directionY*14*1, index10] := 1.0;
> interpolationMatrix[startIndex+directionX*0 + directionY*14*3, index00] := 1.0;
> interpolationMatrix[startIndex+directionX*2 + directionY*14*3, index00] := 2.0/3.0;
> interpolationMatrix[startIndex+directionX*2 + directionY*14*3, index10] := 1.0/3.0;
> interpolationMatrix[startIndex+directionX*4 + directionY*14*3, index00] := 1.0/3.0;
> interpolationMatrix[startIndex+directionX*4 + directionY*14*3, index10] := 2.0/3.0;
> interpolationMatrix[startIndex+directionX*6 + directionY*14*3, index10] := 1.0;
> end proc:
> interpolateX1onB( 1, 12, 7+14*3, -1, -1): #x1 auf Element A
> interpolateX1onB( 4, 6, 7+14*3, 1, -1): #x1 auf Element B
> interpolateX1onB( 7, 6, 7+14*3, 1, 1): #x1 auf Element C
> interpolateX1onB( 10, 12, 7+14*3, -1, 1): #x1 auf Element D
> interpolateX2onD( 3, 2, 7+14*3+1, -1, -1): #x2 auf Element A
> interpolateX2onD( 3, 5, 7+14*3+1, 1, -1): #x2 auf Element B
> interpolateX2onD( 9, 11, 7+14*3+1, -1, 1): #x2 auf Element C
> interpolateX2onD( 9, 8, 7+14*3+1, 1, 1): #x2 auf Element D
> for row from 1 to numberOfRows do
> for col from 1 to numberOfCols do
> scaledInterpolationMatrix[row,col] := round(interpolationMatrix[row,col] * 3.0);
> end do;
> end do;
> Matrix(numberOfRows,numberOfCols,scaledInterpolationMatrix);
> Matrix(numberOfRows,numberOfCols,interpolationMatrix);

```

D A Cache-aware Multigrid Navier-Stokes Solver

Der folgende Anhang beinhaltet ein Paper, das am 13.1.2005 anlässlich der *Twelfth Copper Mountain Conference On Multigrid Methods* unter dem Titel *A Cache-aware Multigrid Navier-Stokes Solver* eingereicht wurde. Er umfasst den kompletten Artikel und fasst damit den Inhalt dieser Arbeit nochmals kompakt in Englischer Sprache zusammen, wurde jedoch passend zu dieser Arbeit umformatiert und verwendet aus Konsistenzgründen das hiesige Literaturverzeichnis.

Aufgrund der geforderten Maximalseitenzahl von zehn Seiten wurden freilich nur einige im Rahmen dieser Veranstaltung interessante Aspekte beleuchtet, jedoch nicht näher ausgeführt. Insbesondere sind die interessanten Mehrgitterimplementierungsdetails, die Darstellung der augmentierten [Blanke 04]-Elemente und die Behandlung allgemeiner θ -Methoden nicht Gegenstand der Abhandlung.

Man beachte, dass sich bei der Darstellung der Poisson-Gleichung ein Vorzeichenfehler unterlaufen ist. Auch weitere Inkonsistenzen mit einigen in der Arbeit dargestellten Fakten ist nicht auszuschließen — schließlich befanden sich sowohl Implementierung als auch schriftliche Ausarbeitung noch in relativ frühen Phasen als dieses Schriftstück verfasst worden ist. Trotz Fehlen einer errata-Liste sollte der Anhang doch einen guten Überblick für den Leser mit wenig Zeit liefern.

D.1 Abstract

Sophisticated numerical algorithms like multigrid techniques on adaptive grids require hierarchical multilevel data, which are commonly organized in tree structures. The resulting non-locality of interconnections between data causes jumps in the memory space during the run of the algorithm. This makes an efficient usage of cache-hierarchies and prefetching algorithms impossible. In this paper, we present a Navier-Stokes solver using an additive multigrid algorithm for the pressure equation resulting from Chorin's projection method. We achieve a very high cache-efficiency by the construction of particular linear data structures based on the concepts of space-filling curves and space-trees. In fact, we can avoid memory jumps completely and, in addition, we get along with only two bits per degree of freedom for administrative information — one for the geometry description and one for the description of the refinement structure. Last but not least new velocity elements are introduced that fulfill the side condition (incompressibility) pointwise. Thus, the algorithm becomes energy and momentum preserving in terms of spatial discretization.

D.2 Introduction

The aim of this paper is to develop a stable and — both in terms of numerical methods and hardware usage — efficient algorithm to solve the Navier-Stokes equations

$$\partial_t u + (u \nabla) u - \frac{1}{Re} \Delta u + \nabla p = 0, \quad (\text{D.1})$$

$$\operatorname{div} u = 0 \quad (\text{D.2})$$

for incompressible flows in two dimensions. To achieve this, we use a special Finite-Element discretization of the momentum equation enabling us to satisfy the continuity equation pointwise and to preserve energy and momentum. We combine it with a sophisticated algorithm and data structure allowing a very efficient implementation of multigrid methods on adaptive grids.

In more detail, we use Chorin's projection method [Chorin 68] combined with an explicit Euler step for simplicity. The explicit Euler step can easily be replaced by more accurate explicit time steps [Emans 03]. This approach leads to a sequence of decoupled PDEs where one is able to calculate the pressure explicitly at any time step.

Our new continuity preserving shape functions for the Finite-Element discretization of the spatial derivatives of the momentum equation enable us to fulfill the continuity equation pointwise. Together with the duality of the discrete gradient and divergence operators this leads to a stable, both energy and momentum preserving system in terms of spatial discretization. Because our discrete solution fulfills the continuous continuity equation exactly, convergence theory is much simpler and does not use the theory of mixed finite element. One can easily observe that a discrete solution in the vicinity of the exact solution exists. Applying Ceas lemma one gets error bounds for the discrete Galerkin solution.

To solve the Poisson equation for the pressure in each time step, we use a multigrid scheme implemented in a very efficient manner in terms of storage usage and access. Hereby, efficiency is achieved by using linear data structures with locally deterministic access. Thus, in contrast to many other approaches, we end up with an algorithm which is highly efficient both in terms of numerics and in terms of hardware usage.

Section D.3 shortly describes the essentials of the grid used, the location of data within the grid and the traversal order of the grid cells. In section D.4, we define our continuity preserving Finite-Element basis functions. The discretization of the pressure equation and the ingredients of the multigrid algorithm are described in D.5. Finally, we give some results for the driven-cavity scenario.

D.3 Space-Trees, Cell-oriented Operator Evaluation and a Cache-aware Hierarchical Domain Traversal

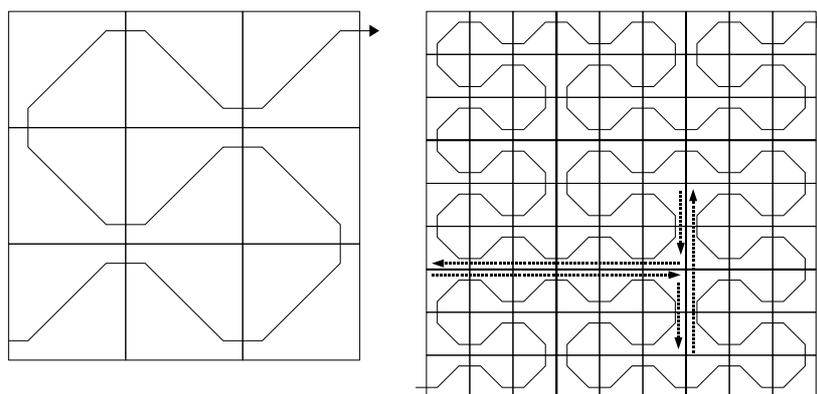


Abbildung D.1: The Peano space filling curve used for the traversal of the spacetree grid. The right picture indicates the processing order of some selected vertices.

The algorithmic concept of this paper is based on the interplay of space-filling curves and space-tree discretizations of a quadratic domain [Frank 00, Sagan 94]. Thus, the computational domain is embedded into the unit square and the grid is created by recursive decomposition of each cell (square) into nine subcells. The refinement rules depend on the geometry and numerical requirements. This allows the construction of very flexible adaptive grids without losing the structuredness of the grid.

During our computation, we process the grid cell-wise and perform a cell-wise operator evaluation [Braess 02]. Hereby, the velocity degrees of freedom are located on the cells vertices and the cells edges (see figure D.2). Thus, we have to establish an algorithm that accesses the vertex and edge data of the cells from suitable data structures in a very efficient way. The first ingredient of such an algorithm is the definition of a traversal order of the cells:

Using a structured grid as defined above, one is able to traverse the grid along the corresponding approximative polygon of the Peano curve [Sagan 94] — a space-filling curve. The recursive definition of the space-filling curve induces an unique ordering of grid cells for adaptive space-tree grids in a top-down-depth-first manner. Figure D.1 shows an example for a regular grid with the fitting Peano polygon.

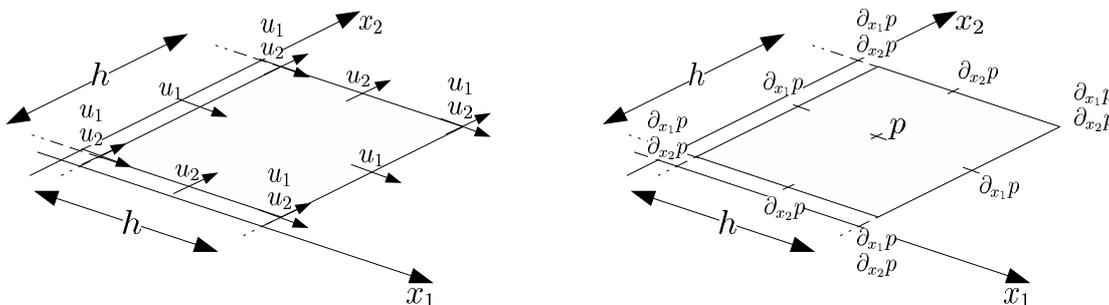


Abbildung D.2: Arrangement of the degrees of freedom per grid cell for both velocity, pressure and pressure derivatives.

If we now examine the order vertex data are used during our traversal, we observe a property which will be fundamental for the construction of our data structures: Let \sqsubseteq_1 be the order the vertices are passed by the curve for the first time. \sqsubseteq_2 is the order the vertices are passed for the second time. Then $x_1 \sqsubseteq_1 x_2 \Leftrightarrow x_1 \sqsubseteq_2 x_2$ holds. Thus, we can store all data on stacks which mirror the resulting requirements: put data on top of a stack after using it for the first time, pop data of the stack when it is needed for the second time. On a regular grid, one is able to split up all the vertices of the grid into two classes — half of the vertices are on the left hand side, the others are on the right hand side of the Peano curve. If we handle nodal data only, one can easily see that in this case it is sufficient to use two stacks representing this two classes of points. Stacks are cache-efficient due to the inherent local memory access. Furthermore, we can show that if we use stacks, the storage of further administrative information like pointers to neighbors and/or subsequent points is obsolete.

In [Günther 04, Pögl 04, Krahnke 04], it was shown that this principle can be generalized to adaptive grids with hierarchical data (generating systems) [Griebel 92] suited for sophisticated multigrid algorithms. The number of stacks is small and independent of the mesh refinement depth. [Hartmann 04] has developed a dimension-generic algorithm. Furthermore [Langlotz 04/2] shows some applications of this concept adding parallel programming, adaptivity and extrapolation techniques. [Günther 04, Langlotz 04/2] have proven that such an algorithm ends up with only 10

percent more cache-misses than absolutely inevitable in a code which does not make explicit use of a prefetching strategy. [Pögl 04] and [Hartmann 04] have shown that this holds for arbitrary dimensions, too.

D.4 Shape Functions Preserving Continuity

Solving both the momentum (D.1) and the continuity equation (D.2) in a weak formulation, the FEM minimizing problem requires an extended theory for mixed elements [Braess 02, Gresho 98]. In contrast to this, we eliminate all complications induced by the side condition by solving it exactly (pointwise). For this, we approximate the velocity space in a very special way:

Let \mathcal{T} be a domain discretization, where every geometric element $V \in \mathcal{T}$ is a square. Then, there is a collocated ansatzspace \mathbb{U} such that

$$Mu_h := \int_V \mathbf{div} u_h \, dx = 0 \quad \forall V \in \mathcal{T} \quad \Leftrightarrow \quad \mathbf{div} u_h = 0 \quad \text{almost everywhere} \quad (\text{D.3})$$

holds for all $u_h \in \mathbb{U}$.

As shown in [Gresho 98], this is the precondition for both energy and momentum conservation with respect to space discretizations¹. Such an ansatzspace with square support for the velocity was presented in [Blanke 04] and extended in [Weinzierl 05]. It was also used in a more complicated setting in [Emans 03]. Figure D.3 shows the two components of one basis function. The idea is to subdivide the support into triangles and couple the two velocity components. On each triangle, the shape function is linear. The "hat" ϕ connecting each velocity component with its own coefficients has the value 1 at the associated grid point, $\frac{1}{4}$ at the midpoints of the adjacent cells and 0 at all other grid points. The coupling function φ connecting each velocity component with the coefficients of the other component has the values $\pm\frac{1}{4}$ at the midpoints of the adjacent cells and 0 at all other grid points. Thus we get the basis representation

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \sum_i u_{1,i} \begin{pmatrix} \phi_i \\ \varphi_i \end{pmatrix} + u_{2,i} \begin{pmatrix} \varphi_i \\ \phi_i \end{pmatrix}.$$

To avoid a decomposition of cells in a checkerboard pattern, we had to introduce additional degrees of freedom at the cells edges (see figure D.2). The corresponding additional basis functions are generated by scaling and translation of the ones described above. To describe the complete ansatzspace is a very technical task exceeding the extent of this paper. For more details see [Weinzierl 05].

We note here in addition that the nonstaggered arrangement of the velocity components is sometimes criticized because it may produce the so called checkerboard instability. But this happens only if the Dirichlet boundary conditions are not introduced in a consistent way. If the boundary conditions are chosen appropriately, this instability is avoided easily.

D.5 The Pressure Equation

According to Chorins projection method with an explicit Euler step, we use the pressure p to achieve a solenoidal solution at time $t + \tau$. Using a FEM approach, equation (D.1) becomes

$$\Omega u_h^{(t+\tau)} = \underbrace{\Omega u_h^{(t)} + \tau \left(\frac{1}{Re} Du_h^{(t)} - C(u_h^{(t)})u_h^{(t)} \right)}_{=:b} - \tau \Omega \nabla_h p. \quad (\text{D.4})$$

¹The development of an optimal time discretization is not content of this work. Thus, conservation of energy and momentum holds only for $\tau \mapsto 0$, but for any grid width!

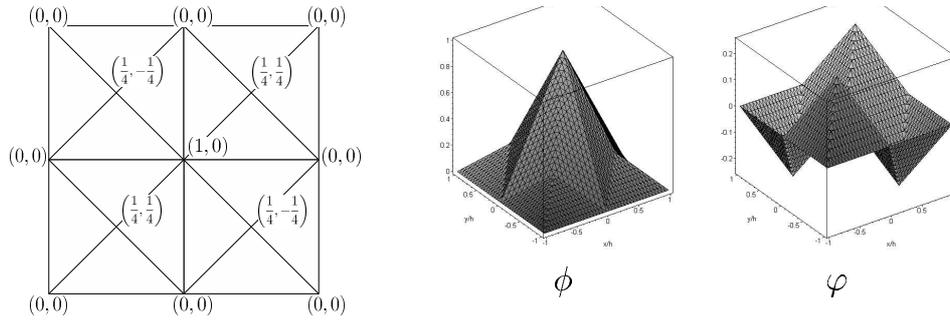


Abbildung D.3: The original velocity shape function presented in [Blanke 04].

Here, Ω denotes the mass matrix, D and $C(u)$ are the discrete diffusion and the convection operator and ∇_h is the discrete gradient. Considering the equivalence given by (D.3) and the duality of divergence and gradient ($\Omega \nabla = M^T$), the continuity equation (D.2) becomes

$$\mathbf{div} u^{(t+\tau)} = 0 \Leftrightarrow M \nabla_h p = \frac{1}{\tau} M \Omega^{-1} b \tag{D.5}$$

$$\Leftrightarrow M \Omega^{-1} M^T p = \frac{1}{\tau} M \Omega^{-1} b, \tag{D.6}$$

resulting in a staggered pressure grid². Such a pressure layout is energy preserving and stable with respect to checkerboard patterns [Trottenberg 00, Verstappen 01]. In (D.6) we used the duality of divergence and gradient operator, which is, in fact, again a precondition for energy and momentum conservation [Verstappen 01]. The entries of the discrete divergence operator M can easily be calculated using the Gauss integral theorem and are given in figure D.4.

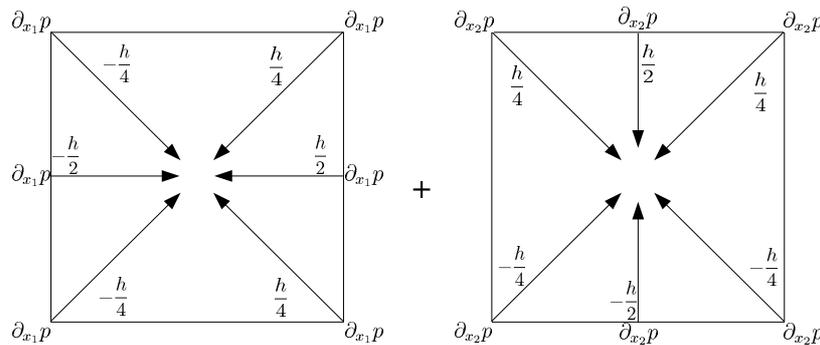


Abbildung D.4: Stencil $M|_V$ on a micro cell V divided up into x_1 and x_2 contributions. Matrix M is composed by all $M|_V$.

²We interpret the pressure being a staggered spline function space.

D.5.1 The Fine Grid Relaxation Scheme

To implement a relaxation method for the pressure, we have to integrate both the storage scheme of the pressure (cell-wise) and the evaluation of the discrete operator $M\Omega^{-1}M^T$ into our algorithmic concept based on stacks and cell-wise operator evaluation. Since the pressure values are visited in a strict linear order, it is sufficient to add one additional stack for the pressure. In contrast, the evaluation of the operator turns out to be a little more complex for the operator offends the locality property, as for the direct evaluation of $M\Omega^{-1}M^T$ we would have the pressure values of the neighbouring cells.

To prevent this, it is a good idea to store the gradients of the pressure, too, which are located at the cell vertices and edges analog to the location of the velocity values. A relaxation smoother for this equation system can be formalized as

$$\begin{aligned} p^{(k+1)} &= p^{(k)} + \omega \text{diag}^{-1} \left(\frac{1}{\tau} M\Omega^{-1}b - M\Omega^{-1}M^T p^{(k)} \right) \\ &=: p^{(k)} + \omega \text{diag}^{-1} \text{res}, \end{aligned} \tag{D.7}$$

where diag is the diagonal entry of $M\Omega^{-1}M^T$. Assume that we start with actual values of p and ∇p , and $\frac{1}{\tau}\Omega^{-1}b$ is given. Then, one iteration of our relaxation method can be decomposed into three steps:

- Evaluate $\text{res}_V = \frac{1}{\tau}M_V\Omega^{-1}b - M_V\nabla p^{(k)}$ (completed when leaving the cell),
- change pressure within cell $p^{(k+1)} = p^{(k)} + \omega \text{diag}^{-1} \text{res}_V$ (completed when leaving the cell) and finally
- update pressure gradients $\nabla p^{(k+1)} = \nabla p^{(k)} + \Omega_V^{-1} M_V^T \omega \text{diag}^{-1} \text{res}_V$. (Accumulation process over adjacent cells. $\nabla p^{(k+1)}$ completed when leaving the last adjacent cell.)

Here Ω_V is the lumped mass matrix. Since the pressure gradient is updated after each correction step immediately, this algorithm equals a Gauss-Seidel smoother and is therefore faster than [Günther 04, Hartmann 04, Pögl 04]. The drawback compared to collocated Jacobi-smoothers is the number of variables to be stored per pressure degree of freedom.

Figure D.6 shows the performance of a SOR Gauss-Seidel smoother for different relaxation factors (see also D.6). For further information on boundary conditions and actual values for ω , diag^{-1} , etc see [Gresho 98, Neckel 05, Weinzierl 05].

D.5.2 The Coarse Grid Correction

The aim of the following chapter is to develop a Galerkin multigrid algorithm for the pressure Poisson equation. Since we are solving a time-dependent problem, the main focus is on reducing low frequency error components fast as they are changing more substantially in each time step than the high frequency components. This can be done using a multigrid approach. Due to our stack concept, the implementation of a multigrid algorithm does not cause any loss of hardware efficiency.

In this paper, all hierarchical values (coefficients of the generating system) and operators are denoted with a hat. The concept of hierarchical generating systems [Griebel 92] fits the concept of space-trees and top-down-depth-first tree traversal in a very natural way. We introduced a generating system for the pressure, whereas the velocity is given on the fine grid only³.

³Future work will concentrate on multilevel representation of the velocities

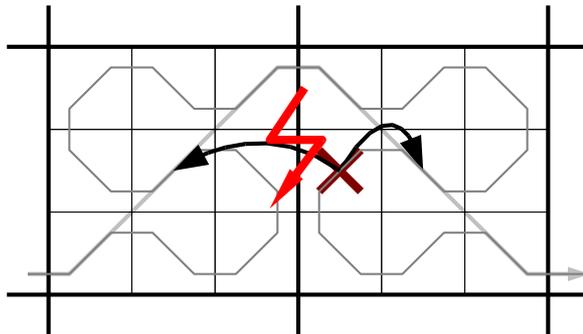


Abbildung D.5: A residual on the micro cell marked would affect the macro cell residual on the left side. This cell is not "available" ascending the space-tree.

Since for the fine grid smoother the nodal pressure gradient has to be computed, let's assume the hierarchical gradient is given on every grid level. Descending the space-tree with the aim of computing the nodal, interpolated values of $\nabla\hat{p}$ on the fine grid, one is able to calculate

$$\Omega^{-1}M^T P_{cell}\hat{p} = P_{\nabla}\nabla\hat{p} \quad (D.8)$$

directly using a suitable gradient interpolation operator P_{∇} . Afterwards, we perform one fine grid relaxation step. So, instead of working on the generating system for the pressure the problem is transformed to a generating system for the pressure gradient.

Unfortunately, one is not able to restrict the pressure residual in one step ascending the spacetree, as every fine grid residual on a micro cell — except the one in the center — would affect the residuals of all neighbouring macro cells within the space-tree as the restriction which is the transposed of the bilinear interpolation offends the locality property of the space filling curve algorithm as illustrated in figure D.5.

The description of the solution of this problem is a very technical task and exceeds the length constraints of this paper. The complete algorithm is described in [Weinzierl 05]. The critical point is to update the hierarchical pressure gradient after each coarse grid smoothing step immediately. This is again a reason for working on the hierarchical pressure gradients instead of the pressure itself.

Finally, we get full multigrid performance. Because we are using the pressure of the preceding time step as starting value, the number of iterations becomes small very soon using exploiting the property of multigrid algorithms that smooth error components which are predominant in our situation are reduced rapidly.

D.6 Results and Conclusion

In this paper, we presented a Navier–Stokes solver based on Chorins projection method [Chorin 68] combined with a new Finite–Element ansatzspace enabling us to solve the incompressibility exactly [Blanke 04]. A highly efficient storage usage and access was achieved by the construction of linear data structures with the help of space–filling curves [Günther 04, Pögl 04].

D A Cache-aware Multigrid Navier-Stokes Solver

Furthermore, by modifying the storage scheme of the pressure (gradients instead of function values) we were able to implement a pressure solver with multigrid efficiency [Braess 02, Griebel 92, Trottenberg 00]. Figure D.6 shows this fact. For all grid sizes, around 20 relaxation steps are needed to reduce the error by a factor of 10. This is much worse compared to multiplicative multigrid schemes but already good enough for our purpose. If we use the method e.g. in the scenario of the time dependant driven-cavity only very few relaxation steps in every time step are enough to reduce the error sufficiently. But it should not be difficult to improve the order of convergence substantially by a more sophisticated relaxation strategy. This will be subject of further investigations. The L2-cache hit-rate is almost minimal in accord with earlier results for simpler PDEs [Günther 04, Hartmann 04, Pögl 04].

In our opinion there are five important facts to point out:

1. Since the side condition is fulfilled exactly, the standard FEM theory can be applied. Particularly with regard to the discretization of the momentum equation no extended theory for mixed/nonconforming elements is required.
2. Current computers suffer from a lack of L2-cache making memory access pretty expensive compared to register access. The sophisticated traversal algorithm used here is by construction cache-optimal with respect to hit-rates on all memory hierarchies, since it uses stacks for data and geometry management.
3. Although the algorithm is designed for adaptive hierarchical meshes, the geometry is stored sequentially using space filling curves. Therefore, one bit per grid point is adequate to store the complete geometry data. In addition a second bit describes whether the cell is refined or not. So, geometry description and refinement structure are stored in a very efficient way.
4. The algorithm provides full multigrid performance.
5. The energy and momentum conservation laws are satisfied as the side condition is fulfilled exactly and all properties of the continuous operators are carried over to the discrete operators.

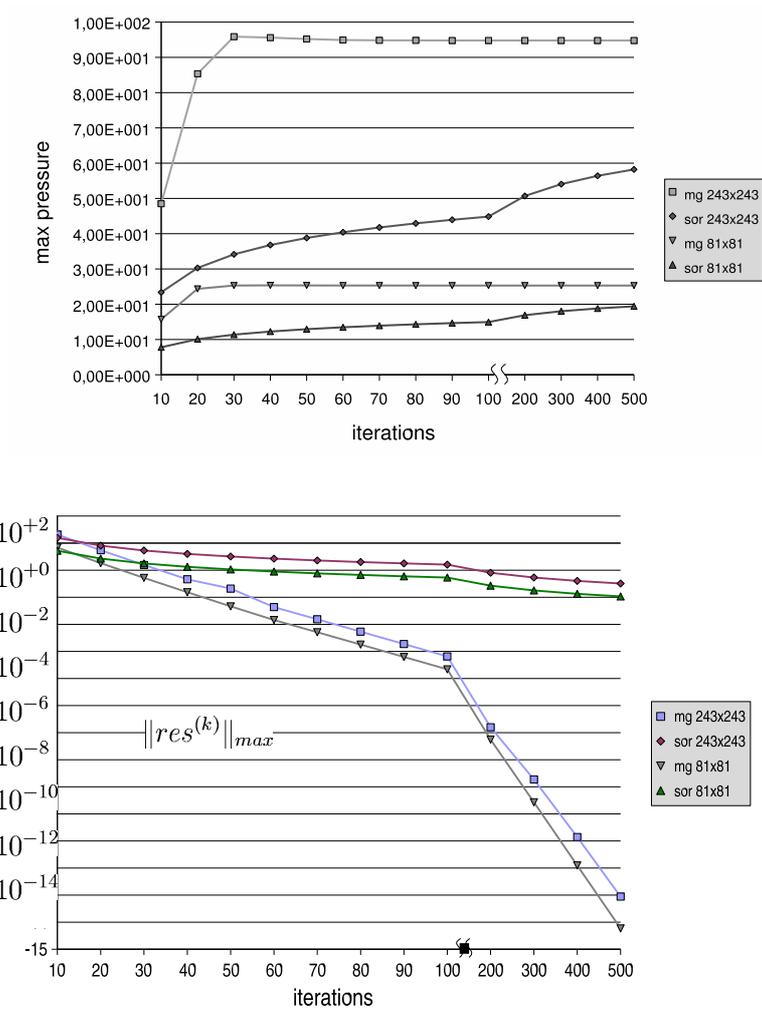


Abbildung D.6: Some results for the driven-cavity scenario with $Re = 10$ comparing the multigrid algorithm with a Gauss-Seidel relaxation scheme on two different grids.

E Fallstudie Driven-Cavity

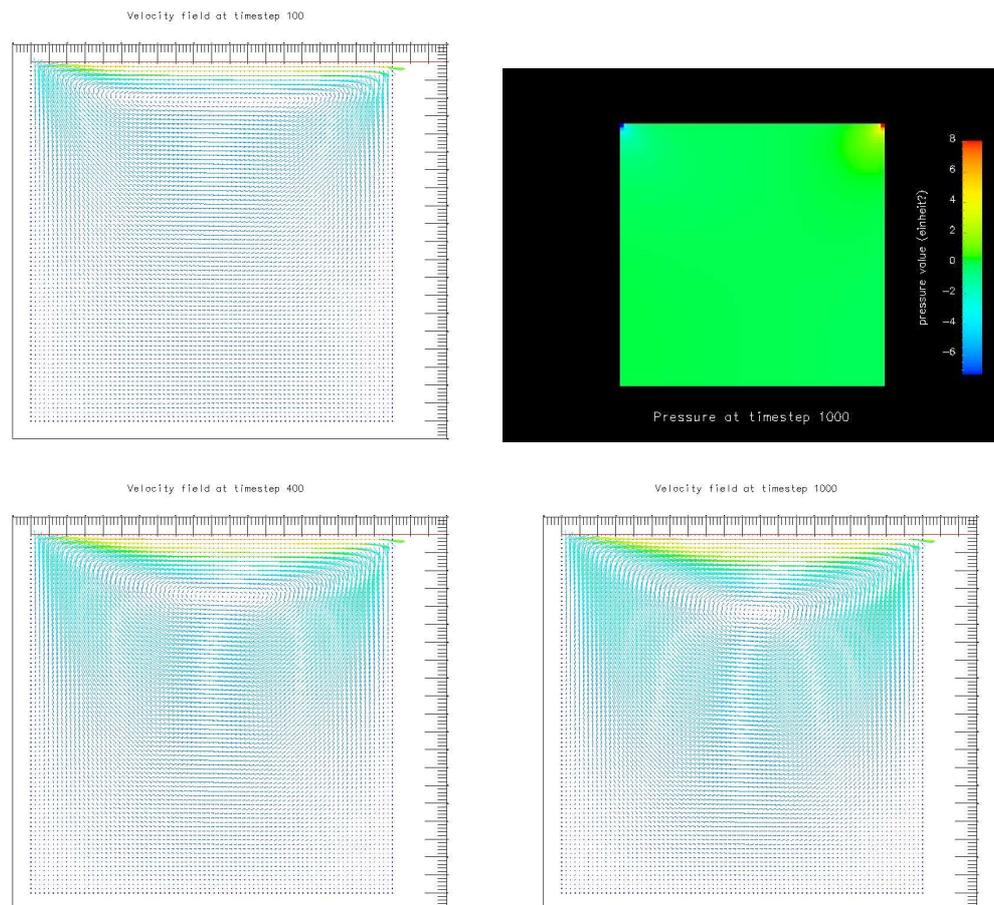


Abbildung E.1: Ein weiteres Mal ein 81^2 Gitter mit einem Fluid der Reynoldszahl $Re = 10$. Die Druckverteilung wird nach wenigen Schritten fast stationär, die Geschwindigkeiten jedoch nicht. Diese Bilder geben nur einen qualitativen Verlauf wieder, zur exakten Zeitskalierung siehe [Neckel 05].

E Fallstudie Driven-Cavity

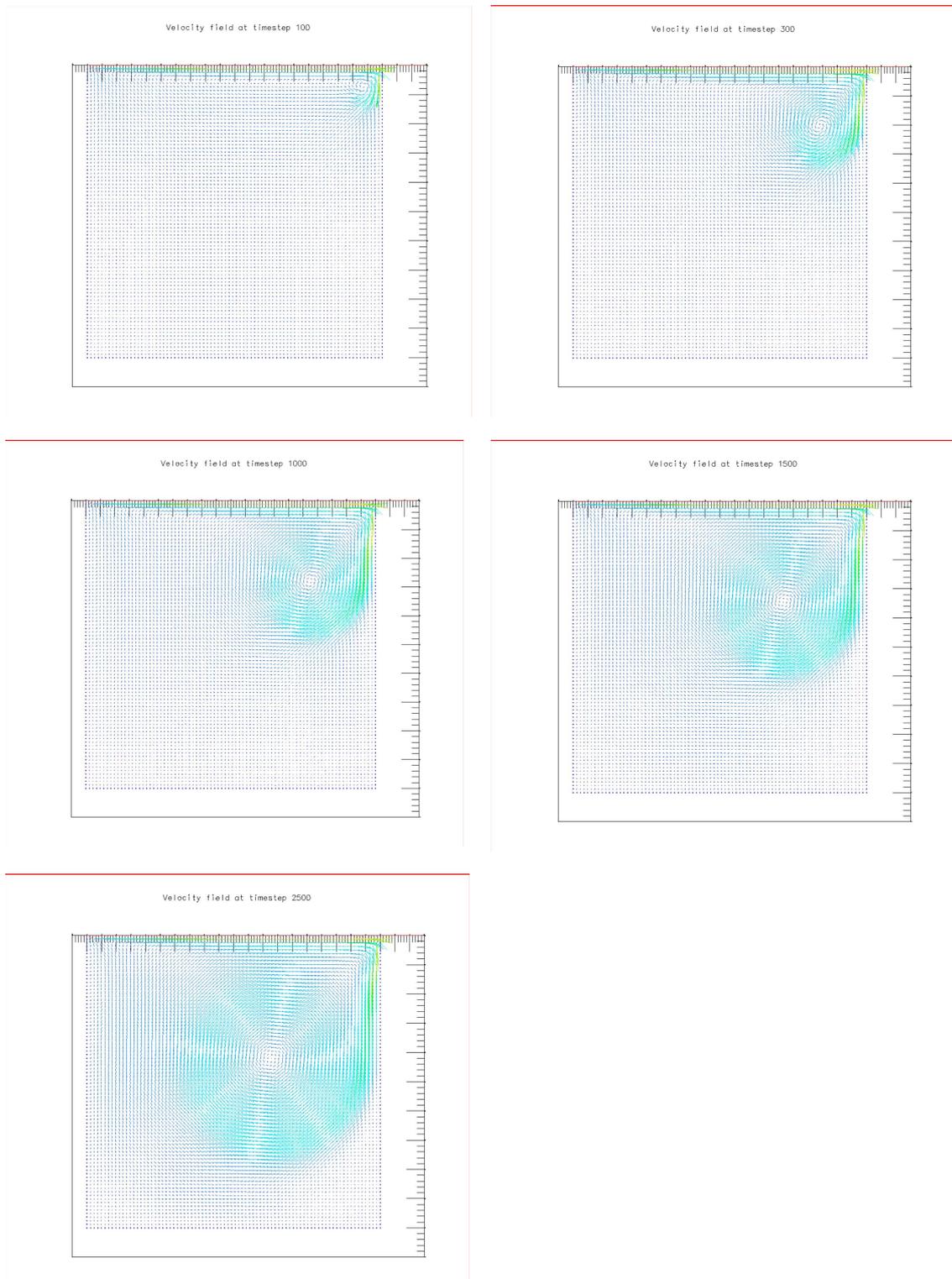


Abbildung E.2: Gleiches Szenario mit der Reynoldszahl $Re = 1500$.

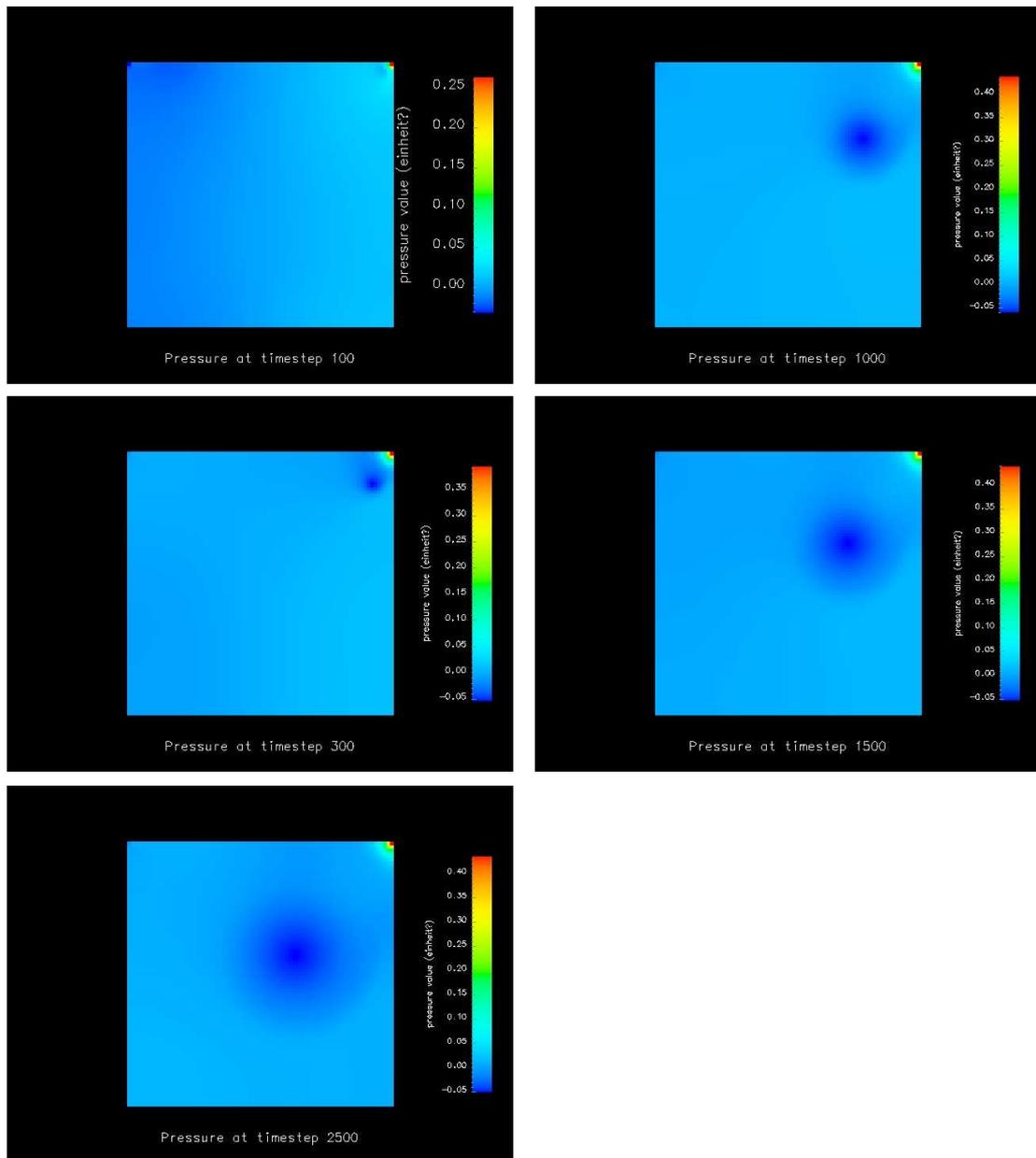


Abbildung E.3: Zugehöriger Druckverlauf.

Literaturverzeichnis

- [Aulbach 04] B. Aulbach: *Gewöhnliche Differentialgleichungen*, Spektrum Akademischer Verlag, Heidelberg, 2004
- [Bader 01] M. Bader, A.C. Frank, Ch. Zenger: *An Octree-Based Approach for Fast Elliptic Solvers*, erschienen in: M. Breuer, F. Durst, Ch. Zenger (Eds.): *High Performance Scientific and Engineering Computing*, Springer-Verlag, Berlin Heidelberg, 2001
- [Balder 91] R. Balder, H.-J. Bungartz (eds.): *Sparse Ideas on Sparse Grids*, Proceedings of the First International Conference on Applied Loricotics held in Castel di Mare (Lago di Bodo), March 11-16, 1991
- [Blanke 04] C. Blanke: *Kontinuitätserhaltende Finite-Elemente-Diskretisierung der Navier-Stokes-Gleichungen*, Diplomarbeit, Technische Universität München, Fakultät für Informatik, Forschungs- und Lehrereinheit Informatik V, 2004
- [Braess 02] D. Braess: *Finite Elemente*, Springer-Verlag, Berlin Heidelberg, 2002
- [Bronstein 01] I.N. Bronstein, K.A.Semendjajew, G. Musiol, H. Mühlig: *Taschenbuch der Mathematik*, Verlag Harri Deutsch, Thun und Frankfurt am Main, 2001
- [Broy 98] M. Broy: *Informatik — eine grundlegende Einführung*, Springer-Verlag, Berlin Heidelberg, 1998
- [Bungartz 96] H.-J. Bungartz, M. Griebel, Ch. Zenger: *Einführung in die Computergraphik - Grundlagen, Geometrische Modellierung, Algorithmen*, Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden, 1996
- [Bungartz 98] H.-J. Bungartz: *Finite Elements of Higher Order on Sparse Grids*, Dissertation, Technische Universität München, Fakultät für Informatik, Forschungs- und Lehrereinheit Informatik V, 1998
- [Bungartz 00] H.-J. Bungartz, S. Zimmer (Hrsg): *Numerische Simulation als interdisziplinäre Herausforderung - Vorträge des Symposiums anlässlich des 60. Geburtstags von Christoph Zenger*, Springer-Verlag, Berlin Heidelberg, 2000
- [Ciarlet 78] P.G. Ciarlet: *The finite element method for elliptic problems*, Series Studies in Mathematics and its Applications, Neuauflage von 1978, SIAM Classics in Applied Mathematics, No. 40, SIAM, Philadelphia, 2002
- [Chibisov 02] D.Chibisov, V. Ganzha, Ch. Zenger: *Objekt-Orientierte Modellierung für Wissenschaftliches Rechnen in Maple*, Wissenschaftlicher Bericht, Technische Universität München, Fakultät für Informatik, 2002
- [Chibisov 03] D.Chibisov: *Computational Models for Finite Element Simulation using Maple*, Diplomarbeit, Technische Universität München, Fakultät für Informatik, Forschungs- und Lehrereinheit Informatik V, 2003

Literaturverzeichnis

- [Chorin 68] A. J. Chorin: *Numerical solution of the Navier-Stokes equations*, Math. Comp. , 22 (1968), pp. 745-762, 1968
- [Deuffhard 02] P. Deuffhard, F. Bornemann: *Numerische Mathematik II - Gewöhnliche Differentialgleichungen*, de Gruyter Verlag, Berlin New York, 2002
- [Donea 03] J. Donea, A. Huerta: *Finite Element Methods for Flow Problems*, John Wiley & Sons Ltd, Chichester, 2003
- [Emans 03] M. Emans: *Numerische Simulation des unterkühlten Blasensiedens in turbulenter Strömung: Ein Euler-Lagrange-Verfahren auf orthogonalen Gittern*, Dissertation, Technische Universität München, Fakultät für Informatik, Forschungs- und Lehrereinheit Informatik V, 2003
- [Engesser 93] H. Engesser (Hrsg): *Duden Informatik 2*, Dudenverlag, Mannheim, Leipzig, Wien, Zürich, 1993
- [Felippa 04] C.A. Felippa: *Introduction to Finite Element Methods*, Material assembled from Lecture Notes on the course *Introduction to Finite Elements Methods* (ASEN 5007), Aerospace Engineering Sciences Department, University of Colorado at Boulder, 2004
- [Ferziger 99] J.H. Ferziger, M. Peric: *Computational Methods for Fluid Dynamics*, Springer-Verlag, Berlin Heidelberg, 1999
- [Fletcher 84] C.A.J. Fletcher, K. Srinivas: *On the role of mass operators in the group finite element formulation*, Computer Methods in Applied Mechanics and Engineering, vol. 46, pp. 313-327, 1984
- [Fletcher 97] C.A.J. Fletcher: *Computational Techniques for Fluid Dynamics (Volume 2) - Specific Techniques for Different Flow Categories*, Springer-Verlag, Berlin Heidelberg New York, 1997
- [Frank 00] A.C. Frank: *Organisationsprinzipien zur Integration von geometrischer Modellierung, numerischen Simulation und Visualisierung*, Dissertation, Technische Universität München, Fakultät für Informatik, Forschungs- und Lehrereinheit Informatik V, 2000
- [Ganzha 96] V.G. Ganzha, E.V. Vorozhtsov: *Computer-Aided Analysis of Difference Schemes for Partial Differential Equations*, John Wiley & Sons Inc, New York, 1996
- [Ganzha 01/1] V.G. Ganzha, D. Chibisov, E.V. Vorozhtsov: *Problem solving for scientific computing: data modelling instead of algorithms?*, Selcuk Journal of Applied Mathematics 2/2, 2001
- [Ganzha 01/2] V.G. Ganzha, D. Chibisov, E.V. Vorozhtsov: *GROOME - Tool Supported Graphical Object Oriented Modelling for Computer Algebra and Scientific Computing*, Computer Algebra in Scientific Computing, CASC'2001. Proc. 4nd Workshop on Computer algebra in Scientific Computing, Konstanz, Sept 22-26, 2001
- [Girault 86] V. Girault, P. Raviart: *Finite Element Methods for Navier-Stokes Equations*, Springer-Verlag, Berlin Heidelberg, 1986
- [Gresho 87] P.M. Gresho, R.L. Sani: *On pressure boundary conditions for the incompressible Navier-Stokes equations*, International Journal for Numerical Methods in Fluids, Vol. 7, p. 1111-1145, 1987

- [Gresho 98] P.M. Gresho, R.L. Sani, M.S. Engelman: *Incompressible Flow and the Finite Element Method - Advection-Diffusion and Isothermal Laminar Flow*, John Wiley & Sons Ltd, Chichester, 1998
- [Griebel 92] M. Griebel: *Multilevel algorithms considered as iterative methods on indefinite systems*, In: Proceedings of the 2nd Copper Mountain Conference on Iterative Methods, University of Colorado at Denver, 1992
- [Griebel 94] M. Griebel: *Multilevelmethoden als Iterationsverfahren über Erzeugendensystemen*, B.G. Teubner Stuttgart, Stuttgart, 1994
- [Griebel 95] M. Griebel, T. Dornseifer, T. Neunhoffer: *Numerische Simulation in der Strömungsmechanik*, Vieweg, 1995
- [Günther 04] F. Günther: *Eine cache-optimale Implementierung der Finite-Elemente-Methode*, Dissertation, Technische Universität München, Fakultät für Informatik, Forschungs- und Lehrereinheit Informatik V, 2004
- [Günther 05] F. Günther, M. Mehl, M. Pögl, Ch. Zenger: *A cache-aware algorithm for PDEs on hierarchical data structures based on space-filling curves*, SIAM Journal on Scientific Computing, in review, 2005
- [Hartmann 04] J. Hartmann: *Entwicklung eines cache-optimalen Finite-Element-Verfahrens zur Lösung d-dimensionaler Probleme*, Diplomarbeit, Technische Universität München, Fakultät für Informatik, Forschungs- und Lehrereinheit Informatik V, 2004
- [Herder 05] W. Herder: *Lastverteilung und parallelisierte Erzeugung von Eingabedaten für ein paralleles Cache-optimales Finite-Element-Verfahren*, Diplomarbeit, Technische Universität München, Fakultät für Informatik, Forschungs- und Lehrereinheit Informatik V, 2005
- [Knabner 00] P. Knabner, L. Angermann: *Numerik partieller Differentialgleichungen - Eine anwendungsorientierte Einführung*, Springer-Verlag, Berlin-Heidelberg, 2000
- [Krahnke 04] A. Krahnke: *Adaptive Verfahren höherer Ordnung auf cache-optimalen Datenstrukturen für dreidimensionale Probleme*, Dissertation, Technische Universität München, Fakultät für Informatik, Forschungs- und Lehrereinheit Informatik V, 2004
- [Langlotz 04/1] M. Langlotz: *Parallelisierung eines Cache-optimalen 3D Finite-Element-Verfahrens*, Diplomarbeit, Technische Universität München, Fakultät für Informatik, Forschungs- und Lehrereinheit Informatik V, 2004
- [Langlotz 04/2] M. Langlotz, M. Mehl, T. Weinzierl, Ch. Zenger: *Cache-Optimal Parallel Solution of PDEs on High Performance Computers Using Space-Trees and Space-Filling Curves*, KON-WIHR Abschlussbericht — Competence Network for Technical, Scientific High Performance Computing in Bavaria, Springer-Verlag, Berlin Heidelberg, in review, 2005
- [Liseikin 99] V.D. Liseikin: *Grid Generation Methods*, Springer-Verlag, Berlin-Heidelberg, 1999
- [Mahler 03] S. Mahler: *Erzeugung und Evaluierung von Oktalbaumstrukturen als Schnittstelle zu CAD-Programme*, Diplomarbeit, Universität Stuttgart, Fakultät Informatik, 2003
- [Marcotty 86] M. Marcotty, H. Ledgard: *The World of Programming Languages*, Springer-Verlag, Berlin, 1986

Literaturverzeichnis

- [Mehl 01] M. Mehl: *Ein interdisziplinärer Ansatz zur dreidimensionalen numerischen Simulation von Strömung, Stofftransport und Wachstum in Biofilmsystemen auf der Mikroskala*, Dissertation, Technische Universität München, Fakultät für Informatik, Forschungs- und Lehrereinheit Informatik V, 2001
- [Meier 00] F. Meier: *Effiziente numerische Behandlung von Strömungen in veränderlichen Geometrien auf kartesischen Gittern*, Shaker Verlag, 2000
- [Neckel 05] T. Neckel: *Einfache 2D-Fluid-Struktur-Wechselwirkungen mit einer cache-optimalen Finite-Element-Methode*, Diplomarbeit, Technische Universität München, Fakultät für Informatik, Forschungs- und Lehrereinheit Informatik V, 2005
- [Pentenrieder 05] S. Pentenrieder: — *Titel noch nicht erschienen* —, Diplomarbeit, Technische Universität München, Fakultät für Informatik, Forschungs- und Lehrereinheit Informatik V, 2005
- [Peyret 83] R. Peyret, T. Taylor: *Computational Methods for Fluid Flow*, Springer Series in Computational Physics, Springer-Verlag, Berlin, 1983
- [Pögl 04] M. Pögl: *Entwicklung eines cache-optimalen 3D Finite-Element-Verfahrens für große Probleme*, Dissertation, Technische Universität München, Fakultät für Informatik, Forschungs- und Lehrereinheit Informatik V, 2004
- [Roache 76] P. Roache: *Computational Fluid Dynamics*, Hermosa Publishers, Albuquerque, 1976
- [Sagan 94] H. Sagan: *Space-Filling Curves*, Springer-Verlag, Berlin Heidelberg, 1994
- [Schöberl 99] J. Schöberl: *Robust Multigrid Methods for Parameter Dependent Problems*, Dissertation, Johannes Kepler Universität Linz, Institut für Analysis und Numerik, Arbeitsgruppe numerische Mathematik und Optimierung, 1999
- [Sivaloganathan 88] S. Sivaloganathan, G.J Shaw, T.M. Shah, D.F. Mayers: *A Comparison of Multigrid Methods for the Incompressible Navier-Stokes Equations*, in K.W. Morton, M.J. Baines (eds.): *Numerical Methods for Fluid Dynamics III*, pp. 410-417, Oxford University Press, 1988
- [Sivaloganathan 91] S. Sivaloganathan: *The Use of Local Mode Analysis in the Design and Comparison of Multigrid Methods*, *Comp. Phys. Comm.*, vol. 65, pp. 246-252, 1991
- [Stoer 90] J. Stoer, R. Bulirsch: *Numerische Mathematik 2*, Springer-Verlag, Berlin, 1990
- [Temam 69] R. Temam: *Sur l'approximation de la solution des equations de Navier-Stokes par la methode des pas fractionnaires (II)*, *Arch. Rational Mech. Anal.*, 33, pp. 377-385, 1969
- [Temam 01] R. Temam: *Navier-Stokes equations. Theory and numerical analysis*, AMS Chelsea Publishing, Amsterdam, 2001
- [Tome 94] M. Tome, S. McKee: *GENSMAC: A computational marker and cell method for free surface flows in general domains*, *Journal of Computational Physics*, vol. 110, pp. 171-186, 1994
- [Trottenberg 00] U. Trottenberg, C. Oosterlee, A. Schüller: *Multigrid*, Academic Press, London 2000
- [Vanka 85] S.P. Vanka: *Block-Implicit Calculation of Steady Turbulent Recirculating Flows*, *Int. J. Heat Mass Transfer*, vol. 28, pp. 2093-2193, 1985

- [Verstappen 01] R.W.C.P. Verstappen, A.E.P. Veldman: *Symmetry-Preserving Discretization of Turbulent Channel Flow*, erschienen in: M. Breuer, F.Durst, C.Zenger (Eds.): High Performance Scientific and Engineering Computing, Springer-Verlag, Berlin Heidelberg, 2001
- [Wagner 05] T. Wagner: *Randbehandlung höherer Ordnung für ein cache-optimales FE-Verfahren auf strukturierten Gittern*, Diplomarbeit, Technische Universität München, Fakultät für Informatik, Forschungs- und Lehrinheit Informatik V, 2005
- [Weinzierl 04] T. Weinzierl: *Finite Elemente und \mathcal{H} -Matrizen*, Ausarbeitung zu einem Interdisziplinären Projekt, Technische Universität München, Fakultät für Mathematik, Lehrstuhl für Numerische Mathematik und Steuerungstheorie, 2004
- [Weinzierl 05] T. Weinzierl: *A Cache-Aware Multigrid Navier-Stokes Solver*, Twelfth Copper Mountain Conference on Multigrid Methods, in review, 2005
- [Werner 02] D. Werner: *Funktionalanalysis*, Springer-Verlag, Berlin, 2002
- [White 04] D. White, A. Fisher, G. Rodrigue: *A Generalized Mass Lumping Scheme for Maxwell's Wave Equation*, Proceedings of the 2004 IEEE International Antennas and Propagation Symposium, vol. 4, pp. 1507-1510, Monterey, California, 2004
- [Zeeuw 97] P.M. de Zeeuw: *Acceleration of Iterative Methods by Coarse Grid Corrections*, Ph.D. Thesis, Centrum voor Wiskunde en Informatica, University of Amsterdam, 1997

Literaturverzeichnis

Abbildungsverzeichnis

0.1	Verschiedene wichtige Arbeiten (ohne Anspruch auf Vollständigkeit).	ii
0.2	Der typische Ablauf der numerischen Simulation und Programmevolution.	iii
1.1	Auswirkung des Laplace-Operators. Man beachte die Beschleunigungs- (oben) respektive die Bremswirkung (unten).	5
1.2	Transport einer Kenngröße (dunkles Gebiet) durch ein Transportmedium (schwarze Pfeile). Nur die Strömung senkrecht zur Kenngrößennormalen (weiße Pfeile) resultiert in einer Verschiebung derselben, eine Tangentialströmung hat keine Auswirkungen.	6
1.3	Der Druckgradient presst das Fluid aus einem Gebiet. Dadurch wird die Inkompressibilität sichergestellt.	6
1.4	Zeitfortschaltung durch den Operator \mathcal{T}_τ	7
1.5	Schematische Darstellung der Driven-Cavity.	9
2.1	Die vertikale Methode als <i>fractional-step method</i>	11
2.2	Zeitfortschaltung durch den Operator \mathcal{T}_τ	13
2.3	Die Foward-Euler-Methode.	14
2.4	Die gemischten Methoden approximieren beide Gleichungen. Daher liegt die Lösung an einem unphysikalischen Ort, entspricht also i.A. unendlicher Energie.	18
2.5	Das beliebte Q^1P^0 -Element: Hier werden bilineare Geschwindigkeitsansatzfunktionen auf den Gitterpunkten in Kombination mit zellweise konstantem Druck verwendet.	19
2.6	Konstruktion der <i>spurious modes</i> zum Q^1P^0 -Element.	23
2.7	Checkerboard-Druckverteilung, wie sie im Programm für einen Kanal mit Hindernissen bei Verwendung des Q^1P^0 -Elements auftreten kann. Hierbei sind an der Ausflusskante Dirichlet-Randbedingungen für die Geschwindigkeiten gegeben. Das Fluid strömt im oberen Bild horizontal von links nach rechts.	24
2.8	Im Driven-Cavity Szenario treten bereits an zwei Stellen an der oberen Kante im Drucknormalenanteil (Halb-) Singularitäten auf. Eine Lösung der Druckpoissongleichung über eine schwache Formulierung ist daher theoretisch nach dem Spursatz nicht untermauert.	26
3.1	Der Raum $\mathbb{U}(\mathcal{F})$ hält im Gegensatz zu \mathbb{U} auch Freiheitsgrade auf dem Rand des Gebietes Ω	33
3.2	Gradient- und Divergenzoperator als Funktionenraumübergang.	37
3.3	Das in [Blanke 04] entwickelte Referenzelement.	39
3.4	Die Augmentierung des in [Blanke 04] entwickelten Raumes um hierarchische Kantenfreiheitsgrade. Der Punkt $u(M)$ ist analog zur konstruktiven Herleitung der 5-Punkte-Elemente ein „virtueller“ hierarchischer Freiheitsgrad.	40

3.5	Teile des Referenzelements zur augmentierten [Blanke 04]–Basis. Oben eine Ansatzfunktion zur nodalen Basis, unten eine zum hierarchischen Überschuss. Die roten Werte bezeichnen die Höhe der Ansatzfunktionen respektive des Überschusses, die schwarzen Werte die Interpolante. Durch Spiegelung und passende Rotation erhält man die restlichen Formfunktionen auf dem Element. Auf den Dreiecken verwende man lineare Interpolation.	41
3.6	<i>Collocated, partially staggered</i> und <i>(fully) staggered grids</i>	44
4.1	Die verschiedenen verwendeten Nummerierungen.	48
4.2	Der Divergenzoperator auf den [Blanke 04]–Elementen. Die gestrichelt markierten Kanten der Elemente sind Indikatoren für einen linearen Wertverlauf.	54
4.3	Der 5–Punkte–Druckstern zu den in [Blanke 04] entworfenen Ansatzfunktionen. Die hier zu beobachtende Auslöschung der vertikalen und horizontalen Nachbardruckbeiträge führt zu der bekannten <i>checkerboard</i> Instabilität.	55
4.4	Der Divergenzoperator auf den augmentierten Geschwindigkeitselementen.	57
4.5	Der 9–Punkte–Druckstern, der zu den in dieser Arbeit neu entworfenen Ansatzfunktionen passt. Im Gegensatz zum 5–Punkte–Stern tritt hier keine Entkopplung von Zellen mit gemeinsamer Kante auf. Der Stern ist also in dieser Hinsicht stabil. Desweiteren beobachte man, dass die vertikalen / horizontalen Nachbarn keine Werte aufgrund von Auslöschung über die Gitterpunkte keine Werte, also ausschließlich über die Kanten Information transportieren.	58
4.6	Nomenklatur der Eingangsgrößen des Divergenzoperators nach Himmelsrichtungen. Der Träger $]-\frac{1}{2}, \frac{1}{2}[\times]-\frac{1}{2}, \frac{1}{2}[$ einer versetzten Pagodenfunktion auf einem „Gitter“ mit Einheitslänge ist farbig markiert.	60
5.1	Wirkung des stationären Glättungsoperators auf dem Fehlerplot.	67
5.2	Elementweise Auswertung der Diskretisierungssterne mit implizit assemblierten Gleichungssystemen am Beispiel des 5–Punkte–Sterns für den Laplace–Operator. . . .	67
6.1	Diskretisierung eines Beispielgebietes mit einem trisektionsbasierten Spacetree der Tiefe eins (links), zwei (Mitte) oder drei (rechts).	72
6.2	Einige Ansatzfunktionen auf einem Teilgebiet mit hängenden Feingitterknoten — einige davon sind durch Pfeile markiert.	73
6.3	Wirbelwanderung bei der Kármánschen Wirbelstraße. Die Bewegung der Wirbel korrespondiert mit wandernden Unterdruckgebieten. Das Beispiel ist [Neckel 05] entnommen.	74
6.4	Das obligatorische [Griebel 94] entstammende Bild zur Einführung hierarchischer Erzeugendensysteme im eindimensionalen Fall.	76
6.5	Dieses Bild entstammt ursprünglich [Pögl 04] und klassifiziert die Gitterelemente in einer Darstellung in einem hierarchischen Erzeugendensystem. Hinzugefügt wurde die Platzierung der zellbasierten Freiheitsgrade. Die Farbfläche repräsentiert exemplarisch den Träger einer Grobgitterdruckfunktion gemäß 6.3.2.	78
6.6	Die Interpolation bilinearer nodaler Ansatzfunktionen.	79
6.7	Illustration der Interpolation eines bilinearen versetzten Drucks durch den Operator P_p . Die Interpolation ist nur möglich, weil der Gradient des Drucks explizit auf den Gitterpunkten zwischengespeichert ist.	80
6.8	Die Anordnung der Fein– und Grobgitterdruckgradientenfreiheitsgrade.	81
6.9	Das Leitmotiv der Peano–Kurve.	84
6.10	Konstruktion des approximierenden Polygons zur Peano–Kurve über das Leitmotiv. Das Bild ist [Günther 04] entnommen.	84

6.11	Die Färbung der Knoten und die damit einhergehende Stackzuweisung. Auch dieses Bild ist [Günther 04] entnommen.	85
6.12	Motivation der Stackidentifikation nach [Günther 04].	86
7.1	Das Glättungsverhalten eines stationären Iterationsverfahrens: Hochfrequente Fehler werden wesentlich schneller gedämpft, als glatte Fehlerkomponenten. Die Abbildung ist [Braess 02] entnommen.	90
7.2	Qualitative Darstellung der Glättung der Fehlerfrequenzen zu verschiedenen Relaxationsparametern relativ zur Gitterweite.	91
7.3	Übergang des Residuums auf eine Grobgitterdarstellung (Restriktion). Bereits in diesem eindimensionalen Fall lässt sich die Nichtlokalität der Restriktion erkennen (roter Pfeil). Desweiteren erkennt man, dass ein Feingitterresiduum beide Gradienten der Nachbargrobzelle beeinflusst.	94
7.4	Restriktionsoperator P_p^T im zweidimensionalen Fall für drei exemplarisch gewählte Feingitterresiduen. Die roten Einflüsse führen zum Verletzen des Lokalisitätsprinzips der Grobzelle. Aufgrund der Symmetrie und des Tensorproduktansatzes ist der komplette Restriktionsfaktor mit dieser Zeichnung spezifiziert.	95
7.5	Transport des Feinresiduums auf Hilfsvariablen im Baumaufstieg, über die dann beim nächsten Baumabstieg summiert wird, um das Residuum zu erhalten.	96
7.6	Umsetzung eines klassischen Mehrgitterzyklus mithilfe des Zweigitteralgorithmus, der dem Lokalisitätsprinzip genügt.	97
7.7	Ablaufschema des schlussendlich implementierten Mehrgitteralgorithmus.	98
7.8	Druckresiduumsverbesserung für ein SOR-Verfahren mit den 5-Punkte-Elementen und den 9-Punkte-Elementen auf einem regulären 81^2 Gitter mit $\omega = 1.2$. Die bessere Konvergenzrate der 5-Punkte-Elemente ist auf das zugrundeliegende reduzierbare Gleichungssystem zurückzuführen.	99
7.9	Konvergenzverhalten des implementierten Mehrgitteralgorithmus auf zwei verschiedenen regulären Gittern.	99
7.10	Entwicklung des Maximaldrucks bei SOR-Verfahren und Mehrgitterverfahren.	100
7.11	Benötigte Iterationszahlen. Es liegt ein reguläres 81^2 Gitter zur Reynoldszahl 10 mit einem absoluten Druckabbruchkriterium von $0.81 \cdot 10^{-3}$ zugrunde. Vorgegebene Maximaliterationenzahl war dabei 1000.	100
7.12	Weiterer Ausschnitt aus dem Verlauf des in Abbildung 7.11 beschriebenen Szenarios.	101
A.1	Die konkurrierenden Ausrichtungen der Konvektionsmatrizen C_j . Die Besetztheitsstruktur ist für bilineare Elemente und $1 \leq j \leq 4$ spezifiziert.	108
B.1	Generierung der Interpolationsmatrix P_{∇} unter Verwendung eines virtuellen Feingitters.	113
D.1	The Peano space filling curve used for the traversal of the spacetree grid. The right picture indicates the processing order of some selected vertices.	122
D.2	Arrangement of the degrees of freedom per grid cell for both velocity, pressure and pressure derivatives.	123
D.3	The original velocity shape function presented in [Blanke 04].	125
D.4	Stencil $M _V$ on a micro cell V divided up into x_1 and x_2 contributions. Matrix M is composed by all $M _V$	125
D.5	A residual on the micro cell marked would affect the macro cell residual on the left side. This cell is not "available" ascending the space-tree.	127

Abbildungsverzeichnis

D.6	Some results for the driven-cavity scenario with $Re = 10$ comparing the multigrid algorithm with a Gauss-Seidel relaxation scheme on two different grids.	129
E.1	Ein weiteres Mal ein 81^2 Gitter mit einem Fluid der Reynoldszahl $Re = 10$. Die Druckverteilung wird nach wenigen Schritten fast stationär, die Geschwindigkeiten jedoch nicht. Diese Bilder geben nur einen qualitativen Verlauf wieder, zur exakten Zeitskalierung siehe [Neckel 05].	131
E.2	Gleiches Szenario mit der Reynoldszahl $Re = 1500$	132
E.3	Zugehöriger Druckverlauf.	133

Index

- θ -Methoden, 13
- Coarse grid principle*, 91
- Computational Fluid Dynamics*, 1, 18
- Smoothing Principle*, 90
- centered-scheme*, 44
- checkerboard pattern*, 23, 54, 57
- grid*
 - collocated*, 43
 - partially staggered*, 43
 - staggered*, 43
- mass lumping*, 51, 107
- pollution*, 75
- row summing*, 52, 107
- spurious modes*, 22, 38, 42, 103
- staggered grid*, 53
- Abbruchbedingung
 - relativ, 69
 - scharf, 69
- Adaptivitätskriterien, 73
- Adjungiertheit, 3, 19, 34, 37
- Approximationsgenauigkeit, 75
- Approximierendes Polygon, 84
- Assemblierung
 - elementweise, 56, 65
 - elementweise Assemblierungsmatrix, 67
 - explizit, 56
 - nodal, 56
- Basis
 - nodale, 48
- Baumknotenconstraint, 78
- Bilinearform, 107
- Cache-Line, 86
- Cache-Miss, 72, 87
- Cauchy-Schritt, 17
- CFD, 18
- CFL, 30
- Computer Aided Software Engineering (CA-SE), 49, 107
- Computer Algebra System (CAS), 47, 80, 107, 113
- Datenlokalität, 72
- Dehierarchisierung, 76, 80
- Differentialgleichung
 - elliptisch, 8, 26
 - hyperbolisch, 8
 - parabolisch, 8
- Diffusionsterm, 5
- Dimensionslose Form, 2
- Diskretisierung, 29
 - Gebietsdiskretisierung, 71
 - Linienmethode, 12
 - Rothe-Methode, 12
 - Semidiskretisierung, 12
 - Volldiskretisierung, 12
- Divergenz, 3
- Divergenzfreiheit, 26
- Divergenzoperator, 7
- Driven-Cavity
 - Definition, 9
- Druckgradient, 56
- Druckpoissongleichung, 25
- Dualität, 37
- Element
 - 5-Punkte-Elemente, 39
 - 9-Punkte-Elemente, 39
 - geometrisches, 38, 53, 56, 67, 71, 77, 113
 - kontinuitätserhaltend, 47
- Energieerhaltung, 50
- Energieminimierung, 8
- Erzeugendensysteme
 - hierarchische, 48
 - nodale, 48
- Euler Betrachtungsweise, 2
- Euler-Gleichungen, 2
- Exkurs
 - Adjungierte Operatoren, 37

Index

- Einbringen von Randbedingungen, 20
- Funktionenräume, 16
- Mass Lumping, 52
- Notation, 4
 - Indices, 4
 - Mathematische, 3
 - Operatoren, 3
 - Symbole, 4
- Nummerierungen, 48, 49
- Symmetrie, 45
- Symmetrie elliptischer Operatoren, 35

- Finite Elemente
 - Definition, 38
 - divergenzfrei, 17
 - Gemischte Verfahren, 18
 - konforme, 17
 - kontinuitätserhaltend, 36
 - nichtkonforme, 17
- Fluid, 1
- Fluid–Struktur–Wechselwirkungen, 2
- Forward–Euler–Verfahren, 14

- Gebietsdiskretisierung, 29
- Gekoppelte Verfahren, 19
- geometrisches Mittel, 69
- Gitter
 - level, 76
 - adaptive, 71, 73
 - Feingitter, 76
 - hierarchische, 71, 74
 - Randbehandlung, 20, 75
 - Traversierung, 76
 - versetzt, 44
 - zeitadaptiv, 89
- Gitterweite, 65
- Glätter, 64, 90
 - additiv, 66
 - Gaus–Seidel, 83
 - Jakobi, 83
 - multiplikativ, 66
- Gradient, 3

- Hängender Punkt, 77
- Hierarchisches Erzeugendensystem, 76, 86

- Impulsgleichung, 2
- inf–sup–Bedingung, 57
- Inkompressibilität, 2, 7
- Interpolation, 76
 - nodale, 79
 - Skripte, 80
- Iterationsverfahren
 - Gauss–Seidel–Verfahren, 64
 - Jakob–Verfahren, 64
 - optimaler Relaxationsfaktor, 66
 - Richardson–Verfahren, 64
 - stationär, 63

- Knoten
 - Dirichlet–, 20, 75
 - hängende, 76
 - Neumann–, 20, 75
- Konsistenzfehler, 17
- Kontinuitätsgleichung, 2
- Konvektions–/Diffusionsteil \mathcal{F}
 - Berechnung, 32
 - Platzierung der Freiheitsgrade, 32
 - Randwerte, 20, 32
- Konvektionsterm, 6
- Konvergenz
 - ordnung, 64
 - rate, 64
 - Mehrgitter, 66
- Konvergenzrate, 69

- Lagrange–Parameter, 7
- Laplace–Operator, 3
- Lipschitz–Gebiet, 16
- Lokalitätsprinzip, 56, 67, 68, 80, 83
 - Definition, 56

- Matrix
 - lumped* Massenmatrix, 107
 - Diffusionsmatrix
 - bilineare Elemente, 108
 - kontinuitätserhaltende Elemente, 110
 - Konvektionsmatrix, 107
 - bilineare Elemente, 108
 - kontinuitätserhaltende Elemente, 110
 - Massenmatrix, 52
 - bilineare Elemente, 108
 - kontinuitätserhaltende Elemente, 110
- Maximaldruck, 92
- Mehrgitter
 - additiv, 97
 - additiv mit multiplikativem Glätter, 90
 - Galerkin–Mehrgitter, 91
 - Grundlage, 90
 - Klassifikation, 91

- Konvergenz, 66, 91
- Methode
 - factional-step-method*, 17
 - Chorin–Temam, 18
 - Druckkorrekturmethode, 18
 - semidiskrete Modelle, 12
- Minimierungsprinzip, 35
- Numerische Strömungssimulation, 1
- Nummerierung, 107
 - gegen Uhrzeigersinn, 107
 - kartesisch, 48
 - Kurvennummerierung, 48
 - lexikographisch, 48, 49, 113
- Oktalbaum, 73
- Operator
 - Diffusions-/Konvektionsoperator \mathcal{F} , 14
 - Zeitoperator, 12
- Phase
 - Einschwingphase, 75
 - quasistationär, 75
 - stationär, 75
 - transient, 75
- Präparationsschritt, 30
 - schwache Formulierung, 32
- Projektionsmethoden
 - Chorinsche, 25
- Quadtree, 73
- Rand
 - Dirichlet–, 20, 75
 - Neumann–, 20, 75
- Randbedingung, 20, 26
 - Druck, 8
 - Einströmbedingung, 8
 - free-slip, 8
 - no-slip, 8
 - schwache, 26
- Relaxation, 66
 - Relaxationsparameterwahl, 90
 - Relaxationsverfahren, 57
- Residuum
 - Definition, 65
- Restriktion, 76, 82
 - Lokalitätsprinzip, 94
 - verschränkt, 90
- Restriktions
 - transponierter Restriktionsoperator, 92
- Reynoldszahl, 2
- Schachbrettmuster, 23, 33
- Schwerkraft, 2
- Singularität, 75
- Spacetree, 73
- Spursatz, 20, 26
- Stömung
 - laminar, 2
 - viskose, 2
- Stokes Betrachtungsweise, 2
- Stokes–Gleichungen, 2
- Tensorprodukt, 79
- Transportterme, 2
- Tripartition, 84
- Turbulenzen, 2
- Viskosität, 2
- Zähigkeit, 2
- Zeitschrittweite, 30
- Zeitschrittweitensteuerung, 31
- Zelle
 - äußere, 75
 - innere, 75
 - Macrozelle, 75
 - Microzelle, 75
 - Randzelle, 59, 75