

INAUGURAL – DISSERTATION
zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich–Mathematischen Gesamtfakultät
der
Ruprecht–Karls–Universität
Heidelberg

vorgelegt von
Diplom–Mathematiker Peter Schreiber
aus Stuttgart im
März 1996
Tag der mündlichen Prüfung:

Eine nichtkonforme Finite-Elemente-Methode zur
Lösung der inkompressiblen 3-D Navier-Stokes
Gleichungen

Gutachter: Prof. Dr. Rolf Rannacher
Prof. Dr. Hans-Georg Bock

Inhaltsverzeichnis

Einleitung und Notationen	11
1 Nichtkonforme Finite-Elemente-Ansatzräume zur Lösung der dreidimensionalen Stokes-Gleichungen	17
1.1 Die Stokes-Gleichungen	18
1.2 Eine Diskretisierung der Stokes-Gleichungen	20
1.3 Die nichtkonformen “rotiert“ trilinearen Elemente	23
1.4 Die nichtparametrische Version	29
1.5 Numerische Ergebnisse für das Stokes-Problem	38
1.5.1 Superkonvergenz für den L^2 -Fehler des Drucks:	43
1.5.2 Superkonvergenz für den H^1 -Fehler der Geschwindigkeiten:	45
2 Der Mehrgitteralgorithmus für das Stokes Problem	53
2.1 Die Formulierung des Mehrgitterverfahrens	54
2.2 Der Vanka-Glätter	59
2.3 Die Gittertransferoperatoren	60
2.4 Die adaptive Schrittweitenkontrolle	64
2.5 Numerische Ergebnisse für das Mehrgitterverfahren	65

3	Die Navier–Stokes–Gleichungen	69
3.1	Die stationären Navier–Stokes–Gleichungen	70
3.1.1	Upwind–Diskretisierung	73
3.1.2	Stromliniendiffusionsmethode	76
3.1.3	Numerischer Vergleich beider Stabilisierungen	79
3.2	Die instationären Navier–Stokes–Gleichungen	83
4	Ein Operator–Splitting Ansatz zur Lösung der instationären Navier–Stokes Gleichungen	89
4.1	Die diskrete Projektionsmethode	90
4.2	Numerische Komplexität	92
4.3	Numerischer Vergleich	96
5	Numerische Testrechnungen und ihre graphische Präsentation	101
5.1	Die 3D–Driven–Cavity	101
5.2	Quader– und Zylinderumströmung	110
5.3	Die “tanzenden Rauchringe“	117
5.4	Fiktive Randtechniken im Mehrgitteralgorithmus	121
5.4.1	Bewegte Ränder 1: Eine Wasserpumpe in 2D	124
5.4.2	Bewegte Ränder 2: Eine Wasserpumpe in 3D	128
5.4.3	Ein drehendes Rotorblatt	130
6	Das Industrieprojekt mit der Firma Vaillant	137
6.1	Problemstellung	138

	3
6.2 Die Physik des Problems	141
6.3 Numerische Ergebnisse für den Brenner	142
6.3.1 Der 2D-Fall	142
6.3.2 Der 3D-Fall	145
7 Zusammenfassung und Ausblick	149
Literaturverzeichnis	151

Abbildungsverzeichnis

1.1	Verfeinerung eines Elementes	22
1.2	Trilineare 1-1-Transformation Ψ_T	24
1.3	Mittelpunkt einer Fläche im \mathbf{R}^3	24
1.4	Darstellung der Gitterstörung	27
1.5	Grobgitter und 3 Verfeinerungen	29
1.6	Element in x-Richtung gestreckt	30
1.7	Element in x- und z-Richtung gestreckt	30
1.8	Lokales Koordinatensystem	31
1.9	Normale einer Fläche eines Elementes	33
1.10	Verzerrtes Element in 2D	34
1.11	Zerlegung in Tetraeder	36
1.12	Gaußsche Integrationspunkte für Tetraeder	37
1.13	Interpolation rotiert bilinear auf biquadratisch	46
1.14	Interpolation rotiert trilinear auf triquadratisch	47
2.1	Verschiedene Zyklen	58
2.2	Die nichtkonforme Prolongation	61

2.3	Grobgitter mit Geschwindigkeitsvektoren	66
3.1	Baryzentrische Fragmente S_{ij} einer Fläche eines Hexaeders	74
3.2	Lumping Gebiet R_l der Fläche l	75
4.1	Druckisoflächen zu unterschiedlichen Zeitpunkten	98
5.1	Definition der 3D-Driven-Cavity	102
5.2	Qualitativer Charakter der 3D-Driven-Cavity	103
5.3	3D-Driven-Cavity, 2D Schnitt	104
5.4	3D-Driven-Cavity zu verschiedenen Zeitpunkten	105
5.5	Angepaßte Grobgitter	108
5.6	Quaderumströmung	110
5.7	Zylinderumströmung	111
5.8	Grobgitter ($8h_1$) für Quader- und Zylinderumströmung	114
5.9	Puff-Puff, Visualisierung mit particle tracing	120
5.10	Kugelumströmung	121
5.11	Grobgitter	122
5.12	“Isolinien des Drucks“	122
5.13	“Isolinien der Geschwindigkeit“	123
5.14	Grobgitter für Wasserpumpe	125
5.15	Ergebnisse für Wasserpumpe	126
5.16	y-Komponente und Druck der Geschwindigkeit in $(0,0)$	127

5.17	y-Komponente der Geschwindigkeit bei Becker	127
5.18	Grobgrid für 3D-Wasserpumpe	129
5.19	Geschwindigkeiten in Schnittebene für 3D-Wasserpumpe	130
5.20	Unterschiedliche Schnittebenen für 3D-Wasserpumpe	131
5.21	x- und y-Komponente der Geschwindigkeit in (0,0)	132
5.22	Druck in (0,0)	132
5.23	Bewegung eines Rotorblattes	133
5.24	Druckisolinien für das Rotorblatt	134
5.25	2D Schnittebene für Geschwindigkeiten	135
5.26	Druckisofläche	136
5.27	Geschwindigkeitsisofläche	136
6.1	Geometrie des Brenners von zwei Seitenansichten	138
6.2	2D Grobgrid für Brennermodell	139
6.3	Grobgrid für Brennermodell	140
6.4	2D Schnitt bei Lochplatte	140
6.5	2D Geschwindigkeiten und Druck	143
6.6	Flux zu verschiedenen Zeitpunkten	144
6.7	Flux zu verschiedenen Konfigurationen	144
6.8	2D Geschwindigkeiten	145
6.9	3D Fluxdarstellung	146
6.10	3D Geschwindigkeitsvektoren	147

6.11 Flux an zwei Punkten gemessen 147

Tabellenverzeichnis

1.1	Gaußsche Integrationskonstanten	37
1.2	Ergebnisse für die parametrischen Elemente	39
1.3	Ergebnisse für die nichtparametrischen Elemente	40
1.4	Ergebnisse für 20% Gitterstörung	41
1.5	Ergebnisse für hohen aspect ratio	42
1.6	Konvergenzraten für “puff-puff“ Stokes-Problem	43
1.7	L^2 -Fehler des Drucks	44
1.8	L^2 -Fehler des Drucks	45
1.9	H^1 -Fehler der Geschwindigkeiten	48
1.10	H^1 -Fehler der Geschwindigkeiten	48
1.11	L^2 -Fehler der Geschwindigkeiten	49
1.12	H^1 -Fehler der Geschwindigkeiten	49
1.13	L^2 -Fehler der Geschwindigkeiten	50
1.14	H^1 -Fehler der Geschwindigkeiten	50
1.15	L^2 - und L^∞ -Fehler für den Druck	51
2.1	Interpolationsgewichte für die Prolongation	62

2.2	Konvergenzraten und Effizienzmaß für i.)–iii.)	67
2.3	Konvergenzraten und Effizienzmaß für die 3D–Driven–Cavity	67
2.4	Konvergenzraten und Effizienzmaß für Test Kapitel 1+2	68
3.1	Fehler für Stromliniendiffusion, $Re = 10$	80
3.2	Fehler für <i>Upwinding</i> , $Re = 10$	81
3.3	Fehler für Stromliniendiffusion, $Re = 250$	81
3.4	Fehler für <i>Upwinding</i> , $Re = 250$	81
3.5	Fehler für <i>Upwinding</i> , $Re = 500$	82
3.6	Fehler für <i>Upwinding</i> , $Re = 10$, <i>aspect ratio</i> =12	82
3.7	Fehler für <i>Stromliniendiffusion</i> , $Re = 10$, <i>aspect ratio</i> =12	82
3.8	Fehler für $\delta^* = 1$, $Re = 250$	83
5.1	Gitter 3D–Q und 3D–Z	114
5.2	Ergebnisse 3D–1Q und 3D–1Z	115
5.3	Ergebnisse 3D–2Q und 3D–2Z	115
5.4	Ergebnisse 3D–3Q und 3D–3Z	116
5.5	Ergebnisse für Gitter 1 und 2	123
5.6	Konvergenzraten Fall 1 und 2	124

Einleitung und Notationen

Gegenstand dieser Arbeit ist die Entwicklung eines effizienten Lösungsverfahrens für die Navier–Stokes–Gleichungen der Form ([20])

$$\mathbf{u}_t - \nu \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p = \mathbf{f} \quad , \quad \nabla \cdot \mathbf{u} = 0 \quad , \quad \text{in } \Omega \times (0, T) \text{ ,}$$

mit gegebenen Rand- und Startwerten.

Wesentlich ist hierbei die Tatsache, daß in dieser Arbeit der Fall $\Omega \subset \mathbf{R}^3$ betrachtet werden soll, wodurch die entstehenden Gleichungssysteme sehr groß werden, da man mit ähnlichen Ortsschrittweiten h arbeiten will wie in zwei Dimensionen, um die gleiche Genauigkeit der Lösung zu erzielen. Um zum Beispiel eine bestimmte Ortsgenauigkeit zu erreichen sind nun statt 100^2 Gitterpunkte in 2D, 100^3 Gitterpunkte in 3D zu verwenden. Und in 2D ist diese Gitterpunktzahl eher das untere Limit.

Bis vor wenigen Jahren konnten Probleme dieser Größenordnung mit bis zu 10 Millionen Unbekannten nur auf Großrechnern bearbeitet werden. Durch die Entwicklung immer leistungsfähigerer Arbeitsplatzrechner mit heute bis zu 1 Gigabyte Arbeitsspeicher und mehr als 100 Mflops (z.B. IBM RS6000/590), bzw. durch Einsatz von Vektor- und Parallelrechnern, hat sich dies grundlegend verändert. Dies bietet die Möglichkeit, die Algorithmen und Lösungsstrategien, die seit geraumer Zeit schon erfolgreich im Falle der 2D Navier–Stokes–Gleichungen verwendet werden, für den 3D–Fall auszuweiten. So ist diese Arbeit entstanden in der Tradition des ursprünglich in Saarbrücken entwickelten Programmpakets **FEAT2D** (siehe [5]) mit der Erweiterung auf **FEAT3D** (siehe [23]) und der speziell für den CFD Bereich abgestimmten Version **FEATFLOW** (siehe [50]).

Das Herzstück einer jeden Finiten Element Methode stellt die Wahl des diskreten Ansatzraumes dar. In dieser Arbeit wird ein “rotiert“ trilinearer nichtkonformer Ansatz für die Geschwindigkeiten, bzw. stückweise konstanter für den Druck gewählt. Die-

ses Stokes–Element wurde in zwei Dimensionen erstmals von Rannacher/Turek in [39] vorgestellt als Vierecks Pendant zum Crouzeix–Raviart Element ([13]) für den Dreiecksfall. Dieses Element wird als “rotiert bilinear“ bezeichnet, rotiert deswegen, da die Freiheitsgrade für die Geschwindigkeiten auf den Seitenmitten liegen, statt wie beim linearen konformen Element auf den Ecken. In drei Raumdimensionen werden Gitter aus Hexaedern bestehend verwendet. Die Freiheitsgrade liegen nun auf den Flächen, bzw. Flächenmitten. Vorteile dieses Ansatzes sind, daß die Stabilitätsbedingung, die sogenannte *Babuška–Brezzi–Bedingung*, unter gewissen Einschränkungen an die Gitter im sogenannten *parametrischen* Fall erfüllt ist, wogegen bei den entsprechenden konformen Elementen (z.B. Q_1/Q_0) diese Stabilitätsbedingung im allgemeinen nicht gilt und Stabilisierungstechniken für den Druck zum Einsatz kommen müssen (siehe hierzu [28] und [2]).

Eine der wichtigsten Aufgaben dieser Arbeit besteht darin, die einschränkenden Bedingungen an die zugrunde liegenden Gitter aufzuheben und sowohl theoretisch, als auch numerisch volle Stabilität und Approximationsgüte zu garantieren. Dies ist ein wichtiger Aspekt, da man sonst kaum in der Lage ist, komplexe Probleme aus den verschiedenen Bereichen der Naturwissenschaften anzugehen. Dies gelingt durch Verwendung eines *nichtparametrischen* Ansatzes. Der Unterschied zwischen diesen beiden Ansätzen besteht darin, daß der *parametrische* Ansatz wie üblich bei Finite Elemente Ansätzen mit einer Transformation auf ein Referenzelement arbeitet. Dort werden dann die Basisfunktionen etc. aufgestellt. Der Nachteil dieses Ansatzes ist bei unserem Elementepaar, daß durch diese Transformationen Störungen eingebracht werden, die sowohl die Stabilität des Ansatzes einschränken, und sich auch in den Fehlerabschätzungen niederschlagen.

Die Idee eines *nichtparametrischen* Ansatzes besteht nun darin, bei dessen Konstruktion durch Erzeugen eines lokalen Koordinatensystemes auf die Transformation auf ein Referenzelement zu verzichten. Hier ist es dann möglich, volle Approximationsgüte zu zeigen, sowie Stabilität auch für große *aspect ratios* (Verhältnis längste/kürzeste Kante eines Elementes) zu garantieren. Dies ist ein sehr wichtiger Punkt, da man gerade in drei Dimensionen bei vielen Anwendungen gezwungen ist, aufgrund des begrenzten Speichers das Gitter geschickt anzupassen, um möglichst nur in “interessanten“ Bereichen, bzw. Richtungen (z.B. normal zum Rand) des Gebietes zu verfeinern. Hier werden in Zukunft immer mehr adaptive Techniken zum Einsatz kommen, die aber nur konsistent arbeiten können bei einer stabilen Diskretisierung.

Basierend auf dieser Ortsdiskretisierung ist es möglich, einen auf die nichtkonformen Elemente abgestimmten Mehrgitteralgorithmus zu konstruieren. Die Grundideen dieses Verfahrens finden sich bei Brenner ([9], [10], [11]) und Turek ([48]). Die Besonderheiten in diesem Fall liegen darin, Druck und Geschwindigkeiten simultan zu behandeln, was zu einem effektiven Lösungsverfahren für die Stokes–Gleichungen führt. Schwierigkeiten bereiten dabei vor allem die Konstruktion der Gittertransferoperatoren, da die diskreten Ansatzräume nicht ineinander geschachtelt sind, wie im konformen Fall. Durch geschickte Wahl dieser Operatoren wird die Effizienz des Verfahrens erheblich gesteigert, zusätzliche Verbesserungen erzielt dabei noch eine sogenannte adaptive Schrittweitenkontrolle im Korrekturschritt. Umfangreiche Testrechnungen bestätigen diese Überlegungen.

Dieses Mehrgitterverfahren liefert dann die Grundlage zur Lösung der inkompressiblen Navier–Stokes–Gleichungen. Eine weitere numerische Schwierigkeit stellt hier die Stabilisierung des konvektiven Terms für hohe Reynoldszahlen dar. In diesem Rahmen werden zwei verschiedene Techniken verwendet, zum einen ein *upwinding*–Verfahren, das sich sehr robust zeigt im Falle hoher Reynoldszahlen, zum anderen die *Stromliniendiffusionsmethode*, die zwar bessere Fehler liefert, dafür aber Instabilitäten zeigt. Angekoppelt an diesen stationären Löser wird dann eine Zeitdiskretisierung zur Lösung der instationären Gleichungen. Wir verwenden hauptsächlich das Fractional–step– ϑ –Verfahren (siehe [34]), das sich in der Praxis Einschnittverfahren gegenüber überlegen zeigt. Vorteile dieses voll gekoppelten Ansatzes ist der implizite Charakter des Verfahrens, der es zuläßt, mit relativ großen Zeitschritten zu arbeiten, und der sich als außerordentlich stabil erweist.

Der entscheidende Nachteil ist aber die doch eher mäßige Effizienz, verursacht durch den verwendeten Vanka–Glätter im Mehrgitterverfahren. Dieser Glättungsoperator, den man als ein Block–Gauß–Seidel Verfahren interpretieren kann, zeigt ein schlechtes Konvergenzverhalten für kleine Zeitschritte. Das bedeutet aber, daß die Einführung einer adaptiven Zeitschrittkontrolle, um feine instationäre Strukturen aufzulösen, auf Kosten der Effizienz geht. Dies führte zur Entwicklung eines anderen Zuganges, des sogenannten *Diskreten Projektionsverfahrens* von S. Turek für den zweidimensionalen Fall (siehe auch Gresho [21]), das in dieser Arbeit auf den 3D Fall erweitert wird. Auch hier zeigt sich die Wahl des Ansatzraumes als äußerst günstig, da dadurch eine sehr effiziente Implementierung möglich ist. Im Gegensatz zum gekoppelten Ansatz hat man es nun mit definiten Teilproblemen für Druck und Geschwindigkeiten zu tun. Damit ist

es möglich, schnelle und robuste Löser vom z.B. *ILU*-Typ einzusetzen, was eine große Beschleunigung bewirkt. Umfangreiche Testrechnungen und Vergleiche mit anderen Programmen und Anwendern werden dies anhand eines Benchmark Tests belegen.

War es bis vor einigen Jahren nur möglich, eingeschränkte Problemklassen eher akademischen Charakters zu betrachten, ist es nun möglich, mit den entwickelten Verfahren eine weit größere Klasse von Problemen anzugehen, wie umfangreiche Testrechnungen bestätigen. Der 3D Strömungscode wird an folgenden Aspekten getestet:

- Komplexe Geometrien,
- Starke Gitteranisotropien –Streckungen und Verzerrungen– ,
- Variierende Randbedingungen in Ort und Zeit,
- Verschiedene Stabilisierungen des konvektiven Terms,
- Projektionsverfahren versus gekoppelten Ansatz in puncto Genauigkeit und Effizienz.

Die numerischen Ergebnisse, vor allem die Vergleiche der Verfahren gegenüber instationären Benchmark Tests (siehe [41]), führen zu dem Resultat, daß mit den in dieser Arbeit analysierten und implementierten Lösern ein Programmpaket vorgestellt wird, mit dem

- schnell –innerhalb eines Tages auf einer Workstation–,
 - flexibel –Anpassung auf neue Probleme sehr einfach–,
 - modular –das Programm ist offen für Ankopplung neuer Routinen–
- Probleme von bis zu 10 Millionen Unbekannten bewältigt werden können.

Die Arbeit ist so gegliedert, daß im ersten Kapitel die *parametrischen* und *nichtparametrischen* Stokes-Elemente hergeleitet und analysiert werden. Die theoretischen Ergebnisse werden durch zahlreiche Tests komplettiert. Das zweite Kapitel stellt den Mehrgitteralgorithmus vor, dabei wird der Aspekt des Gittertransfers und der adaptiven Schrittweitenkontrolle besonders berücksichtigt. Das darauffolgende Kapitel behandelt die stationären und instationären Navier-Stokes Gleichungen unter besonderer Berücksichtigung des konvektiven Terms, sowie der Zeitdiskretisierung.

Im vierten Kapitel wird der Zugang des Operator-Splittings als Alternative zum gekoppelten Ansatz vorgestellt und numerischen Tests unterworfen, die dann im nächsten Kapitel weiter ausgeführt werden. Dieses fünfte Kapitel beginnt mit Ergebnissen zweier

Benchmark Tests, einer *3D-Driven-Cavity* und der Umströmung eines Quaders, bzw. Zylinders. Einen großen Anteil nimmt im weiteren Verlauf dann die Untersuchung verschiedener Randbedingungen ein. Im Vordergrund steht die Anwendung der sogenannten *do nothing*-Bedingung, weiterhin die praktische Umsetzung von fiktiven und bewegten Rändern.

Die Untersuchung der letztgenannten Fragen haben dabei eine Basis geschaffen, mit der ein praxisrelevantes Problem aus der Industrie angegangen werden kann. Dies ist der Inhalt des letzten Kapitels. Mit Hilfe der in den vorangegangenen Kapiteln bereitgestellten Techniken wird ein industrierelevantes Problem behandelt, nämlich die Simulation einer vollständigen Strömung (und Verbrennung) in einem Heizbrenner. Dies ist Teil eines **BMBF**-Verbundprojektes “Anwendungsorientierte Mathematik“. Dabei wird der Teil des Projektes, der sich mit der Strömung beschäftigt, in Heidelberg bearbeitet, die Verbrennungssimulation wird an der TU Cottbus von Professor Bader übernommen. Die Ergebnisse dieses Kapitels lassen die Hoffnung zu, daß mathematische Methoden in der Industrie eine größere Akzeptanz finden und praktisches Handwerkszeug in vielen Bereichen werden können.

Der Autor möchte die Gelegenheit nutzen, sich zu bedanken bei Herrn Prof. Dr. Ranacher für die Bereitstellung der Mittel und die Unterstützung in vielen Fragen, bei Herrn Dr. Turek, für die Zusammenarbeit und Hilfe bei allen Problemen ob technischer oder inhaltlicher Art und vor allem bei Herrn Dr. Zhou für die Beratung und Diskussion bei theoretischen und praktischen Problemen. Darüberhinaus gilt der Dank der ganzen AG Numerik.

Notationen

In der gesamten Arbeit verwenden wir folgende Notationen:

Die Räume $L^2(\Omega)$ und $H^m(\Omega)$ in dieser Arbeit sind die üblichen Lebesgue- und Sobolev-Räume auf dem Gebiet $\Omega \subset \mathbf{R}^d$, $d = 2, 3$ mit den üblichen Normen $\|\cdot\|_0$ und $\|\cdot\|_m$. Die inneren Produkte von $L^2(\Omega)$ werden mit (\cdot, \cdot) gekennzeichnet. Der Raum $H_0^1(\Omega)$ ist die Vervollständigung der Testfunktionen $C_0^\infty(\Omega)$ bzgl. der Norm $\|\cdot\|_1$ und $H^{-1}(\Omega)$ sein Dualraum. Mit $L_0^2(\Omega)$ bezeichnen wir den Unterraum aller $L^2(\Omega)$ -Funktionen auf Ω mit Mittelwert 0. Vektorwertige Funktionen und Räume werden in Fettdruck dargestellt, so daß gilt $\mathbf{L}^2(\Omega) := L^2(\Omega) \times L^2(\Omega) \times L^2(\Omega)$, $\mathbf{H}^m(\Omega) := H^m(\Omega) \times H^m(\Omega) \times H^m(\Omega)$. Normalenvektoren werden mit \mathbf{n} und Tangentialvektoren mit \mathbf{t} gekennzeichnet. Alle übrigen Notationen werden an entsprechender Stelle eingeführt und erklärt.

1. Nichtkonforme Finite-Elemente-Ansatzräume zur Lösung der dreidimensionalen Stokes-Gleichungen

In diesem Kapitel untersuchen wir als Spezialfall der Navier-Stokes-Gleichungen das Stokes-Problem, das den Strömungsverlauf eines stationären, viskosen, inkompressiblen Mediums beschreibt. Vom mathematischen Gesichtspunkt aus betrachtet ist das Stokes-Problem zum einen äquivalent zu einem Optimierungsproblem mit linearen Nebenbedingungen in Form der Inkompressibilität und zum anderen beschreibbar als ein Sattelpunktproblem mit dem Druck als Lagrange-Multiplikator. Die eindeutige Lösbarkeit dieser Probleme wird durch eine Stabilitätsabschätzung impliziert, die von den Lösungsräumen $\mathbf{H}_0^1(\Omega)$ für die Geschwindigkeiten und $L_0^2(\Omega)$ für den Druck erfüllt wird. Zur Lösung der Stokes-Gleichungen mit Hilfe der Finite-Elemente-Methode müssen diskrete Ansatzräume definiert werden, die einem diskreten Analogon dieser Stabilitätsbedingung, der *Babuška-Brezzi-Bedingung*, genügen. Wir beschränken uns dabei auf möglichst einfache Ansatzräume, d.h. verwendet werden nur elementweise konstante Druckapproximationen und „rotiert“ trilineare Geschwindigkeitsansätze. Wichtig hierbei ist die Tatsache, daß ein konformer trilinearer Ansatzraum (Q_1/Q_0) die Stabilitätsbedingung im allgemeinen nicht erfüllt. Daher verwenden wir nichtkonforme Finite-Elemente-Ansatzräume, was dazu führt, daß sowohl die Analyse (fehlende Orthogonalitätsbeziehung zwischen den Ansatzräumen), als auch die Techniken zum Lösen (Prolongation und Restriktion beim Mehrgitteralgorithmus) komplizierter werden. Dafür kommt man aber ohne zusätzliche Stabilisierungstechniken aus (vgl. [2] oder [28]), die ebenfalls Probleme bereiten können.

Für das zweidimensionale Stokes-Problem bieten sich im wesentlichen zwei verschiedene Möglichkeiten zur Diskretisierung an. Für das schon bekannte nichtkonforme *Crouzeix-Raviart-Element* ([13]) (stückweise lineare Geschwindigkeiten, stückweise konstanter Druck) kann man Eindeutigkeit und unter bestimmten Voraussetzungen auch

volle Approximationsgüte der diskreten Lösungen zeigen. Dieses Dreieckselement hat sein Analogon für eine Rechteckzerlegung in dem sogenannten “*rotiert bilinearen*” Ansatz, rotiert deswegen, da die Freiheitsgrade für die Geschwindigkeiten nicht mehr auf den Ecken der Rechtecke liegen, sondern auf den Seitenmitten. Eine ausführliche Analyse dieses Ansatzraumes findet sich in ([39]).

Zur Diskretisierung der Stokes–Gleichungen in drei Raumdimensionen, verwenden wir das dem *rotiert bilinearen* Ansatz entsprechende 3D–Element. Wir haben dann für diese Zerlegung in Hexaeder wieder stückweise konstanten Druck, die Freiheitsgrade für die Geschwindigkeiten liegen nun auf den Mittelpunkten der sechs Flächen, bzw. sind das jeweilige Flächenintegral. Wir bezeichnen dieses Element in Anlehnung an den 2D Fall als *rotiert trilinear*, wobei rotiert hier nicht wörtlich zu nehmen ist wie in zwei Dimensionen (der Widerspruch in der Namensgebung bezieht sich auf die Tatsache, daß das konforme lineare Element die Freiheitsgrade auf den 8 Ecken hat, das nichtkonforme Element aber nur 6 Freiheitsgrade auf den Flächen besitzt). Für das nichtkonforme 3D–Element kann man nun wie in zwei Raumdimensionen volle Approximationsgüte zeigen, wobei man zusätzliche Forderungen an die verwendeten Triangulierungen stellen muß, wenn man einen parametrischen Ansatz wählt. Verwendet man aber einen nichtparametrischen Ansatz, erhält man eine stabile Diskretisierung auch ohne zu einschränkende Forderungen an die Triangulierung des Gebietes. Die daraus resultierenden theoretischen Aussagen werden durch die Ergebnisse numerischer Testrechnungen bestätigt. Darüber hinaus lassen sich für das rotiert trilineare Element Superkonvergenzaussagen ableiten, die ähnlich deren für den konformen Fall sind ([32]).

1.1. Die Stokes–Gleichungen

Sei Ω ein beschränktes Gebiet des \mathbf{R}^3 mit Lipschitz–stetigem Rand $\partial\Omega$. Wir betrachten im folgenden das Stokes–Problem mit Dirichletrandwerten:

Finde ein Paar $\{\mathbf{u}, p\}$, so daß gilt

$$-\nu\Delta\mathbf{u} + \nabla p = \mathbf{f} \quad , \quad \nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega, \quad \mathbf{u} = \mathbf{g} \quad \text{auf } \partial\Omega. \quad (\text{St})$$

Dabei beschreiben $\{\mathbf{u}, p\}$ die unbekannte Geschwindigkeit $\mathbf{u} = (u_1, u_2, u_3)^T$ und den

Druck p einer zähen, inkompressiblen Strömung in Ω mit vorgegebener Volumenkraft $\mathbf{f} = (f_1, f_2, f_3)^T$ und Randgeschwindigkeit $\mathbf{g} = (g_1, g_2, g_3)^T$.

Eine schwache Formulierung von Problem (St) lautet:

Finde ein Paar $\{\mathbf{u}, p\} \in \mathbf{H}^1(\Omega) \times L_0^2(\Omega)$, so daß gilt

$$\begin{aligned} \nu(\nabla \mathbf{u}, \nabla \mathbf{v}) - (p, \nabla \cdot \mathbf{v}) &= (\mathbf{f}, \mathbf{v}) \quad , \forall \mathbf{v} \in \mathbf{H}_0^1(\Omega) , \\ (q, \nabla \cdot \mathbf{u}) &= 0 \quad , \forall q \in L_0^2(\Omega) , \\ \mathbf{u} &= \mathbf{g} \quad , \text{ auf } \partial\Omega . \end{aligned} \tag{W}$$

Seien $\mathbf{f} \in \mathbf{H}^{-1}(\Omega)$ und $\mathbf{g} \in \mathbf{H}^{1/2}(\partial\Omega)$ gegeben mit

$$\oint_{\partial\Omega} \mathbf{g} \cdot \mathbf{n} \, d\sigma = 0 , \tag{1.1}$$

dann ist das Lösungspaar $\{\mathbf{u}, p\}$ von Problem (W) eindeutig bestimmt, und es gilt ([13])

$$\|\mathbf{u}\|_1 + \|p\|_0 \leq c (\|\mathbf{f}\|_{-1} + \|\mathbf{g}\|_{1/2}) . \tag{1.2}$$

Dies ist eine Konsequenz einer bekannten Stabilitätsabschätzung ([20]), die die Räume $\mathbf{H}_0^1(\Omega)$ und $L_0^2(\Omega)$ miteinander koppelt

$$\sup_{\mathbf{v} \in \mathbf{H}_0^1(\Omega)} \frac{(q, \nabla \cdot \mathbf{v})}{\|\nabla \mathbf{v}\|_0} \geq \beta \|q\|_0 > 0 \quad , \forall q \in L_0^2(\Omega) , q \neq 0 . \tag{BB}$$

Im weiteren Verlauf beschränken wir uns aus Gründen der Vereinfachung auf Polyedergebiete $\Omega \subset \mathbf{R}^3$, oder Gebiete deren Ränder hinreichend glatt sind, homogene Randdaten $\mathbf{g} = \mathbf{0}$ und Viskosität $\nu = 1$. Die analogen Ergebnisse für die allgemeineren Fällen können in [12],[13],[20],[47] nachgelesen werden.

Definieren wir zusätzlich die Bilinearformen

$$a(\mathbf{u}, \mathbf{v}) := (\nabla \mathbf{u}, \nabla \mathbf{v}) , \tag{1.3}$$

$$b(p, \mathbf{v}) := -(p, \nabla \cdot \mathbf{v}), \quad (1.4)$$

so läßt sich die schwache Formulierung von Problem (W) kürzer schreiben als:

Finde ein Paar $\{\mathbf{u}, p\} \in \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega)$, so daß gilt

$$a(\mathbf{u}, \mathbf{v}) + b(p, \mathbf{v}) + b(q, \mathbf{u}) = (\mathbf{f}, \mathbf{v}) \quad , \quad \forall \{\mathbf{v}, q\} \in \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega), \quad (V)$$

bzw. in dazu äquivalenter Formulierung ([20]):

Finde ein $\mathbf{u} \in \mathbf{V}(\Omega)$, so daß gilt

$$a(\mathbf{u}, \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \quad , \quad \forall \mathbf{v} \in \mathbf{V}(\Omega), \quad (V^d)$$

mit

$$\mathbf{V}(\Omega) = \{\mathbf{v} \in \mathbf{H}_0^1(\Omega) : \nabla \cdot \mathbf{v} = 0\}. \quad (1.5)$$

Gilt für die rechte Seite $\mathbf{f} \in \mathbf{L}^2(\Omega)$, dann sind $\{\mathbf{u}, p\}$ sogar in $\mathbf{H}^2(\Omega) \times H^1(\Omega)$ und genügen der *a priori* Abschätzung ([20])

$$\|\mathbf{u}\|_2 + \|p\|_1 \leq c \|\mathbf{f}\|_0. \quad (1.6)$$

1.2. Eine Diskretisierung der Stokes–Gleichungen

Sei im folgenden \mathbf{T}_h eine Zerlegung von Ω in (konvexe) Hexaeder mit Gitterweite h , die den maximalen Durchmesser der Elemente von \mathbf{T}_h bezeichnet. Die Elemente, die wir zulassen, sind dadurch charakterisiert, daß sie durch eine trilineare Transformation auf ein Referenzelement abbildbar sind. Wir müssen lediglich die Bedingung stellen, daß die Verbindung zweier Punkte eine Gerade ist. Die Flächen des Hexaeders dürfen gekrümmt sein.

Die Zerlegung sei *regulär* ([1],[12]), d.h., die Schnittmenge zweier Elemente ist entweder leer, ein gemeinsamer Knoten, eine gemeinsame Kante oder eine gemeinsame Fläche.

Weiterhin wird angenommen, daß die Familie $\{\mathbf{T}_h\}_h$ von Zerlegungen der *uniform shape condition* ([1],[12]) genüge, d.h. für jede Zerlegung \mathbf{T}_h sei der Quotient aus maximalen Umkreisradius und minimalen Inkreisradius unabhängig von der Gitterweite beschränkt:

$$d_{\mathbf{T}_h} := \max_{T \in \mathbf{T}_h} \{ \text{Umkreisradius von } T \}$$

$$\Rightarrow c_1 \leq \frac{d_{\mathbf{T}_h}}{q_{\mathbf{T}_h}} \leq c_2 \quad , \quad \forall \mathbf{T}_h \in \{\mathbf{T}_h\}_h . \quad (1.7)$$

$$q_{\mathbf{T}_h} := \min_{T \in \mathbf{T}_h} \{ \text{Inkreisradius von } T \}$$

Folgende Bezeichnungen werden wir im weiteren Verlauf dieser Arbeit verwenden: Die gemeinsame Fläche zweier Elemente $T_i, T_j \in \mathbf{T}_h$ nennen wir Γ_{ij} mit zugehörigem Schwerpunkt m_{ij} . Analog sind $\Gamma_{i0} \subset (\partial\mathbf{T}_h \cap \partial\Omega)$ Randflächen mit zugehörigen Schwerpunkten m_{i0} . Mit $\partial\mathbf{T}_h$ bezeichnen wir die Menge aller Flächen der Elemente. Die gemeinsame Kante zweier Elemente bezeichnen wir mit Υ_{ij} mit Mittelpunkt k_{ij} , entsprechend wieder die Bezeichnungen für die Randkanten.

Unser verwendeter Verfeinerungsalgorithmus sei regulär, zuerst werden auf jeder Fläche die gegenüberliegenden Seitenmitten miteinander verbunden, und dann die gegenüberliegenden Flächenmitten, somit entstehen aus einem Element 8 neue Elemente. Dieser einfache Prozeß hat mehrere Vorteile, zum einen erhalten wir eine strenge Gitterhierarchie, was die Verwendung von Mehrgitteralgorithmen wesentlich vereinfacht, zum anderen erhalten wir bei Grobgittern, die die *uniform shape condition* nur schlecht erfüllen (d.h. mit Konstanten $c_1 \ll c_2$) durch den Verfeinerungsprozeß immer bessere Konstanten, d.h. die Gitter werden immer Parallelogramm ähnlicher. Damit ist auch folgende Strategie gerechtfertigt, im Grobgitter pseudo–adaptiv einzelne Elemente zu verfeinern, indem man sieben Elemente aus ihnen macht, um dann wie oben beschrieben global zu verfeinern (siehe Abbildung 1.1), was den Vorteil hat, daß die volle Mehrgitterhierarchie bewahrt wird und trotzdem in “interessanten“ Bereichen des Gebietes feiner diskretisiert werden kann.

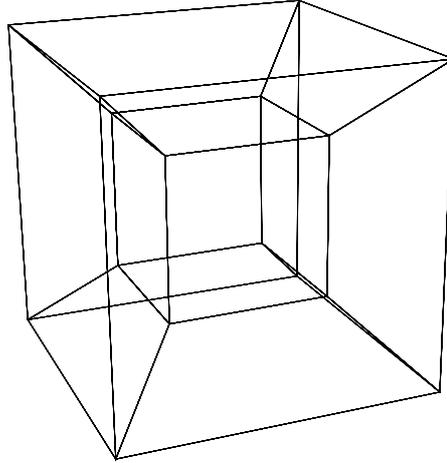


Abbildung 1.1: Verfeinerung eines Elementes

Wie wir in den Abschnitten 1.4 und 1.5 sehen werden, wird die nichtparametrische Version auch bei Gittern mit ungünstigen Winkeln oder großen *aspect ratios* (Verhältnis kürzeste/längste Kante) stabil und mit vollen Fehlerordnungen arbeiten, so daß wir auch bei instationären Rechnungen das Rechengitter nach gewissen Zeiten mit obiger Methode anpassen können, d.h. wir verfeinern eine gewisse Anzahl von Grobgitter Elementen, bauen wieder die volle Gitterhierarchie auf und interpolieren dann die Lösung des alten Gitters auf das Neue und starten dann erneut.

Um Problem (V) mittels der *Finite-Elemente-Methode* approximieren zu können, führen wir diskrete Räume $\mathbf{H}_h \approx \mathbf{H}_0^1(\Omega)$ und $L_h \approx L_0^2(\Omega)$ ein, wobei $h > 0$ die zugehörige Gitterweite zur Zerlegung \mathbf{T}_h ist. Entsprechend den diskreten Räumen \mathbf{H}_h und L_h , die aus elementweise polynomialen Funktionen bestehen, definieren wir diskrete Bilinearformen $a_h(\mathbf{u}, \mathbf{v}) \approx a(\mathbf{u}, \mathbf{v})$ und $b_h(p, \mathbf{v}) \approx b(p, \mathbf{v})$.

Eine diskrete Formulierung von Problem (V) lautet dann:

Finde ein Paar $\{\mathbf{u}_h, p_h\} \in \mathbf{H}_h \times L_h$, so daß gilt

$$a_h(\mathbf{u}_h, \mathbf{v}_h) + b_h(p_h, \mathbf{v}_h) + b_h(q_h, \mathbf{u}_h) = (\mathbf{f}, \mathbf{v}_h) \quad , \quad \forall \{\mathbf{v}_h, q_h\} \in \mathbf{H}_h \times L_h . \quad (\text{V}_h)$$

Hier stellen sich nun die Fragen nach der eindeutigen Lösbarkeit, sowie nach Fehlerabschätzungen für Geschwindigkeit und Druck in geeigneten Normen. Während die erste Frage schon in ([39]) und ([31]) behandelt ist, wollen wir uns in Abschnitt 1.5 vor

allem mit der zweiten Frage beschäftigen.

Zum Nachweis der eindeutigen Lösbarkeit muß die Definitheit der Bilinearform $a_h(\cdot, \cdot)$ auf \mathbf{H}_h und ein diskretes Analogon zu Bedingung (BB), die sogenannte *Babuška-Brezzi-Bedingung* (BB_h), nachgewiesen werden ([20])

$$\max_{\mathbf{v}_h \in \mathbf{H}_h} \frac{b_h(q_h, \mathbf{v}_h)}{\|\mathbf{v}_h\|_h} \geq \tilde{\beta} \|q_h\|_0 > 0, \quad \forall q_h \in L_h, q_h \neq 0. \quad (\text{BB}_h)$$

Die Norm $\|\cdot\|_h$ wird dabei für Funktionen $\mathbf{v}_h \in \mathbf{H}_h \oplus \mathbf{H}_0^1(\Omega)$ definiert durch

$$\|\mathbf{v}_h\|_h := (a_h(\mathbf{v}_h, \mathbf{v}_h))^{1/2}. \quad (1.8)$$

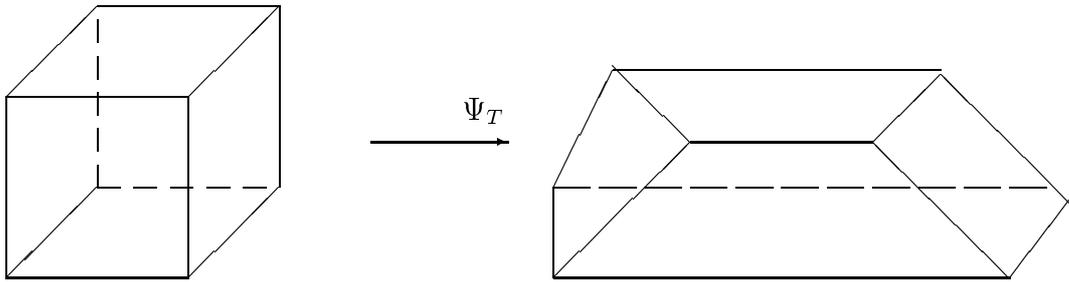
Nach diesen allgemeinen Bemerkungen beginnen wir nun mit der Analyse des nichtkonformen “rotiert“ trilinearen Elements.

1.3. Die nichtkonformen “rotiert“ trilinearen Elemente

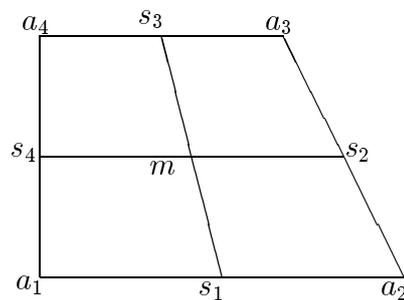
In diesem Paragraphen wollen wir das sogenannte “rotiert“ trilineare Element vorstellen und analysieren. Da dieses Element das 3D Pendant zu dem in zwei Dimensionen verwandten, rotiert bilinearen Element darstellt (siehe [39]), bzw. des Crouzeix–Raviart Elementes ([13]) für den Dreiecksfall, sind viele der nachfolgenden Überlegungen Erweiterungen von entsprechenden des 2D Falles, wobei aber durch die höhere Dimension technisch andere Schwierigkeiten auftauchen. Der erste Teil dieses Abschnittes ist im Wesentlichen eine Wiederholung der Arbeit ([39]) mit den entsprechenden Erweiterungen für den dreidimensionalen Fall.

Sei \mathbf{T}_h eine *reguläre* Zerlegung wie im vorigen Abschnitt beschrieben. Wir beginnen nun mit der Definition der *parametrischen rotiert trilinearen Elemente*. Dazu benötigen wir zunächst das Referenzelement $\hat{T} = [-1, 1] \times [-1, 1] \times [-1, 1]$. Für jedes $T \in \mathbf{T}_h$ definieren wir dann durch $\psi_T : \hat{T} \rightarrow T$ eine zugehörige trilineare 1–1–Transformation und setzen

$$\hat{Q}_1(T) := \{q \circ \psi_T^{-1} \mid q \in \text{span}\langle 1, x, y, z, x^2 - y^2, y^2 - z^2 \rangle\}. \quad (1.9)$$

Abbildung 1.2: Trilineare 1-1-Transformation Ψ_T

Bemerkung 1.1. *Wesentlich für die ganze Definition ist die Tatsache, daß die trilineare Transformation eingeschränkt auf eine Fläche bilinear ist. Damit wird jeder Punkt einer Fläche des Einheitswürfels eindeutig auf einen Punkt des Elementes abgebildet, insbesondere der Mittelpunkt der Fläche, bzw. die Quadraturpunkte für die numerische Integration. Eine einfache geometrische Überlegung verdeutlicht diesen Sachverhalt,*

Abbildung 1.3: Mittelpunkt einer Fläche im \mathbf{R}^3

Wir definieren den Mittelpunkt m als Schnittpunkt der Seitenhalbierenden, verschieben wir den Punkt a_4 in z -Richtung um die Strecke d , so verschieben sich natürlich die Seitenhalbierenden s_1, s_2 um $d/2$, damit der Mittelpunkt m der Fläche um $d/4$, immernoch als Schnittpunkt der Seitenhalbierenden. Es macht also Sinn, weiterhin von einem Mittelpunkt und einer "Fläche" zu reden, auch wenn die 4 Eckpunkte der Fläche nicht mehr in einer Ebene liegen. Damit sind nachfolgende Definitionen, Konstruktionen und Ergebnisse auch für gestörte Elemente, bzw. Gitter gerechtfertigt.

Für die Freiheitsgrade der verwendeten Finiten Elemente haben wir zwei Möglichkeiten, einmal eine punktweise Definition, hier liegen die Freiheitsgrade auf den Schwerpunkten

der Flächen $\{F_\Gamma^{(a)}(\cdot), \Gamma \subset \partial\mathbf{T}_h\}$ oder eine integralmittel orientierte Version, wo die Freiheitsgrade das Integral über den Flächen sind $\{F_\Gamma^{(b)}(\cdot), \Gamma \subset \partial\mathbf{T}_h\}$.

$$F_\Gamma^{(a)}(v) := v(m_\Gamma), \quad (1.10)$$

$$F_\Gamma^{(b)}(v) := |\Gamma|^{-1} \int_\Gamma v d\gamma. \quad (1.11)$$

Die entsprechenden diskreten Räume $\mathbf{H}_h = \mathbf{H}_h^{(a/b)}$ und L_h definieren wir dann als

$$L_h := \{q_h \in L_0^2(\Omega) \mid q_h|_T = \text{const.}, \forall T \in \mathbf{T}_h\}, \quad (1.12)$$

$$H_h := S_h \times S_h \times S_h, \quad (1.13)$$

mit

$$S_h^{(a/b)} := \left\{ \begin{array}{l} v_h \in L^2(\Omega) \mid v_h|_T \in \hat{Q}_1(T), \forall T \in \mathbf{T}_h, v_h \text{ stetig bzgl. der Funk-} \\ \text{tionale } F_{\Gamma_{ij}}^{(a/b)}(\cdot), \forall \text{Flächen } \Gamma_{ij}, \text{ und } F_{\Gamma_{i0}}^{(a/b)}(v_h) = 0, \forall \text{Flächen } \Gamma_{i0} \end{array} \right\}. \quad (1.14)$$

Da wir es hier mit einem nichtkonformen Ansatz zu tun haben, definieren wir die entsprechenden diskreten Bilinearformen $a_h(\cdot, \cdot)$ und $b_h(\cdot, \cdot)$ wie üblich elementweise und erhalten

$$a_h(u_h, v_h) := \sum_{T \in \mathbf{T}_h} \int_T \nabla u_h \cdot \nabla v_h dx, \quad (1.15)$$

$$b_h(q_h, v_h) := - \sum_{T \in \mathbf{T}_h} q_h|_T \int_T \nabla \cdot u_h dx. \quad (1.16)$$

Gemäß der parametrischen Definition der Räume $\mathbf{H}_h^{(a/b)}$ können die zu den Bilinearformen $a_h(\cdot, \cdot)$ und $b_h(\cdot, \cdot)$ entsprechenden Systemmatrizen A_h und B_h effizient durch Rücktransformation auf das Referenzelement \hat{T} berechnet werden.

Sei $j_h : L_0^2 \rightarrow L_h$ der Operator der konstanten Interpolation, der auf jedem Element bestimmt ist durch:

$$j_h p|_T := |T|^{-1} \int_T p dx \quad \forall T \in T_h. \quad (1.17)$$

Dann gilt folgendes Lemma.

Lemma 1.1. *Der konstante Interpolationsoperator $j_h : L_0^2 \rightarrow L_h$ erfüllt die Fehlerabschätzung für $q \in L_0^2 \cap H^1$,*

$$\|q - j_h q\| \leq ch \|q\|_1. \quad (1.18)$$

Der Beweis erfolgt mit Hilfe des Bramble–Hilbert-Lemmas (siehe [39] oder [31]).

Sei weiter $i_h = i_h^{a/b}$ der globale Interpolationsoperator in $\mathbf{H}_h^{(a/b)}$, erzeugt von den lokalen Operatoren i_T . Hier stellt sich das Problem, daß die übliche optimale Fehlerabschätzung für $i_h^{a/b}$ nicht gilt. Das liegt an der Tatsache, daß die Räume $\mathbf{H}_h^{(a/b)}$ nicht “isoparametrisch“ sind, d.h. die multilinearen Transformationen $\Psi_T : \hat{T} \rightarrow T$ sind von einem anderen Polynomtyp als die Formfunktionen auf \hat{T} . Um trotzdem geeignete Approximationseigenschaften für $\mathbf{H}_h^{(a/b)}$ garantieren zu können, müssen wir eine bestimmte schwache Uniformitätsbedingung an die Gitter \mathbf{T}_h stellen. Für jedes Element $T \in \mathbf{T}_h$, sei $\alpha_T \in (0, \pi)$ der maximale eingeschlossene Winkel der Einheitsnormalenvektoren zweier gegenüberliegenden Flächen von T , wobei die Einheitsnormale durch den Mittelpunkt der Fläche verläuft.

Wir definieren dann die Quantität σ_h

$$\sigma_h \equiv \max\{|\pi - \alpha_T|, T \in \mathbf{T}_h\}$$

als Maß für die Störung des Gitters \mathbf{T}_h .

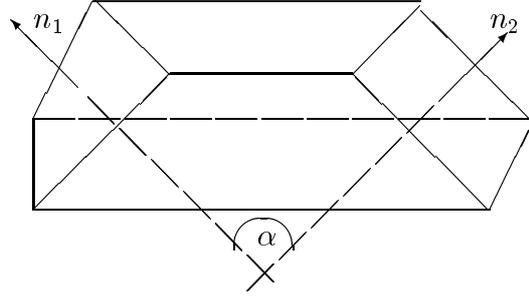


Abbildung 1.4: Darstellung der Gitterstörung

Wir können nun asymptotische Fehlerabschätzungen für den parametrischen Fall angeben. Als Voraussetzungen für folgenden Satz benötigt man Interpolationsabschätzungen für den Operator $i_h = i_h^{a/b}$, sowie eine Stabilitätseigenschaft, die im Falle des mittelpunktorientierten Elementes nur erfüllt ist, wenn die Störung σ_h hinreichend klein ist. Die Details für diese Aussagen, ebenso der Beweis des Satzes finden sich in ([39]). Da wir in Abschnitt 1.4 noch einmal darauf eingehen, geben wir sie hier kurz an.

Lemma 1.2. Für den Interpolationsoperator $i_h = i_h^{a/b}$ gilt die Fehlerabschätzung

$$\|v - i_h v\|_0 + h\|v - i_h v\|_h \leq Ch(h + \sigma_h)\|v\|_2, \quad \forall v \in H_0^1 \cap H^2. \quad (1.19)$$

Der Beweis verläuft analog zum Beweis der Fehlerabschätzung für den Druck. Zuerst wird der Fehler auf das Referenzelement transformiert, dort wendet man das Bramble–Hilbert–Lemma an und bekommt eine Abschätzung gegen die Halbnorm der zweiten Ableitungen. Bei der Rücktransformation auf das Element kommt nun aber die Tatsache zum Tragen, daß die gemischten zweiten Ableitungen der Transformation Ψ_T nur proportional zu $|\pi - \alpha_T|$ sind und nicht zu $O(h^2)$ (siehe [39]). Die zweiten Ableitungen müssen aber die h -Potenzen liefern, damit ist der Interpolationsfehler aber nur noch ($O(h|\pi - \alpha_T|)$) und genau dann optimal, falls die Störung des Gitters $O(h)$ ist.

Für die Lösbarkeit von Problem (V_h) aus Abschnitt 1.2 ist die diskrete *Babuška–Brezzi-Bedingung* (BB_h) zu zeigen:

Lemma 1.3. Für $\mathbf{H}_h = \mathbf{H}_h^{(b)}$ und, wenn die Quantität σ_h hinreichend klein ist, auch für $\mathbf{H}_h = \mathbf{H}_h^{(a)}$ gilt die Stabilitätsabschätzung (mit $\tilde{\beta} \neq \tilde{\beta}(h) > 0$)

$$\max_{\mathbf{v}_h \in \mathbf{H}_h} \frac{b_h(q_h, \mathbf{v}_h)}{\|\mathbf{v}_h\|_h} \geq \tilde{\beta} \|q_h\|_0 > 0, \quad \forall q_h \in L_h, q_h \neq 0. \quad (1.20)$$

Der Beweis findet sich in [31]. Mit den Lemmata 1.1, 1.2 und 1.3 läßt sich der Satz über die asymptotischen Fehlerabschätzungen für die Elementepaare $(\mathbf{H}_h^{(a/b)}, L_h)$ formulieren.

Satz 1.4. *Für $\mathbf{H}_h = \mathbf{H}_h^{(b)}$, und wenn die Quantität σ_h hinreichend klein ist, auch für $\mathbf{H}_h = \mathbf{H}_h^{(a)}$, hat das diskrete Stokes Problem (W) eine eindeutige Lösung $\{u_h, p_h\} \in \mathbf{H}_h^{(a/b)} \times L_h$, und es gilt:*

$$\|u - u_h\|_h + \|p - p_h\| \leq c(h + \sigma_h)\{\|u\|_2 + \|p\|_1\}, \quad (1.21)$$

$$\|u - u_h\| + \|p - p_h\|_{-1} \leq c(h + \sigma_h)^2\{\|u\|_2 + \|p\|_1\} \quad (1.22)$$

(hier bedeutet $\|\cdot\|_{-1}$ die Norm des Dualraumes von $L_0^2 \cap H^1$).

1.4. Die nichtparametrische Version

Wie wir im vorigen Abschnitt gesehen haben, geht die Gitterstörung in die Konstante der Fehlerabschätzung ein, so daß wir für die parametrische Version keine optimale Ordnung erwarten können für Fälle, in denen wir aus praktischen Gründen Gitter verwenden müssen, bei denen diese Störung σ_h groß ist. Ein Beispiel für solch einen Fall zeigen wir in der folgenden Abbildung.

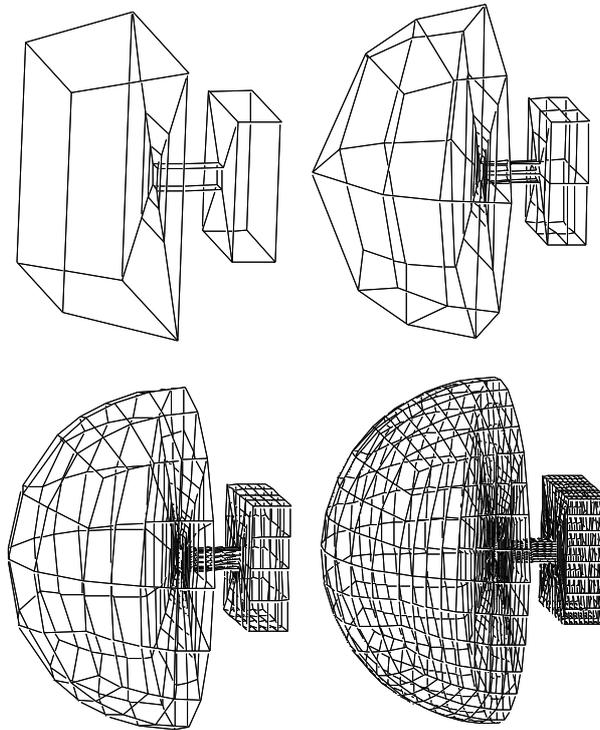


Abbildung 1.5: Grogitter und 3 Verfeinerungen

Man sieht, wie in diesem Fall eine Verfeinerung an einen krummlinigen Rand –in diesem Fall eine Halbkugel– angepaßt wird. Das Grogitter links oben wird durch unseren schon beschriebenen Algorithmus verfeinert, nach jeder Verfeinerung werden die Randknoten des Gitters auf den tatsächlichen Gebietsrand projiziert, so daß man nach

mehreren Verfeinerungen die Geometrie des realen Gebietes sehr gut erfaßt hat (rechts unten). Der Nachteil besteht darin, daß man nun Elemente in dem Gitter hat, bei denen die Störung σ_h sehr groß werden kann. Wie wir in dem letzten Abschnitt aber gesehen haben, geht diese Störung durch die Transformation auf das Referenzelement in die Fehlerabschätzung ein und natürlich auch in die Stabilitätskonstante. Daher wollen wir in diesem Abschnitt eine Konstruktion angeben, bei der man (fast) ohne Transformation auf das Referenzelement auskommt (bis auf Flächen- und Volumenberechnungen). Die Kosten für diese nichtparametrische Version sind dann natürlich höher, da dies aber vor allem im Preprocessing, also Matrixaufbau etc., geschieht, spielt dies vor allem bei instationären Rechnungen keine große Rolle.

Der zweite Aspekt einer möglichen Gitterstörung, den wir bis jetzt noch nicht betrachtet haben, ist der Fall eines großen *aspect ratios*, der in drei Dimensionen folgendermaßen aussehen kann:

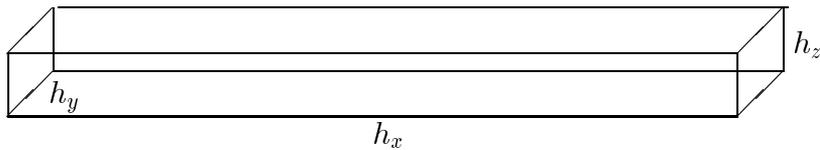


Abbildung 1.6: Element in x-Richtung gestreckt

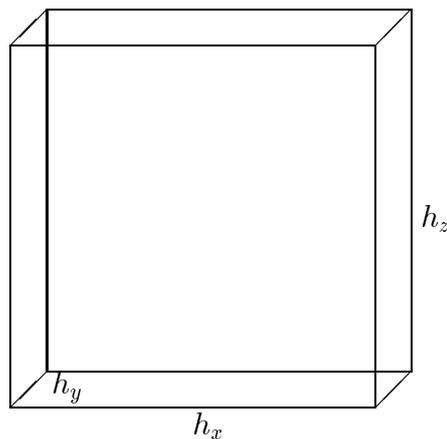


Abbildung 1.7: Element in x- und z-Richtung gestreckt

Oftmals ist man in der Praxis gezwungen, Elemente dieser Form zu benutzen; für die nichtparametrische Definition ist es möglich, Stabilität auch für große *aspect ratios* zu beweisen.

Zur Konstruktion:

Für jedes Element $T \in \mathbf{T}_h$ sei (ξ, μ, λ) ein lokales Koordinatensystem, welches durch Verbinden der Mittelpunkte zweier gegenüberliegenden Flächen erzeugt wird (siehe Abbildung 1.8).

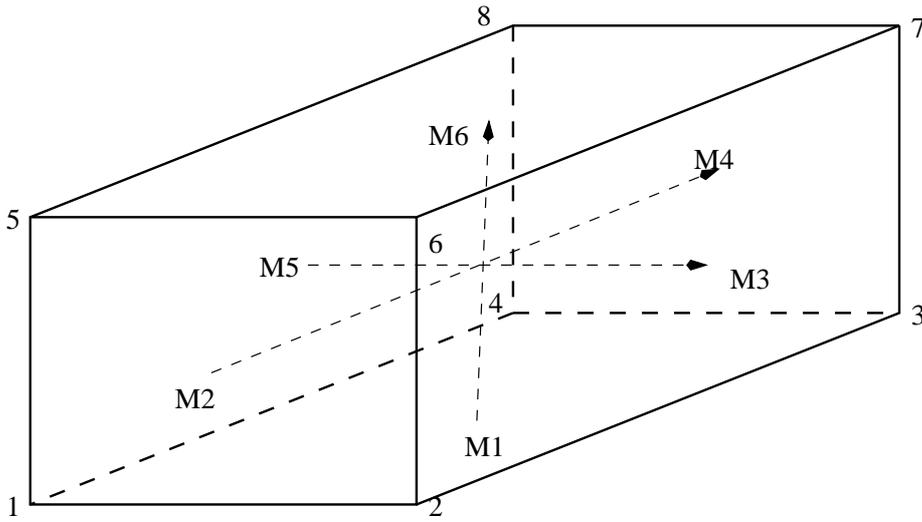


Abbildung 1.8: Lokales Koordinatensystem

Der Ansatzraum $\hat{Q}_1(T)$ ist auf jedem Element $T \in \mathbf{T}_h$ gegeben durch:

$$\hat{Q}_1(T) := \text{span}\langle 1, \xi, \mu, \lambda, \xi^2 - \mu^2, \mu^2 - \lambda^2 \rangle \quad (1.23)$$

mit

$$\xi = a_\xi x + b_\xi y + c_\xi z$$

$$\mu = a_\mu x + b_\mu y + c_\mu z$$

$$\lambda = a_\lambda x + b_\lambda y + c_\lambda z.$$

Wir haben dann

$$(a_\xi, b_\xi, c_\xi) = (XM1 - XM6, YM1 - YM6, ZM1 - ZM6)$$

$$(a_\mu, b_\mu, c_\mu) = (XM2 - XM4, YM2 - YM4, ZM2 - ZM4)$$

$$(a_\lambda, b_\lambda, c_\lambda) = (XM3 - XM5, YM3 - YM5, ZM3 - ZM5).$$

Damit läßt sich unser Ansatzraum $\hat{Q}_1(T)$ schreiben als

$$\begin{aligned} \hat{Q}_1(T) := \text{span} \langle & 1, a_\xi x + b_\xi y + c_\xi z, a_\mu x + b_\mu y + c_\mu z, a_\lambda x + b_\lambda y + c_\lambda z, \\ & d_1 x^2 + d_2 y^2 + d_3 z^2 + d_4 xy + d_5 xz + d_6 yz, \\ & e_1 x^2 + e_2 y^2 + e_3 z^2 + e_4 xy + e_5 xz + e_6 yz \rangle \end{aligned} \quad (1.24)$$

mit Koeffizienten $d_i(a_\xi, b_\xi, c_\xi, a_\mu, b_\mu, c_\mu, a_\lambda, b_\lambda, c_\lambda)$, bzw. $e_i(a_\xi, b_\xi, c_\xi, a_\mu, b_\mu, c_\mu, a_\lambda, b_\lambda, c_\lambda)$. Durch Lösen eines 6×6 Gleichungssystems kann man sich nun auf effiziente Weise die 6 lokalen Basisfunktionen berechnen, welche analog zum parametrischen Fall durch Vorgabe der Funktionale $\{F_T^{(a/b)}(\cdot), \Gamma \subset \partial \mathbf{T}_h\}$ eindeutig bestimmt sind. Wir erhalten damit folgendes Lemma:

Lemma 1.5. *Für den Interpolationsoperator $i_h = i_h^{(a,b)}$ gilt:*

$$\|v - i_h v\|_0 + h \|v - i_h v\|_h \leq ch^2 \|v\|_2, \quad \forall v \in H_0^1(\Omega) \cap H^2(\Omega), \quad (1.25)$$

ohne Abhängigkeit vom Gitterstörungsparameter σ_h .

Beweis.

Da $\hat{Q}_1(T)$ durch obige Konstruktion vollständig ist, d.h. es sind alle linearen Polynome enthalten, kann man das Bramble–Hilbert–Lemma direkt ohne Transformation auf das Referenzelement anwenden. Damit läßt sich die Theorie von Dupont/Scott in [17] auf unseren Fall verallgemeinern. \square

Wir wollen nun, wie eingangs erwähnt, noch auf den Fall großer *aspect ratios* eingehen, wie in den Abbildungen 1.6 und 1.7 beschrieben. Der Beweis für den 2D Fall findet sich in [3].

Satz 1.6. *Die nichtparametrische Version des nichtkonformen rotiert trilinearen Stokes–Elementes ist auf beliebigen Tensorprodukt Gittern stabil, d.h., es gilt:*

$$\sup_{\mathbf{v}_h \in \mathbf{H}_h} \frac{b_h(p_h, \nabla \cdot \mathbf{v}_h)}{\|\nabla \mathbf{v}_h\|_h} \geq \tilde{\gamma} \|p_h\|, \quad \forall p_h \in L_h.$$

mit einer Konstanten $\tilde{\gamma}$ unabhängig von der Gitterweite und des *aspect ratios*.

Beweis.

Für jedes gegebene $p_h \in L_h$ existiert ein $v \in H_0^1$, so daß (siehe [20])

$$(p_h, \nabla \cdot v) \geq \tilde{\gamma} \|p_h\| \|\nabla v\|$$

Dann erfüllt die diskrete Funktion $v_h \in H_h$, die definiert ist durch

$$\int_{\Gamma} v_h ds = \int_{\Gamma} v ds \quad \forall \Gamma \in \Gamma_{ij}$$

folgende Abschätzung: $(p_h, \nabla \cdot v_h) = (p_h, \nabla \cdot v) \geq \tilde{\gamma} \|p_h\| \|\nabla v\|$.

Damit bleibt die H^1 -Stabilität des Interpolationsprozesses zu zeigen, sei dazu

$$\begin{aligned} \int_T \nabla v_h \nabla v_h dx &= - \int_T \Delta v_h v_h dx + \int_{\Gamma} v_h \nabla v_h \cdot n ds \\ &= - \int_T \Delta v_h v dx + \int_{\Gamma} v \nabla v_h \cdot n ds \\ &= \int_T \nabla v \nabla v_h dx. \end{aligned}$$

Hier haben wir die Eigenschaft des nichtparametrischen Ansatzes ausgenutzt, daß Δv_h auf dem Element T verschwindet, was bei dem parametrischen Fall im allgemeinen nicht erfüllt ist. Es bleibt zu zeigen, daß $\nabla v_h \cdot n$ konstant auf den Flächen Γ ist. Dies kann man sich in diesem Fall klar machen durch folgende Überlegung:

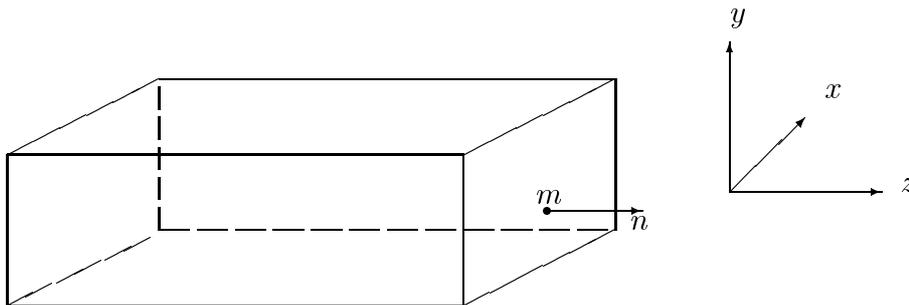


Abbildung 1.9: Normale einer Fläche eines Elementes

Die Normalenableitung ist in diesem Fall eine Linearkombination der z -Ableitungen der Ansatzfunktionen, also für $(1, ax, by, cz, d(x^2 - y^2), e(y^2 - z^2))$, bleibt: $c - 2ez$ als Normalenableitung, da z aber auf dieser Fläche konstant ist, ist auch die Ableitung konstant und wir können den Beweis mit der Cauchy-Schwarzschen Ungleichung ab-

schließen. □

Bemerkung 1.2. *Wesentlich in obigem Beweis ist die Tatsache, daß Δv_h auf dem Element T verschwindet. Bei einem allgemeinen Gitter erfordert diese Eigenschaft einige Modifikationen. Zum besseren Verständnis demonstrieren wir dies am analogen 2D Fall.*

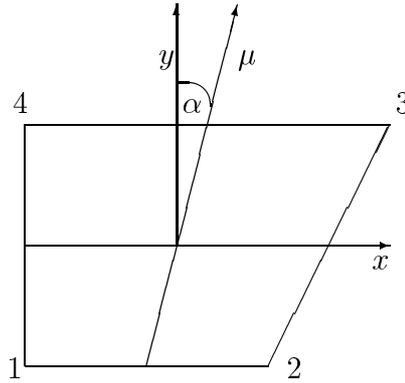


Abbildung 1.10: Verzerrtes Element in 2D

Unsere Basisfunktionen des nichtparametrischen Ansatzes sehen hier aus wie folgt:

$$\hat{Q}_1(T) := \text{span}\langle 1, \xi, \mu, \xi^2 - \mu^2 \rangle = \text{span}\langle 1, x, \mu, x^2 - \mu^2 \rangle =$$

$$= \text{span}\langle 1, x, x \sin \alpha + y \cos \alpha, x^2 - (x \sin \alpha + y \cos \alpha)^2 \rangle, \text{ mit } \mu = (x \sin \alpha + y \cos \alpha).$$

Die letzte Funktion läßt sich umformen zu:

$$x^2 - (x \sin \alpha + y \cos \alpha)^2 = x^2 \cos^2 \alpha - xy \sin 2\alpha - y^2 \cos^2 \alpha,$$

darauf den Laplace Operator angewandt ergibt wieder die Behauptung.

Da sich bei der Fehlerabschätzung für den konstanten Interpolationsoperator $j_h : L_0^2 \rightarrow L_h$ nichts ändert und auch die Stabilitätsbedingung gezeigt ist, läßt sich nun Satz 1.4 neu formulieren:

Satz 1.7. Für die nichtparametrischen Elemente $\mathbf{H}_h = \mathbf{H}_h^{(b)}$, und wenn die Quantität σ_h hinreichend klein ist auch für $\mathbf{H}_h = \mathbf{H}_h^{(a)}$, hat das diskrete Stokes–Problem (W) eine eindeutige Lösung $\{u_h, p_h\} \in \mathbf{H}_h^{(a/b)} \times L_h$, und es gilt:

$$\|u - u_h\|_h + \|p - p_h\| \leq ch\{\|u\|_2 + \|p\|_1\},$$

$$\|u - u_h\| + \|p - p_h\|_{-1} \leq ch^2\{\|u\|_2 + \|p\|_1\}.$$

Beweis.

Der Beweis bei [39] für den parametrischen Fall läßt sich für diesen Satz vollständig übertragen, da die einzige Änderung in der Fehlerabschätzung für den Interpolationsoperator $i_h = i_h^{(a,b)}$ (vgl. Lemma 1.2 und Satz 1.4) liegt. \square

Es bleibt noch, das praktische Problem der numerischen Integration zu besprechen. Wir verwenden hier zwei Möglichkeiten. Die erste arbeitet wieder **mit** Transformation, d.h. es werden nur die Quadraturpunkte auf das Referenzelement transformiert, dort wird die Integration durchgeführt, diese Möglichkeit verwenden wir auch bei den Testrechnungen, siehe Bemerkung 1.1. Die zweite Möglichkeit ist die, daß auch die numerische Integration auf jedem Element ausgeführt wird, was aber zu technischen Schwierigkeiten führt, da es keine Integrationsformeln für beliebige Hexaeder gibt. Daher muß jedes Element erst in fünf Tetraeder zerlegt werden (siehe Abbildung 1.11), auf jedem Tetraeder kann man dann numerisch integrieren.

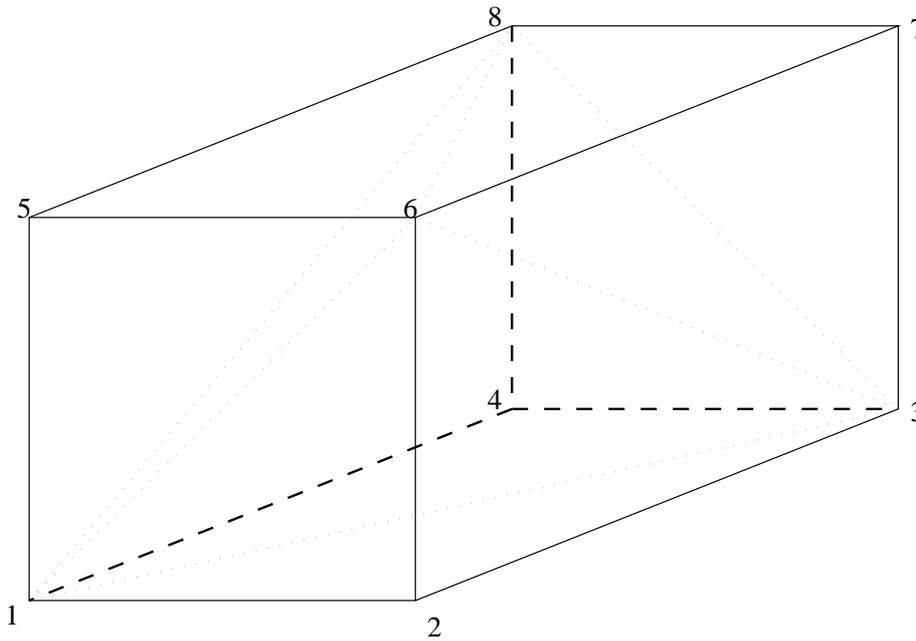


Abbildung 1.11: Zerlegung in Tetraeder

Die Zerlegung ist also wie folgt:

Tetraeder 1: Ecken 1, 3, 4, 8;

Tetraeder 2: Ecken 1, 2, 3, 6;

Tetraeder 3: Ecken 1, 5, 6, 8;

Tetraeder 4: Ecken 3, 6, 7, 8;

Tetraeder 5: Ecken 1, 3, 6, 8.

Bemerkung 1.3. Da wir im Mehrgitter für das Stokes–Problem (siehe nächstes Kapitel) an verschiedenen Stellen (z.B. Prolongation/ Restriktion) das Volumen der Hexaeder berechnen müssen, verwenden wir auch dafür obige Zerlegung. Zu beachten ist dabei aber, daß für deformierte Elemente die Zerlegung, bzw. das Volumen mit einem Fehler der Größe h^3 behaftet ist. Zudem ist die Zerlegung nicht eindeutig damit ebenso wenig die Lage der Integrationspunkte.

Die Gaußschen Integrationskonstanten kann man finden z.B. in ([18]). Der Vollständigkeit halber geben wir sie hier kurz an:

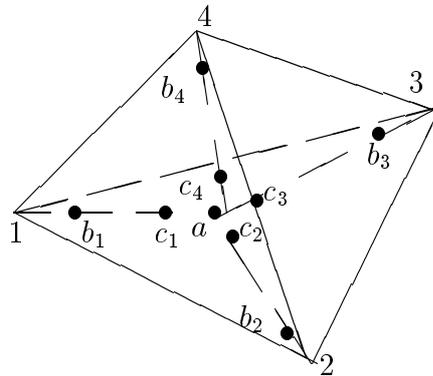


Abbildung 1.12: Gaußsche Integrationspunkte für Tetraeder

Wir haben dann:

$$I = \int \int \int f(x, y, z) dx dy dz = \int \int \int \theta(\xi, \mu, \lambda) |J| d\xi d\mu d\lambda =$$

$$V \sum_{j=1}^n w_j \theta(\xi_1, \xi_2, \xi_3, \xi_4)_j$$

Es ergibt sich mit obiger Darstellung die folgende Tabelle (mit den vier Koordinaten $\xi_j = L_j$, $j = 1, \dots, 4$; und $\alpha = 0.58541020$, $\beta = 0.13819660$):

Tabelle 1.1: Gaußsche Integrationskonstanten

n	Ordnung	Punkte	L_1	L_2	L_3	L_4	w_j
1	Linear, $O(h^2)$	a	0.25	0.25	0.25	0.25	1
4	Quadr. $O(h^3)$	b_1	α	β	β	β	0.25
		b_2	β	α	β	β	0.25
		b_3	β	β	α	β	0.25
		b_4	β	β	β	α	0.25
5	Kubi. $O(h^4)$	a	1/4	1/4	1/4	1/4	-0.8
		c_1	1/3	1/6	1/6	1/6	0.45
		c_2	1/6	1/3	1/6	1/6	0.45
		c_3	1/6	1/6	1/3	1/6	0.45
		c_4	1/6	1/6	1/6	1/3	0.45

Bemerkung 1.4. Die kubische Integrationsformel zeigt numerische Instabilitäten, was an dem negativen Gewicht liegt, und ist daher in der Praxis nur bedingt einsetzbar.

Der zusätzliche Aufwand für diese nichtparametrische Version liegt also im wesentlichen an dem Lösen der 6×6 Gleichungssysteme für jedes Element, sowie an der numerischen Integration, sofern man sie auch auf den Elementen durchführt. Unsere numerischen Ergebnisse im nächsten Abschnitt zeigen aber, daß dieser Aufwand sehr wohl gerechtfertigt ist und zu vollen Fehlerordnungen führt auch bei Störungen des Gitters.

1.5. Numerische Ergebnisse für das Stokes–Problem

In diesem Abschnitt präsentieren wir die Ergebnisse für die in den Abschnitten 1.3 und 1.4 analysierten 3D Stokes–Elemente. Um die Erhaltung der Fehlerordnungen zu untersuchen nehmen wir ein übliches Testproblem auf dem Einheitskubus, $\Omega = (0, 1) \times (0, 1) \times (0, 1)$, mit der exakten Lösung:

$$u_1(x_1, x_2, x_3) = -256x_1^2(x_1 - 1)^2x_2(x_2 - 1)(2x_2 - 1)x_3(x_3 - 1)(2x_3 - 1)$$

$$u_2(x_1, x_2, x_3) = 128x_2^2(x_2 - 1)^2x_1(x_1 - 1)(2x_1 - 1)x_3(x_3 - 1)(2x_3 - 1)$$

$$u_3(x_1, x_2, x_3) = 128x_3^2(x_3 - 1)^2x_1(x_1 - 1)(2x_1 - 1)x_2(x_2 - 1)(2x_2 - 1)$$

$$p(x_1, x_2, x_3) = 150(x_1 - 0.5)(x_2 - 0.5)(x_3 - 0.5).$$

Wir beginnen nun mit den Ergebnissen für ein ungestörtes Grobgitter für die parametrischen Elemente (punkt- bzw. integralorientiert), sowie der nichtparametrischen Pendants auf gestörten Gittern. In den folgenden Tabellen stellen wir jeweils den L^2 - und H^1 -Fehler für die Geschwindigkeiten, den L^2 -Fehler des Drucks, sowie die Konvergenzrate des verwendeten Mehrgitterverfahrens (*cvg*) dar. Wichtig zu erwähnen ist noch, daß wir für die parametrischen Rechnungen in allen Fällen eine $3 \times 3 \times 3$ Gauß Quadraturformel verwenden, im Falle der nichtparametrischen Elemente die Quadratur, wie in Abschnitt 1.4 beschrieben, durchführen.

Tabelle 1.2: Ergebnisse für die parametrischen Elemente

parametrisch, punktorientiert							
h	cvg	$L^2(u)$	Ord.	$H^1(u)$	Ord.	$L^2(p)$	Ord.
1/4	0.02	1.2170		1.678		0.6200	
1/8	0.06	0.5780	1.07	1.251	0.42	0.3520	0.81
1/16	0.13	0.2030	1.51	0.753	0.73	0.1590	1.15
1/32	0.22	0.0580	1.81	0.405	0.89	0.0662	1.27
1/64	0.20	0.0152	1.93	0.207	0.96	0.0294	1.17
parametrisch, integralorientiert							
h	cvg	$L^2(u)$	Ord.	$H^1(u)$	Ord.	$L^2(p)$	Ord.
1/4	0.02	0.9410		1.470		0.551	
1/8	0.01	0.3480	1.44	0.972	0.59	0.276	0.99
1/16	0.17	0.1020	1.77	0.543	0.84	0.124	1.15
1/32	0.19	0.0273	1.90	0.282	0.94	0.057	1.12
1/64	0.18	0.0069	1.98	0.143	0.98	0.027	1.08

Wir sehen an diesem einfachen Problem bereits, daß wir für alle verwendeten Elemente die vollen Fehlerordnungen erhalten, also $O(h^2)$ für den L^2 -Fehler der Geschwindigkeiten, sowie $O(h)$ für den H^1 -Fehler der Geschwindigkeiten und den L^2 -Fehler des Drucks. Wir werden die beiden letztgenannten Fehler aber noch verbessern können durch Interpolationstechniken, mit denen Superkonvergenzaussagen möglich sind (siehe Ende des Abschnittes in 1.5.1 und 1.5.2). Wir sehen hier aber bereits schon, daß das integralmittelorientierte Element absolut gesehen fast um den Faktor $1/2$ kleinere Fehler liefert, als das punktorientierte. Die minimalen Unterschiede zwischen den Zahlen der obigen Tabelle stammen von den unterschiedlichen Quadraturformeln, wie bereits in Abschnitt 1.4 erwähnt, haben wir für Tensorproduktgitter die Option, die Integration auch auf dem Element auszuführen, was wir in obigem Fall getan haben, später bei gestörten Gittern, arbeiten wir wieder mit Transformation der Quadraturpunkte auf das Referenzelement.

Der nächste Test, den wir jetzt durchführen, besteht darin, das gleiche Problem auf gestörten Gittern zu behandeln. Da wir bereits gesehen haben, daß bei sowohl dem parametrischem, als auch dem nichtparametrischem Fall, das integralorientierte Element weit besser abschneidet, beschränken wir uns im folgenden auf einen Vergleich der beiden integralmittelorientierten Elemente. Dazu stören wir jeweils das Rechen-

Tabelle 1.3: Ergebnisse für die nichtparametrischen Elemente

nichtparametrisch, punktorientiert							
h	cvg	$L^2(u)$	Ord.	$H^1(u)$	Ord.	$L^2(p)$	Ord.
1/4	0.02	1.2170		1.676		0.6200	
1/8	0.06	0.5870	1.05	1.251	0.42	0.3520	0.81
1/16	0.13	0.2070	1.50	0.753	0.73	0.1590	1.15
1/32	0.22	0.0590	1.83	0.405	0.89	0.0662	1.27
1/64	0.22	0.0155	1.93	0.207	0.96	0.0294	1.17
nichtparametrisch, integralorientiert							
h	cvg	$L^2(u)$	Ord.	$H^1(u)$	Ord.	$L^2(p)$	Ord.
1/4	0.02	0.9520		1.690		0.551	
1/8	0.01	0.3580	1.41	0.972	0.59	0.276	0.99
1/16	0.18	0.1060	1.76	0.543	0.84	0.124	1.15
1/32	0.20	0.0285	1.90	0.282	0.94	0.057	1.12
1/64	0.20	0.0072	1.98	0.143	0.98	0.027	1.08

gitter um 20%, was bedeutet, das jeweils nur das feinste Gitter der Gitterhierarchie, die aus einem regelmäßigen Grobgitter durch einen regelmäßigen Verfeinerungsprozeß entstanden ist, gestört wird. Wir verfeinern also nicht ein gestörtes Gitter weiter und stören dann noch einmal. Somit bekommen wir folgende Resultate:

Wir sehen an diesem Beispiel, daß die Fehler für die Geschwindigkeiten im parametrischen Fall für $h \rightarrow 0$ stehenbleiben, wogegen beim nichtparametrischen Element die Fehler fast mit voller Ordnung heruntergehen. Das “fast“ rührt von der Tatsache, daß wir das jeweils feinste Gitter zufällig stören, so daß man von Level zu Level nicht unbedingt eine Halbierung der Gitterweite hat.

Der zusätzliche Aufwand, der betrieben werden muß für das nichtparametrische Element (v.a. numerische Integration) wird durch dieses akademische Beispiel natürlich noch nicht gerechtfertigt. Wie bereits in Abschnitt 1.2 erwähnt, werden die Gitter durch unseren Verfeinerungsalgorithmus immer “besser“, was heißt, selbst bei einem Grobgitter mit schlechten Winkleigenschaften entstehen nach mehreren Verfeinerungen immer Tensorprodukt ähnlicher werdende Feingitter, bei denen wir auch für das parametrische Element volle Ordnung erhalten (dann sogar etwas mehr als $O(h^2)$, bzw. $O(h)$, da das jeweils verfeinerte Gitter näher am optimalen Fehler ist).

parametrisch, integralorientiert							
h	cvg	$L^2(u)$	Ord.	$H^1(u)$	Ord.	$L^2(p)$	Ord.
1/4	0.74	1.170		1.693		0.642	
1/8	0.86	0.505	1.21	1.172	0.53	0.335	0.91
1/16	0.68	0.172	1.55	0.697	0.75	0.150	1.16
1/32	0.43	0.088	0.96	0.436	0.67	0.068	1.14
1/64	0.22	0.076	0.20	0.325	0.42	0.033	1.03
nichtparametrisch, integralorientiert							
h	cvg	$L^2(u)$	Ord.	$H^1(u)$	Ord.	$L^2(p)$	Ord.
1/4	0.52	1.649		1.919		0.610	
1/8	0.85	0.608	1.44	1.232	0.63	0.319	0.93
1/16	0.62	0.182	1.74	0.698	0.82	0.139	1.20
1/32	0.45	0.052	1.81	0.390	0.84	0.063	1.14
1/64	0.22	0.016	1.63	0.221	0.82	0.033	0.92

Tabelle 1.4: Ergebnisse für 20% Gitterstörung

Da wir aber durch unsere Mehrgitterstruktur nur global verfeinern können, sind wir bei vielen Problemen gezwungen, das Grobgitter so anzupassen, daß in Bereichen des Rechengebietes, in denen “wenig“ passiert (z.B. Vorlaufbereich bei Umströmung eines Hindernisses), relativ grob diskretisiert wird. Dadurch handelt man sich dann aber Elemente ein, die einen relativ hohen *aspect ratio* –d.h. Verhältnis kürzeste Kante zu längster Kante– haben. Dieser *aspect ratio* bleibt aber im Gegensatz zu den oben besprochenen “besser“ werdenden Winkeln erhalten. Daher führen wir als nächsten Test Rechnungen auf Gittern mit einem *aspect ratio* von 8, bzw. 16 durch. In der nächsten Tabelle geben wir nur die Ergebnisse des nichtparametrischen Elementes an, da das Mehrgitterverfahren für die parametrischen Elemente auf diesen Gittern nur noch schlecht, bzw. gar nicht mehr konvergierte. Die Konvergenzraten beziehen sich in diesem Fall auf einen Mehrgitteralgorithmus mit 8 Vor- und Nachglättungsschritten mit dem Vanka–Glätter (siehe nächstes Kapitel).

Tabelle 1.5: Ergebnisse für hohen aspect ratio

aspect ratio 8							
h	cvg	$L^2(u)$	Ord.	$H^1(u)$	Ord.	$L^2(p)$	Ord.
1/2	0.17	1.971		2.352		0.827	
1/4	0.15	1.049	0.91	1.737	0.43	0.465	0.82
1/8	0.34	0.341	1.62	1.011	0.78	0.220	1.08
1/16	0.44	0.095	1.84	0.537	0.91	0.099	1.15
1/32	0.41	0.024	1.94	0.238	1.17	0.046	1.10
aspect ratio 16							
h	cvg	$L^2(u)$	Ord.	$H^1(u)$	Ord.	$L^2(p)$	Ord.
1/2	0.52	2.033		2.390		0.824	
1/4	0.57	1.063	0.93	1.752	0.44	0.464	0.82
1/8	0.54	0.344	1.63	1.017	0.78	0.220	1.08
1/16	0.52	0.096	1.84	0.540	0.91	0.099	1.10

Damit beenden wir diese Beispiele auf dem Einheitskubus mit exakter Lösung und wenden uns einem Problem zu, das wir bereits in Abschnitt 1.4 angesprochen haben, dem sogenannten ‘‘puff–puff‘‘ Problem. Eine ausführliche Beschreibung dieses Problems findet sich in Abschnitt 5.3. Hier haben wir das Problem, das wir einen krummlinigen Körper, in diesem Fall eine Halbkugel, diskretisieren wollen, was dadurch geschieht, daß wir ausgehend von einem Grobgitter, daß wie üblich in Mehrgitterkontext relativ wenig Zellen hat und damit auch noch wenig Ähnlichkeit mit einer Kugel, Feingitter erzeugen, die dann an die Kugeloberfläche angepaßt werden (siehe Abbildung 1.5). Dadurch handeln wir uns aber Gitter mit starken Deformationen ein, bei denen wir Schwierigkeiten haben eine stationäre Stokes–Lösung zu erhalten (v.a. bei höheren Reynoldszahlen), die wir aber später gerne als Startlösung für den instationären Navier–Stokes Löser hätten.

Das Problem stellt sich nun wie folgt (vgl. [27]): Wir geben an dem Rand des Quaders einen Druckwert von Null vor, am Rand der Halbkugel einen negativen Wert, die Geschwindigkeiten stellen sich dann von hohem Druck nach niederem Druck ein. Wir lösen nun das Stokes–Problem immer auf einem Level (11776 Elemente) mit verschiedenen Viskositäten (ν) und vergleichen die Konvergenzraten des Mehrgitters (mit 8 Vor– und Nachglättungsschritten).

Tabelle 1.6: Konvergenzraten für “puff–puff“ Stokes–Problem

	11776 Elemente					
ν	1	1/10	1/100	1/250	1/500	1/1000
<i>para.</i>	0.47	0.48	0.53	0.58	0.62	0.89
<i>nichtpara.</i>	0.055	0.079	0.13	0.17	0.18	0.21

Bei diesem praktisch relevanten Problem sehen wir die Robustheit der nichtparametrischen Version sehr deutlich, während die Konvergenzraten für die parametrische Version gegen 1 gehen, bleiben sie im anderen Fall nahezu konstant auf niedrigem Niveau. Damit ist auch der zusätzliche Aufwand gerechtfertigt, selbst für stationäre Probleme, da man im nichtparametrischen Fall mit erheblich weniger Glättungsschritten auskommt als bei der parametrischen Version.

1.5.1. Superkonvergenz für den L^2 –Fehler des Drucks:

Wie bereits erwähnt, können wir für den L^2 –Fehler des Drucks Superkonvergenzaussagen herleiten. Wir haben dabei drei Möglichkeiten:

1. Wir benutzen eine 1–Punkt Gauß Quadraturformel für die Fehlerauswertung und erhalten,

$$\|p_h - j_h p\|_2 \leq ch^2 .$$

2. Der Druck der konstant auf jedem Element ist, wird auf die Knoten interpoliert, jeder Knoten ist Ecke von 8 Elementen und bekommt gewichtet mit dem Volumen dieser Elemente ein Mittel der Druckwerte. Dann führen wir auf dem konformen linearen Element die Fehlerauswertung durch. Wir haben damit also,

$$\|p - I_h p_h\|_2 \leq \|p - I_h p\|_2 + \|I_h p - I_h p_h\|_2 .$$

Der erste Term auf der rechten Seite ist von der Ordnung h^2 , der zweite läßt sich beschränken durch $\|j_h p - p_h\|_2 \leq ch^2$.

3. Der Druck wird auf das nichtkonforme trilineare Element interpoliert, diesmal liegt jede Flächenmitte auf 2 Elementen und bekommt wie oben ein Mittel der Druckwerte dieser beiden Elemente.

Bemerkung 1.5. Die theoretischen Nachweise dieser Aussagen und der noch folgenden für den H^1 -Fehler der Geschwindigkeiten müssen als offenes Problem angesehen werden.

Wir erhalten folgende Ergebnisse auf dem Einheitskubus mit regelmässiger Verfeinerung:

Tabelle 1.7: L^2 -Fehler des Drucks

h	regelmässiges Gitter					
	wie 1.	Ord.	wie 2.	Ord.	wie 3.	Ord.
1/4	0.3943		0.6367		0.5008	
1/8	0.1782	1.14	0.2856	1.16	0.2225	1.17
1/16	0.0631	1.50	0.1009	1.51	0.0784	1.50
1/32	0.0194	1.72	0.0312	1.68	0.0244	1.68
1/64	0.0055	1.82	0.0093	1.75	0.0072	1.76

Wir sehen bei allen drei Möglichkeiten der Druckauswertung, daß der Fehler asymptotisch gegen $O(h^2)$ geht, wobei wie zu erwarten, die erste die absolut gesehen besten Fehler liefert. Trotzdem haben die beiden anderen Verfahren ihre Berechtigung, da man bei verschiedenen ingenieurtechnischen Anwendungen an den Druckwerten direkt auf dem Rand des Gebietes interessiert ist und man daher gerne auch den Druck linear behandelt, was wie unsere Rechnungen zeigen nicht notwendig ist, da man durch einfache Interpolationen bereits eine optimale Fehlerordnung erreichen kann. Ähnliche Techniken verwenden wir übrigens auch für das Mehrgitterverfahren. Wir interpolieren den konstanten Druck auf die Ecken, bzw. Flächenmitten, und führen dann Prolongation, bzw. Restriktion durch. Durch diese Technik lassen sich die Konvergenzraten des Mehrgitters deutlich verbessern.

Um die Stabilität obiger Interpolationen zu testen, führen wir die gleiche Rechnung auf gestörten Gittern durch. Es ist jeweils das feinste Gitter um 20 % gestört, wir erhalten dann:

Tabelle 1.8: L^2 –Fehler des Drucks

h	20 % Gitterstörung					
	wie 1.	Ord.	wie 2.	Ord.	wie 3.	Ord.
1/4	0.4203		0.6670		0.5370	
1/8	0.2042	1.04	0.3098	1.09	0.2518	1.10
1/16	0.0723	1.50	0.1112	1.49	0.0941	1.42
1/32	0.0244	1.57	0.0376	1.56	0.0323	1.54
1/64	0.0076	1.68	0.0112	1.76	0.0102	1.66

Wir sehen wieder wie oben, daß wir fast volle $O(h^2)$ Ordnung erhalten, wobei die Fehler absolut gesehen nur minimal schlechter werden gegenüber dem ungestörten Fall.

1.5.2. Superkonvergenz für den H^1 –Fehler der Geschwindigkeiten:

Um die Ergebnisse dieses Abschnittes abzuschließen, beschäftigen wir uns jetzt noch mit Superkonvergenzaussagen für den H^1 –Fehler der Geschwindigkeiten. Wir lehnen unsere Aussagen dabei an die Arbeit von Lin ([32]) an, der ähnliche Superkonvergenzaussagen für den konformen Fall untersucht hat. Die Schwierigkeit im nichtkonformen Fall liegt v.a. in der Behandlung des Interpolationsoperators, der die Geschwindigkeiten, die auf den 6 Flächenmitten jedes Elementes liegen, auf die Ecken des konformen triquadratischen Elementes interpoliert. Um die Vorgehensweise besser illustrieren zu können, beginnen wir mit dem 2D–Fall (siehe [48]). Dort liegen die Freiheitsgrade des rotiert bilinearen Elementes auf den Seitenmitten. Wir haben also folgende Situation:

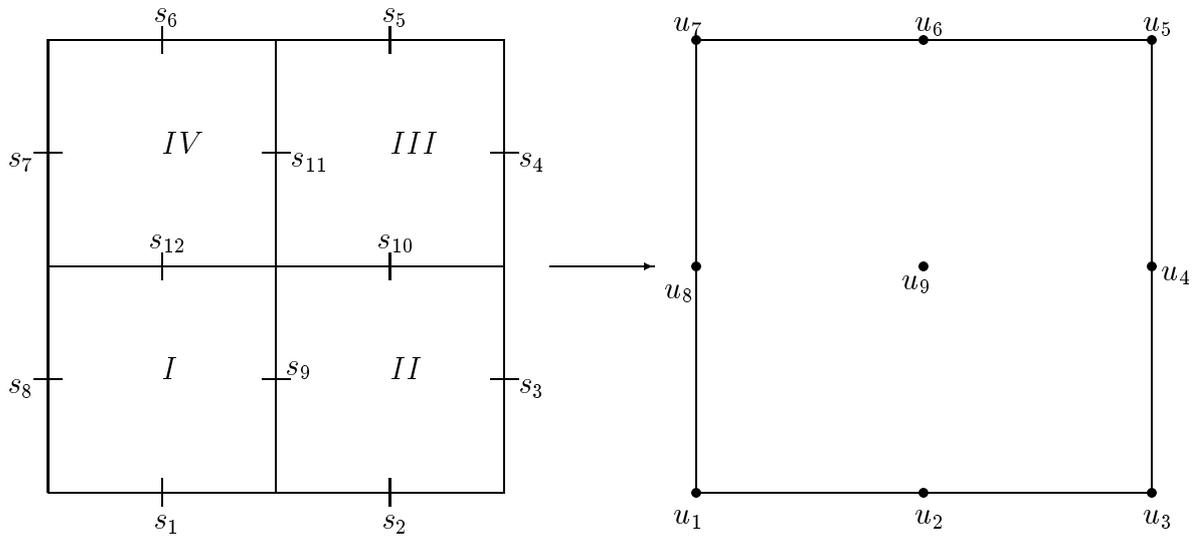


Abbildung 1.13: Interpolation rotiert bilinear auf biquadratisch

Unser Vorgehen sieht nun folgendermaßen aus, wir rechnen auf dem Feingitter mit den Elementen *I*, *II*, *III* und *IV*, die Freiheitsgrade liegen damit auf den Seitenmitten s_1 – s_{12} . Nun interpolieren wir die Geschwindigkeitswerte dieser Seitenmitten auf die Knoten u_1 – u_9 des Makroelementes. Da im Inneren unseres Gitters stets die Situation vorliegt, daß ein Knoten auf vier Elementen liegt (wie im Bild oben u_9), geben wir für diesen Knoten die Interpolation an (am Rand sind entsprechend andere Gewichte zu verwenden):

$$u_9 = \frac{1}{4} \left(\frac{3}{4} (s_9 + s_{10} + s_{11} + s_{12}) - \frac{1}{4} (s_1 + s_2 + s_3 + s_4 + s_5 + s_6 + s_7 + s_8) \right)$$

Auf dem Makroelement rechts oben im Bild, läßt sich damit die H^1 -Fehlerauswertung biquadratisch durchführen. Der Unterschied liegt genau in der Analyse des Interpolationsoperators (vgl. [32]), der im konformen Fall natürlich trivial ist, da die Freiheitsgrade des bilinear konformen Elementes genau auf den Freiheitsgraden des biquadratisch konformen Elementes ein Level tiefer liegen. Damit ist im konformen Fall gar keine Interpolation notwendig.

Im 3D Fall stellt sich nun folgende Situation dar:

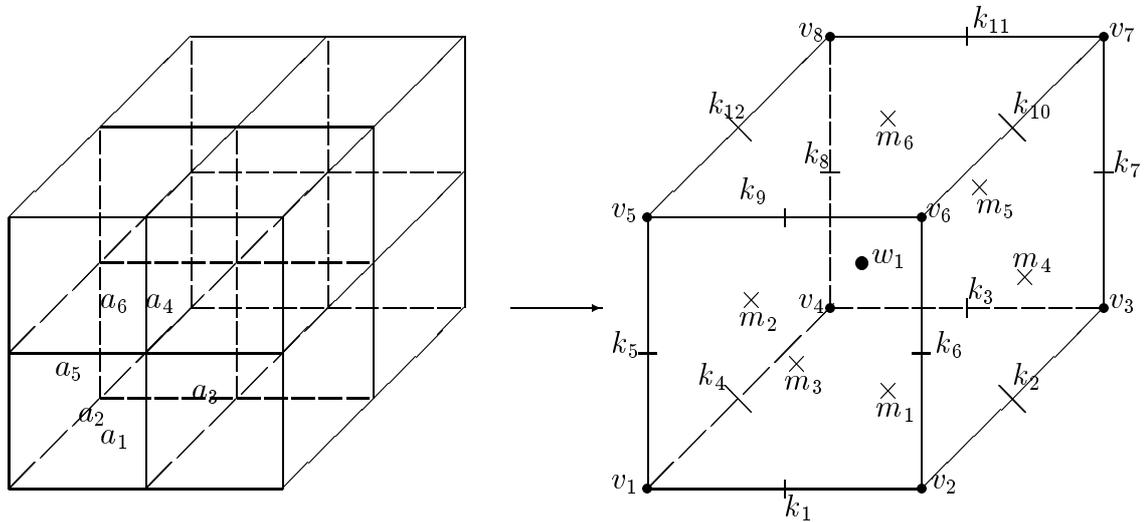


Abbildung 1.14: Interpolation rotiert trilinear auf triquadratisch

Wieder sollen die Geschwindigkeitswerte (insgesamt 36 Flächenmitten) des nichtkonformen trilinearen Elementes auf den 8 Mikroelementen (im Bild links) auf die Knoten des triquadratischen Makroelementes interpoliert werden. Im Gegensatz zum 2D Fall stellt sich hier aber das Problem, daß es bei dem konformen triquadratischen Element vier Kategorien von Knoten gibt:

Es sind die 8 Ecken (v_1 – v_8), die 6 Flächenmitten (m_1 – m_6), die 12 Kantenmitten (k_1 – k_{12}), sowie der Schwerpunkt (w_1), insgesamt 27 Freiheitsgrade. Diese 27 Knoten entsprechen aber genau wieder den Knoten (Ecken) des trilinearen konformen Elementes auf einem Level höher. Wir müssen mit unserer Interpolation die Geschwindigkeitswerte des nichtkonformen trilinearen Elementes, die auf den Flächenmitten liegen, auf die Ecken des gleichen Levels interpolieren, um dann wieder ein Level tiefer gehen zu können, um dort die Fehlerauswertung triquadratisch durchzuführen.

Die Interpolation sieht nun folgendermaßen aus. Wir geben wieder die Interpolation exemplarisch für einen inneren Punkt an, in diesem Fall für den Schwerpunkt w_1 , für Randpunkte müssen entsprechend andere Vorfaktoren gewählt werden. Der Anteil von

dem Mikroelement links unten in Abbildung 1.14 berechnet sich wie folgt:

$$w_1 = \frac{1}{8} \left(\frac{2}{3} (a_3 + a_4 + a_6) - \frac{1}{3} (a_1 + a_2 + a_5) \right)$$

Dieser Prozeß wird für den Punkt w_1 auf allen 8 Elementen durchgeführt.

Nun geben wir die numerischen Resultate an, zuerst für den 2D-Fall (die Rechnungen sind mit dem nichtparametrischen Element durchgeführt):

Tabelle 1.9: H^1 -Fehler der Geschwindigkeiten

h	regelmässiges Gitter			
	H^1 ,lin.	Ord.	H^1 ,quadr.	Ord.
1/4	5.88-1		1.92-1	
1/8	3.27-1	0.84	8.92-2	1.11
1/16	1.69-1	0.95	2.55-2	1.80
1/32	8.56-2	0.98	8.45-3	1.59
1/64	4.29-2	0.99	2.86-3	1.56
1/128	2.15-2	0.99	9.85-4	1.54
1/256	1.07-2	1.01	3.00-4	1.71

Auf gestörten Gittern (immer das feinste, vgl. Tabelle 1.4) bekommen wir folgende Resultate:

Tabelle 1.10: H^1 -Fehler der Geschwindigkeiten

h	20 % Gitterstörung			
	H^1 ,lin.	Ord.	H^1 ,quadr.	Ord.
1/4	6.19-1		2.47-1	
1/8	3.68-1	0.75	2.24-1	0.14
1/16	2.21-1	0.73	2.42-1	-0.11
1/32	1.24-1	0.83	2.57-1	-0.08
1/64	7.89-2	0.66	2.70-1	-0.07
1/128	4.29-2	0.89	2.20-1	0.02
1/256	2.40-2	0.80	2.11-1	0.06

Hier sehen wir die Theorie bestätigt (vgl. [32]), die für Superkonvergenz fordert, daß

das Gitter zumindest lokal regelmäßig ist. Dies ist in diesem Fall verletzt, da wir das feinste Gitter willkürlich um 20% stören. Führen wir die gleiche Rechnung auf einem Gitter durch, wo nur das Grobgitter gestört ist und danach wieder regelmäßig verfeinert wird, erhalten wir wieder die Ordnung h^2 für den H^1 –Fehler der Geschwindigkeiten. Dies werden wir aber für den 3D–Fall genauer untersuchen.

In den Ergebnissen für den 3D–Fall geben wir auch noch den L^2 –Fehler für die trilineare nichtkonforme Fehlerauswertung (nk), sowie für die trilineare konforme (ko), ebenso für den H^1 –Fehler, hier dann natürlich auch für die triquadratische (qu), an. Wie oben bereits geschildert, interpolieren wir die Geschwindigkeiten des nichtkonformen Elementes von den Flächenmitten auf die Knoten des konform trilinearen Elementes, um dann ein Level tiefer zu gehen. Die Interpolation können wir dann schon testen, wenn wir eine konform trilineare Fehlerauswertung vornehmen; kommt hier die richtige Ordnung heraus, befinden wir uns in der Situation wie bei ([32]). Das heißt auch, es reicht, die Interpolation nichtkonform–konform zu untersuchen, was wir als erstes in folgender Tabelle sehen bei einem Vergleich der L^2 –Fehler.

Tabelle 1.11: L^2 –Fehler der Geschwindigkeiten

	regelmässiges Gitter			
h	$L^2(u),nk$	Ord.	$L^2(u),ko$	Ord.
1/4	9.37-1		6.82-1	
1/8	3.48-1	1.43	3.22-1	1.08
1/16	1.02-1	1.77	9.90-2	1.70
1/32	2.73-2	1.90	2.66-2	1.91
1/64	6.90-3	1.98	6.40-3	2.05

Tabelle 1.12: H^1 –Fehler der Geschwindigkeiten

	regelmässiges Gitter					
h	$H^1(u),nk$	Ord.	$H^1(u),ko$	Ord.	$H^1(u),qu$	Ord.
1/4	1.46		8.48-1		5.02-1	
1/8	9.72-1	0.59	4.63-1	0.87	1.92-1	1.39
1/16	5.43-1	0.84	1.92-1	1.27	5.21-2	1.88
1/32	2.82-1	0.94	8.07-2	1.25	1.52-2	1.78
1/64	1.43-1	0.98	4.11-2	0.97	4.12-3	1.89

Wir sehen in den obigen Tabellen sehr schön, wie wir überall volle Fehlerordnungen erhalten. Außerdem erkennen wir die Richtigkeit unserer Interpolation, da die Zahlen für den L^2 -Fehler nahezu übereinstimmen. Interessant ist die Tatsache, daß der H^1 -Fehler, konform trilinear ausgewertet absolut gesehen erheblich bessere Ergebnisse liefert, als bei der nichtkonformen Auswertung.

Um nun noch zu zeigen, daß es für die Superkonvergenz ausreicht, lokal regelmäßige Gitter zu haben, verwenden wir für nächsten Test einen Einheitskubus, verfeinern ihn dreimal, stören dieses Gitter um 20%, und nehmen es dann als Grobgitter. Die weiteren Gitter die dann durch regelmäßige Verfeinerung entstehen sind nur noch lokal regulär. Wir bekommen so folgende Zahlen:

Tabelle 1.13: L^2 -Fehler der Geschwindigkeiten

h	Grogitter 20% gestört			
	$L^2(u),nk$	Ord.	$L^2(u),ko$	Ord.
1/8	4.33-1		3.82-1	
1/16	1.29-1	1.75	1.22-1	1.65
1/32	3.47-2	1.92	3.36-2	1.89
1/64	9.20-3	1.92	8.72-3	1.95

Tabelle 1.14: H^1 -Fehler der Geschwindigkeiten

h	Grogitter 20% gestört					
	$H^1(u),nk$	Ord.	$H^1(u),ko$	Ord.	$H^1(u),qu$	Ord.
1/8	1.08		5.86-1		2.73-1	
1/16	6.07-1	0.83	2.53-1	1.21	7.04-2	1.88
1/32	3.17-1	0.93	1.03-1	1.30	2.23-2	1.73
1/64	1.65-1	0.94	4.56-2	1.19	5.90-3	1.93

Auch hier sehen wir wieder die Theorie bestätigt. Die Fehlerordnungen sind sogar etwas besser als im ungestörten Fall, was wie schon erwähnt, daran liegt, daß die Gitter durch den regelmäßigen Verfeinerungsprozeß immer Parallelogramm ähnlicher werden.

Bemerkung 1.6. *Man kann mit Hilfe einfacher Extrapolationstechniken bereits auf niedrigen Levels absolut gesehen bessere Ergebnisse erzielen. Dies ist v.a. aus prakti-*

schem Interesse wichtig, da man die Visualisierung der Ergebnisse meist nicht auf dem feinsten Gitter durchführen will, bzw. kann, da dort die Datenmengen oft zu groß werden können. Auch kann man sich auf diese Weise gute Startlösungen für instationäre Rechnungen bereits auf niederen Leveln mit relativ geringem Aufwand verschaffen. Diese Startlösungen werden dann auf das feinste Level prolongiert. Wir zeigen dies exemplarisch für den L^2 - und L^∞ -Fehler des Drucks, wobei wir eine einfache Richardson Extrapolation durchführen (siehe z.B. [6]) :

$$\|p - \frac{I_h p_h - 4I_{h/2} p_{h/2}}{3}\|_{2(\infty)}$$

Wir erhalten so folgende Ergebnisse:

Tabelle 1.15: L^2 - und L^∞ -Fehler für den Druck

h	L^2	L^2 , extrap.	L^∞	L^∞ , extrap.
1/4	2.735	3.26-1	5.11	6.55-1
1/8	9.35-1	8.15-2	3.12	3.21-1
1/16	3.33-1	2.12-2	1.79	1.69-1
1/32	9.98-2	5.32-3	9.94-1	8.46-2
1/64	2.86-2	1.32-3	4.82-1	4.22-2
1/128	7.38-3	3.33-4	2.44-1	2.11-2

Man hat somit mit geringem Aufwand bereits beim L^2 -Fehler ein besseres Ergebnis für $h=1/16$ (extrapoliert mit der Lösung auf $h=1/32$), als für $h=1/64$. Beim L^∞ -Fehler ist der Unterschied noch größer, hier erreicht man schon für $h=1/16$ punktweise den gleichen Fehler wie sonst bei $h=1/128$. Da der Aufwand von Verfeinerung zu Verfeinerung um den Faktor 8 wächst, ist diese Extrapolation ein nützliches Hilfsmittel.

Damit schließen wir dieses Kapitel ab. Wir können als Fazit ziehen, daß die von uns verwendeten nichtkonformen trilinearen Finite-Elemente-Ansätze sowohl stabil gegen Gitterstörungen, wie hoher *aspect ratio* und Verzerrungen sind, und vor allem darüber hinaus mit dem nur unwesentlichen Mehraufwand der nichtparametrischen Versionen, volle Fehlerordnungen liefern. Im nächsten Kapitel werden wir einen Mehrgitteralgorithmus für das Stokes–Problem vorstellen und analysieren, bevor wir dann zu den stationären und instationären Navier–Stokes–Gleichungen kommen.

2. Der Mehrgitteralgorithmus für das Stokes Problem

Ziel dieses Kapitels ist es, ein Mehrgitterverfahren zur Lösung der Stokes–Gleichungen, basierend auf einer Diskretisierung mit den nichtkonformen 3D–Elementen aus dem letzten Kapitel einzuführen. Diesen Mehrgitteralgorithmus werden wir später in Kapitel 3 auch zur Lösung der linearisierten Navier–Stokes–Gleichungen einsetzen, bzw. auch zur Lösung oder als Vorkonditionierer der Teilprobleme im Projektionsverfahren in Kapitel 4. Der Algorithmus besteht im wesentlichen aus den gleichen Komponenten wie ein entsprechendes Standardverfahren zur Lösung von skalaren Poisson–Gleichungen: Glättung, Gittertransfer und Schrittweitenkontrolle der Korrektur. Ein Problem bei Mehrgitterverfahren für nichtkonforme Finite Elemente liegt hierbei im Nachweis der Approximationseigenschaft, daher müssen die verwendeten Gittertransferoperatoren gewissen Bedingungen genügen. Unsere Konstruktion lehnt sich dabei an die Arbeit von Brenner [9], Turek [48], Schreiber [45] und Morley [35] an.

Das weitaus größere Problem für unseren Ansatz ist aber der Nachweis der Glättungseigenschaft. Durch die Kopplung von Geschwindigkeiten und Druck entsteht ein Sattelpunktproblem, hier ist es sehr schwierig überhaupt einen Glätter zu definieren. Eine Möglichkeit stellt dabei der sogenannte *Vanka*–Glätter dar (siehe [54]), der aber aufgrund seiner relativ schlechten Effizienz von uns hauptsächlich im stationären Fall eingesetzt wird. Der theoretische Nachweis der Glättungseigenschaft muß als Problem offen bleiben. Eine andere Möglichkeit ergibt sich durch Benutzung von diskret divergenzfreien Ansatzfunktionen, man erhält dann eine Formulierung bei der der diskrete Druck eliminiert worden ist, so daß man ihn in einem postprocessing aus den bekannten Geschwindigkeiten berechnen kann. Diese Möglichkeit ist von Turek in [48] für den 2D–Fall realisiert worden, Langer in [31] gab eine 3D Formulierung an (siehe auch [26]), deren Implementierung aber für die Praxis zu aufwendig erscheint.

Dieses Kapitel ist nun so gegliedert, daß wir zunächst den Mehrgitteralgorithmus vor-

stellen und dabei die Besonderheiten des nichtkonformen Elements und des gekoppelten Ansatzes herausstellen, anschließend betrachten wir den Vanka–Glätter, wobei wir aber den Nachweis der Glättungseigenschaft als Problem offen lassen müssen. In Abschnitt 2.3 gehen wir eingehend auf die Gittertransferoperatoren ein, da deren Konstruktion quasi die Basis eines jeden Mehrgitterverfahrens für nichtkonforme Elemente darstellt. Mit Testrechnungen schließen wir dieses Kapitel ab.

2.1. Die Formulierung des Mehrgitterverfahrens

Sei $\{\mathbf{T}_h\}_h$ eine Familie von Triangulierungen, die gemäß dem *regulären* Verfeinerungsprozeß aus Abschnitt 1.2 generiert werden. Die Grundschnittweite bezeichnen wir mit h_1 und die Gitterweite auf Level k mit h_k . Die zugehörigen Gitter heißen dann \mathbf{T}_{h_k} bzw. kürzer \mathbf{T}_k und die Ansatzräume $\mathbf{H}_{h_k} \times \mathbf{L}_{h_k}$ bzw. $\mathbf{H}_k \times \mathbf{L}_k$. Entsprechende Notationen werden bei gitterabhängigen Konstrukten verwendet, wie beispielsweise Normen, Bilinearformen, Skalarprodukten, Transferoperatoren. Beschränken wir uns auf den Fall von zwei Gittern, so werden der Kürze wegen die Indizes h und $2h$ benutzt.

Das diskrete Stokes–Problem auf Level k , das nun mittels eines Mehrgitterverfahrens gelöst werden soll, lautet damit

Finde ein Paar $(\mathbf{u}_k, p_k) \in \mathbf{H}_k \times L_k$, so daß gilt

$$a_k(\mathbf{u}_k, \mathbf{v}_k) + b_k(p_k, v_k) + b_k(q_k, \mathbf{u}_k) = (\mathbf{f}, \mathbf{v}_k) \quad , \quad \forall \mathbf{v}_k \in \mathbf{H}_k, \forall q_k \in L_k. \quad (\text{V}_k)$$

In Matrix–Vektor Schreibweise schreibt sich Problem (V_k) dann als

$$\begin{aligned} A\mathbf{u} + Bp &= f, \\ B^T\mathbf{u} &= 0 \end{aligned} \tag{2.1}$$

d.h. wir haben als Systemmatrix eine indefinite Blockmatrix S ,

$$S = \begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix}. \quad (2.2)$$

Als nächstes definieren wir *Prolongationsoperatoren* $I_{k-1}^{k,v} : \mathbf{H}_{k-1} \rightarrow \mathbf{H}_k$ und $I_{k-1}^{k,p} : L_{k-1} \rightarrow L_k$ für Geschwindigkeit und Druck, bzw. die dazu adjungierten *Restriktionsoperatoren* $I_{k,v}^{k-1} : \mathbf{H}_k \rightarrow \mathbf{H}_{k-1}$ und $I_{k,p}^{k-1} : L_k \rightarrow L_{k-1}$ durch

$$(I_{k,v}^{k-1} \mathbf{w}_k, \mathbf{v}_{k-1})_{k-1} = (\mathbf{w}_k, I_{k-1}^{k,v} \mathbf{v}_{k-1})_k, \quad \forall \mathbf{v}_{k-1} \in \mathbf{H}_{k-1}, \forall \mathbf{w}_k \in \mathbf{H}_k. \quad (2.3)$$

und

$$(I_{k,p}^{k-1} p_k, q_{k-1})_{k-1} = (p_k, I_{k-1}^{k,p} q_{k-1})_k, \quad \forall q_{k-1} \in L_{k-1}, \forall p_k \in L_k. \quad (2.4)$$

Aus rein formalen Gründen verwenden wir hier nur *Prolongations-* und *Restriktionsoperatoren*, die nach dieser Definition adjungiert zueinander sind. Bei der praktischen Durchführung kann man natürlich auch mit verschiedenen Operatoren arbeiten.

Nach Einführung dieser Bezeichnungen können wir nun unseren k -Level Mehrgitteralgorithmus $MG(k, \cdot, \cdot)$ zur Lösung von Problem (V_k) formulieren.

Die k -Level Iteration $MG(k, (\mathbf{u}_k^0, p_k^0), \mathbf{g}_k)$

Die Parameter eines vollen Iterationsschrittes $MG(\cdot, \cdot, \cdot)$ sind die Levelnummer k , die Startnäherung \mathbf{u}_k^0 und die rechte Seite \mathbf{g}_k mit Koeffizienten

$$G_k^i = (\mathbf{f}, \mathbf{v}_k^{i,d}). \quad (2.5)$$

Wiederholtes Anwenden des Algorithmus liefert dann nach endlich vielen Schritten die Lösung (\mathbf{u}_k, p_k) des Problems (V_k) , d.h.

$$S_k \begin{bmatrix} \mathbf{u}_k \\ p_k \end{bmatrix} = g_k. \quad (2.6)$$

Sei also (\mathbf{u}_k^0, p_k^0) eine Startnäherung zur exakten diskreten Lösung (\mathbf{u}_k, p_k) auf Level k , dann arbeitet ein Schritt der k -Level Iteration $MG(k, (\mathbf{u}_k^0, p_k^0), \mathbf{g}_k)$ wie folgt:

Wenn $k = 1$, dann liefert $MG(1, (\mathbf{u}_1^0, p_1^0), \mathbf{g}_1)$ die exakte Lösung, d.h.

$$MG(1, (\mathbf{u}_1^0, p_1^0), \mathbf{g}_1) = S_1^{-1} \mathbf{g}_1 . \quad (2.7)$$

Für $k > 1$, zerfällt ein Iterationszyklus in die folgenden vier Teilschritte:

1) m -Vorglättungsschritte

Wende m Glättungsschritte auf (\mathbf{u}_k^0, p_k^0) an und erhalte (\mathbf{u}_k^m, p_k^m) .

2) Korrekturschritt

Berechne den *restringierten* Defekt \mathbf{g}_{k-1} durch

$$\mathbf{g}_{k-1} = I_k^{k-1}(\mathbf{g}_k - S_k(\mathbf{u}_k^m, p_k^m)) \quad (2.8)$$

mit I_k^{k-1} als Verkettung von $I_{k,v}^{k-1}$ und $I_{k,p}^{k-1}$ und bestimme rekursiv die Funktionen $(\mathbf{u}_{k-1}^i, p_{k-1}^i) \in \mathbf{H}_{k-1} \times L_{k-1}$ ($1 \leq i \leq q$, $q \geq 2$) durch

$$\mathbf{u}_{k-1}^i = MG(k-1, (\mathbf{u}_{k-1}^{i-1}, p_{k-1}^{i-1}, \mathbf{g}_{k-1}), 1 \leq i \leq q, \quad (2.9)$$

mit Startnäherung $(\mathbf{u}_{k-1}^0, p_{k-1}^0) = (\mathbf{0}, 0)$.

3) Schrittweitenkontrolle

Berechne \mathbf{u}_k^{m+1} nach der Vorschrift

$$\mathbf{u}_k^{m+1} = \mathbf{u}_k^m + \alpha_k I_{k-1}^{k,v} \mathbf{u}_{k-1}^q, \quad (2.10)$$

bzw. p_k^{m+1} durch

$$p_k^{m+1} = p_k^m + \alpha_k I_{k-1}^{k,p} p_{k-1}^q . \quad (2.11)$$

Dabei kann der Parameter α_k als fester Wert oder auch adaptiv so gewählt werden, daß der Fehler $\mathbf{u}_k^{m+1} - \mathbf{u}_k$ bzw. $p_k^{m+1} - p_k$ in einer geeigneten Norm $||| \cdot |||_t$ minimiert wird. Wählt man $t = 1$, also gerade die Energienorm, so läßt sich α_k bestimmen als

$$\alpha_k = \frac{(\mathbf{g}_k - S_k(\mathbf{u}_k^m, p_k^m), I_{k-1}^k(\mathbf{u}_{k-1}^q, p_{k-1}^q))_k}{((S_k I_{k-1}^k(\mathbf{u}_{k-1}^q, p_{k-1}^q), I_{k-1}^k(\mathbf{u}_{k-1}^q, p_{k-1}^q))_k)} . \quad (2.12)$$

4) m -Nachglättungsschritte

Analog zu Schritt 1) wende m Glättungsschritte auf $(\mathbf{u}_k^{m+1}, p_k^{m+1})$ an und erhalte $(\mathbf{u}_k^{2m+1}, p_k^{2m+1})$.

Ein solcher Iterationsschritt $MG(\cdot, \cdot, \cdot)$ liefert damit ausgehend von einer Startnäherung \mathbf{u}_k^0 die neue Approximierende $(\mathbf{u}_k^{2m+1}, p_k^{2m+1})$, d.h.

$$MG(k, (\mathbf{u}_k^0, p_k^0), \mathbf{g}_k) = (\mathbf{u}_k^{2m+1}, p_k^{2m+1}). \quad (2.13)$$

Aufbauend auf dieser k -Level Iteration kann nun ein *voller* Mehrgitteralgorithmus beschrieben werden, der optimale Effizienz hat, d.h., der die diskrete Lösung mit $O(n_k)$ arithmetischen Schritten liefert (siehe [1],[9],[22]).

Der volle Mehrgitteralgorithmus mit r Schritten

- 1) Löse auf dem größten Level exakt und erhalte $(\hat{\mathbf{u}}_1, \hat{p}_1)$.
- 2) Berechne rekursiv die Approximationen $(\hat{\mathbf{u}}_k, \hat{p}_k)$ zu (\mathbf{u}_k, p_k) ($k \geq 2$) durch

$$\text{a) } (\mathbf{u}_k^0, p_k^0) = I_{k-1}^k(\hat{\mathbf{u}}_{k-1}, \hat{p}_{k-1})$$

$$\text{b) } (\mathbf{u}_k^l, p_k^l) = MG(k, (\mathbf{u}_k^{l-1}, p_k^{l-1}), \mathbf{g}_k), l = 1 \dots, r$$

$$\text{c) } (\hat{\mathbf{u}}_k, \hat{p}_k) = (\mathbf{u}_k^r, p_k^r).$$

Zu diesen beiden vorgestellten Algorithmen müssen nun noch einige Bemerkungen gemacht werden.

Bemerkung 2.1. Die Zahl der Glättungsschritte muß beim Vor- und Nachglätten nicht übereinstimmen. Wir benutzen in der Praxis außerdem die Option, die Zahl der Glättungsschritte levelabhängig variabel vorzuschreiben. Wie im nächsten Abschnitt genauer ausgeführt verwenden wir sowohl zum Glätten als auch zum Lösen eine Art Block-Gauß-Seidel Verfahren für die elementweise entstehenden Schur Komplement

Gleichungen.

Bemerkung 2.2. *Wie schon vorhin bemerkt wurde, müssen in der Praxis Restriktionsoperator und Prolongationsoperator nicht unbedingt adjungiert zueinander sein. Vielmehr macht man auch bei der Lösung von anderen Problemen, wie beispielsweise Poisson- oder Plattengleichungen, die praktische Erfahrung, daß der Restriktionsoperator weniger genau als der Prolongationsoperator sein darf. Dies kann man sich damit erklären, daß der Restriktionsoperator auf einen schon geglätteten Defekt, der Prolongationsoperator dagegen auf eine noch raue Grobgitterlösung angewandt wird. In numerischen Testrechnungen verwenden wir zusätzlich zum V-Zyklus ($q = 1$, siehe Formel (2.9)) auch den F-Zyklus, der sich aber nicht durch die Wahlen $q = 1$ oder $q = 2$ beschreiben läßt. In Abbildung 2.1 sind die einzelnen Zyklusarten graphisch dargestellt.*

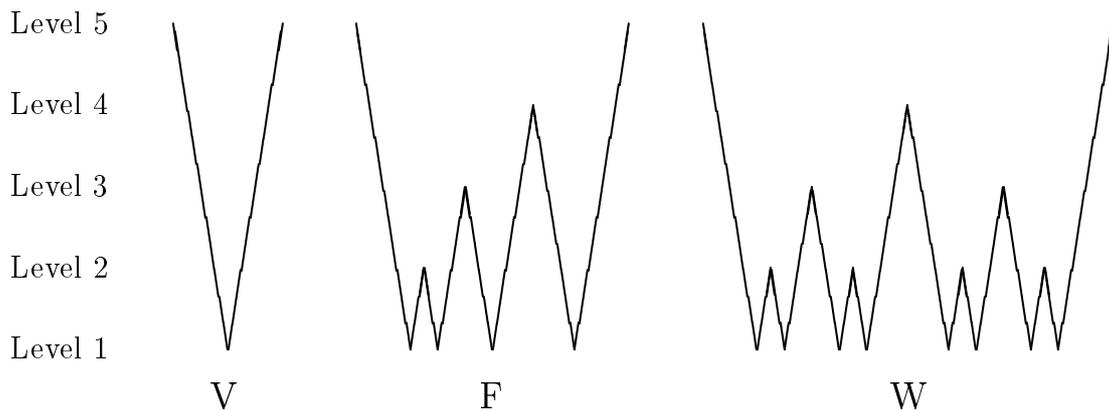


Abbildung 2.1: Verschiedene Zyklen

Bemerkung 2.3. *Wie üblich bei nichtkonformen Mehrgitterverfahren (vgl. z.B. [48] oder [45]) wirkt sich eine Schrittweitenkontrolle positiv auf die Konvergenzraten des Mehrgitters aus. Diese Technik ist im konformen Fall nicht nötig, da die diskreten Ansatzräume ineinander verschachtelt sind und die Lösungen der Korrekturgleichungen Bestapproximationen der Feingitterfehler sind (vergleiche dazu [7]). Daher liefert die feste Wahl $\alpha_k = 1$ die optimalen Ergebnisse. Dies ist aber nicht mehr der Fall bei Verwendung von nichtkonformen Ansatzfunktionen. Wie schon in [8] beschrieben wurde, liefert auch hier ein fester Wert $\alpha_k \neq 1$ oder ein adaptiv berechnetes α_k wesentlich bessere Konvergenzresultate*

Bemerkung 2.4. Da wir unsere Mehrgitterverfahren hauptsächlich als Vorkonditionierer in einer Fixpunktiteration zur Lösung der stationären und instationären Navier–Stokes–Gleichungen einsetzen, spielt der volle Mehrgitteralgorithmus nur eine untergeordnete Rolle. Wir starten stattdessen meistens auf dem feinsten Level mit einer gegebenen Startlösung zu einer kleineren Reynoldszahl oder im instationären Fall mit der Lösung des vorherigen Zeitschrittes.

2.2. Der Vanka–Glätter

Dieser von uns verwendete Glätter geht auf Vanka in [54] zurück. Grundsätzlich liegt diesem Schema, das von Vanka als **S**ymmetrical **C**oupled **G**auß–**S**eidel (**SCGS**) Schema bezeichnet wird, die Idee zugrunde, Geschwindigkeiten und Druck voll gekoppelt zu behandeln, und läßt sich als Erweiterung des traditionellen Gauß–Seidel Verfahrens für die Laplace–Gleichung ansehen.

In unserem Fall verwenden wir eine modifizierte Variante. Wie wir im letzten Abschnitt gesehen haben, besteht unser volles System aus Blöcken S_e (der Index e steht für elementweise):

$$S_e = \begin{bmatrix} A_e & B_e \\ B_e^T & 0 \end{bmatrix}. \quad (2.14)$$

Diese Blöcke sind im 2D–Fall 9×9 Matrizen (2×4 Geschwindigkeiten + Druck) und im 3D–Fall 19×19 Matrizen (3×6 Geschwindigkeiten + Druck).

Wir betrachten nun das lokale *Schur*–Komplement:

$$C_e p = B_e^T A_e^{D-1} B_e p \quad \text{mit: } A^D = \text{diag} A \quad (2.15)$$

Wichtig hierbei ist die Tatsache, daß C_e nur noch ein skalarer Wert ist, die Geschwindigkeiten werden dann berechnet durch $A_e^D u = g(f, u, p) - B_e p$, d.h. wir berechnen erst den Druck und dann die Geschwindigkeiten.

Da wir nur den Diagonalanteil der Matrix A mit den Einträgen der Geschwindigkeit be-

trachten und die Nicht-Diagonal Einträge auf die rechte Seite schaffen, können wir nun schnelle lokale Schur-Komplement Löser einsetzen, in unserem Fall das Gauß-Seidel-Verfahren, und wir müssen in unseren Iterationen nur Vektor-Vektor Multiplikationen durchführen. Wir verwenden deswegen das Gauß-Seidel-Verfahren, da wir für die Berechnung auf einem neuen Element, die bereits bekannten Werte der anderen Elemente benutzen können. Bei einer parallelen Version dieses Glättungsoperators, bei der die lokalen Schur-Komplemente simultan behandelt werden, wird man dann das Jacobi-Verfahren einsetzen.

Vorteile:

- Stabilität, auch im Falle der Navier-Stokes-Gleichungen, für hohe Reynoldszahlen und in Verbindung mit dem nichtparametrischen Element (siehe Abschnitt 1.4) auch für anisotrope Gitter.
- Effizienz und gutes Konvergenzverhalten im stationären Fall.

Nachteile:

- Im instationären Fall nur bedingt einsetzbar, da in den einzelnen Zeitschritten fast die gesamte Rechenzeit für das Glätten verbraucht wird, außerdem schlechtes Konvergenzverhalten bei Verwendung kleiner Zeitschritte (d.h. keine Verbesserung für $\Delta t \rightarrow 0$).

Zusammenfassend läßt sich damit sagen, daß der von uns verwendete Glätter für die stationären Stokes und Navier-Stokes Gleichungen sehr gute Eigenschaften besitzt, dies zeigen auch unsere Testrechnungen in Abschnitt 2.5, aber im instationären Fall vor allem in drei Dimensionen einfach eine zu schlechte Effizienz zeigt, worauf wir in den Kapiteln 3 und 4 noch genauer eingehen.

2.3. Die Gittertransferoperatoren

Wir wollen die Konstruktion des Gittertransfers ausführlich beschreiben, da dieser Prozeß bei Mehrgitterverfahren für nichtkonforme Elemente ein wesentlicher Bestandteil darstellt. Durch die fehlende Teilraumbeziehungen sind hier die Techniken anders als im konformen Fall. Wir beschränken uns aber auf eine Beschreibung der Prolongationsoperatoren, da die Restriktionsoperatoren wie schon erwähnt gerade adjungiert zu

den Prolongationsoperatoren sind. Wir behandeln dabei den Druck und die Geschwindigkeiten separat.

Die volle makroweise nichtkonforme Prolongation der Geschwindigkeiten

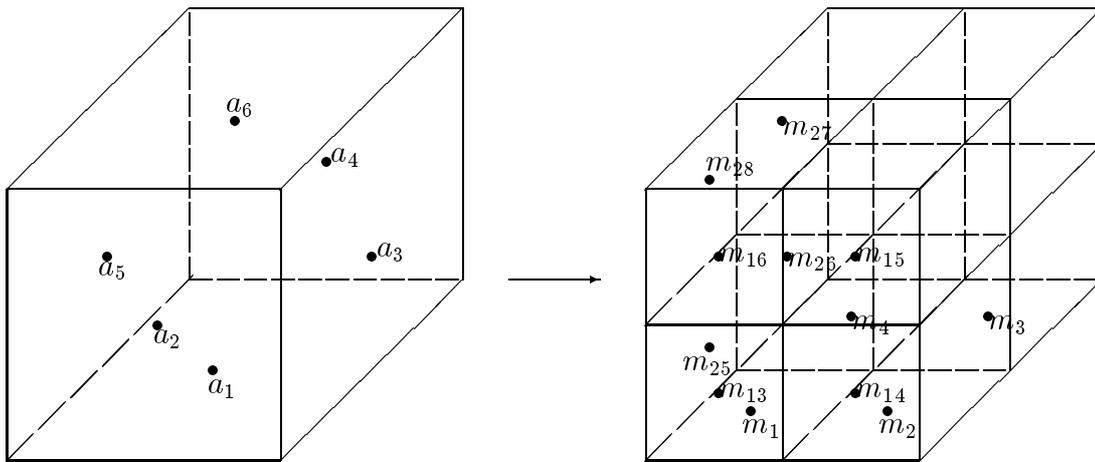


Abbildung 2.2: Die nichtkonforme Prolongation

Die 36 neuen Flächenmitten, bzw. Flächenintegrale, der 8 entstandenen Elemente (Zerlegung wie in Abschnitt 1.2) beschrieben ergeben sich mit obigem Bild wie folgt:

Die Flächenmitten m_1 – m_4 , m_{13} – m_{14} und m_{25} – m_{28} sind dargestellt, die in der Abbildung fehlenden Mitten liegen immer in Blöcken von vier auf den entsprechenden Parallellflächen. Die Beziehung zwischen Grob- und Feingitter Flächenmitten ergibt sich aus der folgenden Tabelle, wobei wir den Fall eines inneren Elementes betrachten. Für Flächenmitten, die auf dem Rand liegen müssen dementsprechend doppelt so große Gewichte gewählt werden.

Wir haben folgende Gewichte:

a.) Punktorientierte Version:

$$A_1=11/24; A_2=7/48; A_3=-5/48; A_4=-1/24; A_5=11/24; A_6=1/12; A_7=-1/24.$$

b.) Integralmittellorientierte Version:

Tabelle 2.1: Interpolationsgewichte für die Prolongation

	a_1	a_2	a_3	a_4	a_5	a_6
m_1	A1	A2	A3	A3	A2	A4
m_2	A1	A2	A2	A3	A3	A4
m_3	A1	A3	A2	A2	A3	A4
m_4	A1	A3	A3	A2	A2	A4
m_5	A6	A5	A7	A7	A5	A6
m_6	A6	A5	A5	A7	A7	A6
m_7	A6	A7	A5	A5	A7	A6
m_8	A6	A7	A7	A5	A5	A6
m_9	A4	A2	A3	A3	A2	A1
m_{10}	A4	A2	A2	A3	A3	A1
m_{11}	A4	A3	A2	A2	A3	A1
m_{12}	A4	A3	A3	A2	A2	A1
m_{13}	A2	A1	A2	A4	A2	A3
m_{14}	A2	A1	A2	A4	A2	A3
m_{15}	A3	A1	A2	A4	A3	A2
m_{16}	A3	A1	A3	A4	A3	A2
m_{17}	A5	A6	A7	A6	A5	A7
m_{18}	A5	A6	A5	A6	A7	A7

	a_1	a_2	a_3	a_4	a_5	a_6
m_{19}	A7	A6	A5	A6	A7	A5
m_{20}	A7	A6	A7	A6	A5	A5
m_{21}	A2	A4	A3	A1	A2	A3
m_{22}	A2	A4	A2	A1	A3	A3
m_{23}	A3	A4	A2	A1	A3	A2
m_{24}	A3	A4	A3	A1	A2	A2
m_{25}	A2	A2	A4	A3	A1	A3
m_{26}	A2	A3	A4	A2	A1	A3
m_{27}	A3	A3	A4	A2	A1	A2
m_{28}	A3	A2	A4	A3	A1	A2
m_{29}	A5	A5	A6	A7	A6	A7
m_{30}	A5	A7	A6	A5	A6	A7
m_{31}	A7	A7	A6	A5	A6	A5
m_{32}	A7	A5	A6	A7	A6	A5
m_{33}	A2	A2	A1	A3	A4	A3
m_{34}	A2	A3	A1	A2	A4	A3
m_{35}	A3	A3	A1	A2	A4	A2
m_{36}	A3	A2	A1	A3	A4	A2

$A1=0.5$; $A2=0.125$; $A3=-0.125$; $A4=0$; $A5= 0.5$; $A6=0$; $A7=0$.

Diese Gewichte berechnen sich aus den Ansatzfunktionen der nichtkonformen Elemente, vgl. auch [48] und [45]. Die Technik zur Berechnung dieser Gewichte ist, daß man das Makroelement auf das Referenzelement transformiert. Dort werden mittels der Basisfunktionen des nichtkonformen trilinearen Elementes die Werte der neu entstehenden Flächenmitten aus den vorhandenen Werten der Flächenmitten des Makroelementes berechnet.

Bemerkung 2.5. *Da wir bei unseren Gitterinterpolationsoperatoren Geschwindigkeiten und Druck separat behandeln, gelten obige Gewichte auch bei Mehrgitterverfahren für Laplace Probleme, wie wir sie in Kapitel 4 bei unserem Projektionsverfahren zu lösen haben.*

Bemerkung 2.6. *Im Falle von Triangulierungen, bei denen Elemente mit großem Volumen an Elemente mit kleinem Volumen grenzen, was z.B. nötig ist um Randschichten aufzulösen, oder bei der Diskretisierung komplexer Geometrien wie in Kapitel 5 be-*

schrieben, verwenden wir eine mit dem Volumen gewichtete Variante. Flächenmitten, die auf zwei Elementen liegen, bekommen nicht die Hälfte der jeweiligen Werte, sondern einen volumenabhängigen Anteil. Bei unseren numerischen Ergebnissen zeigt sich, daß man in solchen Fällen die Konvergenzraten deutlich verbessern kann, die Effizienz wird dabei nicht beeinträchtigt, da wir das Volumen der Elemente im Preprocessing nur einmal berechnen müssen und dann an gegebener Stelle mit übergeben können.

Die Prolongation für den Druck

Wir verwenden für den Gittertransfer des Drucks drei verschiedenen Varianten.

A.) Konstante Interpolation:

Bei dieser einfachen Variante geht der Druck eines Makroelementes konstant auf alle 8 Mikroelemente über, bzw. wird bei der Restriktion von den Mikroelementen eingesammelt.

B.) Lineare nichtkonforme Interpolation:

Wie schon in Abschnitt 1.5 beschrieben, extrapolieren wir den Druck auf die 6 Flächenmitten eines jeden Elementes, um dann mit der gleichen Methode wie bei den Geschwindigkeiten beschrieben den Druck makroweise zu prolongieren, bzw. zu restringieren, danach wird der Druck wieder auf die Elemente extrapoliert.

C.) Lineare konforme Interpolation:

Selbstverständlich können wir die Strategie wie in B.) auch konform linear durchführen, indem wir den Druck erst auf die nun 8 Knoten des Elementes extrapolieren, dann linear prolongieren, bzw. restringieren, und dann wieder zurück extrapolieren.

Bemerkung 2.7. *Natürlich ist Variante A.) die sparsamste, dafür zeigt sich bei Testrechnungen, daß der Aufwand für B.) und C.) zu besserem Konvergenzverhalten des Mehrgitters führt, wobei sich bei den letzten beiden Varianten keine signifikanten Unterschiede ausmachen lassen. Bei allen drei Möglichkeiten können wir wieder wie für die Geschwindigkeiten beschrieben eine gewichtete Option verwenden. Diese Gittertransfer Routinen scheinen sehr aufwendig zu sein, wenn man aber bedenkt, daß bei komplexen Geometrien und daraus resultierenden anisotropen Triangulierungen der weitaus größte Teil der Rechenzeit für das Glätten verbraucht wird, spielt die geschickte Wahl des Gittertransfers eine entscheidende Rolle in der Konvergenzbeschleunigung. Der Mehraufwand wird, wie sich bei unseren numerischen Tests zeigt, durch Einsparen an Glättungsschritten mehr als kompensiert, vor allem in Verbindung mit der adaptiven Schrittweitenkontrolle, wie sie im nächsten Abschnitt beschrieben ist.*

2.4. Die adaptive Schrittweitenkontrolle

Wie schon bei der Formulierung des Mehrgitterverfahrens in Abschnitt 2.1 beschrieben, bestimmen wir den Parameter α_k als:

$$\alpha_k = \frac{(\mathbf{g}_k - S_k(\mathbf{u}_k^m, p_k^m), I_{k-1}^k(\mathbf{u}_{k-1}^q, p_{k-1}^q))_k}{((S_k I_{k-1}^k(\mathbf{u}_{k-1}^q, p_{k-1}^q), I_{k-1}^k(\mathbf{u}_{k-1}^q, p_{k-1}^q))_k)}. \quad (2.16)$$

In Matrix-Vektor Schreibweise ist dies dann gerade:

$$\alpha_k = \frac{\left(\mathbf{g}_k - \begin{bmatrix} A_k & B_k \\ -B_k^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_k^m \\ p_k^m \end{bmatrix}, \begin{bmatrix} I_{k-1}^k \mathbf{u}_{k-1}^q \\ I_{k-1}^k p_{k-1}^q \end{bmatrix} \right)_k}{\left(\begin{bmatrix} A_k & B_k \\ -B_k^T & 0 \end{bmatrix} \begin{bmatrix} I_{k-1}^k \mathbf{u}_{k-1}^q \\ I_{k-1}^k p_{k-1}^q \end{bmatrix}, \begin{bmatrix} I_{k-1}^k \mathbf{u}_{k-1}^q \\ I_{k-1}^k p_{k-1}^q \end{bmatrix} \right)_k} \quad (2.17)$$

Entscheidend ist hier die Wahl des gekoppelten Operators S_k als

$$S_k = \begin{bmatrix} A_k & B_k \\ -B_k^T & 0 \end{bmatrix},$$

dadurch erhalten wir einen definiten Operator, eine andere Wahl würde zur Divergenz des Verfahrens führen. Im nächsten Abschnitt untersuchen wir diese Komponenten des Mehrgitterverfahrens numerisch.

2.5. Numerische Ergebnisse für das Mehrgitterverfahren

Wir untersuchen das Verhalten der in den letzten Abschnitten vorgestellten Gittertransfertechniken, sowie der adaptiven *step length control* anhand eines praktisch relevanten Beispiels, der Umströmung einer Platte in einem Kanal. Für alle Rechnungen verwenden wir den Vanka–Glätter und die nichtparametrische Version des nichtkonformen Elements, wie in Abschnitt 1.4 beschrieben. Wir haben ein angepaßtes Grobgitter, was heißt, daß im Nachlaufbereich des Hindernisses das Gitter feiner ist, als im übrigen Gebiet. Am Einströmrand haben wir Dirichlet-, am Ausströmrand Neumanndaten, die sogenannte “do nothing“ Bedingung, auf die wir in Abschnitt 5.3 noch genauer zu sprechen kommen. Das folgende Bild zeigt Grobgitter und einen 2D–Schnitt der Geschwindigkeitsvektoren. Die Strömung verläuft von rechts nach links, wir erkennen dabei die schräggestellte Platte.

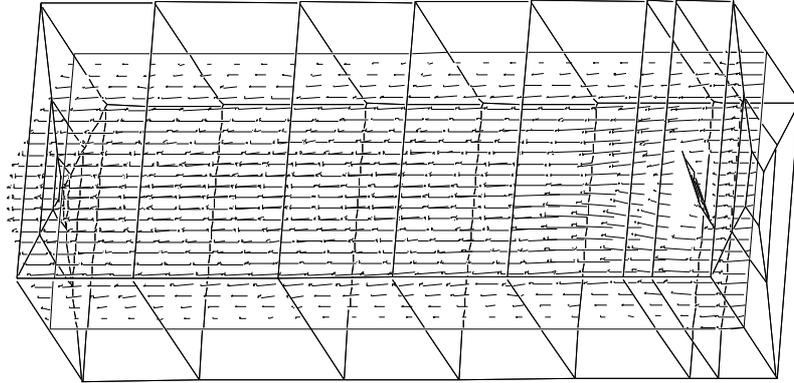


Abbildung 2.3: Grobgitter mit Geschwindigkeitsvektoren

Wir haben folgende Ergebnisse für das stationäre Stokes–Problem, wobei wir drei verschiedene Konfigurationen betrachten:

- i.) Ungewichtete Interpolation für die Geschwindigkeiten, konstante für den Druck, ohne adaptive step length control.
- ii.) Gewichtete Interpolationen für Geschwindigkeiten und Druck, Druck zudem linear interpoliert, ohne adaptive step length control.
- iii.) Mit allen Optionen.

Die Viskosität bei diesem Beispiel beträgt 1, ebenso die Einströmgeschwindigkeit, wir verwenden 4 Vor- und Nachglättungsschritte und wir führen je 10 Mehrgitterschritte durch. Um die Effizienz der Optionen vergleichen zu können, definieren wir ein Effizienzmaß γ (siehe [48] und [45]) mit folgender Formel, NIT (hier =10) bezeichne dabei die Anzahl der Iterationen, NEL die Anzahl der Elemente und $TTMG$ die Gesamtzeit.

$$\text{Effizienzmaß: } \gamma = -\frac{1000 TTMG}{NIT NEL \log(cvg)}$$

Dieses Effizienzmaß mißt damit die benötigte Zeit in Millisekunden pro Unbekannte, um eine Stelle an Genauigkeit zu gewinnen und ist im Mehrgitterkontext neben der Konvergenzrate ein wichtiges Kriterium, d.h. die Raten sollten beide konstant bleiben für $h \rightarrow 0$.

Tabelle 2.2: Konvergenzraten und Effizienzmaß für i.)–iii.)

<i>NEL</i>	i.)		ii.)		iii.)	
	cvg.	γ	cvg.	γ	cvg.	γ
696	0.47	5.51	0.41	4.64	0.24	2.74
5568	0.46	5.56	0.45	5.11	0.31	3.80
44544	0.69	12.96	0.61	10.10	0.35	4.94
356544	0.72	16.16	0.63	12.88	0.33	5.68

Wie man sieht, erzielen diese “Konvergenzbeschleuniger“ die beabsichtigte Wirkung, wird das Gitter noch anisotroper werden auch die Unterschiede größer. Bei der dritten Option bleiben nicht nur die Konvergenzraten konstant, sondern auch das Effizienzmaß, d.h. wir erzielen mit diesen Techniken auch eine Verringerung der Rechenzeiten.

Als zweites Beispiel betrachten wir eine *3D-Driven-Cavity*, auf die wir in Abschnitt 5.1 noch genauer eingehen. Hier haben wir zwar ein Gitter, wo alle gegenüberliegenden Flächen parallel sind, dafür aber einen hohen *aspect ratio* haben, d.h. das Verhältnis kürzeste/längste Kante ist hier der Faktor 10, dieses Verhältnis bleibt bei allen Gittern bestehen (siehe Abschnitt 1.5). Die Viskosität bei diesem Beispiel beträgt 1, wir führen je 4 Vor- und Nachglättungsschritte durch. Wir erhalten dann folgende Ergebnisse:

Tabelle 2.3: Konvergenzraten und Effizienzmaß für die 3D-Driven-Cavity

<i>NEL</i>	i.)		iii.)	
	cvg.	γ	cvg.	γ
1024	0.54	7.51	0.38	4.63
8192	0.58	11.16	0.36	5.13
65536	0.66	17.99	0.39	6.02

Die Konvergenzraten sind deutlich besser bei iii.); legt man die Faustregel zugrunde, die besagt, daß bei einer Konvergenzrate größer als 0.5 eine Verdoppelung der Glättungsschritte eine Halbierung der Konvergenzrate bewirkt, so müßten wir bei Variante i.) den doppelten Aufwand betreiben zur Erzielung der gleichen Konvergenzrate.

Test für Kapitel 1+2

Zum Abschluß der ersten beiden Kapitel über das Stokes-Problem führen wir noch eine Rechnung für die Umströmung einer Platte durch, wobei das Gitter zum Einen

stark anisotrop ist im Bereich der Platte, zum Anderen einen hohen *aspect ratio* hat, im weiteren Nachlaufbereich des Kanals. Wir haben damit die Situation aus Abbildung 2.3, nur verwenden wir eben ein modifiziertes Gitter.

Wir führen 8 Vor- und Nachglättungsschritte durch, im ersten Testfall verwenden wir die parametrische Version der nichtkonformen Elemente, keine gewichtete Interpolation, Druck nur konstant interpoliert und keine *adaptive step lenght control*. Im nächsten Test verwenden wir alle Optionen für dieses Element. Dasselbe führen wir dann auch für die nichtparametrischen Elemente durch (diese Rechnungen wurden auf einer IBM RS6000/590 durchgeführt, obige Rechnungen auf einer Sun SS10/xx).

Tabelle 2.4: Konvergenzraten und Effizienzmaß für Test Kapitel 1+2

	param.ohne		param. mit		nichtparam.ohne		nichtparam. mit	
	cvg.	γ	cvg.	γ	cvg.	γ	cvg.	γ
<i>NEL</i>								
648	0.40	1.97	0.42	2.68	0.27	1.15	0.16	0.95
5184	0.84	14.01	0.55	4.14	0.53	3.82	0.19	0.89
41472	0.86	13.51	0.73	8.08	0.56	4.29	0.18	0.84
331776	0.88	15.19	0.78	10.80	0.62	5.41	0.19	0.88

Die Zahlen belegen eindeutig den Effekt der von uns verwandten Techniken zur Stabilisierung, Konvergenzbeschleunigung und daraus resultierende Verkürzung der Rechenzeiten. Damit beenden wir unsere Analyse des Mehrgitterverfahrens für das 3D Stokes–Problem und wenden uns den stationären und instationären Navier–Stokes–Gleichungen zu, für deren Lösung natürlich die bisher beschriebenen Techniken benutzt werden.

3. Die Navier–Stokes–Gleichungen

In diesem Kapitel wollen wir ein robustes und effizientes Verfahren zur Lösung der instationären Navier–Stokes–Gleichungen in der Form

$$\begin{aligned} \mathbf{u}_t - \nu \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p = \mathbf{f} \quad , \quad \nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \times (0, T) \, , \\ \mathbf{u} = \mathbf{g} \quad \text{auf } \partial\Omega \quad , \quad \mathbf{u}|_{t=0} = \mathbf{u}_0 \, , \end{aligned}$$

entwickeln. Dabei beschreiben wieder $\{\mathbf{u}(x, t), p(x, t)\}$ die unbekannte Geschwindigkeit und den Druck einer zähen, inkompressiblen Strömung in Ω mit vorgegebener Viskosität ν , Volumenkraft \mathbf{f} , Randgeschwindigkeit \mathbf{g} und Anfangsgeschwindigkeit \mathbf{u}_0 (siehe [20],[47]). Wie gewöhnlich wird die Dichte der Flüssigkeit normalisiert.

Zunächst betrachten wir jedoch den stationären Fall, für den wir einen effizienten Löser konstruieren auf der Basis des im vorangegangenen Kapitel vorgestellten Mehrgitterverfahrens, um dann in einem nächsten Schritt die instationären Navier–Stokes–Gleichungen unter Verwendung eines Zeitdiskretisierungsverfahrens anzukoppeln. Die Hauptkomponenten unseres Verfahrens sind dabei eine *Upwind*–Diskretisierung, bzw. die Stromliniendiffusionsmethode zur Stabilisierung der Konvektionsterme und ein effizienter Mehrgitterlöser für die linearisierten Navier–Stokes–Gleichungen, hier verwenden wir natürlich den im vorigen Kapitel vorgestellten Stokes–Löser.

Dieser stationäre Navier–Stokes–Löser wird im instationären Fall als Grundlage dienen, sowohl bei dem gekoppelten Lösungsansatz (simultane Behandlung der Geschwindigkeiten und des Drucks), der durch seinen impliziten Charakter für Anwendungen im niederen Reynoldsbereich bei stationären Lösungen (die wir aber nicht! stationär berechnen können), als auch im voll instationären Fall, d.h. höheren Reynoldszahlbereich $Re > 1000$, bei dem wir einen Splittingansatz verwenden, den wir im nächsten Kapitel vorstellen werden. Außerdem wollen wir die beiden Stabilisierungstechniken –*Upwind*–Diskretisierung und Stromliniendiffusionsmethode– numerisch vergleichen.

3.1. Die stationären Navier–Stokes–Gleichungen

Wir beschränken uns in diesem Abschnitt wie in Kapitel 1 auf die Vereinfachung homogener Randdaten $\mathbf{g} = \mathbf{0}$. Die analogen Resultate für inhomogene Randwerte können in [20] und [47] nachgelesen werden. Eine schwache Formulierung der stationären Navier–Stokes–Gleichungen schreibt sich dann mit den Notationen aus Kapitel 1 wie folgt:

Finde ein Paar $\{\mathbf{u}, p\} \in \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega)$, so daß gilt

$$\nu a(\mathbf{u}, \mathbf{v}) + n(\mathbf{u}, \mathbf{u}, \mathbf{v}) + b(p, \mathbf{v}) + b(q, \mathbf{u}) = (\mathbf{f}, \mathbf{v}) \quad , \quad \forall \{\mathbf{v}, q\} \in \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega) \quad , \quad (\text{NS})$$

Hierbei ist die Trilinearform $n(\mathbf{u}, \mathbf{v}, \mathbf{w})$ für Funktionen $\mathbf{u} \in \mathbf{V}(\Omega)$ und $\mathbf{v}, \mathbf{w} \in \mathbf{H}_0^1(\Omega)$ in Anlehnung an die Arbeiten von [20] und [52] definiert als

$$n(\mathbf{u}, \mathbf{v}, \mathbf{w}) := \frac{1}{2} \int_{\Omega} \left(u_i \frac{\partial v_j}{\partial x_i} w_j - u_i v_j \frac{\partial w_j}{\partial x_i} \right) dx \quad . \quad (3.1)$$

Für hinreichend kleine Daten (z.B. ν hinreichend groß, $\|f\|_{-1}$ hinreichend klein) besitzt das Problem (NS) eine eindeutige Lösung (siehe [20],[47]).

Zur Diskretisierung der Navier–Stokes–Gleichungen sei wieder eine Zerlegung \mathbf{T}_h von Ω mit den Eigenschaften aus Abschnitt 1.2 gegeben. Als diskrete Ansatzräume verwenden wir die vorher eingeführten Geschwindigkeitsräume \mathbf{H}_h und Druckräume L_h (siehe Abschnitte 1.3 und 1.4). Neben den entsprechenden Bilinearformen $a_h(\cdot, \cdot)$ und $b_h(\cdot, \cdot)$ definieren wir auch die diskrete Trilinearform $n_h(\cdot, \cdot, \cdot)$ durch

$$n_h(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) := \frac{1}{2} \sum_{T \in \mathbf{T}_h} \int_T \left(u_{h,i} \frac{\partial v_{h,j}}{\partial x_i} w_{h,j} - u_{h,i} v_{h,j} \frac{\partial w_{h,j}}{\partial x_i} \right) dx \quad . \quad (3.2)$$

Man sieht sofort, daß diese diskrete Form für Funktionen $\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h \in \mathbf{H}_h$ die Eigenschaften

$$n_h(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) = -n_h(\mathbf{u}_h, \mathbf{w}_h, \mathbf{v}_h) \quad (3.3)$$

und damit auch

$$n_h(\mathbf{u}_h, \mathbf{v}_h, \mathbf{v}_h) = 0 \quad (3.4)$$

besitzt. Eine diskrete Formulierung von Problem (NS) lautet dann:

Finde ein Paar $\{\mathbf{u}_h, p_h\} \in \mathbf{H}_h \times L_h$, so daß gilt

$$\nu a_h(\mathbf{u}_h, \mathbf{v}_h) + n_h(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h) + b_h(p_h, \mathbf{v}_h) + b_h(q_h, \mathbf{u}_h) = (\mathbf{f}, \mathbf{v}_h) \quad , \forall \{\mathbf{v}_h, q_h\} \in \mathbf{H}_h \times L_h \quad , \quad (\text{NS}_h)$$

Aufgrund der Eigenschaften von $n_h(\cdot, \cdot, \cdot)$, den Stabilitäts- und Approximationseigenschaften von \mathbf{H}_h und L_h , die in den vorherigen Kapiteln hergeleitet wurden, gilt dann folgendes Standardresultat ([20],[47]) für die zentrale Diskretisierung $n_h(\cdot, \cdot, \cdot)$:

Satz 3.1. *Die Probleme (NS) und (NS_h) besitzen unter den obigen Voraussetzungen eindeutig bestimmte Lösungspaare. Wenn zusätzlich die kontinuierlichen Lösungen $\{\mathbf{u}, p\} \in \mathbf{H}^2(\Omega) \times H^1(\Omega)$ sind, dann gelten die Fehlerabschätzungen*

$$\|\mathbf{u} - \mathbf{u}_h\|_h + \|p - p_h\|_0 \leq ch \{ \|\mathbf{u}\|_2 + \|p\|_1 \} \quad , \quad (3.5)$$

$$\|\mathbf{u} - \mathbf{u}_h\|_0 + \|p - p_h\|_{-1} \leq ch^2 \{ \|\mathbf{u}\|_2 + \|p\|_1 \} \quad . \quad (3.6)$$

Als Lösungsverfahren der diskreten Probleme (NS_h) verwenden wir eine *Fixpunkt-Defektkorrektur*-Methode mit Dämpfung ([42],[52]). Dazu führen wir Matrizen A_h und $N_h(U_h)$ entsprechend den diskreten Bi- bzw. Trilinearformen ein und können damit Problem (NS_h) umformulieren als:

Finde $\mathbf{u}_h = \sum U_h^i \mathbf{v}_h^i \in \mathbf{H}_h$, so daß gilt

$$\nu A_h U_h + N_h(U_h) U_h + B_h P_h = F_h, B_h^T U_h = 0 \quad (\text{NS}_h)$$

Die Iterationsvorschrift des Verfahrens lautet dann:

Gegeben sei ein Startvektor (U_h^0, P_h^0) , berechne rekursiv die weiteren Iterierten durch

$$\begin{bmatrix} U_h^{n+1} \\ P_h^{n+1} \end{bmatrix} = \begin{bmatrix} U_h^n \\ P_h^n \end{bmatrix} + \quad (3.7)$$

$$\omega_h^{n+1} \begin{bmatrix} \nu A_h + N_h(U_h^n) & B_h \\ B_h^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} F_h - \nu A_h U_h^n - N_h(U_h^n) U_h^n - B_h P_h^n \\ -B_h^T U_h^n \end{bmatrix}$$

Dabei ist $\omega_h^{n+1} > 0$ ein Schrittweitenparameter, auf den wir später noch eingehen werden. Für $\omega_h = 1$ ist dieses Iterationsverfahren gerade die *gewöhnliche* Fixpunktiteration

$$\begin{bmatrix} \nu A_h + N_h(U_h^n) & B_h \\ B_h^T & 0 \end{bmatrix} \begin{bmatrix} U_h^{n+1} \\ P_h^{n+1} \end{bmatrix} = \begin{bmatrix} F_h \\ 0 \end{bmatrix}. \quad (3.8)$$

In jedem nichtlinearen Iterationsschritt muß damit eine indefinite nichtsymmetrische Matrix der Form:

$$\begin{bmatrix} \nu A_h + N_h(U_h^n) & B_h \\ B_h^T & 0 \end{bmatrix}$$

invertiert werden, die der Diskretisierungsmatrix eines linearisierten Navier–Stokes bzw. eines Oseen–Problems entspricht:

$$-\nu \Delta \mathbf{u} + (\mathbf{u}^n \cdot \nabla) \mathbf{u} + \nabla p = \text{ rechte Seite } , \quad \nabla \cdot \mathbf{u} = 0. \quad (3.9)$$

Durch die zentrale Diskretisierung der Konvektionsterme $n_h(\mathbf{u}_h^n, \mathbf{v}_h, \mathbf{w}_h)$ entstehen für kleines ν numerische Instabilitäten. Daher können wir den Mehrgitteralgorithmus für die Stokes–Gleichungen aus Kapitel 2 nicht einsetzen. Da die Systemmatrizen $N_h(U_h^n)$ sämtliche günstigen Matrizeneigenschaften wie Symmetrie und M–Matrizeigenschaft verlieren, können wir keine Standarditerationen wie Jacobi-, SOR- oder CG–Verfahren verwenden. Abhilfe schafft hier eine modifizierte Diskretisierung des Konvektionsterms $n(\cdot, \cdot, \cdot)$, die zur Stabilisierung beiträgt. Im Folgenden bezeichnen wir mit $n_h^1(\cdot, \cdot, \cdot)$ und $n_h^2(\cdot, \cdot, \cdot)$ zwei mögliche Diskretisierungen (siehe nachfolgende *Upwind*–Diskretisierung oder *Stromliniendiffusionsverfahren*), dann schreibt sich das Iterationsverfahren als

$$\begin{bmatrix} U_h^{n+1} \\ P_h^{n+1} \end{bmatrix} = \begin{bmatrix} U_h^n \\ P_h^n \end{bmatrix} + \quad (3.10)$$

$$\omega_h^{n+1} \begin{bmatrix} \nu A_h + N_h^1(U_h^n) & B_h \\ B_h^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} F_h - \nu A_h U_h^n - N_h^2(U_h^n) U_h^n - B_h P_h^n \\ -B_h^T U_h^n \end{bmatrix}$$

Dies bedeutet, daß man die Lösung des Problems beispielsweise zu einer genaueren, aber instabileren Diskretisierung bestimmt, als Vorkonditionierer aber in jedem nicht-linearen Teilschritt eine stabilere, dafür ungenauere Diskretisierung verwendet. Wir verwenden zwei verschiedene Möglichkeiten der Stabilisierung, zum Einen eine *Upwind*-Diskretisierung, zum Anderen die *Stromliniendiffusionsmethode*.

3.1.1. Upwind–Diskretisierung

Wir folgen nun den Arbeiten von Ohmori/Ushijima [36], die die *Upwind*-Diskretisierung für nichtkonforme Finite-Elemente-Ansatzräume zur Lösung der Konvektions-Diffusionsgleichungen entwickelt haben, und der Arbeiten von Tobiska/Schieweck in [52] und [43], die das Verfahren auf die Navier–Stokes–Gleichungen in primaler Darstellung übertragen haben. Die genauen Ergebnisse für das nichtkonforme Dreieckselement können in diesen beiden Arbeiten nachgelesen werden; wir zeigen nachfolgend die Herleitung der *Upwind*-Diskretisierung im Falle der Hexaederelemente.

Der Grundgedanke dabei ist die Einführung von geeigneten flächenorientierten *Lumping Gebieten* und *Lumping Operatoren*. Wir teilen dazu jeden Hexaeder $T \in \mathbf{T}_h$ in 24 baryzentrische Fragmente S_{ij} auf (siehe Abbildung 3.1) und definieren für jede Fläche Γ_l bzw. jede Flächenmitte m_l das *Lumping Gebiet* R_l (siehe Abbildung 3.2) durch

$$R_l := \bigcup_{k \in \Lambda_l} S_{lk}, \quad (3.11)$$

wobei Λ_l die Menge von Indizes k ist, für die m_l und m_k benachbarte Flächenmitten sind.

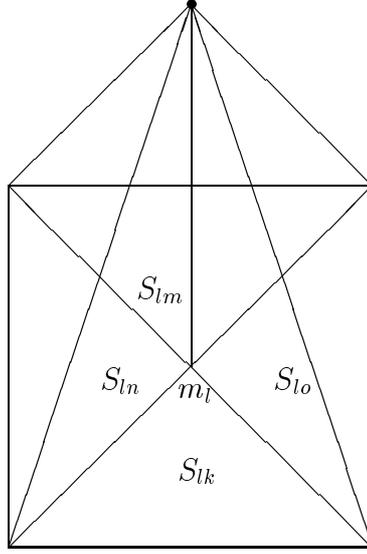


Abbildung 3.1: Baryzentrische Fragmente S_{ij} einer Fläche eines Hexaeders

Definieren wir weiterhin die Fläche Γ_{lk} als

$$\Gamma_{lk} := \partial S_{lk} \cap \partial S_{kl}, \quad (3.12)$$

so erhalten wir für den Rand ∂R_l des *Lumping Gebiets* R_l die Charakterisierung (siehe Abbildung 3.2)

$$\partial R_l = \bigcup_{k \in \Lambda_l} \Gamma_{lk}. \quad (3.13)$$

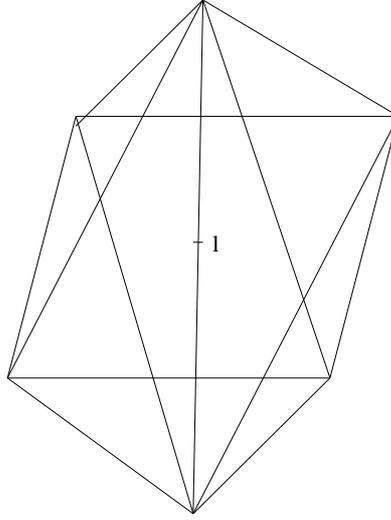
Damit erhalten wir eine neue flächenmittensbezogene Partitionierung von $\bar{\Omega} = \cup_{T \in \mathbf{T}_h} \bar{T}$ in Form von

$$\bar{\Omega} = \bigcup_l \bar{R}_l. \quad (3.14)$$

Auf diesen *Lumping Gebieten* führen wir nun einen stückweise konstanten *Lumping Operator* L_h für Funktionen $\mathbf{v}_h \in \mathbf{H}_h$ gemäß der Vorschrift

$$(L_h \mathbf{v}_h)(x) = \mathbf{v}_h(m_l) \quad , \quad \forall x \in R_l \quad (3.15)$$

ein. Man sieht sofort (siehe [36] im Dreiecksfall), daß für diesen Interpolationsoperator

Abbildung 3.2: Lumping Gebiet R_l der Fläche l

gilt:

$$\|\mathbf{v}_h\|_0 \sim \|L_h \mathbf{v}_h\|_0, \quad \forall \mathbf{v}_h \in \mathbf{H}_h, \quad (3.16)$$

$$\|L_h \mathbf{v}_h - \mathbf{v}_h\|_0 \leq ch \|\mathbf{v}_h\|_h, \quad \forall \mathbf{v}_h \in \mathbf{H}_h. \quad (3.17)$$

Wir spalten nun die diskrete Trilinearform $n_h(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h)$ auf in

$$n_h(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) = n_h^1(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) + n_h^2(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) \quad (3.18)$$

mit

$$n_h^1(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) := \sum_{T \in \mathbf{T}_h} \int_T \frac{\partial u_{h,i} v_{h,j}}{\partial x_i} w_{h,j} dx, \quad (3.19)$$

$$n_h^2(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) := - \sum_{T \in \mathbf{T}_h} \int_T \frac{\partial u_{h,i}}{\partial x_i} v_{h,j} w_{h,j} dx. \quad (3.20)$$

Anschließend ersetzen wir $n_h^1(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h)$ durch $n_h^1(\mathbf{u}_h, \mathbf{v}_h, L_h \mathbf{w}_h)$ und $n_h^2(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h)$ durch $n_h^2(\mathbf{u}_h, L_h \mathbf{v}_h, L_h \mathbf{w}_h)$ und modifizieren beide zu (siehe [36])

$$\tilde{n}_h^1(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) := \sum_l \sum_{k \in \Lambda_l} \oint_{\Gamma_{lk}} (\mathbf{u}_h \cdot \mathbf{n}_{lk}) \mathbf{v}_h^{lk} d\gamma \mathbf{w}_h(m_l), \quad (3.21)$$

$$\tilde{n}_h^2(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) := - \sum_l \sum_{k \in \Lambda_l} \oint_{\Gamma_{lk}} \mathbf{u}_h \cdot \mathbf{n}_{lk} d\gamma \mathbf{v}_h(m_l) \mathbf{w}_h(m_l), \quad (3.22)$$

mit

$$\mathbf{v}_h^{lk} := \lambda_{lk} \mathbf{v}_h(m_l) + (1 - \lambda_{lk}) \mathbf{v}_h(m_k) \quad (3.23)$$

und Schalterfunktionen λ_{lk} mit

$$\lambda_{lk} = 1 - \lambda_{kl} \quad , \quad |\lambda_{lk}| \leq c. \quad (3.24)$$

Schließlich definieren wir die neue Form $\tilde{n}_h(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h)$ durch

$$\tilde{n}_h(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) := \tilde{n}_h^1(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) + \tilde{n}_h^2(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h), \quad (3.25)$$

d.h., es gilt insgesamt

$$\tilde{n}_h(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) = \sum_l \sum_{k \in \Lambda_l} \oint_{\Gamma_{lk}} \mathbf{u}_h \cdot \mathbf{n}_{lk} d\gamma (1 - \lambda_{lk}(\mathbf{u}_h)) (\mathbf{v}_h(m_k) - \mathbf{v}_h(m_l)) \mathbf{w}_h(m_l). \quad (3.26)$$

Man hat nun verschiedene Möglichkeiten, den Schalter $\lambda_{lk}(\mathbf{u}_h)$ zu setzen. Die beiden von uns verwendeten sind (mit $x = \frac{1}{\nu} h_{lk} u_{lk}$, mit $h_{lk} =$ lokales h auf Γ_{lk} , $u_{lk} =$ lokale Geschwindigkeit auf Γ_{lk}):

- 1) *einfaches Upwind*: $\lambda_{lk}(\mathbf{u}_h) = \left\{ \begin{array}{ll} 1 & \text{wenn } x \geq 0 \\ 0 & \text{sonst} \end{array} \right\}.$
- 2) *Modifiziertes Samarskij-Upwind* ([51]): $\lambda_{lk}(\alpha, \mathbf{u}_h) = \left\{ \begin{array}{ll} \frac{\frac{1}{2} + x^\alpha}{1 + x^\alpha} & \text{wenn } x \geq 0 \\ \frac{1}{2(1 - x^\alpha)} & \text{sonst} \end{array} \right\}.$

($\alpha = 1$: klassisches $\alpha = 0$: zentrales $\alpha \rightarrow \infty$: einfaches *Upwind*)

3.1.2. Stromliniendiffusionsmethode

Wir folgen hier der Arbeit von Lube in [33] und betrachten dazu wieder die stationären Navier–Stokes–Gleichungen:

$$-\nu\Delta\mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega, \quad (3.27)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega, \quad (3.28)$$

Für gegebenes \mathbf{a} mit $\mathbf{a} \in L^\infty(\Omega)^d \cap H_{div}(\Omega)$, $\nabla \cdot \mathbf{a} = 0$, in Ω , $H_{div}(\Omega) := \{\mathbf{v} \in L^2(\Omega)^d, \nabla \cdot \mathbf{v} \in L^2(\Omega)\}$ sei die Bilinearform B folgendermaßen definiert:

$$B(\mathbf{a}; \hat{\mathbf{u}}, \mathbf{v}) := \nu(\nabla\mathbf{u}, \nabla\mathbf{v}) + \frac{1}{2}\{((\mathbf{a} \cdot \nabla)\mathbf{u}, \mathbf{v})_\Omega - ((\mathbf{a} \cdot \nabla)\mathbf{v}, \mathbf{u})_\Omega\} - (p, \nabla \cdot \mathbf{v})_\Omega. \quad (3.29)$$

Dann lautet die schwache Formulierung unseres kontinuierlichen Problems folgendermaßen:

$$\hat{\mathbf{u}} \in \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega) : B(\mathbf{u}, \hat{\mathbf{u}}, \mathbf{v}) = (\mathbf{f}, \mathbf{v})_\Omega, \quad \forall \mathbf{v} \in \mathbf{H}_0^1(\Omega), \quad (3.30)$$

$$(\nabla \cdot \mathbf{u}, q)_\Omega = 0, \quad \forall q \in L_0^2(\Omega), \quad (3.31)$$

mit $\hat{\mathbf{u}} = (\mathbf{u}, p)$.

Die entsprechende diskrete Formulierung von (3.30), (3.31) mit $\hat{\mathbf{u}}_h = (\mathbf{u}_h, p_h) \in \mathbf{H}_h \times L_h$ schreibt sich somit als:

$$B(\mathbf{u}_h, \hat{\mathbf{u}}_h, \mathbf{v}_h) = (\mathbf{f}, \mathbf{v}_h)_\Omega, \quad \forall \mathbf{v}_h \in \mathbf{H}_h, \quad (3.32)$$

$$(\nabla \cdot \mathbf{u}_h, q_h)_\Omega = 0, \quad \forall q_h \in L_h, \quad (3.33)$$

Bemerkung 3.1. Für die nachfolgenden Überlegungen ist es wesentlich, daß unser Elementepaar die Babuška–Brezzi–Bedingung erfüllt (siehe Abschnitt 1.2) und dabei der Druck stückweise konstant ist. Dadurch ergeben sich wesentliche Unterschiede zu der Argumentation bei Lube, der in seinem Beweis von stabilisierten Elementepaaren wie z.B. $Q1/Q1$ ausgeht und außerdem einen Spezialfall der Galerkin Least–Square Methode zeigt.

Mit der Stromliniendiffusionsmethode erhalten wir nun mit der neuen Bilinearform \hat{B} , die folgendermaßen aussieht,

$$\hat{B}(\mathbf{a}; \hat{\mathbf{u}}, \hat{\mathbf{v}}) := B(\mathbf{a}; \hat{\mathbf{u}}, \mathbf{v}) + \sum_T ((\mathbf{a} \cdot \nabla)\mathbf{u}, \delta_T(\mathbf{a} \cdot \nabla)\mathbf{v}) + (\nabla \cdot \mathbf{u}, q)_\Omega, \quad (3.34)$$

für $\hat{\mathbf{u}}_h = (\mathbf{u}_h, p_h) \in \mathbf{H}_h \times L_h$ die diskrete Formulierung:

$$\hat{B}(\mathbf{u}_h; \hat{\mathbf{u}}_h, \hat{\mathbf{v}}_h) = L(\mathbf{u}_h; \hat{\mathbf{v}}_h), \quad \forall \hat{\mathbf{v}}_h = (v_h, q_h) \in \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega) \quad (3.35)$$

mit $L(\mathbf{a}; \hat{\mathbf{v}}_h) := (\mathbf{f}, \mathbf{v})_\Omega + \sum_T (\mathbf{f}, \delta_T(\mathbf{a} \cdot \nabla) \mathbf{v})_T$.

Wie im Unterabschnitt 3.1.1 wollen wir wieder unsere Fixpunktiteration verwenden und müssen dazu ganz analog linearisierte Navier–Stokes–Gleichungen lösen.

Im folgenden setzen wir daher \mathbf{a} als gegeben voraus und wollen eine Fehlerabschätzung für die obige linearisierte Methode durchführen. Zuerst zeigen wir die positive Definitheit der Bilinearform $\hat{B}(\mathbf{a}; \cdot, \cdot)$.

Lemma 3.2. *Für $\delta_T C^2 \nu \leq h^2$ (mit der gitterabhängigen Konstante C), ist die Bilinearform $\hat{B}(\mathbf{a}; \cdot, \cdot)$ positiv definit, d.h.*

$$\hat{B}(\mathbf{a}; \hat{\mathbf{v}}_h, \hat{\mathbf{v}}_h) \geq \frac{1}{2} \|\hat{\mathbf{v}}_h\|^2, \quad \forall \hat{\mathbf{v}}_h = (\mathbf{v}_h, q_h) \in \mathbf{H}_h \times L_h \quad (3.36)$$

Beweis.

Wir setzen $\hat{\mathbf{v}}_h = \hat{\mathbf{u}}_h$ in (3.34) und haben dann

$$\hat{B}(\mathbf{a}; \hat{\mathbf{v}}_h, \hat{\mathbf{v}}_h) = \nu \|\nabla \mathbf{v}_h\|^2 + \sum_T \delta_T \|\mathbf{a} \cdot \mathbf{v}_h\|^2 + \sum_T \delta_T (-\nu \Delta \mathbf{v}_h, (\mathbf{a} \cdot \nabla) \mathbf{v}_h)$$

Den letzten Term können wir weiter abschätzen:

$$\delta_T (-\nu \Delta \mathbf{v}_h, \mathbf{a} \cdot \nabla \mathbf{v}_h) \leq \sum_T \delta_T \nu \|\Delta \mathbf{v}_h\| \|\mathbf{a} \cdot \mathbf{v}_h\|$$

Für den verbleibenden Rest nützen wir eine inverse Beziehung und erhalten:

$$\delta_T \nu \|\Delta \mathbf{v}\| \|\mathbf{a} \cdot \mathbf{v}_h\| \leq C \delta_T \frac{\nu}{h} \|\nabla \mathbf{v}_h\| \|(\mathbf{a} \cdot \nabla) \mathbf{v}_h\| \leq \frac{1}{2} \nu \|\nabla \mathbf{v}_h\|^2 + \frac{1}{2} \frac{\delta_T^2 C^2 \nu}{h^2} \|(\mathbf{a} \cdot \nabla) \mathbf{v}_h\|^2$$

Damit haben wir mit der Bedingung

$$\frac{\delta_T C^2 \nu}{h^2} \leq 1 \quad (3.37)$$

und

$$|||\hat{\mathbf{u}}_h|||^2 := \nu |||\nabla \mathbf{u}_h|||^2 + \sum_T \delta_T |(\mathbf{a} \cdot \nabla) \mathbf{u}_h|^2 \quad (3.38)$$

Lemma 3.2 bewiesen. \square

Mit dieser Norm können wir die Fehlerabschätzungen des Verfahrens angeben und wollen dies als Satz formulieren:

Satz 3.3. *Unter den Bedingungen aus Lemma 3.2 gelten für die Stromliniendiffusionsmethode mit dem nichtkonformen rotiert trilinearen Element die folgenden Fehlerabschätzungen für Druck und Geschwindigkeiten:*

$$\begin{aligned} |||\mathbf{u} - \mathbf{u}_h||| &\leq C \left(\nu h + \nu^{1/2} h + \delta^{-1/2} h^2 + \delta^{1/2} h \|a\|_{L^\infty} + \nu \delta^{1/2} \right) \|\nabla^2 \mathbf{u}\| \\ &\quad + C \left(h + \min\{\delta^{1/2}, \delta \nu^{-1/2}\} \|a\|_{L^\infty} \right) \|\nabla p\|, \\ \|p - p_h\| &\leq C \left(\nu h + h^2 \|a\|_{L^\infty} + \delta h \|a\|_{L^\infty} \right) \|\nabla^2 \mathbf{u}\| + C \left(h + \delta \|a\|_{L^\infty} \right) \|\nabla p\| \\ &\quad + C \left(\nu + \delta \|a\|_{L^\infty}^2 \right) |||\mathbf{u} - \mathbf{u}_h|||. \end{aligned}$$

Der Beweis verläuft mit Hilfe von Standardargumenten.

Bemerkung 3.2. *Für hohe Reynoldszahlen ist die Konvergenzordnung für den H^1 -Fehler der Geschwindigkeit $O(\delta^{1/2})$, die Ordnung für den Druck ist in diesem Fall besser. Für niedrige Reynoldszahlen haben wir die Ordnung $O(h)$ für Druck und Geschwindigkeiten.*

Bemerkung 3.3. *In den nachfolgenden Testrechnungen zeigt sich, daß die Stromliniendiffusionsmethode sensibel vom Parameter $\delta = O(h)$ abhängt und die Methode nur einen kleinen Stabilitätsbereich hat.*

3.1.3. Numerischer Vergleich beider Stabilisierungen

Als Vergleichsrechnung betrachten wir wieder wie schon in Abschnitt 2.5 die Umströmung einer Platte, dabei geben wir auf den Rändern eine polynomiale Lösung vor, um die Fehler messen zu können. Wir berechnen die stationäre Lösung zu verschiedenen

Reynoldszahlen, zu unterschiedlichem δ^* , sowie zu verschiedenen Gittern (regelmäßiges Grobgitter versus angepaßtes, und damit anisotropes Grobgitter).

Dabei entspricht der Vorfaktor δ^* einer numerische Konstanten, mit der die Parameter δ der beiden Stabilisierungen multipliziert werden. Dabei berechnet sich der Parameter δ für das *Upwinding* wie in Unterabschnitt 3.1.1 angegeben, bei der Stromliniendiffusionsmethode ist δ definiert als:

$$\delta = \frac{\delta^* h 2^{\frac{Re_{loc}}{1+Re_{loc}}}}{\|u\|_{L^\infty}},$$

mit Re_{loc} =lokaler Reynoldszahl.

Zuerst betrachten wir eine regelmäßige Triangulierung des Gebietes, das dem Benchmark Problem aus Abschnitt 5.2 entspricht. Wir geben immer als Ordnung den Vergleich der beiden feinsten Level an.

Tabelle 3.1: Fehler für Stromliniendiffusion, $Re = 10$

	$\delta^*=1$			$\delta^*=0.5$			$\delta^*=0.1$		
Level	$L^2(u)$	$H^1(u)$	$L^2(p)$	$L^2(u)$	$H^1(u)$	$L^2(p)$	$L^2(u)$	$H^1(u)$	$L^2(p)$
2	2.47-2	2.02-1	8.41-2	2.98-2	2.16-1	9.96-2	3.55-2	2.35-1	1.16-1
3	9.02-3	1.15-1	3.61-2	9.84-3	1.20-1	3.85-2	1.06-2	1.24-1	4.08-2
4	2.65-3	6.20-2	1.34-2	2.74-3	6.29-2	1.37-2	2.84-3	6.37-2	1.40-2
<i>Ord.</i>	1.76	0.90	1.42	1.84	0.93	1.49	1.90	0.96	1.54

Tabelle 3.2: Fehler für *Upwinding*, $Re = 10$

Level	$\delta^*=1$			$\delta^*=0.5$			$\delta^*=0.1$		
	$L^2(u)$	$H^1(u)$	$L^2(p)$	$L^2(u)$	$H^1(u)$	$L^2(p)$	$L^2(u)$	$H^1(u)$	$L^2(p)$
2	3.78-2	2.41-1	1.18-1	3.82-2	2.41-1	1.21-1	3.75-2	2.41-1	1.22-1
3	1.09-2	1.26-1	4.01-2	1.09-2	1.28-1	4.09-2	1.07-2	1.26-1	4.13-2
4	2.90-3	6.39-2	1.37-2	2.82-3	6.39-2	1.41-2	2.86-3	6.40-2	1.41-2
<i>Ord.</i>	1.91	0.97	1.54	1.95	1.00	1.53	1.90	0.97	1.55

Bei diesem einfachen Problem sehen wir zwischen beiden Stabilisierungen kaum Unterschiede, das *Upwinding* zeigt jedoch ein stabileres Verhalten gegenüber der Wahl von δ^* .

Als nächstes betrachten wir eine höhere Reynoldszahl, bei gleichem Gitter.

Tabelle 3.3: Fehler für Stromliniendiffusion, $Re = 250$

Level	$\delta^*=1$			$\delta^*=0.5$			$\delta^*=0.1$		
	$L^2(u)$	$H^1(u)$	$L^2(p)$	$L^2(u)$	$H^1(u)$	$L^2(p)$	$L^2(u)$	$H^1(u)$	$L^2(p)$
2	1.31-2	1.61-1	4.88-2	2.14-2	1.57-1	2.84-2	1.59-2	1.85-1	6.05-2
3	2.06-3	8.45-2	1.28-2	2.36-3	8.69-2	1.68-2	7.17-3	1.03-1	3.29-2
4	1.17-3	4.43-2	8.27-3	6.09-4	4.76-2	1.03-2	2.25-3	5.75-2	1.30-2
<i>Ord.</i>	0.86	0.93	0.63	1.96	0.86	0.70	1.67	0.85	1.33

Tabelle 3.4: Fehler für *Upwinding*, $Re = 250$

Level	$\delta^*=1$			$\delta^*=0.5$			$\delta^*=0.1$		
	$L^2(u)$	$H^1(u)$	$L^2(p)$	$L^2(u)$	$H^1(u)$	$L^2(p)$	$L^2(u)$	$H^1(u)$	$L^2(p)$
2	5.57-2	2.81-1	7.79-2	5.03-2	2.63-1	8.02-2	4.09-2	2.44-1	8.54-2
3	2.62-2	1.44-1	4.58-2	2.37-2	1.32-1	4.09-2	1.26-2	1.26-1	3.02-2
4	9.81-3	6.91-2	1.87-2	8.07-3	6.70-2	1.51-2	5.40-3	6.39-2	1.10-2
<i>Ord.</i>	1.41	1.05	1.37	1.55	0.98	1.43	1.22	0.97	1.45

Hier läßt sich bereits eine Tendenz ersehen, die Stromliniendiffusion liefert gute Fehler, ist aber sehr sensibel gegenüber der Wahl von δ^* , das *Upwinding* ist stabiler, dafür wie erwartet mit einer schlechteren Fehlerordnung.

Bei noch höherer Reynoldszahl konvergiert das mit der Stromliniendiffusionsmethode stabilisierte Verfahren nicht mehr, bzw. der Bereich eines optimalen δ^* wird so klein, daß man umfangreiche Testrechnungen mit verschiedenen Parametern durchführen muß,

um überhaupt Konvergenz zu erzielen. Anders verhält sich hier das *Upwinding*, wie aus der folgenden Tabelle ersichtlich wird ($\delta^* = 1$).

Tabelle 3.5: Fehler für *Upwinding*, $Re = 500$

Level	$L^2(u)$	$H^1(u)$	$L^2(p)$
2	9.02-2	4.09-1	1.25
3	7.71-2	3.17-1	6.05-1
4	4.03-2	1.88-1	2.64-1
<i>Ord.</i>	0.95	0.75	1.19

Die Fehlerordnung des Verfahrens wird zwar schlecht in diesem Fall, dafür erzielt man wenigstens Konvergenz ohne durch Parameterstudien einen optimalen Bereich für δ^* abklären zu müssen.

Der nächste Test untersucht die Abhängigkeit der Fehlerordnung bezüglich eines hohen *aspect ratio*, das Grobgitter besteht aus 256 Elementen.

Tabelle 3.6: Fehler für *Upwinding*, $Re = 10$, *aspect ratio*=12

Level	$\delta^*=1$			$\delta^*=0.1$		
	$L^2(u)$	$H^1(u)$	$L^2(p)$	$L^2(u)$	$H^1(u)$	$L^2(p)$
2	2.58-1	4.90-1	5.97-1	2.58-1	4.91-1	5.97-1
3	1.14-1	3.26-1	2.74-1	1.14-1	3.27-1	2.74-1
4	3.54-2	1.82-1	9.14-2	3.54-2	1.82-1	9.14-1
<i>Ord.</i>	1.68	0.84	1.58	1.68	0.84	1.58

Tabelle 3.7: Fehler für *Stromliniendiffusion*, $Re = 10$, *aspect ratio*=12

Level	$\delta^*=1$			$\delta^*=0.1$		
	$L^2(u)$	$H^1(u)$	$L^2(p)$	$L^2(u)$	$H^1(u)$	$L^2(p)$
2	2.45-1	4.67-1	5.66-1	2.57-1	4.88-1	5.91-1
3	1.10-1	3.15-1	2.76-1	1.11-1	3.25-1	2.71-1
4	3.49-2	1.79-1	8.96-2	3.54-2	1.82-1	9.06-1
<i>Ord.</i>	1.65	0.81	1.62	1.64	0.83	1.58

Bemerkung 3.4. Die Tatsache, daß wir keine besseren Ordnungen erhalten liegt mit daran, daß wir schlichtweg nicht fein genug rechnen können. In obigen Testrechnungen besteht das feinste Gitter bereits aus 131072 Elementen. Mit einem Speicherbedarf von

ungefähr 1 Kilobyte pro Zelle und einem Arbeitsspeicher von 1 Gigabyte, können wir nicht nocheinmal verfeinern. Vergleiche mit entsprechenden Tests in 2D zeigen aber, daß die Ordnungen bei vergleichbarer Ortsschrittweite ähnlichen Charakter haben.

Bei einem an die Geometrie angepaßten und damit deformierten Gitter erhalten wir für beide Diskretisierungen im Falle der Konvergenz ähnliche Ergebnisse, wie wir aus folgender Tabelle ablesen können. Das zugrunde liegende Gitter stammt von dem 3D Benchmark Test aus Abschnitt 5.2.

Tabelle 3.8: Fehler für $\delta^* = 1$, $Re = 250$

Level	<i>Upwinding</i>			<i>Stromliniendiffusion</i>		
	$L^2(u)$	$H^1(u)$	$L^2(p)$	$L^2(u)$	$H^1(u)$	$L^2(p)$
2	1.69-1	3.82-1	2.29-1	7.70-2	2.81-1	1.85-1
3	8.60-2	2.53-1	9.63-2	3.68-2	1.15-1	1.23-1
4	3.64-2	1.32-1	4.23-2	1.82-2	8.32-2	6.51-2
<i>Ord.</i>	1.24	0.93	1.18	1.01	0.46	0.91

Beide Verfahren liefern als Fehlerordnung nur noch h , der absolute Wert des Fehlers $L^2(u)$ ist bei der Stromliniendiffusionsmethode zwar besser, beim $H^1(u)$ -Fehler ist aber kaum ein Unterschied festzustellen, der Druckfehler ist beim *Upwinding* sogar besser.

Unser Fazit dieses Vergleiches fällt eher positiv für das *Upwinding* aus, die Stromliniendiffusionsmethode scheint zu Instabilitäten zu neigen. Diese Vermutung wird im instationären Fall bestätigt, so daß wir bei allen folgenden Testrechnungen auf diese Methode verzichten und mit dem *Upwinding* arbeiten.

3.2. Die instationären Navier–Stokes–Gleichungen

Wir wollen in diesem Paragraphen mit Hilfe des im letzten Abschnitt entwickelten stationären Lösers ein robustes Verfahren zur Lösung der instationären Navier–Stokes–Gleichungen in der Form

$$\begin{aligned} \mathbf{u}_t - \nu \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p &= \mathbf{f} \quad , \quad \nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \times (0, T) \text{ ,} \\ \mathbf{u} &= \mathbf{g} \quad \text{auf } \partial\Omega \quad , \quad \mathbf{u}|_{t=0} = \mathbf{u}_0 \end{aligned}$$

entwickeln. Im Gegensatz zu dem Charakteristikenverfahren von [37] behandeln wir Orts- und Zeitableitungen separat und können daher eines der üblichen Zeitdiskretisierungsschemata für Systeme von gewöhnlichen Differentialgleichungen verwenden. Bei der Auswahl dieser Schemata ist die Steifheit der Probleme zu berücksichtigen, die *A-stabile* Methoden erfordert.

Im folgenden zeigen wir einige Eigenschaften der verwendeten Verfahren, die schon von der numerischen Analyse gewöhnlicher Differentialgleichungen hinreichend bekannt sind ([19],[30]). Dazu beschränken wir uns zuerst einmal auf die übliche skalare Testgleichung

$$\dot{x}(t) + \lambda x(t) = 0 \quad , \quad t \geq 0 \quad , \quad (3.39)$$

mit $\lambda \in \mathbf{C}$, $Re\lambda \geq 0$.

Wenden wir ein Zeitdiskretisierungsschema mit konstanter Zeitschrittweite Δt auf diese Gleichung an, so erhalten wir eine Sequenz von Werten $x_n \sim x(t_n)$ mit $t_n := n\Delta t$. Das Verhalten dieser Schemata für $t \rightarrow \infty$ in Abhängigkeit von dem Parameter λ wird üblicherweise durch den *Verstärkungsfaktor* $\omega = \omega(\lambda\Delta t)$ gekennzeichnet. Für Einschrittverfahren, auf die wir uns beschränken werden, gilt dann $x_n = \omega^n x_0$.

Man kann nun mittels der Terme ω die folgenden wünschenswerten Eigenschaften von Zeitschrittverfahren formulieren ([38]):

- 1) $|\omega(\lambda\Delta t)| \leq 1$ (lokale Stabilität: *A-Stabilität*)
- 2) $\lim_{Re\lambda \rightarrow \infty} |\omega(\lambda\Delta t)| \leq 1 - O(\Delta t)$ (globale Regularität)
- 3) $\lim_{Re\lambda \rightarrow \infty} |\omega(\lambda\Delta t)| \leq 1 - \delta < 1$ (Glättungseigenschaft: *starke A-Stabilität*)
- 4) $|\omega(\lambda\Delta t)| \sim 1$ für $Re\lambda = 0$ (nicht dissipativ)

Die Einschrittverfahren schreiben wir so, daß sie auf das folgende allgemeine lineare System angewendet werden

$$u_t + A(t)u = f(t) \quad . \quad (3.40)$$

Wir benutzen dazu die Bezeichnungen $A_n = A(t_n)$ und $f_n = f(t_n)$. Die Approximationseigenschaften können dann wieder mit Hilfe der Faktoren $\omega(\lambda\Delta t)$, λ Eigenwert von $A(t)$, beschrieben werden. Die Hoffnung ist nun, diese Ergebnisse auch auf die nichtlinearen Navier–Stokes–Gleichungen übertragen zu können.

Zuerst wollen wir uns mit den *Einschritt- ϑ -Schemata* befassen:

I) Die **Einschritt- ϑ -Verfahren**:

Angewandt auf das obige lineare System lauten diese Verfahren

$$[I + \vartheta \Delta t A_{n+1}] u_{n+1} = [I - (1 - \vartheta) \Delta t A_n] u_n + \vartheta \Delta t f_{n+1} + (1 - \vartheta) \Delta t f_n \quad (3.41)$$

mit *Verstärkungsfaktoren*

$$\omega(z) = \frac{1 - (1 - \vartheta)z}{(1 + \vartheta)z}. \quad (3.42)$$

Den Fall $\vartheta = 0$, der gerade das explizite Eulerverfahren darstellt, schließen wir aus, da er nur bedingte Stabilität liefert ($\Delta t \leq 1/\lambda$). Die beiden für uns interessanten Fälle sind $\vartheta = 1$ (implizites Eulerverfahren) und $\vartheta = 1/2$ (Crank–Nicolson–Verfahren).

Das implizite Eulerverfahren ist stark *A-stabil* mit $|\omega(z)| \rightarrow 0$ für $Re\lambda \rightarrow \infty$, neigt aber dazu, sehr stark zu dämpfen, d.h. $|\omega(i\Delta t)| < 1$ (beispielsweise $|\omega| = 0.995$ für $\Delta t = 0.1$). Dieses sehr robuste Verfahren ist auch nur von erster Ordnung–Genauigkeit. Von zweiter Ordnung dagegen ist das Crank–Nicolson–Verfahren, das aber dafür nur geringe Dämpfungseigenschaften besitzt ($|\omega(z)| \rightarrow 1$ für $Re\lambda \rightarrow \infty$) und auch nur *A-stabil* ist. Dafür werden freie Oszillationen sehr gut wiedergegeben ($|\omega(i\Delta t)| = 1$).

Beide Verfahren sind also mit Nachteilen behaftet: Das implizite Eulerverfahren ist zwar sehr robust, doch ist es zu ungenau und zu stark dämpfend, während das Crank–Nicolson–Verfahren zwar genauer arbeitet, dafür aber leicht zu Instabilitäten neigt.

Ein anderes Schema, das die Vorteile dieser beiden Verfahren miteinander verknüpft, ist das *Fractional-step- ϑ -Schema*, wie in [34] beschrieben.

II) Das **Fractional-step- ϑ -Verfahren**:

Seien Konstanten $\vartheta \in (0, 1)$, $\vartheta' = 1 - 2\vartheta$, und $\alpha \in [0, 1]$, $\beta = 1 - \alpha$, gegeben; ein Zeitschritt $t_n \rightarrow t_{n+1}$ spaltet sich dann in die folgenden drei Teilschritte auf ([38])

(wobei die Zeitschritte $t_n \rightarrow t_{n+1}$ dann gerade dreimal größer wie oben sind):

$$\begin{aligned}
[I + \alpha\vartheta\Delta t A_{n+\vartheta}] u_{n+\vartheta} &= [I - \beta\vartheta\Delta t A_n] u_n + \vartheta\Delta t f_n \\
[I + \beta\vartheta'\Delta t A_{n+1-\vartheta}] u_{n+1-\vartheta} &= [I - \alpha\vartheta'\Delta t A_{n+\vartheta}] u_{n+\vartheta} + \vartheta'\Delta t f_{n+1-\vartheta} \\
[I + \alpha\vartheta\Delta t A_{n+1}] u_{n+1} &= [I - \beta\vartheta\Delta t A_{n+1-\vartheta}] u_{n+1-\vartheta} + \vartheta\Delta t f_{n+1}
\end{aligned} \tag{3.43}$$

mit

$$\omega(z) = \frac{(1 - \beta\vartheta z)^2 (1 - \alpha\vartheta' z)}{(1 + \alpha\vartheta z)^2 (1 + \beta\vartheta' z)}. \tag{3.44}$$

Wählt man $\alpha = \frac{1-2\vartheta}{1-\vartheta}$, dann sind die Koeffizientenmatrizen in allen drei Teilschritten die gleichen. Für die Parameterwahl $\vartheta = 1 - \frac{\sqrt{2}}{2}$ ist das Verfahren zudem von zweiter Ordnung–Genauigkeit. Außerdem sehen wir, daß das Verfahren stark *A-stabil* ist mit $\lim_{Re\lambda \rightarrow \infty} |\omega(z)| = \frac{\beta}{\alpha} \sim 0.7$ und freie Oszillationen gut wiedergegeben werden ($|\omega(i\Delta t)| \sim 0.9998$ für $\Delta t = 0.8$). In den folgenden Tests wird sich auch numerisch bestätigen, daß dieses Verfahren gegenüber den beiden vorherigen zu bevorzugen ist, da es ohne größeren numerischen Aufwand gleichzeitig genau und robust arbeitet.

Wenden wir diese ϑ -Verfahren auf unsere Fixpunktiteration zur Lösung der Navier–Stokes–Gleichungen an, so erhalten wir in jedem Zeitschritt Gleichungssysteme vom Typ

$$\begin{aligned}
&\begin{bmatrix} M_h + \alpha\vartheta\Delta t(\nu A_h + \bar{N}_h(U_{h,n+1})) & B_h \\ B_h^T & 0 \end{bmatrix} \begin{bmatrix} U_{h,n+1} \\ P_{h,n+1} \end{bmatrix} = \\
&\begin{bmatrix} M_h + \alpha\vartheta\Delta t(\nu A_h + \bar{N}_h(U_{h,n})) & B_h \\ B_h^T & 0 \end{bmatrix} \begin{bmatrix} U_{h,n} \\ P_{h,n} \end{bmatrix} + \begin{bmatrix} \vartheta\Delta t F_{h,n+1} + \vartheta'\Delta t F_{h,n} \\ 0 \end{bmatrix}
\end{aligned}$$

mit gewissen Konstanten $\alpha, \beta, \vartheta, \vartheta'$, die in Abhängigkeit von der Wahl des Verfahrens bestimmt sind, und einer Diskretisierung $\bar{N}_h(\cdot)$ des Konvektionsterms entsprechen. Da alle Verfahren impliziten Charakter haben, müssen wir daher in jedem Zeitschritt ein nichtlineares Teilproblem

$$\begin{bmatrix} M_h + c\nu\Delta t A_h + c\Delta t \bar{N}_h(X_h) & B_h \\ B_h^T & 0 \end{bmatrix} \begin{bmatrix} X_h \\ 0 \end{bmatrix} = \text{rechte Seite} \quad (3.45)$$

lösen. Für diese Problemklasse bietet es sich natürlich an, unseren Löser für die stationären Navier–Stokes–Gleichungen zu verwenden. Die Hauptkomponenten dieses gekoppelten Ansatzes sind damit :

–*Fractional-step- ϑ -Schema* für die Zeitdiskretisierung

–*Fixpunkt-Defektkorrektur-Methode* für die nichtlinearen Teilprobleme

–*Upwind-, bzw. Stromliniendiffusion-Diskretisierung* für den Konvektionsterm

– Mehrgittertechniken für die resultierenden linearen, nichtsymmetrischen Gleichungssysteme

Unsere Erfahrungen mit diesem Ansatz sind im stationären Fall sehr gut, d.h. auch bei instationären Rechnungen, die in einen stationären Limes konvergieren. Probleme haben wir dagegen im voll instationären Fall, d.h. in Situationen, bei denen die Lösung instationär, z.B. periodisch wird (siehe z.B. die Abschnitte 5.1 und 5.2). In solchen Fällen verhält sich diese Methode sehr ineffizient, da der Aufwand in jedem Zeitschritt sehr groß ist und außerdem die Konvergenz des Mehrgitters schlechter wird bei zu kleinen Zeitschritten. Dies kommt von dem impliziten Charakter des gekoppelten Ansatzes sowie des verwendeten Vanka–Glätters (siehe Abschnitt 2.2). Hier schafft auch eine adaptive Wahl des Zeitschrittes nur bedingt Abhilfe. Aus diesen Gründen stellen wir im nächsten Kapitel einen anderen Zugang für die Behandlung der instationären Navier–Stokes–Gleichungen vor.

4. Ein Operator–Splitting Ansatz zur Lösung der instationären Navier–Stokes Gleichungen

Der gekoppelte Ansatz, d.h. Druck und Geschwindigkeit werden gleichzeitig behandelt, zeigt sich im voll instationären Fall als relativ unbrauchbar, allein deswegen, da in jedem Zeitschritt eine stationäre Gleichung vom Oseen Typ gelöst werden muß. Wir verwenden diesen Ansatz daher nur im stationären Fall, auch bei instationären Rechnungen, die in einen stationären Limes konvergieren, da hier der implizite Charakter es erlaubt, mit sehr großen Zeitschritten zu rechnen. Im voll instationären Fall, d.h. auch die zu erwartende Lösung wird instationär, ist dies aber kein Vorteil mehr, da ein zu großer Zeitschritt eine eventuelle Periode gar nicht erfassen kann. Auch eine Zeitschrittkontrolle kann hier nur wenig verbessern, da man sich bei zu kleinen Zeitschritten wieder im Mehrgitterverfahren “totiteriert“, und außerdem das Konvergenzverhalten sehr schlecht wird. Daher dieser andere Zugang des Splittens, der in zwei Raumdimensionen von Turek entwickelt wurde (siehe dazu auch [21]). Da die Elemente des Finite–Elemente–Codes in 2D und 3D analog entwickelt wurden, ließen sich die Strategien des Splittens, bzw. der Projektionsmethode in 2D mit den entsprechenden Änderungen auch in 3D verwirklichen. Ein wichtiger Gesichtspunkt ist hierbei das verwendete nichtkonforme Stokeseselement \hat{Q}_1 , das eine besonders effiziente Algorithmik und Implementierung zuläßt.

In den nächsten Abschnitten geben wir eine kurze Zusammenfassung der Projektionsmethode, wie sie von Turek in [49] beschrieben ist. Erste Ergebnisse für den 3D Fall sind in [46] erschienen. Der Vollständigkeit halber müssen wir aber bemerken, daß in drei Raumdimensionen eine voll instationäre Strömung zu simulieren eine Aufgabe ist, die an die Grenze der uns verfügbaren Mittel geht. Abhilfe wird hier in Zukunft der Einsatz massiver Parallelisierung, sowie adaptiver Techniken bringen, so daß in Zukunft auch voll instationäre Simulationen in angemessener Zeit durchgeführt werden können.

4.1. Die diskrete Projektionsmethode

Wie im vorigen Abschnitt 3.2 beschrieben, diskretisieren wir die Zeitableitung in den Navier–Stokes–Gleichungen mit einem der vorgestellten Schemata. Dies führt zusammen mit der Ortsdiskretisierung mit unseren nichtkonformen Finiten Elementen auf das folgende algebraische System:

Für gegebenes $\mathbf{u}(t_n)$, rechter Seite $\mathbf{g}(t_{n+1}, t_n)$ und einem Zeitschritt $k = t_{n+1} - t_n$, löse für $\mathbf{u} = \mathbf{u}(t_{n+1})$ und $p = p(t_{n+1})$ das folgende nichtlineare Problem:

$$S(\mathbf{u}) + kBp = \mathbf{g}, \quad B^T \mathbf{u} = 0, \quad (4.1)$$

mit dem Operator S und der rechten Seite \mathbf{g} definiert durch

$$S(\mathbf{u}) = [M - \vartheta\nu kA + \vartheta kN(\mathbf{u})]\mathbf{u} \quad (4.2)$$

$$\mathbf{g} = [M - \vartheta_1 k(-\nu A + N(\mathbf{u}(t_n)))]\mathbf{u}(t_n) + \vartheta_2 \mathbf{f}(t_{n+1}) + \vartheta_3 k\mathbf{f}(t_n). \quad (4.3)$$

Wir verwenden wieder die gleichen Bezeichnungen wie im letzten Kapitel, d.h. M bezeichne die *Masse* Matrix, A die *Laplace* Matrix und $N(\cdot)$ die Advektionsmatrix, die den nichtlinearen konvektiven Anteil beinhalte (und zusätzliche Stabilisierung bei dominantem Transport, siehe Abschnitt 3.1). Schließlich bezeichne B die *Gradienten* Matrix und deren Transponierte B^T die *Divergenz* Matrix.

Bei der numerischen Behandlung des diskreten Problems (4.1) stellen sich die beiden Hauptprobleme:

- Der nichtlineare advektive Anteil erfordert spezielle iterative quasi–Newton Techniken
- Die Indefinitheit hervorgerufen durch die Geschwindigkeits–Druck Kopplung erfordert besondere Sorgfalt.

Diese Probleme werden üblicherweise getrennt behandelt. In Kapitel 3.2 haben wir dabei die Nichtlinearität in einer äußeren Fixpunktiteration behandelt und dann die resultierenden linearen indefiniten Probleme mit einem gekoppelten Ansatz gelöst. Bei

dem Projektionsansatz gehen wir einen anderen Weg:

Wir splitten das gekoppelte Problem und erhalten definite Probleme sowohl in \mathbf{u} (Burgers Gleichung), als auch in p (lineare Poisson Gleichungen für den Druck). Danach behandeln wir die nichtlinearen Probleme in \mathbf{u} mit einer geeigneten nichtlinearen Iteration oder Linearisierungstechniken.

In der folgenden Diskussion wählen wir die gelumpfte Massematrix als Vorkonditionierer. Andere Möglichkeiten werden wir später angeben. Der (nichtlineare) diskrete Projektions Algorithmus läßt sich dann folgendermaßen schreiben, wobei der Parameter ϑ von der benutzten Zeitdiskretisierung abhängt, wie sie in Abschnitt 3.2 angegeben ist.

Instationäre nichtlineare Version

Gegeben sei $\mathbf{u}(t_n)$ und $p(t_n)$, $k = k(t_{n+1})$, und

$$S(\cdot) := M - \vartheta \nu k A + \vartheta k N(\cdot).$$

Schritt 1: Löse für $\tilde{\mathbf{u}}$ die Transport–Diffusions Gleichung

$$S\tilde{\mathbf{u}} = \mathbf{g}(t_{n+1}, t_n) - kBp(t_n). \quad (4.4)$$

Schritt 2: Löse für q die diskrete Poisson Gleichung (mit $P = B^T M_l^{-1} B$)

$$Pq = \frac{1}{k} B^T \tilde{\mathbf{u}}. \quad (4.5)$$

Schritt 3: Update $p(t_{n+1}) := p(t_n) + \alpha q$, mit $\alpha > 0$, und berechne

$$\mathbf{u}(t_{n+1}) = \tilde{\mathbf{u}} - k M_l^{-1} B q. \quad (4.6)$$

Die Iteration (4.4) – (4.6) kann betrachtet werden als nichtlineares Iterationsschema für die Lösung des verallgemeinerten stationären Navier–Stokes–Problems: Zuerst wird die Geschwindigkeit \mathbf{u} und der Druck p entkoppelt als eine äußere Iteration, während die nichtlineare Transport–Diffusions Gleichung in einer inneren Iteration gelöst wird. Dieser Zugang ist wohlbekannt und ähnlich einer bestimmten Klasse von Iterations-

schemata, nämlich den “klassischen“ Projektionsverfahren. Da die Matrix $B^T M_l^{-1} B$ korrespondiert mit einer gemischten Diskretisierung des Laplace Operators, ist diese Methode ein diskretes Analogon zu den Verfahren von Chorin [14] (für $p_0 = 0$) und von Van Kan [55] (für $p_0 = p(t_n)$). Der Hauptunterschied ist, daß wir erst die Gleichungen diskretisieren, wie in Abschnitt 3.1 und dann führen wir die Projektionsschritte (4.4) – (4.6) durch. Dieser Unterschied ist deshalb so wichtig, da wir mit unseren nichtkonformen Elementen die Matrix $P = B^T M_l^{-1} B$ erheblich verkleinern und schnelle Löser einsetzen können. Daher ist auch unsere Behandlung der Randwerte für den Druck p , was die Hauptschwierigkeit bei den kontinuierlichen Projektionsverfahren darstellt, etwas anders. Wenn wir die selben (Dirichlet-) Randbedingungen sowohl für die Matrix M_l , als auch für die volle Matrix S nehmen, zeigen unsere numerischen Tests gute Ergebnisse d.h. keine signifikanten Randschichten für höhere Reynoldszahlen. Wichtig bei diesem Verfahren ist außerdem, daß $P = B^T M_l^{-1} B$ explizit angegeben werden kann für unser Elementepaar \hat{Q}_1/Q_0 .

4.2. Numerische Komplexität

Um die vorgestellten Methoden vergleichen zu können, betrachten wir als Maß für den numerischen Aufwand die Kosten für eine Matrix–Vektor Multiplikation (MV), da dies den Hauptanteil an Rechenzeit bildet. Bei der gekoppelten Methode mit Matrix A ,

$$A = \begin{bmatrix} S & kB \\ B^T & 0 \end{bmatrix}, \quad (4.7)$$

haben wir für drei Stokes Elemente die Kosten in Anzahl der Elemente (NEL) ausgedrückt. Die Elemente sind:

- a.) Q_1/Q_1 lineare Geschwindigkeit und Druck Approximation auf den Ecken ([20],[28])
- b.) Q_1/Q_0 lineare Geschwindigkeit auf den Ecken, konstanter Druck in den Zellen ([20],[28])
- c.) \hat{Q}_1/Q_0 nichtkonforme rotiert lineare Geschwindigkeit, konstanter Druck in den Zellen (1.3)

Die folgenden Komplexitäts Abschätzungen sind für Tensor Produkt Gitter, auf allge-

meinen Gittern gelten ähnliche Ergebnisse. Wir stellen hier nur den für uns interessanten 3D Fall dar, für den 2D Fall verweisen wir auf Turek in [49].

Fall Q_1/Q_1 :

d.o.f. (jede Geschwindigkeitskomponente) = *d.o.f.* (Druck) \approx Zahl der Elemente = NEL

stencil(S) = 27, stencil(B) = stencil(S)

- 1 MV: $3 \cdot (27 (\sim Su_i) + 27 (\sim B_i p)) + 3 \cdot 27 (\sim B_i^T u_i) \approx \mathbf{243 \text{ NEL}}$

Fall Q_1/O_0 :

d.o.f. (jede Geschwindigkeitskomponente) \approx *d.o.f.* (Druck) = NEL

stencil(S) = 27, stencil(B) = 8

- 1 MV in 3D: $3 \cdot (27 (\sim Su_i) + 8 (\sim B_i p)) + 3 \cdot 8 (\sim B_i^T u_i) \approx \mathbf{129 \text{ NEL}}$

Fall \hat{Q}_1/Q_0 :

d.o.f. (jede Geschwindigkeitskomponente) ≈ 3 *d.o.f.* (Druck), *d.o.f.*(Druck) = NEL

stencil(S) = 11, stencil(B) = 6

- 1 MV in 3D: $3 \cdot (3 \cdot 11 (\sim Su_i) + 6 (\sim B_i p)) + 3 \cdot 6 (\sim B_i^T u_i) \approx \mathbf{135 \text{ NEL}}$

Wir notieren, daß die beiden Elementpaare mit konstantem Druck nur die Hälfte des Aufwands benötigen, das einzige Element das generell ohne zusätzliche Stabilisierung stabil bei der Druckapproximation ist, ist das nichtkonforme Elementepaar \hat{Q}_1/Q_0 .

Wir wollen nun die gleiche Betrachtung für die diskrete Projektionsmethode anstellen. Wie im vorigen Abschnitt angesprochen, erhoffen wir uns für den voll instationären Fall, d.h. große Reynoldszahlen und damit kleine Zeitschritte, mit dem Projektionsverfahren eine bessere Effizienz als mit den voll gekoppelten Ansätzen. Daher setzen wir $k \ll 1$ voraus. Da S sich verhält wie eine Störung der Massematrix mit der Ordnung k , können wir sehr robuste und effiziente Löser einsetzen, wenn nötig mit einer nichtlinearen Standard Iteration. Auch auf sehr anisotropen Gittern haben wir mit modifizierten Glättern/Vorkonditionierern von ILU-Typ (siehe [56]) mit Umsortierung [15], geeignete Löser zur Hand.

Der Rechenzeit schluckende Teil ist der zeitinvariante Anteil mit der Matrix $B^T M_i^{-1} B$

(M_l Diagonalmatrix). Mögliche Lösungsstrategien sind wieder Mehrgitter- oder konjugierte Gradientenverfahren mit diagonalem Glätten bzw. Vorkonditionieren, da die Iterationsmatrix nur implizit gegeben ist. Wir betrachten nun die Kosten für eine Matrix-Vektor Multiplikation (MV), als natürliches Maß für die implizit gegebene Matrix:

$$\sum_{i=1}^{d=3} B_i^T M_l^{-1} B_i. \quad (4.8)$$

Fall a : Q1/Q1

$$\Rightarrow 1 \text{ MV: } 3 \cdot (27 (\sim B_i^T) + 1 (\sim M_l^{-1}) + 27 (\sim B_i)) \approx 165 \text{ NEL}$$

Fall b : Q1/Q0

$$\Rightarrow 1 \text{ MV: } 3 \cdot (8 (\sim B_i^T) + 1 (\sim M_l^{-1}) + 8 (\sim B_i)) \approx 51 \text{ NEL}$$

Fall c : $\tilde{Q}1/Q0$

$$\Rightarrow 1 \text{ MV: } 3 \cdot (6 (\sim B_i^T) + 3 (\sim M_l^{-1}) + 6 (\sim B_i)) \approx 45 \text{ NEL}$$

Dies stellt eine Verbesserung dar, aber ist noch nicht voll befriedigend, da ausschließlich diagonale Vorkonditionierer/Glätter benutzt werden können, da die Matrix nur in Produktform gegeben ist. Bei anisotropen Gittern kann es dann zu Problemen mit der Robustheit kommen. Daher ist der letzte Schritt, das Matrixprodukt P nur einmal zu entwickeln und das Ergebnis in kompakter Form abzuspeichern als:

$$P = \sum_{i=1}^{d=3} B_i^T M_l^{-1} B_i. \quad (4.9)$$

Damit ergibt sich der folgende Gewinn an Speicher und numerischer Effizienz für eine Matrix-Vektor Multiplikation mit P :

Lemma 4.1. *(Speicher und numerischer Aufwand für Projektionsmethode)*

Wir erhalten die folgenden Größen der Matrizen P für den Druck und damit die (asymptotischen) Kosten für eine Matrix-Vektor Multiplikation für kleine Zeitschritte k . Als Einheiten verwenden wir die Anzahl der Gitterzellen (NEL), die Verbesserung ist auf den gekoppelten Lösungsalgorithmus bezogen:

Für Q1/Q1: 125 Nel (Faktor 2).

Für Q1/Q0: 27 Nel (Faktor 5).

Für $\tilde{Q}1/Q0$: 7 Nel (Faktor 19).

Für das Elemente Paar Q1/Q1 erhalten wir einen sehr großen Stern, der einer Diskretisierung des Laplace-Operators mit quadratischen Finiten Elementen entspricht. Daher scheinen kontinuierliche Projektionsverfahren, die zu kleineren Sternen mit 9, bzw. 27 Einträgen führen besser geeignet zu sein, wenn man allerdings die Randschichten des Drucks in Kauf nimmt. Ähnliches gilt für Q1/Q0, zusätzlich benötigen beide Paare Stabilitätstechniken (siehe [28], [3]). Das einzige immer stabile Elementepaar ist unser nichtkonformes, das auch den geringsten numerischen Aufwand bedarf. Man erhält bei kartesischen Gittern einen 7 Punkte Stern, der einer Diskretisierung des Laplace-Operators mit Finiten Differenzen entspricht. Man stelle sich nun eine dreidimensionale "Matrix" vor, deren Mittelebene folgende Einträge hat.

$$\textit{Stencil}(P) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 6 & -1 \\ 0 & -1 & 0 \end{bmatrix}. \quad (4.10)$$

Diese 7 Punkte Diskretisierung stellt die einfachste stabile Diskretisierung dar. In diesem Fall lassen sich dann auch bekannte ILU-Techniken (siehe [56]) anwenden.

Es bleibt die Frage nach der totalen Effizienz dieser Splitting Technik verglichen mit der voll gekoppelten Methode. Theoretisch haben wir einen Gewinn pro Iteration um mindestens den Faktor 19. Dafür haben wir nun ein Schema, das einen eher expliziteren Charakter hat, was zu einer Reduktion des Zeitschrittes führen wird.

Im Folgenden vergleichen wir die nötigen Zeitschritte und die verbrauchte Gesamtzeit der folgenden Kombinationen:

- 1) Voll gekoppelte Strategie (CC),
- 2) Projektionsverfahren (PP).

Wir untersuchen folgende Fragestellungen:

- a.) Unterschiede in Genauigkeit und Robustheit,
- b.) Unterschiede in Genauigkeit und Robustheit zwischen Crank–Nicolson, Backward Euler und Fractional step– θ –Diskretisierungen der Momenten Gleichung für \mathbf{u} ,
- c.) Einfluß der Behandlung der Nichtlinearität.

4.3. Numerischer Vergleich

Wir wollen in diesem Abschnitt beide Methoden –gekoppelten Ansatz versus diskrete Projektionsmethode– numerisch vergleichen, um die in den letzten Abschnitten gemachten Aussagen zu belegen. Die numerischen Tests beziehen sich dabei auf Genauigkeit und Effizienz

- Wir berechnen mit beiden Methoden den stationären Limes einer bekannten Lösung.
- Wir führen eine instationäre Rechnung mit hoher Reynoldszahl mit beiden Methoden durch bis Erreichen einer vorgegebener Endzeit T .

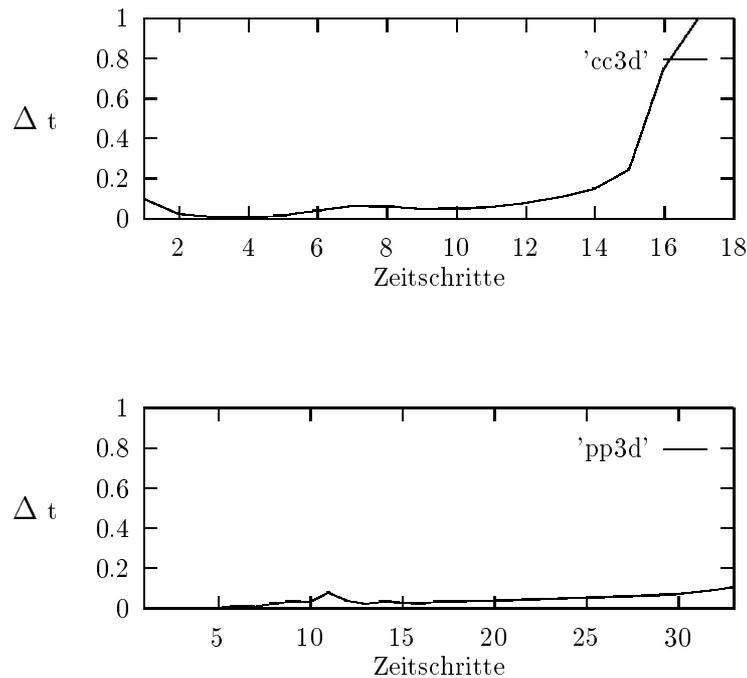
Als Beispiel dient uns hier wieder wie schon in Abschnitt 2.5 die Umströmung einer Platte.

Wir beginnen zuerst mit der Frage der Grundanforderung der unterschiedlichen Verfahren, d.h. beide Verfahren müssen dasselbe Ergebnis liefern im Falle einer stationären Lösung, die wir mit dem Lösungsverfahren aus Abschnitt 3.1 berechnet haben. Die Daten für diese Vergleiche sind wie folgt:

- Grobgrid: 87 Elemente, Rechengitter: 44544 Elemente
- Abmessungen des Gebietes: $(0; 6.5) \times (0; 5) \times (0; 5)$, die Platte hat eine Grundfläche von 0.4×0.4 , Anströmbereich: 1cm
- Viskosität: 1/10
- Einströmgeschwindigkeit: 1
- Neumann “do nothing“ Bedingungen am Ausströmrand

Als Referenzlösung für diesen Test dient uns dabei die stationäre Lösung, d.h. wir führen mit dem gekoppelten Ansatz **CC3D**, bzw. mit der diskreten Projektionsmethode **PP3D**, instationäre Rechnungen durch, bis die stationäre Lösung auf einen Fehler

von 5% erreicht ist. Bei beiden Methoden haben wir die gleichen Abbruchkriterien, die gleiche adaptive Zeitschrittkontrolle und wir starten jeweils mit Nulldaten. Das Zeitschrittverhalten beider Methoden ist in folgender Abbildung dargestellt. Das gekoppelte Verfahren benötigt 18 Zeitschritte, das Projektionsverfahren fast doppelt so viele, um in die stationäre Lösung zu konvergieren.



Wir haben hier den “bestmöglichen“ Fall für den gekoppelten Löser **CC3D**, die adaptive Zeitschrittkontrolle fährt den Zeitschritt bis auf 1 hoch, das Konvergenzverhalten und damit die Effizienz wird besser. Dies erkennt man an den Rechenzeiten für einen Zeitschritt: Zu Beginn der Rechnung benötigt das Verfahren 482 CPU Sekunden für einen Makroschritt (bei einem Zeitschritt von 0.0078s), gegen Ende nur noch 292 CPU Sekunden (bei einem Zeitschritt von 0.245). Dies bestätigt die Aussagen von Kapitel 3; durch den voll impliziten Charakter des Ansatzes entfaltet die Methode ihre Möglichkeiten bei einem großen Zeitschritt.

Ganz anders verhält sich dagegen die diskrete Projektionsmethode **PP3D**, der Zeitschritt bleibt relativ klein, daher braucht das Verfahren auch fast doppelt so viele Zeitschritte, um in den stationären Limes zu konvergieren, bzw. das gesetzte Abbruchkriterium zu erreichen. Bedingt durch den eher expliziten Charakter des Verfahrens wird die Effizienz schlechter bei ansteigenden Zeitschritten. Ein Vergleich der CPU

Zeiten belegt dies: 144 CPU Sekunden (Zeitschritt von 0.001) stehen 189 Sekunden (Zeitschritt von 0.2) gegenüber. Im Endeffekt haben beide Verfahren praktisch die gleiche Rechenzeit verbraucht, um das Abbruchkriterium zu erreichen, d.h. hier sehen wir noch keine spürbare Beschleunigung der Projektionsmethode.

Die nächste Frage ist die nach der Effizienz für instationäre Rechnungen bei hohen Reynoldszahlen. Dazu lösen wir das gleiche Problem wie oben, diesmal aber mit einer Reynoldszahl von 1000. Zudem verfeinern wir das Gitter noch einmal, so daß das Feinste nunmehr aus 356352 Elementen besteht. Nur mit dieser feinen Auflösung haben wir eine Chance, eine sogenannte Kármán'sche Wirbelstraße (siehe z.B. [53]) zu bekommen, die in der folgenden Abbildung und auf Seite 154 durch Darstellung von Druckisoflächen gezeigt wird.

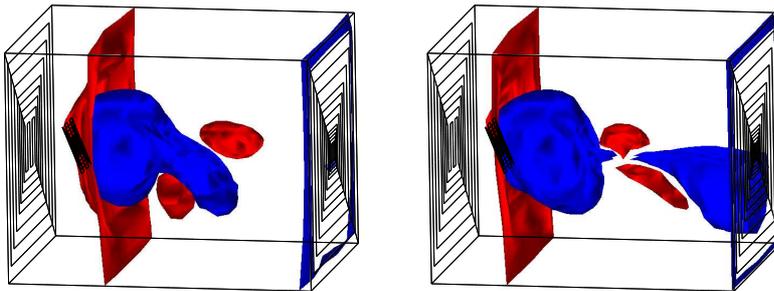


Abbildung 4.1: Druckisoflächen zu unterschiedlichen Zeitpunkten

Die Lösung wird bei beiden Verfahren instationär, d.h. zeigt ein periodisches Verhalten. Interessant sind die Aufwandsbetrachtungen beider Verfahren für eine gewisse Zeitspanne. Die Kriterien für den Vergleich sind dabei:

- Wir starten mit einer Stokes Lösung vom Level tiefer, d.h. die Lösung wird auf das feine Gitter prolongiert.
- Wir messen die Anzahl und Zeit der Zeitschritte, die die Verfahren benötigen, um von $T = 5$ auf $T = 10$ zu kommen.
- Bei beiden Verfahren verwenden wir die adaptive Zeitschrittkontrolle und soweit möglich die gleichen Abbruchkriterien.

Dies führt zu folgenden Ergebnissen:

Die voll gekoppelte Methode benötigt 39 Makrozeitschritte, mit einem mittleren Zeitschritt von 0.128s. Dabei liegt die Gesamtzeit des Verfahrens bei 222846 Sekunden. Im Gegensatz dazu benötigt die diskrete Projektionsmethode 58 Zeitschritte, mit einem mittleren Zeitschritt von 0.086s. Die Gesamtzeit ist dabei 93066 Sekunden. Das Verfahren liefert also einen speed up vom Faktor 2.4. Diese Ergebnisse sind vergleichbar mit denen von Turek in [49] für den 2D-Fall und zeigen eine große Effizienzsteigerung bei der Behandlung von instationären Problemen.

In dem nächsten Kapitel werden wir noch mehr numerische Vergleiche zu Fragen der Zeitdiskretisierung in Zusammenhang mit den Benchmark Resultaten in den Abschnitten 5.1 und 5.2 durchführen.

5. Numerische Testrechnungen und ihre graphische Präsentation

Wir wollen unsere numerischen Verfahren zuerst an zwei sogenannten “Benchmark“-Problemen testen, um eine Sicherheit darüber zu haben, daß die Ergebnisse für Probleme bei denen keine experimentellen oder gerechneten Daten vorliegen, die Realität wiedergeben. Zudem lassen sich bei Benchmark-Problemen auch die Güte der Löser, der Diskretisierung und der numerische Aufwand mit anderen Strömungscodes vergleichen. Bei den Benchmark Problemen handelt es sich um eine *3D-Driven-Cavity*, wir folgen dabei der Definition aus Deville (siehe [16]), das andere Beispiel ist ein aktuelles Problem, es handelt sich um die Umströmung eines Quaders, bzw. Zylinders (siehe [41]).

Nach diesen beiden Beispielen vertiefen wir die Frage der Randbedingungen, da dies bei dem Industrieprojekt mit der Firma Vaillant im nächsten Kapitel eine zentrale Rolle spielt. Wir betrachten dabei sowohl “künstliche“ Ränder, dies quasi gezwungenermaßen, um für das Mehrgitterverfahren genügend “grobe“ Grobgitter (bis zu maximal 1000 Zellen) zu haben, andererseits aber trotzdem komplexe Geometrie Strukturen erfassen zu können. Da diese Technik gute Ergebnisse bringt (qualitativ und quantitativ) ist der Schritt zu bewegten, d.h. in der Zeit veränderlichen Rändern nicht mehr weit, dies demonstrieren wir am Beispiel eines Rotorblattes und schaffen damit die Voraussetzungen zur Erweiterung der Verfahren auf die Behandlung von Vermischungsproblemen.

5.1. Die 3D-Driven-Cavity

Als erstes Problem behandeln wir eine *Driven-Cavity*, wie sie in [16] und [44] beschrieben ist. Stationäre Ergebnisse hierzu finden sich auch in [24] und [25].

Wie man in der folgenden Zeichnung sehen kann, haben wir eine Box mit den Maßen $(0;1) \times (0;1) \times (0;3)$, Dirichletrandwerte 0 überall, bis auf die obere Seite, dort ist die Geschwindigkeit 1 in x-Richtung. Als Reynoldszahl wird ein Wert von 3200 vorgegeben, wobei hier:

$$Re = \frac{L_x U}{\nu}.$$

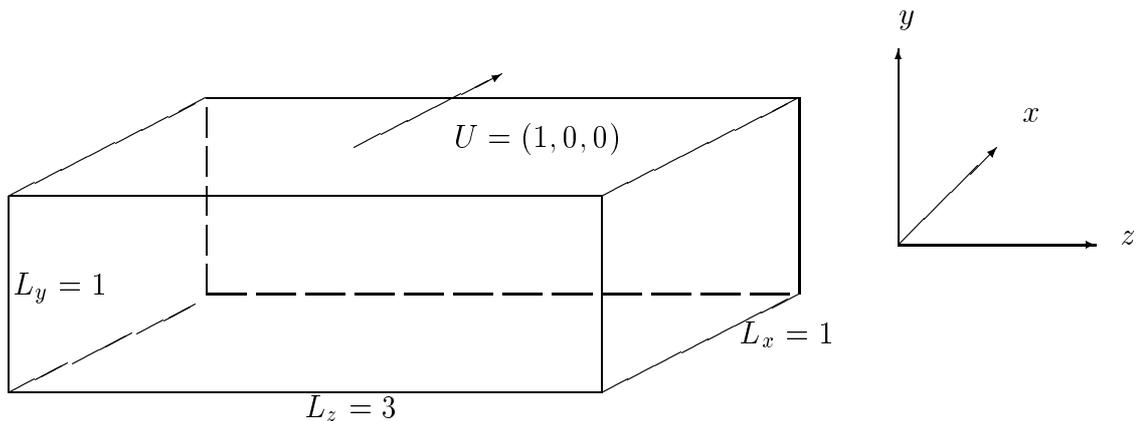


Abbildung 5.1: Definition der 3D-Driven-Cavity

Die Fragestellungen bei diesem Problem sind die folgenden:

1. Methode

- Formulierung
- Diskretisierung in Zeit und Raum
- Stabilität
- Randbedingung
- Zeitschritt
- Ortsgitter

2. Computerkosten

- verwendeter Computer
- Gesamtkosten

- Anzahl der Freiheitsgrade
- Anzahl der Zeitschritte

Da für dieses Problem sowohl numerische Ergebnisse anderer Strömungscodes (siehe [16]), als auch experimentelle Daten vorliegen (siehe [29]), lassen sich unsere Ergebnisse nicht nur qualitativ sondern vor allem auch quantitativ einordnen. In nachfolgender Abbildung haben wir die auftretenden Wirbel eingezeichnet, die Grobstruktur der Lösung wird im 2D-Modell bereits wiedergegeben, dies entspricht der y - z -Ebene. Hier erhalten wir die Hauptwirbelzelle, sowie die 3 Sekundärwirbel. Zusätzlich werden im Experiment bei dieser Reynoldszahl bis zu 9 sogenannte *Taylor-Görtler* Längswirbelpaare beobachtet, die sich mit der Zeit entwickeln und bewegen. Sie entstehen durch Rotationen, die durch das Zusammenwirken der Reibung der Seitenwände und der konkaven Separationsfläche zwischen Primär- und Sekundärwirbel an der Ausströmungswand hervorgerufen werden. Außerdem treten noch zwei Eckenwirbel am Boden auf, die entstehen durch das Zusammenwirken vom herunterströmenden Medium und den Reibungskräften in den Ecken. Die folgende Abbildung veranschaulicht den Sachverhalt.

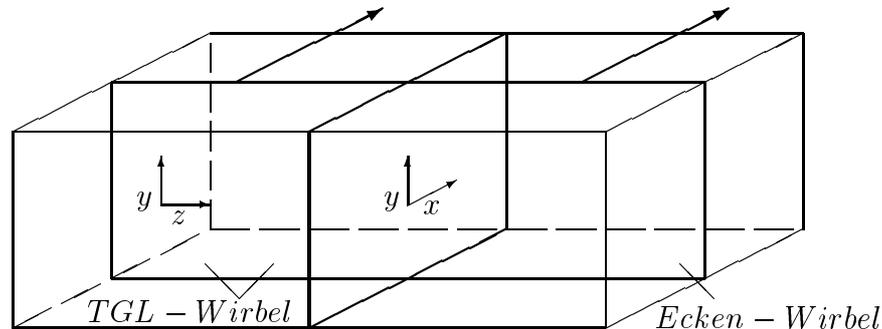


Abbildung 5.2: Qualitativer Charakter der 3D-Driven-Cavity

Um die Ergebnisse graphisch darzustellen, betrachten wir, wie oben dargestellt, zum einen die x - y -Ebene, diese Darstellung entspricht der 2D-Driven-Cavity auf dem Einheitsquadrat. Die hier weit relevantere Darstellung ist die y - z -Ebene, bzw. die x - z -Ebene, hier erkennt man die 3D-Effekte.

Wir führen zwei Simulationen für das instationäre Problem durch:

Test 1

Das Grobgitter besteht aus 24 Elementen, die Elemente sind Einheitswürfel der Kantenlänge 0.5cm. Dieses Grobgitter verfeinern wir noch fünfmal und erhalten im feinsten Gitter dann 786432 Elemente. Wir arbeiten mit der diskreten Projektionsmethode mit Extrapolation für die adaptive Schrittweitenkontrolle, dabei wird der Zeitschritt von 0.01s auf 0.33s hochgefahren und bleibt dann auch in dieser Größenordnung. Für die Zeitdiskretisierung verwenden wir das Fractional-step- ϑ -Verfahren aus Abschnitt 3.2. Gerechnet wird bis zu einer Gesamtzeit von 200s.

Das folgende Bild zeigt den oben erwähnten 2D-Schnitt in der x-y-Ebene, wir sehen den Hauptwirbel in der Mitte, sowie die Sekundärwirbel in den Ecken. Dieser Hauptwirbel wandert dabei in der Zeit von oben nach unten.

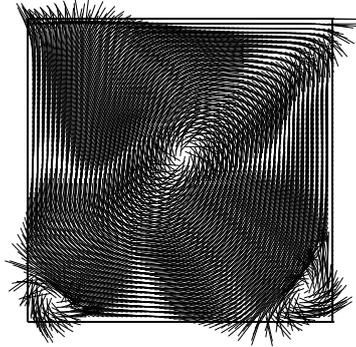


Abbildung 5.3: 3D-Driven-Cavity, 2D Schnitt

Das nächste Bild zeigt die 3D-Effekte, wir sehen vier Zeitpunkte ($T=10;50;100;200$) von oben nach unten, die Ebene die betrachtet wird ist die y-z-Ebene für $x=0.767$.

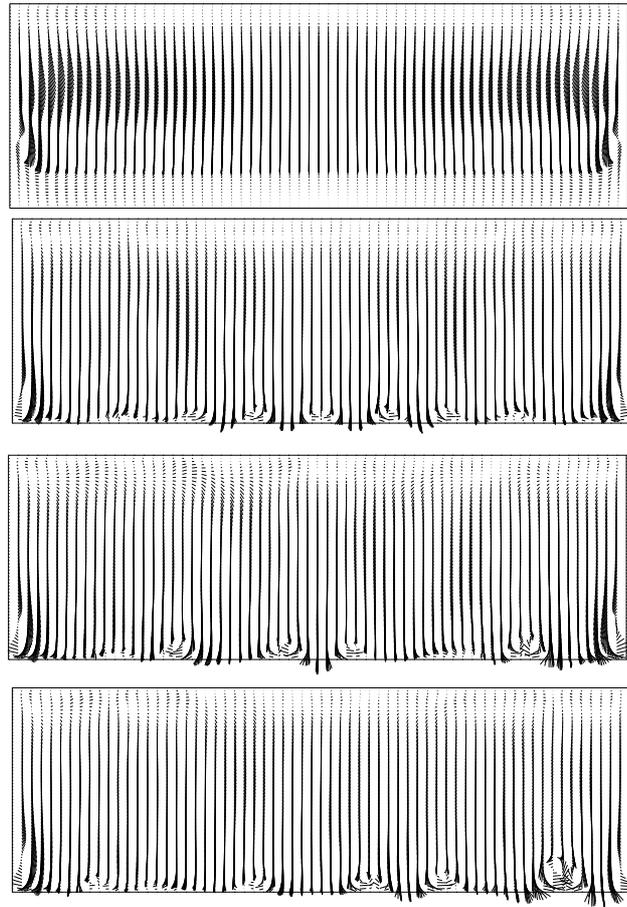
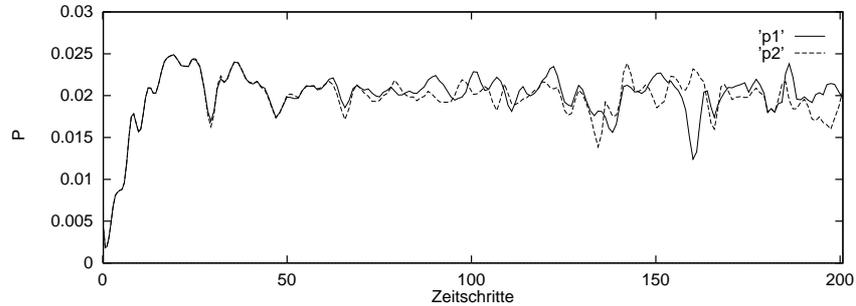
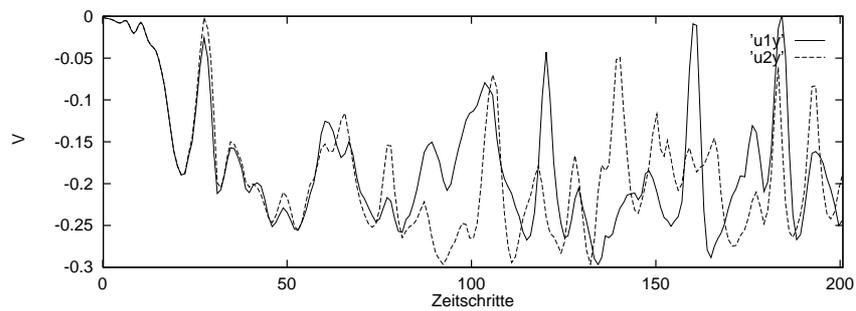
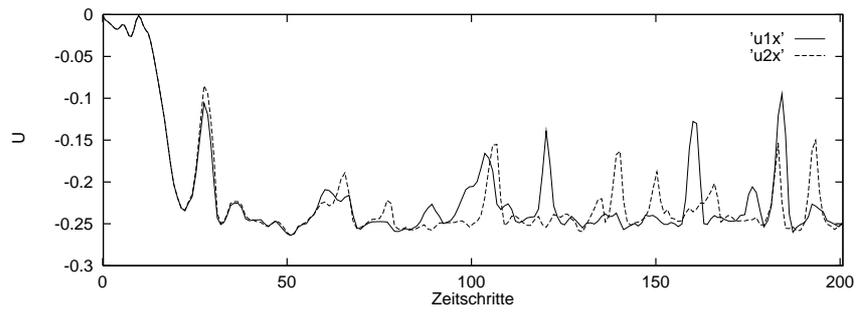


Abbildung 5.4: 3D-Driven-Cavity zu verschiedenen Zeitpunkten

Es entstehen schon nach wenigen Sekunden die Eckenwirbel, die *Taylor-Görtler* Längswirbelpaare entstehen nach ungefähr 20s und wandern in der Zeit, wie man auf den Abbildungen erkennt.

Interessant ist die Frage, ob es ausreicht nur die Hälfte der Box zu rechnen und in der Mittelebene bei $z = 1.5$ Symmetriebedingungen anzunehmen, um die Größe des Problems zu reduzieren. Einen Test dazu machen wir wie folgt: Wir vergleichen Geschwindigkeits- und Druckwerte zweier symmetrischer Punkte in der Zeit.



Man sieht deutliche Unterschiede in den jeweiligen Kurven, d.h. die Symmetrieanahme einiger Teilnehmer am Test in [16] ist offensichtlich falsch, nur in der Einschwingphase bis ca. $T=50s$ stimmen die Kurven überein, danach ergeben sich relativ große Unterschiede.

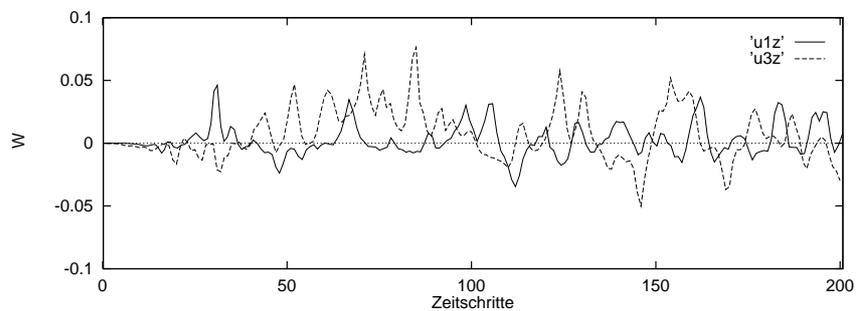
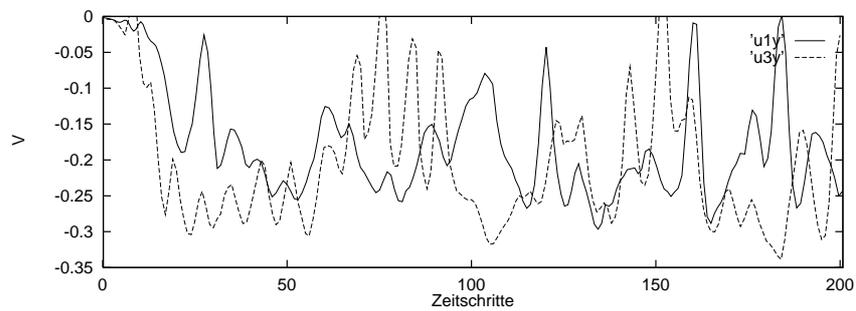
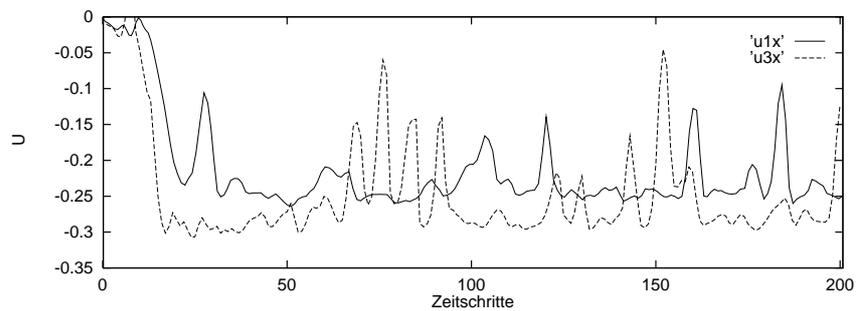
Test 2

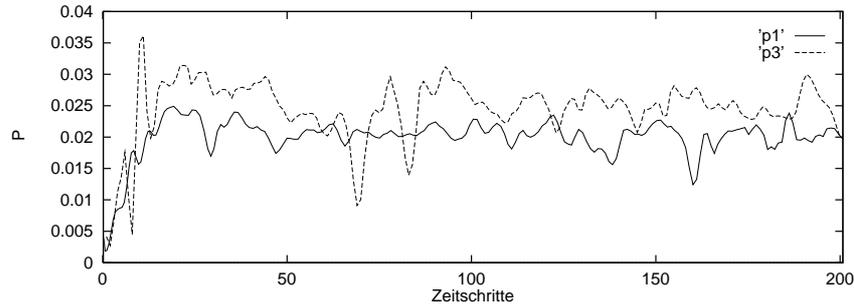
Wir führen die gleiche Rechnung auf einem anderen Gitter durch, diesmal besteht das Grobgitter aus 28 Elementen, dies ergibt nach wiederum 5 Verfeinerungen 917504 Elemente auf dem feinsten Level. Das Grobgitter ist dabei konstruiert wie folgt:

In y -Richtung besteht es aus 3 Schichten bei 0; $1/3$; 1, in x -Richtung bei 0; 0.5; 1, in z -Richtung: 0; 0.25; 0.616; 1.16; 1.83; 2.38; 2.75; 3.

In Unterschied zur ersten Rechnung arbeiten wir diesmal mit einem konstanten Zeitschritt von $1/3s$.

Qualitativ ergeben sich die gleichen Resultate wie oben, um das Problem hinlänglich Gitterabhängigkeit zu testen, vergleichen wir die Punktwerte für Druck und Geschwindigkeiten mit denen der anderen Rechnung.





Auch hier sehen wir Abweichungen der Werte, zum Einen bedingt durch das Zeitschritt-kriterium, d.h. die erste Rechnung hat in der Einschwingphase wesentlich kleinere Zeitschritte gewählt und dürfte damit genauer sein, das zweite Kriterium ist die Wahl des Gitters. Es zeigte sich bei zahlreichen vergeblichen Versuchen mit “angepaßten“ Gittern, daß obwohl die erwarteten *Taylor-Görtler* Längswirbelpaare im unteren Drittel der Cavity auftreten, eine Anpassung des Grobgitters, die dieser Tatsache Rechnung trägt, nicht zum Ziel führt. Es ist im Gegenteil entscheidend, daß auch im oberen Bereich der Cavity sehr fein diskretisiert wird, damit quasi die Störung (in Form der später auftretenden Wirbel) hinunter transportiert werden kann. Es wäre interessant herauszufinden, was ein adaptiver Gittergenerator in diesem Fall anstellt, wir haben die besten Erfahrungen mit dem Gitter aus **Test 1** gemacht, in dem wir nur Einheitswürfel verwendet haben. Beispiele von Gittern, bei denen das Grobgitter adaptiv in der x-y-Ebene, bzw. an den Rändern verfeinert ist, sehen wir in der folgenden Abbildung. Bei diesen Gittern konvergiert die Lösung in einen stationären Limes und es treten nur die Eckenwirbel auf, jedoch keine *Taylor-Görtler* Längswirbel.

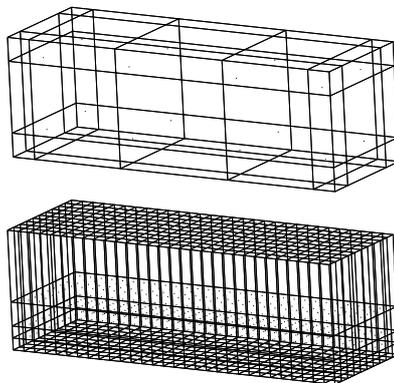


Abbildung 5.5: Angepaßte Grobgitter

Die Fragestellungen nach Art der Methode haben wir damit weitgehend beantwortet, es bleibt noch eine Bewertung des Aufwandes anzustellen. Beide Ergebnisse wurden auf einer **IBM RS6000/590** erzielt.

Test 1

- Anzahl der Freiheitsgrade: ca. 800000
- Anzahl der Makrozeitschritte: 210 (d.h. 840 Zeitschritte)
- Gesamtkosten: 127 CPU-Stunden
- Arbeitsspeicher: 920 MByte

Test 2

- Anzahl der Freiheitsgrade: ca. 920000
- Anzahl der Makrozeitschritte: 200 (d.h. 800 Zeitschritte)
- Gesamtkosten: 159 CPU-Stunden
- Arbeitsspeicher: 940 MByte

Wie schon erwähnt kostet die adaptive Zeitschrittkontrolle sowohl Rechenzeit als auch Speicherplatz, daher erklären sich die Unterschiede.

Verglichen mit den Ergebnissen der anderer Teilnehmer in [16], erzielten wir ähnliche qualitative Ergebnisse (man kann hier nicht von **der** Lösung sprechen). Die Anzahl der Freiheitsgrade ist dabei sehr hoch, dies liegt zum einen daran, daß wir die volle Box rechnen, d.h. keine Symmetrieanahmen stellen, was auch nicht gerechtfertigt scheint. Zum anderen verwenden wir einen linearen Ansatz im Gegensatz zu anderen Gruppen, die teilweise Ansätze höherer Ordnung haben. Die Rechenzeiten sind nur bedingt vergleichbar, da zum Teil Großrechner verwendet wurden, wir dagegen auf einer Workstation arbeiten. Zusammenfassend läßt sich sagen, daß wir im Vergleich des Aufwandes nicht besonders gut abschneiden. Dies liegt hauptsächlich daran, daß dieses Problem der *Driven Cavity* geradezu prädestiniert ist für die Verwendung von Finiten Differenzenverfahren. Hier können spezielle Verfahren auf Tensorproduktgittern mit explizitem Charakter ihre ganzen Vorteile zeigen (es brauchen z.B. nicht einmal Matrizen im Speicher gehalten werden). Unsere Methode, die für eine große Klasse von Problemen entwickelt ist, verliert bei diesem Problem dann natürlich an Effizienz. Eine Betrachtung der Rechenzeit verdeutlicht dies, bei unserem Verfahren benötigt der Matrixaufbau fast 75% der CPU Zeit.

Damit beenden wir diesen Abschnitt mit dem Fazit, daß unser Verfahren in der Lage

ist, auch im instationären Fall Ergebnisse zu liefern, die sowohl mit Experimenten, als auch mit vergleichbaren Codes wie in [16] zumindest qualitative Übereinstimmung zu erzielen. Im nächsten Abschnitt wenden wir uns verstärkt einem quantitativen Vergleich zu. Hier zeigt sich dann auch die Überlegenheit eines Finite-Elemente-Ansatz bei einem Problem mit komplexer Geometrie.

5.2. Quader- und Zylinderumströmung

Dieses Benchmarkproblem wurde unter dem Titel, “Definition von Benchmarkproblemen (Inkompressible laminare Strömungen)“, der Arbeitsgruppe von Professor Durst in Erlangen gestellt, und gehört in den allgemeinen Rahmen der DFG-SPP: “Strömungssimulation mit Hochleistungsrechnern“. Dieses Benchmarkproblem umfaßt auch den entsprechenden 2D-Fall, die Ergebnisse hierzu finden sich bei Schäfer/Turek in [41].

- Geometrie und Randbedingungen

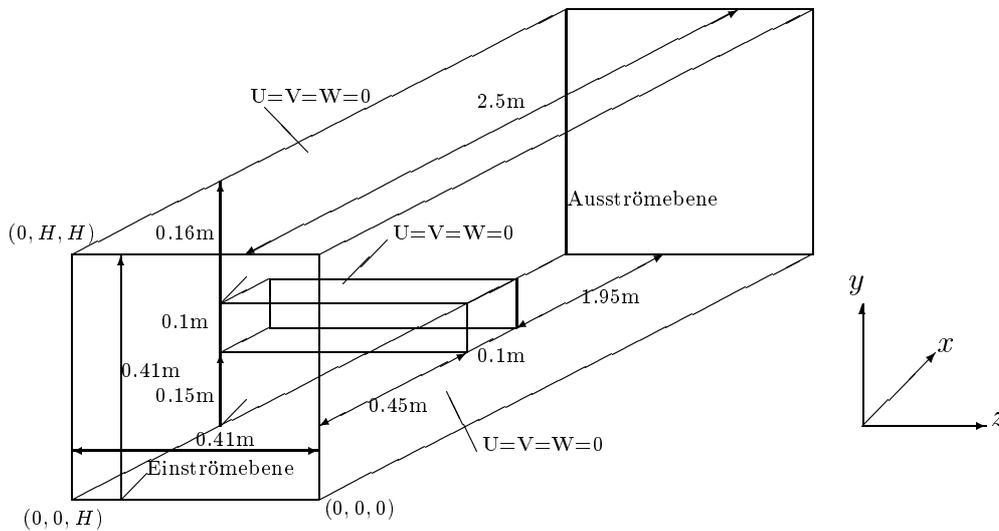


Abbildung 5.6: Quaderumströmung

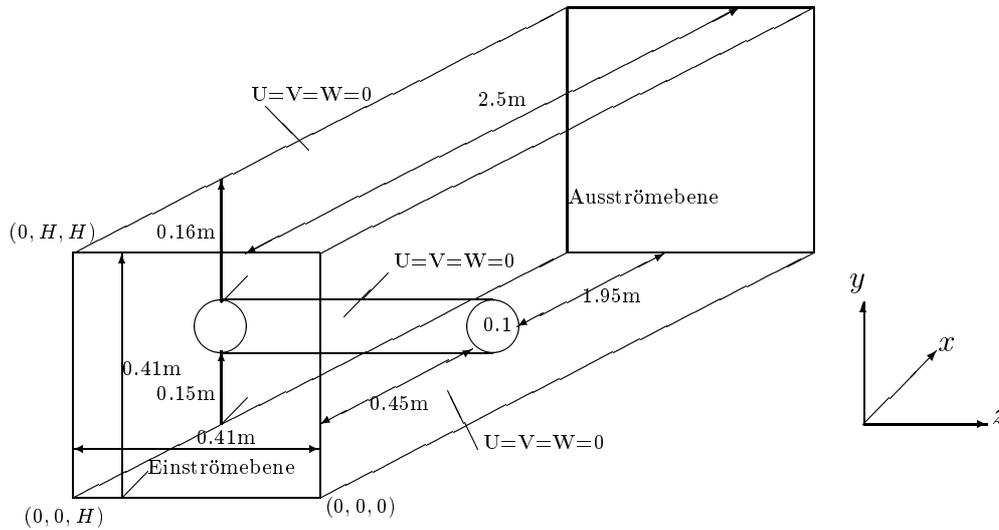


Abbildung 5.7: Zylinderumströmung

–Kanalhöhe und -breite: $H = 0.41m$

–Quaderseitenlänge und Zylinderdurchmesser: $D = 0.1m$

–Charakteristische Geschwindigkeit: $\bar{U} = \max\{4U(0, H/2, H/2, t)/9, \forall t\}$

–Reynoldszahl: $Re = \bar{U}D/\nu$

–Widerstandskraft: $F_w = \int_S (\rho\nu \frac{\partial v_t}{\partial n} n_y - pn_x) dS$

(Quader- bzw. Zylinderoberfläche S , Normale n zu S mit x-Komponente n_x und y-Komponente n_y , Tangentialgeschwindigkeit v_t zu S , Tangente $t = (n_y, -n_x, 0)$)

–Widerstandsbeiwert: $c_w = \frac{2F_w}{\rho\bar{U}^2DH}$

–Auftriebskraft: $F_a = - \int_S (\rho\nu \frac{\partial v_t}{\partial n} n_x + Pn_y) dS$

–Auftriebsbeiwert: $c_a = \frac{2F_a}{\rho\bar{U}^2DH}$

–Strouhalzahl: $St = Df/\bar{U}$

–Druckdifferenz: $\Delta P = \Delta P(t) = P(x_a, y_a, z_a, t) - P(x_e, y_e, z_e, t)$

(Koordinaten $(x_a, y_a, z_a) = (0.45, 0.205, 0.205)$ und $(x_e, y_e, z_e) = (0.55, 0.205, 0.205)$)

• Testfälle und zu berechnende Größen

a.) Testfall 3D–1Q und 3D–1Z (stationär):

– Einströmbedingung:

$$U(0, y, z) = 16U_m yz(H - y)(H - z)/H^4 \text{ mit } U_m = 0.45 \text{ m/s; } V = W = 0$$

– Ausströmbedingung: Vom Anwender selbst zu wählen

– Reynoldszahl: $Re = 20$

– Anfangswerte ($t = 0$): $U = V = W = P = 0$

– Zu berechnen ist: Der Widerstandsbeiwert c_w , der Auftriebswert c_a

b.) Testfall 3D–2Q und 3D–2Z (instationär):

– Einströmbedingung:

$$U(0, y, z, t) = 16U_m yz(H - y)(H - z)/H^4 \text{ mit } U_m = 2.25 \text{ m/s; } V = W = 0$$

– Ausströmbedingung: Vom Anwender selbst zu wählen

– Reynoldszahl: $Re = 100$

– Zu berechnen ist

- der Widerstandsbeiwert c_w , der Auftriebsbeiwert c_a und die Druckdifferenz ΔP
in Abhängigkeit von der Zeit für drei Perioden $[t_0, t_0 + 3/f]$ mit $f = f(c_a)$
- der maximale Widerstandsbeiwert c_{wmax}
- der maximale Auftriebsbeiwert c_{amax}

- die Strouhalzahl St
- Anfangswerte ($t = 0$) sind beliebig, jedoch für voll entwickelten Fluß

c.) **Testfall 3D-3Q und 3D-3Z (instationär):**

-Einströmbedingung:

$$U(0, y, z, t) = 4U_m yz(H - y)(H - z) \sin(\pi t/8)/H^2 \text{ mit } U_m = 2.25 \text{ m/s; } V = W = 0$$

-Zeitintervall: $0 \leq t \leq 8 \text{ s}$

-Ausströmbedingung: Vom Anwender selbst zu wählen

-Reynoldszahl: $0 \leq Re(t) \leq 100$

-Anfangswerte ($t = 0$) : $U = V = P = 0$

-Zu berechnen ist

- der Widerstandsbeiwert c_w und der Auftriebsbeiwert c_a in Abhängigkeit von der Zeit für $0 \leq t \leq 8 \text{ s}$
- der maximale Widerstandsbeiwert c_{wmax}
- der maximale Auftriebsbeiwert c_{amax}
- die Druckdifferenz $\Delta P(t)$ für $t = 8 \text{ s}$

Alle Rechnungen wurden auf einer **IBM RS6000/590** durchgeführt mit 1024 MB Arbeitsspeicher.

Die folgenden Grobgitter werden für obige Testfälle verwendet:

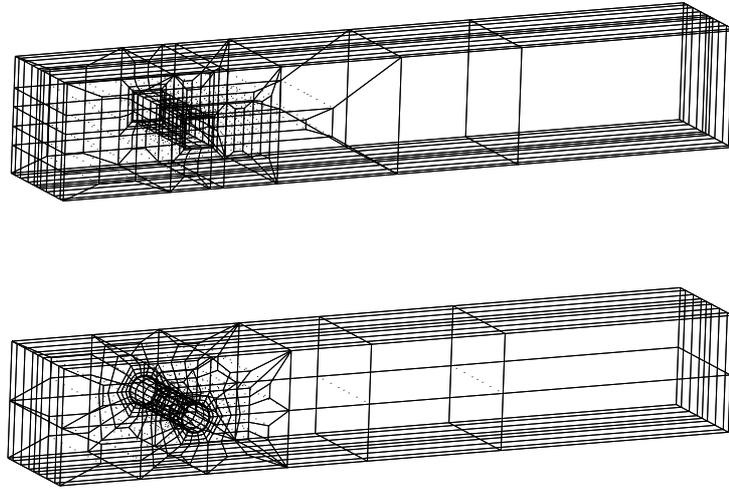


Abbildung 5.8: Grobgitter ($8h_1$) für Quader- und Zylinderumströmung

Damit ist die Ortsdiskretisierung für alle obigen Testfälle wie folgt:

Tabelle 5.1: Gitter 3D-Q und 3D-Z

	Quader			Zylinder		
	h_1	$2h_1$	$4h_1$	h_1	$2h_1$	$4h_1$
Elemente	604160	75520	9440	606208	75776	9472
Knoten	622080	80032	10584	624520	80388	10642
Knoten auf Q/Z	7776	1968	504	9360	2376	612
h_{min} auf Q/Z	$3.125 \cdot 10^{-3}$	$6.250 \cdot 10^{-3}$	$1.500 \cdot 10^{-2}$	$2.173 \cdot 10^{-3}$	$4.347 \cdot 10^{-3}$	$8.695 \cdot 10^{-3}$
h_{max} auf Q/Z	$7.500 \cdot 10^{-3}$	$1.500 \cdot 10^{-3}$	$3.000 \cdot 10^{-3}$	$7.812 \cdot 10^{-3}$	$1.562 \cdot 10^{-2}$	$3.125 \cdot 10^{-2}$
h_{min} zu Q/Z	$6.250 \cdot 10^{-4}$	$1.250 \cdot 10^{-4}$	$2.500 \cdot 10^{-3}$	$5.208 \cdot 10^{-4}$	$1.041 \cdot 10^{-3}$	$2.083 \cdot 10^{-3}$
h_{max} zu Q/Z	$6.250 \cdot 10^{-4}$	$1.250 \cdot 10^{-4}$	$2.500 \cdot 10^{-3}$	$5.208 \cdot 10^{-4}$	$1.041 \cdot 10^{-3}$	$2.083 \cdot 10^{-3}$

Für den Testfall a.) setzen wir den stationären Navier–Stokes Löser aus Abschnitt 3.1 ein.

Für die Zeitdiskretisierung für die Testfälle b.) und c.) verwenden wir das *Fractional-step- ϑ -scheme* (siehe Abschnitt 3.2) und eine adaptive Schrittweitensteuerung über den lokalen Zeitdiskretisierungsfehler (Abschnitt 4.1). Als Lösungsverfahren für die instationären Probleme wählen wir die diskrete Projektionsmethode aus Abschnitt 4 mit

der Massematrix als Schur-Komplement-Vorkonditionierer und einer voll-impliziten Behandlung der nichtlinearen Konvektionsterme. Zur Lösung der linearen Teilprobleme verwenden wir Mehrgitterverfahren, wobei wir ILU-Verfahren als Glätter und Löser einsetzen.

Das Abbruchkriterium für den relativen Zeitfehler:

$$\frac{\|u - u_k\|}{\|u_k\|} \leq \epsilon_3^i, \text{ mit } \epsilon_3^1 = 1.25 \cdot 10^{-4}, \epsilon_3^2 = 1.25 \cdot 10^{-3}, \epsilon_3^3 = 1.25 \cdot 10^{-2}$$

Für die stationären Testfälle erhalten wir folgende Ergebnisse:

Testfall a.)

Tabelle 5.2: Ergebnisse 3D-1Q und 3D-1Z

	3D-1Q			3D-1Z		
	h_1	$2h_1$	$4h_1$	h_1	$2h_1$	$4h_1$
c_w	7.6148	7.5622	7.3069	6.1043	5.9731	5.8431
c_a	0.0600	0.0503	0.0348	0.00792	0.00595	0.00614
Δp_t	0.1729	0.1683	0.1590	0.1672	0.1605	0.1482
Speicher (MB)	690	88	10	700	89	11
CPU Zeit	8244	1267	380	8440	1466	290

In beiden Fällen wird jeweils die Lösung des größeren Gitters als Startwert eingelesen.

Testfall b.)

Tabelle 5.3: Ergebnisse 3D-2Q und 3D-2Z

	3D-2Q			3D-2Z		
	h_1/ϵ_2^1	h_1/ϵ_2^2	h_1/ϵ_2^3	h_1/ϵ_2^1	h_1/ϵ_2^2	h_1/ϵ_2^3
Zeitschritte	142	124	84	128	120	80
$c_{w_{max}}$	4.3923	4.3932	4.4071	3.2950	3.2970	3.3200
$c_{a_{max}}$	0.0146	0.0191	0.0896	-0.0081	-0.0025	0.0480
ST	0.2777	0.2806	0.2400	0.2912	0.2830	0.2684
Speicher (MB)	840	840	840	840	840	840
CPU Zeit	29428	29945	30372	31145	31730	21686

Testfall c.)

Tabelle 5.4: Ergebnisse 3D–3Q und 3D–3Z

	3D–3Q			3D–3Z		
	h_1/ϵ_2^1	h_1/ϵ_2^2	h_1/ϵ_2^3	h_1/ϵ_2^1	h_1/ϵ_2^2	h_1/ϵ_2^3
Zeitschritte	772	392	82	668	272	60
$c_{w_{max}}$	4.4086	4.5698	5.5709	3.2802	3.3748	2.7312
$c_{a_{max}}$	0.0133	0.0262	0.1230	0.0034	0.0360	0.0069
Δp_{max}	-0.1264	-0.1213	0.0183	-0.0959	-0.0603	-0.0682
Speicher (MB)	840	840	840	840	840	840
CPU Zeit	164749	89679	35600	164837	77538	29742

Ein Fazit des Vergleiches mit den anderen teilnehmenden Gruppen stellt sich folgendermaßen dar:

–Der Einsatz impliziter Methoden kann für die Berechnung inkompressibler Strömungen eine weit effizientere Methode darstellen als explizite Algorithmen. Die Beschränkung des Zeitschrittes bei der expliziten Methode läßt diesen Ansatz ineffizient werden, da die physikalische Zeitskala größer werden kann als der maximale Zeitschritt im expliziten Algorithmus. Dieses Fazit bedarf aber noch weitgehendere Untersuchungen.

–Moderne iterative Methoden wie Mehrgitterverfahren schneiden weit besser ab als Verfahren, die auf konventionellen Methoden beruhen. Unser Verfahren benötigt z.B. im Testfall 3D–3Z weit weniger CPU Zeit auf einer Workstation, bei gleicher Zahl Unbekannter, als Ergebnisse, die auf Supercomputern erzielt wurden.

–Finite–Elemente, bzw. Finite–Volumen Diskretisierungen auf angepaßten Gittern sind sehr effizient und genau. Dies liegt natürlich an der Geometrie der Hindernisse, hier sind Finite–Elemente–Verfahren flexibler und besser geeignet. Dies steht im Gegensatz zur Driven–Cavity aus dem letzten Abschnitt. Dort entfalten Differenzenverfahren ihre ganzen Möglichkeiten.

–Die Berechnung von stationären Lösungen mit pseudo Zeitschrittverfahren ist ineffizient verglichen mit quasi Newton Verfahren als stationärem Löser.

–Der konvektive Term sollte mit Verfahren höherer Ordnung behandelt werden. Upwinding von nur erster Ordnung Genauigkeit führt zu nicht zufriedenstellender Genauigkeit auch nicht auf sehr feinen Gittern.

Andere Fragen wie z.B. der Vergleich gekoppelter Methoden mit Operator Splitting

oder Verwendung von Ortsdiskretisierungen höherer Ordnung bedürfen weiterer Untersuchung.

Die Ergebnisse des 2D-Falles, sowie die der anderen teilnehmenden Gruppen, finden sich in [41].

5.3. Die “tanzenden Rauchringe“

Auf dieses Beispiel sind wir bereits kurz in Abschnitt 1.4 eingegangen, dort befindet sich auch eine Abbildung der Gitterhierarchie. Dieses Beispiel ist aus mehreren Gründen interessant, zum ersten natürlich durch die Geometrie und der daraus resultierenden deformierten Gitter, wodurch dieses Beispiel auch interessant wurde für die Einführung der nichtparametrischen Elemente. Der zweite Aspekt ist hier bei diesem Problem die Wahl der Randbedingungen, und der dritte und nicht zu unterschätzende Punkt ist die Visualisierung, die besondere Techniken erfordert.

In der Frage der Randbedingungen folgen wir der Arbeit von Heywood et al. in ([27]), dort sind die entsprechende Techniken theoretisch und für den 2D-Fall auch praktisch analysiert. Das Problem, das wir zu lösen haben, sieht folgendermaßen aus:

Pressure Drop Problem: Finde $\mathbf{u}(t)$ und $p(t)$ so daß,

$$\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = 0, \quad \nabla \cdot \mathbf{u} = 0, \quad (5.1)$$

$$\mathbf{u}|_{\Gamma} = 0, \quad \frac{1}{|S_i|} \int_{S_i} p \, ds = P_i(t), \quad (5.2)$$

für jeden vorgeschriebenen Wert von $P_i(t)$.

Die dazu entsprechende variationelle Formulierung hat damit folgende Gestalt:

Variationelles Pressure Drop Problem: Finde $\mathbf{u}(t)$ so daß für alle t ,

$$\mathbf{u}(t) \in \mathbf{V}_1(\Omega) \equiv \{\varphi \in \mathbf{H}^1(\Omega) : \varphi|_{\Gamma} = 0\}, \quad p(t) \in L^2(\Omega), \quad (5.3)$$

und

$$\nu(\nabla \mathbf{u}, \nabla \varphi) + (\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u}, \varphi) - (p, \nabla \cdot \varphi) = - \sum_j P_j(t) \int_{S_j} \varphi \cdot \mathbf{n} ds \quad \forall \varphi \in \mathbf{V}_1(\Omega), \quad (5.4)$$

$$(\chi, \nabla \cdot \mathbf{u}) = 0, \quad \forall \chi \in L^2(\Omega). \quad (5.5)$$

Ein Spezialfall dieser Formulierung ist die sogenannte “do nothing“-Bedingung an freien Rändern, an denen wir a-priori nicht wissen können, welche Bedingungen vorzuschreiben sind. Diese Neumann ähnliche Randbedingung erweist sich schon bei einer einfachen Kanalströmung mit Hindernis als nützlich (siehe die Ergebnisse in Abschnitt 2.5), da man bei solch einem Problem keine Dirichletdaten am Ausflußrand vorschreiben kann, ohne die Strömung im Inneren dadurch zu beeinflussen, bzw. müßte man den Kanal entsprechend lang machen, um den Einfluß auszuschalten, was dann aber wieder unnötig Speicher und Rechenzeit kostet. Für Einzelheiten verweisen wir auf ([27]).

Anwendung:

Diese Formulierungen erlauben eine Fülle von Anwendungen, speziell in drei Dimensionen bei komplizierten Geometrien. Das Beispiel der tanzenden Rauchringe findet sich in ([53]). Im physikalischen Experiment werden dabei nacheinander 2 “Puffs“ durch einen Kolben mit Durchmesser 8cm gepustet, der zweite Rauchring ist schneller und wandert fast durch den ersten hindurch.

Unser numerisches Modell sieht nun folgendermaßen aus:

Wir geben an den Rändern beider Kammern einen unterschiedlichen Druckmittelwert gemäß obiger Formulierung vor. Am Rand des schmalen Verbindungskanals schreiben wir Dirichlet Nullranddaten vor. Durch den “pressure drop“ angetrieben, entsteht eine Strömung aus der kleinen Kammer in die Halbkugel, dies entspricht dem ersten “Puff“. Diese Randdaten werden zusätzlich zeitabhängig vorgeschrieben, d.h. wir geben eine bestimmte Anzahl Zeitschritte diesen Druckunterschied vor, schalten ihn dann ein paar Zeitschritte lang aus, um ihn dann erneut leicht erhöht wieder vorzuschreiben. Dadurch hat der zweite “Puff“ eine höhere Geschwindigkeit als der erste und kann somit durch den ersten hindurchwandern.

Das Grobgitter für diese Rechnung besteht aus 23 Elementen, das feinste Rechengitter aus 94208 Elementen, was ungefähr 900000 Unbekannten entspricht. Wir rechnen mit

einer festen Schrittweite von $\Delta t = 0.1$, bis zu einer Gesamtzeit von $T = 20$, also 200 Zeitschritte. Die Reynoldszahl ist bei dieser Rechnung ungefähr $Re = 600$ und damit etwas geringer als im Experiment (siehe [53]). Dies hat aber allein praktische Gründe, wählte man die gleiche Reynoldszahl wie im Experiment (bei gleicher Geometrie), wanderten die Ringe zu schnell durch das Gebiet, und hätten keine Möglichkeit, sich zu durchdringen. Diese Rechnung ist mit dem gekoppelten Ansatz (siehe Abschnitt 3.2) durchgeführt, da wegen der geringen Reynoldszahl eine relativ große Zeitschrittweite verwendet werden kann. Die Rechenzeit für die gesamte Simulation liegt im Bereich von 20 Stunden auf einer IBM RS6000/590.

Die nachfolgenden Bilder entstanden in Zusammenarbeit mit Herrn Dipl.Math. H. Oswald, der das 3D-**Particle Tracing** aufbauend auf das Graphikpaket Advanced Visualisation Systems (AVS) entwickelte. Diese Bilder sind Momentaufnahmen einer Sequenz, die aus 75 Einzelbildern besteht und als Videofilm verfügbar ist. Wir erkennen den ersten “Puff“ im Bild links oben. Auf den darauffolgenden Bildern wandert der zweite “Puff“ durch den ersten hindurch, bis beide dann schließlich aus dem Rechengebiet hinauswandern. Ein Vergleich mit 2D Ergebnissen bei Turek in ([48]) und Experimenten in ([53]) zeigen gute Übereinstimmung. Die Hauptarbeit bestand in der Visualisierung der Daten. Standarddarstellungen mit Vektorpfeilen oder Isoflächen können diesen Prozeß der Durchdringung der Ringe nur schlecht, bzw. gar nicht wiedergeben (siehe auch Seite 155).

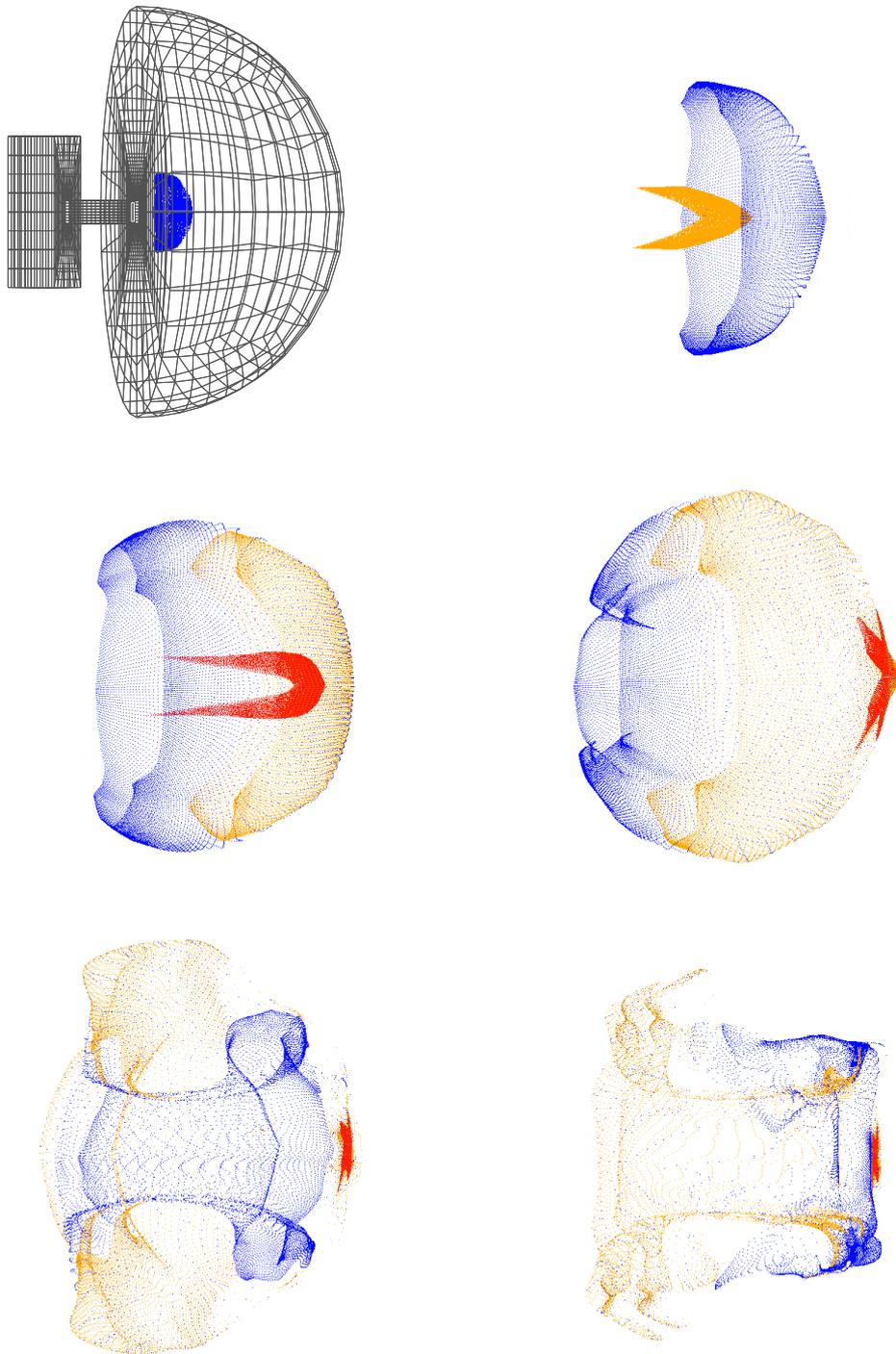


Abbildung 5.9: Puff-Puff, Visualisierung mit particle tracing

5.4. Fiktive Randtechniken im Mehrgitteralgorithmus

Eine zentrale Frage bei der praktischen Umsetzung der vorgestellten Mehrgittertechniken stellt sich bei der Diskretisierung komplexer Geometrien mit mehreren Randkomponenten. Dieses Problem wird für den 3D-Fall ganz wesentlich vor allem beim Projekt für die Strömungsberechnung in einem Heizbrenner, siehe dazu Abschnitt 6.3. Vom Mehrgitterstandpunkt aus gesehen, wollen wir ein möglichst “grobes“ Grobgitter, d.h. mit maximal bis zu 1000 Zellen, ein feineres Gitter wäre nicht mehr effektiv, da das Grobgitterproblem dann zu groß wäre. Andererseits sollen in diesem Grobgitter bereits alle Informationen enthalten sein, die das Problem beschreiben. Bei sehr komplexen Geometrien, wie z.B. das Brennermodell aus dem nächsten Kapitel, in dem eine Keramikplatte mit ca. 5000 Kanälen diskretisiert werden soll, ist das natürlich nicht möglich. Die nachfolgende Technik, die wir für solche Probleme verwenden, beschreiben und testen wir hier an dem 2D Benchmark Problem, das wir in diesem Kapitel bereits für den analogen 3D-Fall vorgestellt haben. Wir präsentieren deswegen den 2D-Fall, da er graphisch besser darstellbar ist, im weiteren Verlauf dieses Abschnittes werden wir die Techniken auch für dreidimensionale Konfigurationen anwenden.

Wir haben also das Problem der Kugelumströmung, wie folgt:

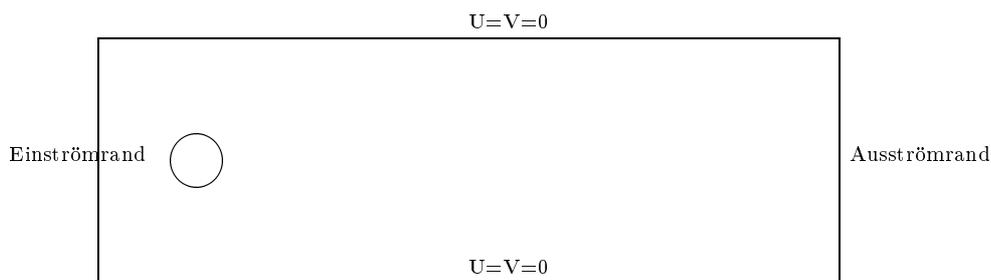


Abbildung 5.10: Kugelumströmung

Wir lösen dieses stationäre Problem nun mit drei verschiedenen Grobgittern, das erste ist ein an die Geometrie voll angepaßtes, die anderen zwei Gitter erfassen die Geometrie nicht, in dem zweiten dieser Gitter sind jedoch im Bereich des Kreises mehr Elemente vorhanden.

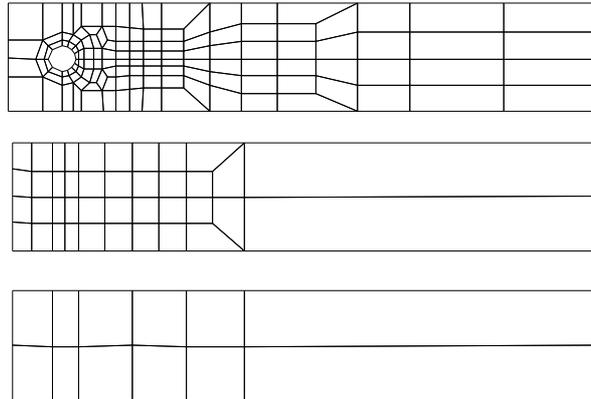


Abbildung 5.11: Grobgitter

Die Ergebnisse präsentieren wir nun zuerst qualitativ, wir zeigen für das erste und das zweite Gitter die Isolinien des Drucks, sowie der x-Komponenten der Geschwindigkeiten:

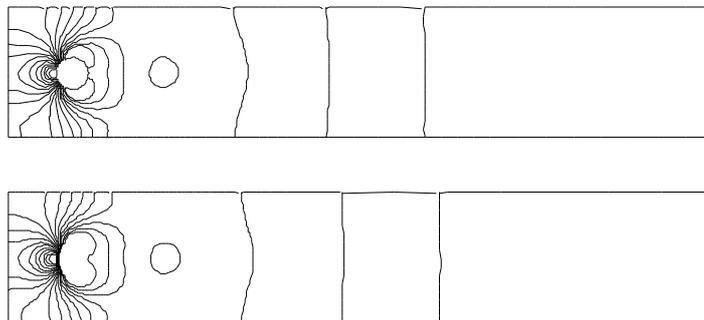


Abbildung 5.12: "Isolinien des Drucks"

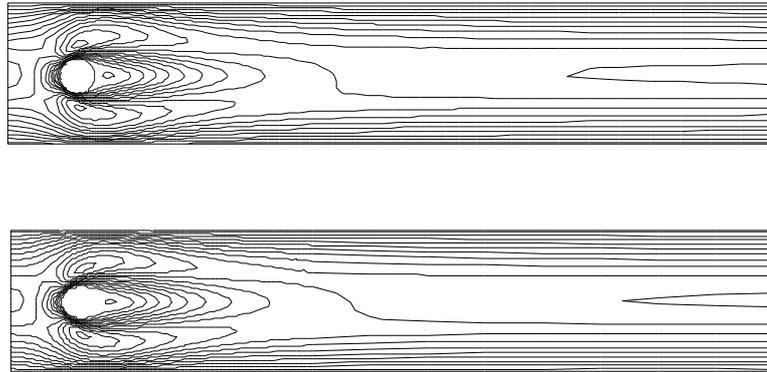


Abbildung 5.13: “Isolinien der Geschwindigkeit“

Wir sehen in diesen Abbildungen eine sehr gute Übereinstimmung der Ergebnisse, um das Ganze aber auch quantitativ zu beurteilen, vergleichen wir die Geschwindigkeits- und Druckwerte an ausgewählten Punkten des Gitters, wobei wir besonderes Augenmerk auf die Druckdifferenz vor und hinter dem Kreis richten, da nach diesem Wert in der Definition des Benchmark Problems gefragt wird.

Tabelle 5.5: Ergebnisse für Gitter 1 und 2

Elem.	voll angepaßt				nicht angepaßt			
	2080	8320	33280	133120	2560	10240	40960	163840
Pres 1	0.1377	0.1343	0.1327	0.1322	0.1293	0.1299	0.1304	0.1312
Pres 2	0.2315-1	0.1916-1	0.1661-1	0.1538-1	0.1521-1	0.1513-1	0.1485-1	0.1468-1
Pres 3	0.2855-1	0.2847	0.2812-1	0.2787-1	0.2442-1	0.2665-1	0.2739-1	0.2761-1
Pres 4	0.2244-1	0.2271	0.2272-1	0.2267-1	0.1867-1	0.2112-1	0.2207-1	0.2237-1
P1-P2	0.1145	0.1151	0.1161	0.1168	0.1141	0.1148	0.1155	0.1165
Velo 1	0.2349	0.2156	0.2062	0.2029	0.2254	0.2113	0.2051	0.2027
Velo 2	-0.3261-3	0.1372-3	0.5247-3	0.7310-3	-0.3747-3	0.4633-3	0.7053-3	0.7982-3
Velo 3	0.2671	0.2569	0.2516	0.2497	0.2580	0.2535	0.2508	0.2496
Velo 4	-0.3777-3	-0.3247-3	-0.2294-3	-0.1584-3	-0.6518-3	-0.3807-3	-0.2067-3	-0.1370-3

Wir sehen, daß wir mit diesem Vorgehen eine sehr gute Übereinstimmung der Ergebnisse erzielen. Darauf zu achten ist jedoch, daß wir, um im Mehrgitter effektiv arbeiten zu können, auf den Gittern unterhalb des feinsten Gitters dasselbe Problem lösen. Wir dürfen also nicht den Fehler machen, die Randwerte nur auf dem feinsten Level vorzugeben, oder Gitter zu benutzen, auf denen im Gebiet des definierten Randes keine Gitterpunkte mehr liegen. Was sonst passieren kann, sehen wir in der nächsten Tabelle.

Wir betrachten hier die Konvergenzraten des linearen Mehrgitterverfahrens für 9 nicht-lineare Iterationen, einmal für den Fall, daß wir die Randbedingungen auf allen Gittern vorgeben (Fall 1), zum zweiten für den Fall, daß wir die Randbedingung nur auf dem feinsten Gitter definieren, auf den gröberem Gittern damit eine reine Kanalströmung haben (Fall 2).

Tabelle 5.6: Konvergenzraten Fall 1 und 2

Elem.	Fall 1			Fall 2		
	10240	40960	163840	10240	40960	163840
RHOMG 1	0.19	0.07	0.05	0.54	0.44	0.52
RHOMG 2	0.26	0.13	0.09	0.72	0.94	0.89
RHOMG 3	0.29	0.19	0.15	0.75	0.89	0.88
RHOMG 4	0.27	0.18	0.16	0.71	0.81	0.88
RHOMG 5	0.32	0.26	0.16	0.72	0.83	0.91
RHOMG 6	0.36	0.23	0.17	0.70	0.90	0.99
RHOMG 7	0.36	0.27	0.20	0.66	0.86	0.97
RHOMG 8	0.39	0.29	0.19	0.67	0.86	1.00
RHOMG 9	0.40	0.25	0.15	0.64	0.85	1.00

Es ist also zu berücksichtigen, daß in dem Bereich des Randes, der nachträglich definiert wird, auch auf den gröberem Gittern Punkte liegen, da sonst keine Konvergenz mehr für das Mehrgitterverfahren erzielt wird. Bei diesem numerischen Beispiel reichte es aus, auf den 3 feinsten Gittern den Rand vorzuschreiben, die Mehrgitterkonvergenzraten waren für diesen Fall nur unwesentlich schlechter, als für Fall 1. Natürlich soll diese Technik kein Gegenbeispiel für die Verwendung angepaßter Gitter sein, es geht nur darum, in Fällen wie in Kapitel 6 beschrieben, überhaupt unsere Mehrgittertechniken verwenden zu können und eine gewisse numerische Rechtfertigung zu haben, daß die dann erzielten Ergebnisse auch quantitativ in Ordnung sind.

5.4.1. Bewegte Ränder 1: Eine Wasserpumpe in 2D

Als weitere Anwendung oben beschriebener Strategie können wir nun eine weitere Problemklasse behandeln. Wir haben jetzt die Möglichkeit, auch Probleme mit in der Zeit veränderter Geometrie durch Definition von fiktiven Rändern zu berechnen. Auch hier beginnen wir wieder mit dem zweidimensionalen Fall, da wir für das erste Beispiel,

einem pumpenden Kolben, eine Vergleichsmöglichkeit in der Arbeit von Becker ([2]) aus dem eigenen Haus haben. Das Problem ist damit folgendermaßen charakterisiert:

Die Geometrie der Pumpe wird durch einen Zylinder beschrieben, in dem sich ein Kolben auf und ab bewegt. Durch Steuerung der Randbedingungen, d.h. zeitlichen Wechsel von Dirichlet Nullrandbedingungen zu Neumann Randbedingungen, wird an den Enden des Zylinders eine Klappe simuliert, die je nach Bewegung des Kolbens auf oder zu ist. Die Bewegung des Kolbens wird durch eine trigonometrische Funktion $\alpha(t)$ definiert, die Randbedingung auf dem Kolben ist dann gerade durch die Ableitungsfunktion $\alpha'(t)$ definiert, für technische Details siehe [2]. Der Kolben bewegt sich dann zwischen $y = -2$ und $y = -0.25$. Der einzige Unterschied zu Becker ist der, daß wir die Geschwindigkeit des Kolbens höher vorgeben (Faktor 10). Das erste Bild zeigt nun das Grobgitter, das während der Rechnung nicht verändert wird.

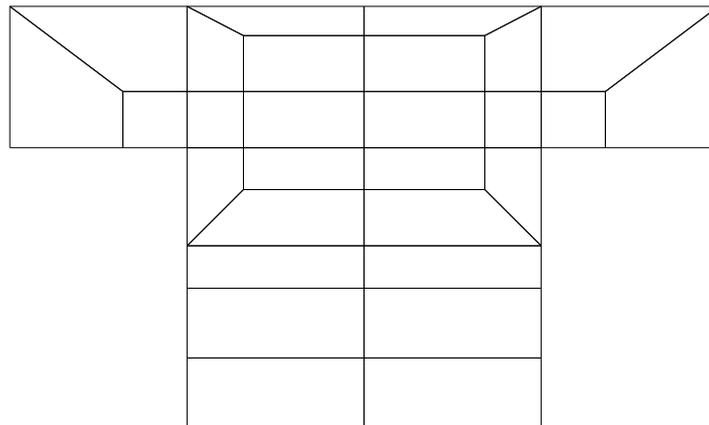


Abbildung 5.14: Grobgitter für Wasserpumpe

Im Gegensatz zu der Arbeit von Becker haben wir keine Möglichkeit einer adaptiven Verfeinerung, es ist daher von Interesse, inwieweit die auftretenden Singularitäten durch den Einfluß der Ecken und des Bodens des Kolbens die Ergebnisse hier beeinflussen. Zuerst wollen wir die qualitativen Ergebnisse veranschaulichen, es ist hier jeweils eine Abbildung der Druckisoflächen und Geschwindigkeitspfeile (normalisiert) dargestellt, links für die Abwärts-, rechts für die Aufwärtsbewegung. Bei der Abwärtsbewegung ist der rechte Rand offen, der linke zu, bei der Aufwärtsbewegung genau umgekehrt.

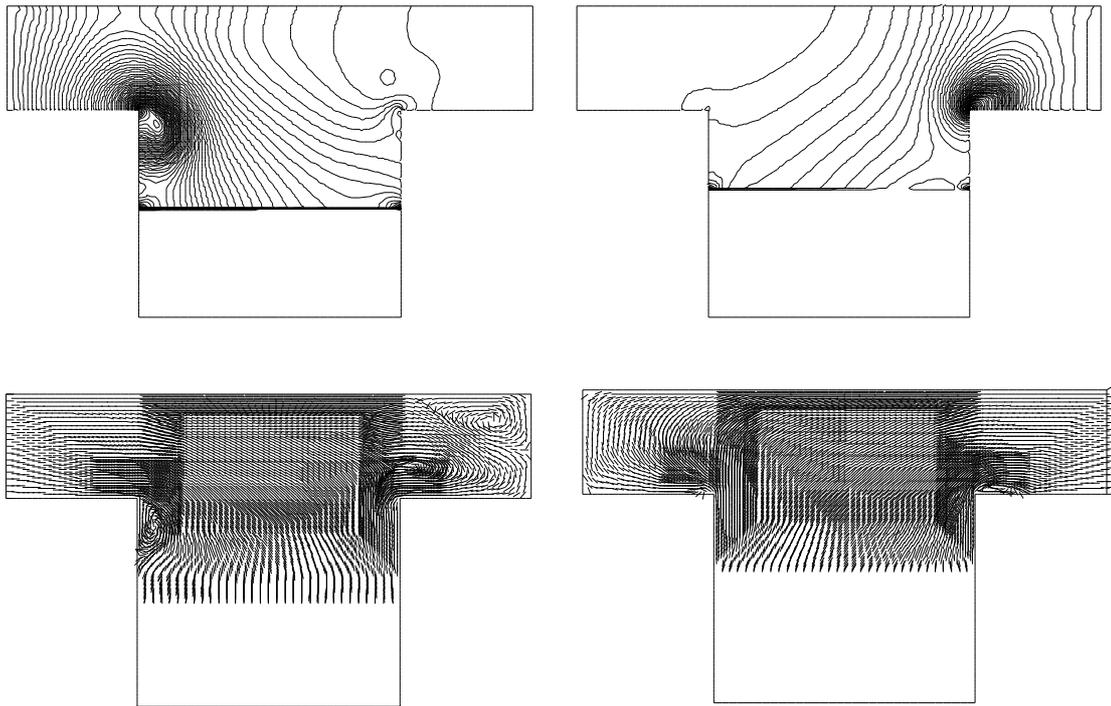
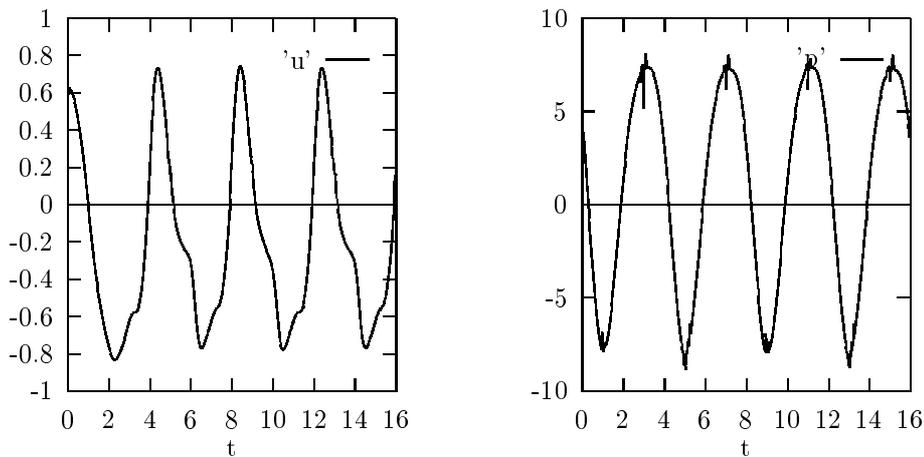
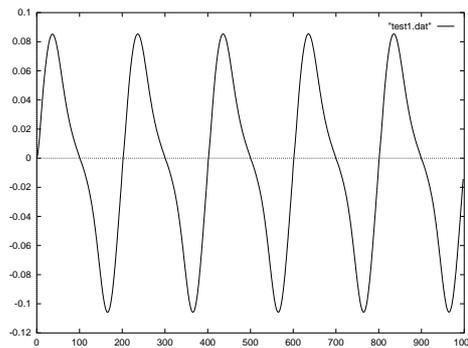


Abbildung 5.15: Ergebnisse für Wasserpumpe

In diesen Abbildungen erkennen wir eine gute Übereinstimmung mit den Ergebnissen von Becker, interessanter sind aber die Messungen am Punkt(0,0) für Druck und Geschwindigkeit.

Abbildung 5.16: y -Komponente und Druck der Geschwindigkeit in $(0,0)$ Abbildung 5.17: y -Komponente der Geschwindigkeit bei Becker

Verglichen mit den Ergebnissen bei Becker in Abbildung 5.4.1, bekommen wir für die Geschwindigkeit eine gute Übereinstimmung, Probleme bereitet der Druck, der bei den Umkehrpunkten in kleine Unstetigkeiten gerät, hier machen sich die Singularitäten bedingt durch die Ecken und des Bodens des Kolbens mehr bemerkbar als bei der Geschwindigkeit. Zudem spielt natürlich auch die Trägheit des Mediums eine Rolle. Am Umkehrpunkt wirkt keine, bzw. nur noch eine minimale Geschwindigkeit des Kolbens, es scheinen dortige Unstetigkeiten durch Trägheitskräfte zu kommen. Nach diesen Testrechnungen wollen wir das gleiche Beispiel auch in drei Dimensionen behandeln.

5.4.2. Bewegte Ränder 2: Eine Wasserpumpe in 3D

Wir haben die gleichen Abmessungen wie in zwei Dimensionen, die zusätzliche z -Komponente ist 2cm lang, damit hat der Kolben eine quadratische Grundfläche. Es wäre natürlich interessanter eine Pumpe zu betrachten, deren Grundfläche an allen 4 Seiten echt kleiner ist, als die des darüber liegenden Gebietes, aber um die Ergebnisse mit denen des 2D-Falles vergleichen zu können, haben wir die zuerst angegebene Konfiguration, die dann mit einem Grobgitter von 520 Elementen sich darstellt in folgender Abbildung, in der auch ein 2D-Schnitt der Mittelebene zu sehen ist:

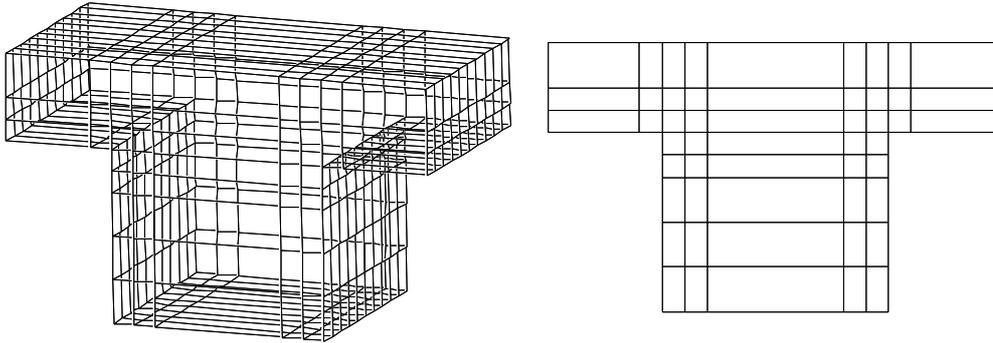


Abbildung 5.18: Grobgitter für 3D-Wasserpumpe

Die anderen Parameter für die instationäre Rechnung wählen wir wie oben, d.h. wir haben gleiche Geschwindigkeit und Auslenkung des Kolbens, gleiche Viskosität und wir wählen auch den gleichen festen Zeitschritt. Ein 2D-Schnitt in der Mittelebene zeigt qualitativ die gleichen Ergebnisse wie für die 2D-Rechnung:

Interessanter ist folgende Betrachtung, wir lassen die Visualisierungsebene in z -Richtung wandern, so daß sie fast am Rand ist, die folgende Abbildung zeigt für den selben Zeitpunkt erst den Schnitt in der Mittelebene, dann den Schnitt fast am Rand.

Wir erkennen deutlich den zusätzlichen Einfluß des Randes, es finden weit stärkere Verwirbelungen statt, als in der Mittelebene, wir haben sogar Rückflußgebiete (siehe vor allem im unteren Bild am linken offenen Rand).

Wenn wir wie in zwei Dimensionen wieder die Geschwindigkeits- bzw. Druckkomponente im Punkt $(0,0)$ betrachten, so erhalten wir die folgenden Graphiken:

Wir haben für die x - und y -Komponente eine sehr gute Übereinstimmung zu den Ergebnissen in zwei Raumdimensionen, der Druck dagegen zeigt ein sehr unstetiges Verhalten, hier erkennt man das weitaus größere h gegenüber den 2D Rechnungen.

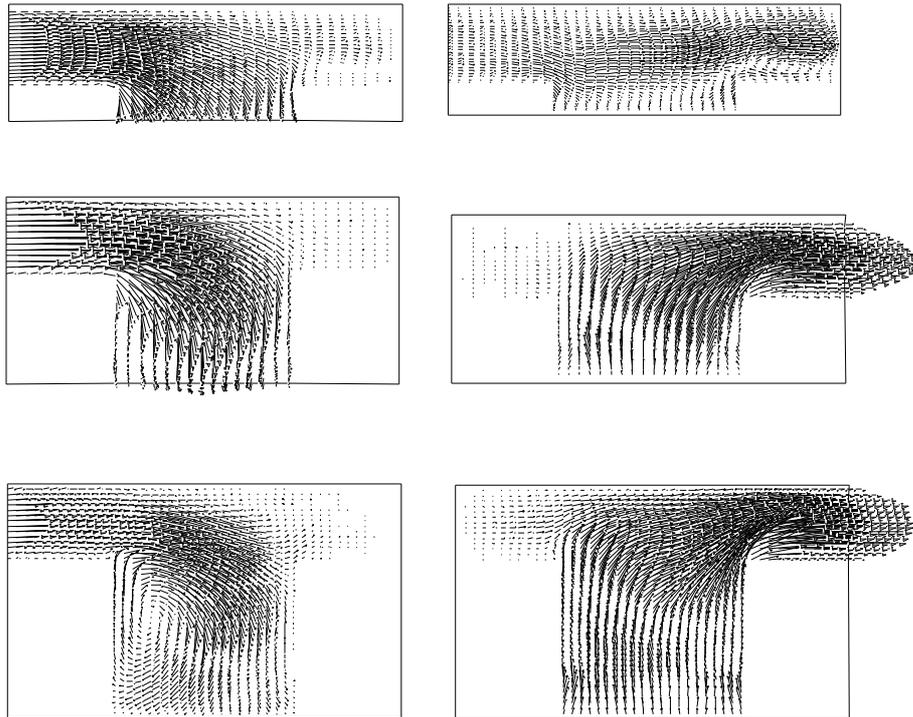


Abbildung 5.19: Geschwindigkeiten in Schnittebene für 3D-Wasserpumpe

5.4.3. Ein drehendes Rotorblatt

Wir greifen mit diesem numerischen Beispiel schon zum Teil dem nächsten Kapitel vorweg. Da wir diese Ausführungen aber nur als gewisse Vorarbeiten betrachten, finden die nachfolgenden Resultate und Techniken noch keinen Eingang in die Umsetzung des praktischen Problems der numerischen Simulation der Brennerströmung.

In folgendem definieren wir einen Rotor mit 4 Rotorblättern. Der Rotor dreht sich dabei nach einer Umdrehungsvorschrift, die im wesentlichen durch trigonometrische Funktionen beschrieben ist. Als Randbedingung auf den Rotorblättern geben wir als Dirichletbedingungen gerade die Ableitung der Umdrehungsfunktion an. In unserem Beispiel dreht sich der Rotor gegen den Uhrzeigersinn, am rechten Rand haben wir eine Dirichleteinströmbedingung, oben, unten und links am Rand haben wir Neumannrandbedingung (siehe [27]).

Wir erkennen, daß durch die Drehung des Rotors bedingt, die Strömung auch gegen den Uhrzeigersinn verläuft. Am oberen Ausströmrand geht deutlich mehr hinaus, als am Unteren. Der Ausfluß am linken Rand kommt hauptsächlich durch die Strömung

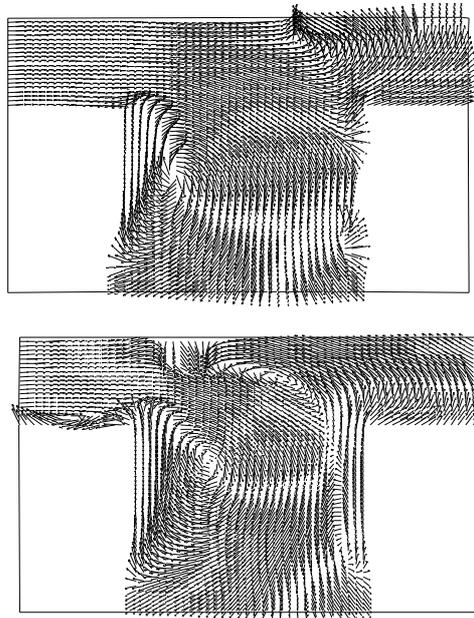


Abbildung 5.20: Unterschiedliche Schnittebenen für 3D–Wasserpumpe

von oben zustande. Die Druckisolines sehen folgendermaßen aus:

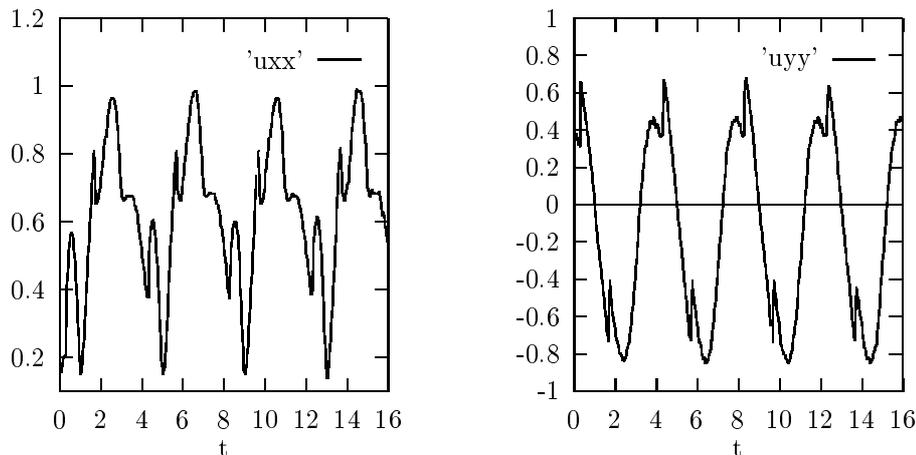


Abbildung 5.21: x- und y-Komponente der Geschwindigkeit in (0,0)

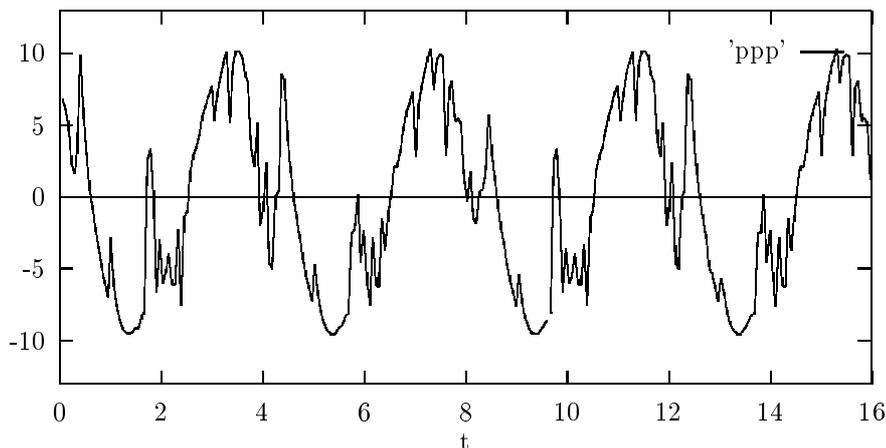


Abbildung 5.22: Druck in (0,0)

Die bisherigen Resultate sind aus Gründen der Visualisierung (in drei Dimensionen können wir z.B. keine Isolinien darstellen) auf zwei Dimensionen beschränkt gewesen, das Problem, das wir tatsächlich angehen wollen, betrifft die Annahme einer vollständigen Vermischung einer Zweiphasenströmung durch die Verwirbelung und Beschleunigung in einem Mischrad (siehe Kapitel 6). Wir greifen mit den nachfolgenden Betrachtungen zum Teil dem nächsten Kapitel vorweg, da wir aber hier noch keine physikalisch relevanten Daten verwenden, sondern lediglich die Techniken zum späteren Einbau in ein Mehrphasenmodell bereit stellen wollen, kann man dies als separates Problem formulieren.

Als Test und quasi Vorlaufrechnung zu diesem Problem betrachten wir einen Kanal,

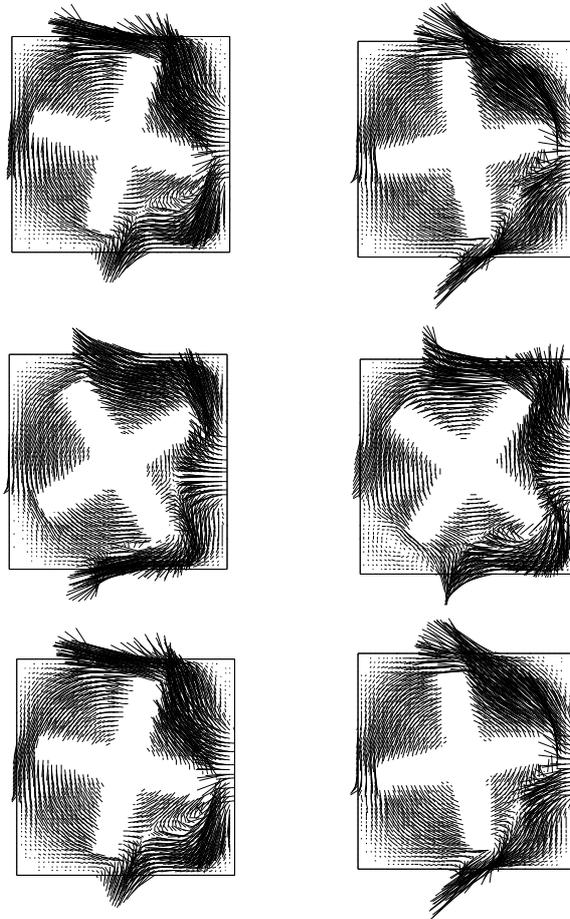


Abbildung 5.23: Bewegung eines Rotorblattes

in dessen Mitte sich ein Laufrad befindet. Wir geben als Randbedingung auf der einen Seite die Neumann'schen "do nothing" Bedingungen vor, auf der anderen Seite zwei Randbereiche auf denen wir Dirichlet Einströmbedingungen vorschreiben. Dies soll später einmal die Situation der Luft/Methan Einströmung simulieren. An Geometriedaten haben wir (in Zentimetern):

- Kanallänge –breite –höhe: $1.8 \times 0.6 \times 0.6$
- Abmessungen des Laufrades: Radius: 0.2, Breite: 0.4, Dicke: 0.1
- Umdrehungsgeschwindigkeit: Π s für eine Umdrehung
- Viskosität: $\nu = 1/100$
- Einströmbedingung: 1cm/s horizontal, 1.4cm/s vertikal

Damit erhalten wir die folgenden Ergebnisse, das erste Bild zeigt einen Zyklus einer

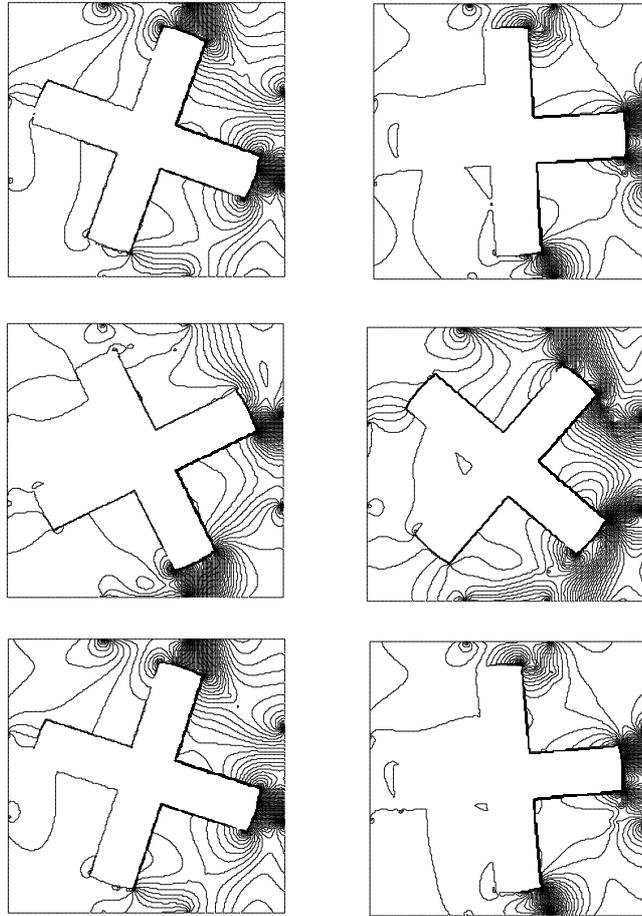


Abbildung 5.24: Druckisolines für das Rotorblatt

Umdrehung.

Die weiteren Abbildungen zeigen eine Druckisofläche (hier sieht man gut die Kontur des Laufrades) und eine Geschwindigkeitsisofläche.

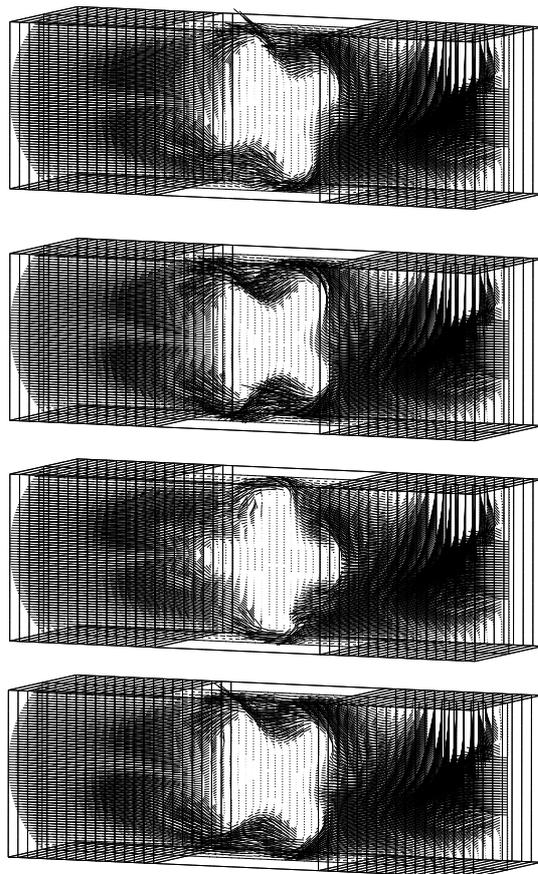


Abbildung 5.25: 2D Schnittebene für Geschwindigkeiten

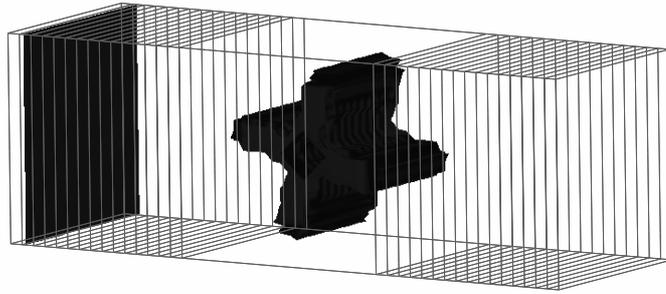


Abbildung 5.26: Druckisoffläche

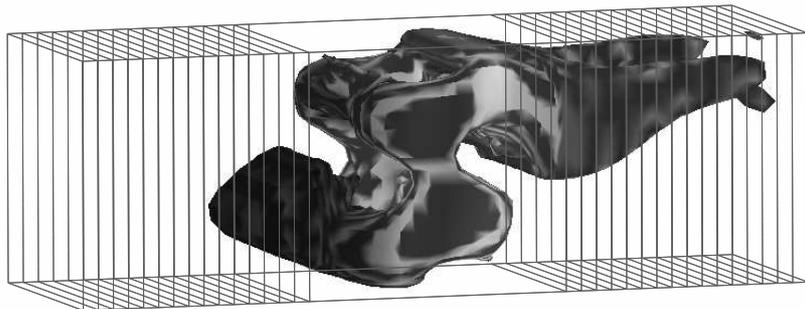


Abbildung 5.27: Geschwindigkeitsisoffläche

6. Das Industrieprojekt mit der Firma Vaillant

Da diese Arbeit im Rahmen des BMBF–Verbundprojektes “Anwendungsorientierte Mathematik“ entstand, soll dieses Kapitel dazu dienen, das Projekt vorzustellen und zu zeigen, inwieweit die vorgestellten Verfahren und Algorithmen in der Lage sind ein praktisches Problem zu bewältigen. In diesem Falle handelt es sich um eine Zusammenarbeit mit der Firma Joh. Vaillant (Remscheid). Ziel dieses Projektes ist es, ein numerisches Modell zu entwickeln, das die Strömung (und Verbrennung) in einem Heizbrenner simuliert. Das Projekt ist dabei in zwei Bereiche unterteilt, einmal die Strömung des Luft–Gas Gemisches im Mischraum bis zu einer Keramikplatte, auf der die Flamme steht, diese Hälfte wird hier in Heidelberg bearbeitet, die Simulation der Flamme wird von der Gruppe von Professor Bader in Cottbus bearbeitet (siehe auch [40]).

Die Fragestellung des Industriepartners beziehen sich dabei auf eine Optimierung der Geometrie des Brenners dahingehend, daß der Schadstoffausstoß minimiert wird. Dies ist gleichbedeutend mit einer gleichmäßigen Flammenausbildung auf der Keramikplatte und dies wiederum mit einer gleichmäßigen Ausbreitung des Luft–Gas Gemisches unter der Keramikplatte im Brennerraum. Dies ist das Problem, daß wir mit Hilfe unserer in den letzten Kapitel vorgestellten Algorithmen und Lösungsansätzen angehen können. Ein Problem stellt sich hierbei in der Frage der Randbedingungen für die Modellierung der Keramikplatte, dies läßt sich mit den schon in dem letzten Kapitel besprochenen Formulierungen behandeln, hier werden wir sogenannte “künstliche Randbedingungen“ verwenden, die bei numerischen Tests, sowohl qualitativ, als auch quantitativ gute Ergebnisse liefern. Es sei an dieser Stelle aber noch erwähnt, daß wir mit den heutigen Mitteln noch nicht in der Lage sind, die Physik eines solchen Problems vollständig zu erfassen, sondern, daß diese Arbeiten unter dem Aspekt zu bewerten sind, Vorarbeiten zu leisten, die in Zukunft dazu dienen sollen, immer bessere Methoden zu entwickeln, um ein solch komplexes Problem tatsächlich numerisch simulieren zu können.

6.1. Problemstellung

Eine wesentliche Arbeit stellt die Gittergenerierung des Brennermodells dar. Da wir, wie in den vorangegangenen Kapiteln beschrieben, zum Lösen der auftretenden Gleichungssysteme Mehrgitterverfahren benutzen, benötigen wir eine Gitterhierarchie, und vor allem ein relativ “grobes“ Grobgitter. Üblicherweise verwenden wir Grobgitter bis zu 500 Zellen, feinere Grobgitter würden das Lösen der Grobgitterkorrekturgleichungen so erschweren, daß das ganze Mehrgitterverfahren nicht mehr effektiv genug wäre. Andererseits hat man dadurch aber das Problem, daß man feine Strukturen der Geometrie nicht exakt wiedergeben kann. Wir verwenden daher wieder die Strategie, wie schon beim “puff-puff“ Problem (siehe Abschnitt 5.3), daß wir das Gitter von Level zu Level anpassen an die vorhandene Geometrie. Die folgenden Zeichnungen der Geometrie des Brenners wurden uns von der Firma Vaillant zur Verfügung gestellt. Wir verzichten aber auf die Angaben der exakten Abmessungen und beschränken uns auf die reine Geometrie.

1.) Die Geometrie des Brenners:

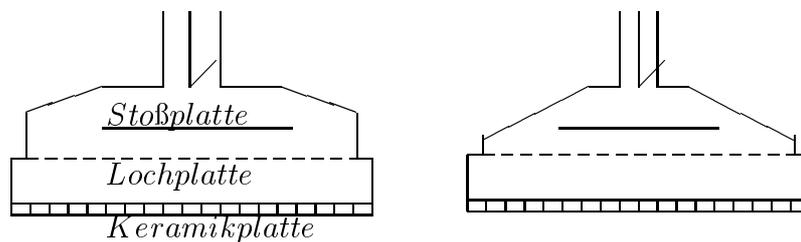


Abbildung 6.1: Geometrie des Brenners von zwei Seitenansichten

Die Strömung die es für uns zu berechnen gilt hat folgenden Charakter: Das Luft-Gas Gemisch (unter Annahme einer optimalen Vermischung) kommt mit einer relativ hohen Geschwindigkeit in den Brennerraum, dort trifft es auf die Stoßplatte, die quasi Energie aus dem System nimmt, hinter der Stoßplatte geht es dann durch die Lochplatte unterhalb der Keramikplatte und durch diese dann hindurch. Um ein gewisses Gefühl für die Physik des Problems zu bekommen, arbeiten wir zuerst mit einem 2D Modell.

2.) Die 2D Diskretisierung:

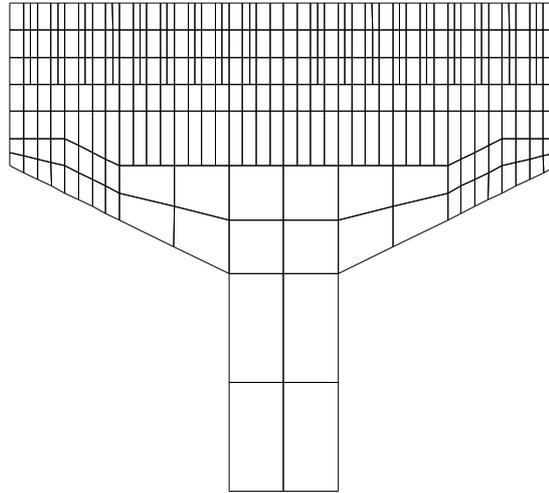


Abbildung 6.2: 2D Grobgitter für Brennermodell

In dieses Grobgitter ist die Stoßplatte eingebaut, sowie die Lochplatte, die wir im 2D Fall auflösen können, und die hier aus 20 Löchern besteht. Die Keramikplatte können wir natürlich mit ihren ca.7000 Löchern (in 2D ca.100 Löcher) nicht auflösen. Hier haben wir folgende Strategie: Wir verfeinern obiges Grobgitter und bauen dann Level-, bzw. h -abhängigen Dirichletrand ein (siehe Abschnitt 5.4). Mit dieser Strategie können dann auf dem feinsten Gitter sehr einfach die Vorgaben der tatsächlichen Geometrie erfüllt werden. Gleichzeitig haben wir aber für unsere Mehrgitterstrategie ein "grobes" Grobgitter und lösen auf allen Leveln der Hierarchie das gleiche Dirichletproblem, d.h. wir können gute Konvergenz erwarten. Diese Technik ist vor allem im 3D Fall sehr nützlich. Der einzige Nachteil ist, daß wir auch dort verfeinern und Elemente haben, wo wir Dirichletrand haben, d.h. wir "verschenken" damit Unbekannte, bzw. machen das Problem größer. Im vorigen Kapitel haben wir diese Techniken ausreichend behandelt.

3.) Die 3D Diskretisierung:

Mit obigen Zeichnungen und den entsprechenden Abmessungen, konstruieren wir folgendes Grobgitter mit 248 Elementen. Man erkennt hier die Stoßplatte, die Keramikplatte wird nicht in die Triangulation mit einbezogen, da wir hier entsprechende Randbedingungen wählen können.

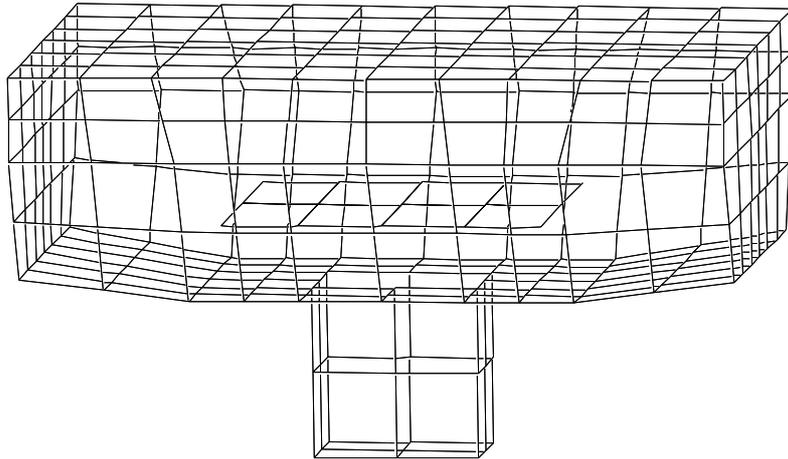


Abbildung 6.3: Grobgitter für Brennermodell

Wie oben bereits erwähnt, haben wir mit solch einem Grobgitter keine Möglichkeit die Struktur der Lochplatte wiederzugeben. Die reale Platte hat ungefähr 700 Löcher. Wir gehen daher anders vor:

Ein 2D Schnitt unseres Grobgitter an der Plazierung der Lochplatte sieht folgendermaßen aus.

	×		×		×		×		×
×		×		×		×		×	
	×		×		×		×		×
×		×		×		×		×	
	×		×		×		×		×
×		×		×		×		×	

Abbildung 6.4: 2D Schnitt bei Lochplatte

Wir schreiben also für das Grobgitter überall dort wo Kreuze sind einen fiktiven Rand vor. Wir haben dann natürlich nur 30 Löcher. Dieselbe Option machen wir aber auch bei den nächsten beiden Verfeinerungen, wir haben dann 121 Löcher auf dem nächsten und 481 Löcher auf dem übernächsten Level. Diese 481 Löcher stimmen dann sehr

gut mit der Wirklichkeit überein, vor allem auch noch unter dem Aspekt, daß sie eine Größe von ca. $0.5\text{cm} \times 0.5\text{cm}$ haben, was ebenfalls nahezu der tatsächlichen Geometrie entspricht. Dieses Vorgehen ist nötig, damit wir auf allen Leveln die richtigen physikalischen Informationen mitgeben. Würden wir für alle Level die 481 Löcher vorgeben, hätten wir unter Umständen die Tatsache, daß auf den niedrigsten Leveln keine Flächenmitten in einem Loch liegen, d.h. wir würden auf diesen Leveln nicht das volle Problem lösen, was zu einer Beeinträchtigung der Konvergenzgeschwindigkeit führen würde.

6.2. Die Physik des Problems

Nachdem wir die Geometrie und die Umsetzung in eine Finite-Elemente-Diskretisierung betrachtet haben, gehen wir nun auf die Physik des Problems ein. Die folgenden Daten wurden uns hierbei von der Firma Vaillant zur Verfügung gestellt.

Bei einer Belastung von 20 kW haben wir folgenden Volumenstrom und Konzentrationen:

- Volumenstrom: $0.007188\text{ m}^3/\text{s}$ (273)K
- CH_4 : 7.75 Vol.
- O_2 : 19.37 Vol.
- N_2 : 72.88 Vol.

Bei der Teillast von 10 kW ergibt sich die Hälfte des Volumenstromes, bei gleichen Konzentrationen.

Der gemessene Druckverlust beträgt 3 Pascal bei $20\text{W}/\text{cm}^2$, bei einer Gemischtemperatur von 50 C.

Unter Einbeziehung dieser Angaben errechnet sich die Viskosität als (siehe [4]):

- $\nu = 18.6 \cdot 10^{-6}\text{ m}^2/\text{s}$

Legt man zur Berechnung der Geschwindigkeit ein Rohrprofil zugrunde, so erhält man bei einer angenommenen Vollast eine Einströmgeschwindigkeit von ca. 5 m/s.

Damit haben wir es mit einem Problem zu tun, dessen *Reynoldszahl* im Bereich von $10^3 - 10^4$ liegt. Damit ist es auch gerechtfertigt, zunächst mit unserem Navier–Stokes Löser das Problem zu behandeln, also einen laminaren Ansatz zu verwenden. Die Ankopplung eines Turbulenzmodells wird in einem der nächsten Schritte erfolgen.

Weitere Annahmen (bzw. Vereinfachungen) sind:

–Verzicht auf Einbeziehung der Temperatur. Da das Gemisch mit Zimmertemperatur aus der Leitung kommt und die Temperatur im Vormischraum nur ca. 50 C beträgt, kann man den Temperaturgradienten vernachlässigen.

–Wir setzen optimale Vermischung voraus, d.h. wir rechnen mit einem einkomponentigen Medium.

6.3. Numerische Ergebnisse für den Brenner

Unsere numerischen Ergebnisse beziehen sich vor allem auf die Frage, welche Veränderungen an der Geometrie erfolgversprechende Auswirkungen auf das grundsätzliche Problem einer optimalen Verteilung (und damit Minimierung des Schadstoffausstoßes) des Luft/Gas–Gemisches unterhalb der Keramikplatte haben können. Dabei sind wir aber an Vorgaben gebunden. Wir können im wesentlichen nur die Position und Größe der beiden Platten (Stoß– und Lochplatte) verändern, das Gehäuse des Brenners, sowie die Rechteckform der Keramikplatte können aus bautechnischen Gründen nicht verändert werden.

Die Frage der Verteilung untersuchen wir, indem wir den Flux am Austrittsrand (also an der Keramikplatte) messen, unter verschiedenen Positionen und Größen der Platten, sowie unter verschiedenen Einströmbedingungen, die je nach Betriebsart (Voll– bzw. Teillast) schwanken kann.

6.3.1. Der 2D–Fall

Mit dem Grobgitter aus Abschnitt 6.1 erhalten wir bei Vollast folgende qualitativen Ergebnisse für Druck und Geschwindigkeiten (siehe auch Seite 156):

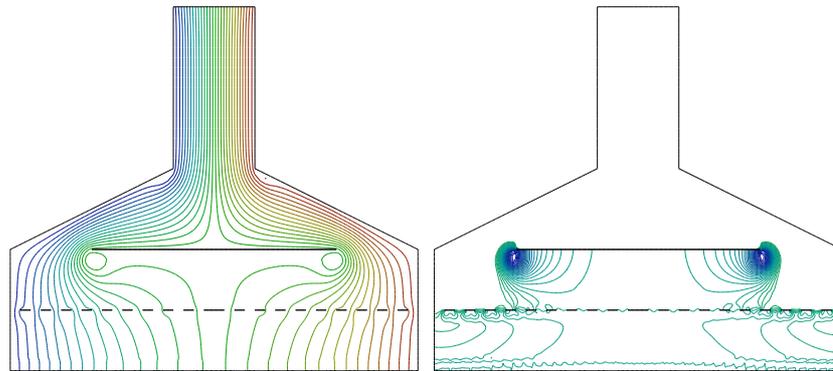


Abbildung 6.5: 2D Geschwindigkeiten und Druck

Man erkennt schon in diesen Bildern, daß die Ausflußmenge am Ausströmrand starke Schwankungen aufweist, dies wird untermauert durch die graphische Darstellung des Fluxes.

Quantitativ gemessen erhalten wir folgende Darstellungen für den Flux, das erste Bild zeigt Ergebnisse für die gleiche Geometrie zu unterschiedlichen Zeitpunkten (Flux 1 zu Beginn,...,Flux 4 am Ende der Rechnung), das zweite Schaubild zeigt Ergebnisse zum gleichen Zeitpunkt, aber unterschiedliche Konfigurationen. Flux 1 ist dabei wie im ersten Bild, 0.5vel bei halber Einströmgeschwindigkeit und plate+0.5 bedeutet, daß die Stoßplatte um 0.5cm nach vorne gerückt ist

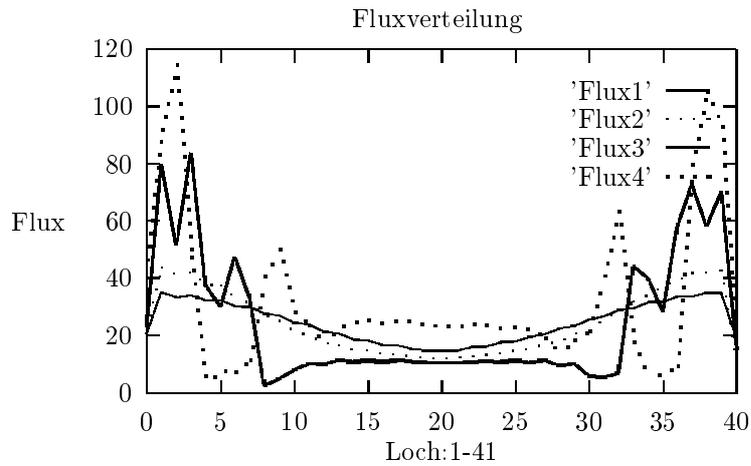


Abbildung 6.6: Flux zu verschiedenen Zeitpunkten

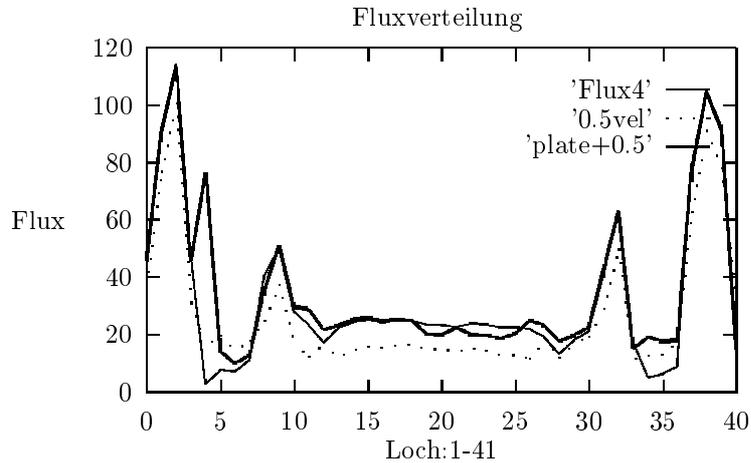


Abbildung 6.7: Flux zu verschiedenen Konfigurationen

Diese Ergebnisse in zwei Dimensionen, lassen erste Rückschlüsse auf die Struktur der Strömung zu. Den Verlauf der instationären Rechnung zu obigen Graphiken sieht man in den nächsten Bildern, auf denen normalisierte Geschwindigkeitsvektoren (d.h. Betrag der Geschwindigkeit wird durch Einfärbung dargestellt) zu zwei verschiedenen Zeitpunkten (entsprechend zu “Flux1“ und “Flux4“) abgebildet sind (siehe auch Seite 157).

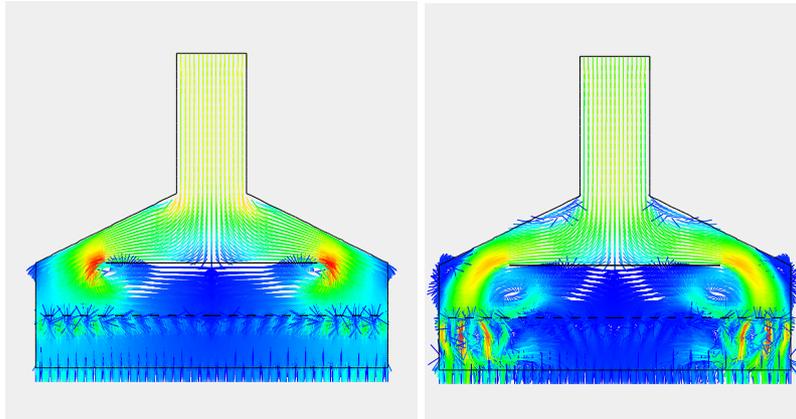


Abbildung 6.8: 2D Geschwindigkeiten

Durch die Komplexität der Geometrie mit der Stoßplatte entstehen starke Verwirbelungen, die teilweise durch den Einbau der Lochplatte wieder abgedämpft werden. Dies zeigen auch numerische Tests, die ohne die Lochplatte durchgeführt wurden und bei denen die Verwirbelung hinter der Stoßplatte so stark waren, daß der instationäre Löser nach wenigen Zeitschritten nicht mehr konvergierte. Die Lochplatte zerstört praktisch diese Effekte und homogenisiert die Strömung. Weiter sehen wir an den Bildern bereits, daß die Lösung instationären Charakter hat, was dann natürlich in drei Dimensionen enormen Rechenaufwand erfordert, um aussagekräftige Ergebnisse zu bekommen.

6.3.2. Der 3D-Fall

Hier verwenden wir ebenfalls das Grobgitter aus Abschnitt 6.1, einen ersten qualitativen Eindruck der Fluxverteilung vermittelt dabei folgendes Bild (siehe auch Seite 157)

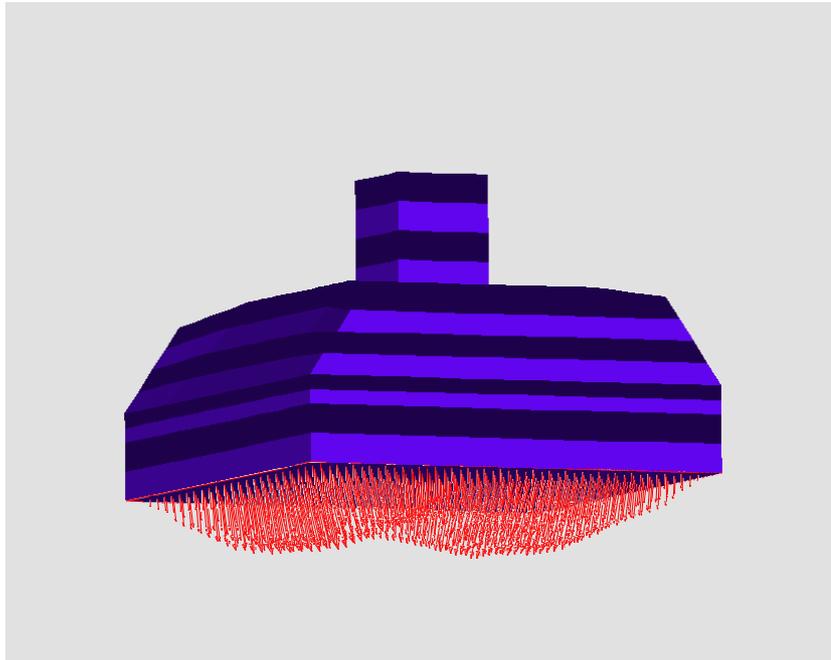


Abbildung 6.9: 3D Fluxdarstellung

In der Darstellung mit normalisierten Geschwindigkeitsvektoren erhalten wir folgende Darstellung (siehe auch Seite 157):

Man sieht auch hier die gleichen Verhältnisse wie in 2 Dimensionen, d.h. wir haben an den Rändern stark ausgeprägte peaks, in der Mitte dagegen wenig Ausströmung.

Messen wir wieder wie in zwei Dimensionen den Betrag des Fluxes am Ausströmrand, ergibt sich folgendes Verhalten des Fluxes an zwei ausgewählten Punkten des Ausströmrandes. Der erste Punkt (Flux1), liegt genau in der Mitte, der zweite Punkt (Flux2) liegt ziemlich nahe am Rand. Wir geben in folgender Abbildung das Ausströmverhalten an diesen zwei Punkten für einen Zeitraum von 0–13 Sekunden wieder.

Es bestätigen sich die Eindrücke des 2D-Falles. In der Mitte des Ausströmrandes geht wesentlich weniger hinaus als in der Nähe des Randes. Auch zeigt sich ein instationäres Verhalten, d.h. es ist nicht zu erwarten, daß die Lösung in einen stationären Limes konvergiert.

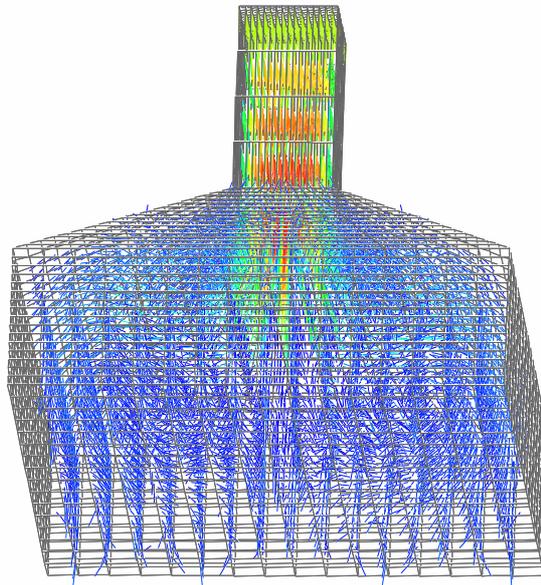


Abbildung 6.10: 3D Geschwindigkeitsvektoren

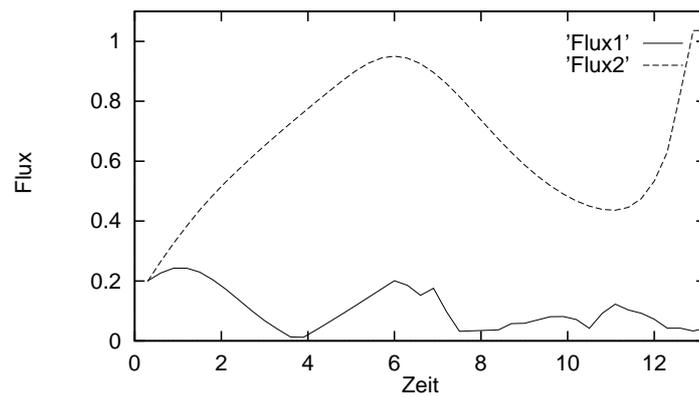


Abbildung 6.11: Flux an zwei Punkten gemessen

7. Zusammenfassung und Ausblick

Eine Zusammenfassung der vorherigen Resultate zeigt, daß wir auf der Grundlage des nichtkonformen *rotiert trilinearen* Finite-Element-Ansatzes ein Lösungsverfahren erhalten haben, daß den Ansprüchen eines praxisnahen und problemorientierten Lösungsverfahrens für die instationären, inkompressiblen Navier-Stokes-Gleichungen in drei Raumdimensionen genügt.

Es arbeitet mit einer festen (fast) problemunabhängigen Parametereingabe sehr effizient (instationäre Rechnungen können auf einer IBM RS6000/590 Workstation mit bis zu 10 Millionen Unbekannten innerhalb eines Tages ausgeführt werden), robust (gegenüber unterschiedlichen Gebietsformen, Gittertypen, Viskositäten und Eingabedaten) und hinreichend genau (Vergleiche mit Benchmark Tests und Experimenten). Darüberhinaus findet mit Hilfe von Visualisierungstechniken (Filmgenerierung, 3D Particle Tracing) eine gezielte Aufbereitung der gerechneten Daten statt, so daß das gesamte Programmpaket ein effizientes Werkzeug zur Simulation und Darstellung dreidimensionaler laminarer Strömungen anbietet.

Wichtig zu erwähnen bleibt die Tatsache, daß diese Arbeit ein Baustein in einem Gesamtrahmen darstellt. Viele Techniken und Algorithmen, die bereits für andere Probleme entwickelt und getestet wurden, konnten hier mit entsprechenden Modifikationen weiterverwendet werden. Da die erstellten Programme auf der Basis der Programmpakete **FEAT2D**, **FEAT3D** und **FEATFLOW** erstellt wurden, ist eine natürliche Fortsetzung für weitere Verwendung gewährleistet.

Dies wird dokumentiert durch die Arbeiten von H.Oswald, der an einer parallelisierten Version des *Diskreten Projektionsverfahrens* arbeitet. Hier treten dann andere Schwerpunkte in den Vordergrund, wie z.B. Implementierung eines parallelen Mehrgitterverfahrens, blockstrukturierte Iterationsverfahren und *Domain Decomposition* Verfahren. Erste Ergebnisse am Beispiel der Driven Cavity geben Anlaß zur Hoffnung, daß die Parallelisierung eine derartige Effizienzsteigerung bewirkt, daß Probleme mit bis zu

100 Millionen Unbekannten in wenigen Stunden gelöst werden können.

Eine weitere Ausdehnung der Techniken für die Behandlung von zeitabhängigen Rändern wird von D.Engelmann benutzt, der das Problem von Zweiphasenströmungen damit behandelt. Hier soll ein innerer Rand die Trennschicht zwischen Medien unterschiedlicher Viskosität beschreiben. Damit soll im wesentlichen der Impuls- und Gasaustausch beschrieben werden. Praktische Anwendung ist dabei z.B der Gasaustausch in Abhängigkeit der Strömungsdynamik nahe einer bewegten Wasseroberfläche.

Schließlich werden basierend auf diese Lösungsverfahren zusätzliche Komponenten zum besseren Verständnis des Brennermodells eingebaut. Dabei ist vor allem an eine schwachkompressible Erweiterung gedacht, d.h. die Bedingung $\nabla \cdot u = 0$ wird abgeschwächt, wodurch das Gesamtproblem weniger steif wird. Als weiterer Schritt ist dann die Ankopplung der Temperatur vorgesehen, was zu den *Boussinesq-Gleichungen* führt. Dies ist als erster Schritt in Richtung eines praxisnahen Turbulenzmodells zu sehen.

Literaturverzeichnis

- [1] Bank, R.E., Dupont, T.: *An optimal order process for solving finite element equations*, Math. Comp., 36, 35–51 (1981)
- [2] Becker, R.: *An Adaptive Finite Element Method for the Incompressible Navier-Stokes Equations on Time-dependent Domains*, Dissertation, Heidelberg (1995)
- [3] Becker, R., Rannacher, R.: *Finite element discretization of the Stokes and Navier-Stokes equations on anisotropic grids*, Proc. 10th GAMM-Seminar, Kiel, January 14–16, 1994 (G. Wittum, W. Hackbusch, eds.), Vieweg
- [4] Beitz, W., Küttner, K.-H.: *Dubbel*, Taschenbuch für den Maschinenbau, 18. Auflage, Springer
- [5] Blum, H., Harig, J., Müller, S.: **FEAT** . *Finite element analysis tools. Release 1.3. User Manual*, Technischer Report Nr. 554, SFB 123, Universität Heidelberg, 1992
- [6] Blum, H.: *Asymptotic error expansion and defect correction in the Finite Element Method*, Habilitationsschrift, Heidelberg (1991)
- [7] Braess, D., Hackbusch, W.: *A new convergence proof for the multigrid method including the V-cycle*, SIAM J. Numer. Anal., 20, 967–975 (1983)
- [8] Braess, D., Verfürth, R.: *Multi-grid methods for non-conforming finite element methods*, Technischer Report Nr. 453, SFB 123, Universität Heidelberg, März 1988
- [9] Brenner, S.C.: *A nonconforming multigrid method for the stationary Stokes equations*, Math. Comp., 55, 411–437 (1990)
- [10] Brenner, S.C.: *An optimal order multigrid method for P1 nonconforming finite elements*, Math. Comp., 52, 1–15 (1989)

- [11] Brenner, S.C.: *An optimal order nonconforming multigrid method for the biharmonic equation*, SIAM J. Numer. Anal., 26, 1124–1138 (1989)
- [12] Ciarlet, Ph.G.: *The Finite Element Method for Elliptic Problems*, North-Holland 1976
- [13] Crouzeix, M., Raviart, P.A.: *Conforming and non-conforming finite element methods for solving the stationary Stokes equations*, R.A.I.R.O. **R-3**, 77–104 (1973)
- [14] Chorin, A.J.: *Numerical solution of the Navier–Stokes equations*, Math. Comp., 22, 745–762 (1968)
- [15] Cuthill, E., McKee, J.: *Reducing the bandwidth of sparse symmetric matrices*, in Proc. ACM Nat. Conf., 157–172, New York 1969
- [16] Deville, M., L e, T-H., Morchoisne, Y.: *Numerical Simulation of 3-D Incompressible Unsteady Viscous Laminar Flows*, Notes on Numerical Fluid Mechanics, Volume 36
- [17] Dupont, T., Scott, R.: *Polynomial Approximation of Functions in Sobolev Spaces*, Math. Comp., 34, 441–463 (1980)
- [18] Finite Element Handbook *Discretization Methods for Stable Initial Value Problems*, Springer, LNM Vol. 1044, 1984
- [19] Gekeler, E.: *Discretization Methods for Stable Initial Value Problems*, Springer, LNM Vol. 1044, 1984
- [20] Girault, V., Raviart, P.A.: *Finite Element Methods for Navier–Stokes equations*, Springer, Berlin–Heidelberg 1986
- [21] Gresho, P.M.: *On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix, Part 1: Theory*, Int. J. Numer. Meth. Fluids, 11, 587–620 (1990). *Part 2: Implementation*, Int. J. Numer. Meth. Fluids, 11, 621–659 (1990)
- [22] Hackbusch, W.: *Multi-Grid Methods and Applications*, Springer, Berlin–Heidelberg 1985

- [23] Harig, J., Schreiber, P., Turek, S.: **FEAT3D** . *Finite element analysis tools in 3 Dimensions. Release 1.2. User Manual*, Technischer Report Nr. 94–19, SFB 359, Universität Heidelberg, 1994
- [24] Harig, J.: *Eine robuste und effiziente Finite Elemente Methode zur Lösung der inkompressiblen 3-D Navier–Stokes Gleichungen auf Vektorrechnern*, Dissertation, Heidelberg 1991
- [25] Harig, J.: *Visualization of 3-D Finite Element Solutions of the Navier–Stokes Equations*, Technischer Report Nr. 2, IWR Heidelberg, 1990
- [26] Hecht, F.: *Construction d’une base de fonctions P_1 non conformes á divergence nulle dans R^3* , R.A.I.R.O. Numer. Anal., 15, 119–150 (1981)
- [27] Heywood, J.G., Rannacher, R., Turek, S.,: *Artificial Boundaries and Flux and Pressure Conditions for the Incompressible Navier–Stokes Equations*, Int. J. Numer. Meth. Fluids, 22, 325–352 (1996)
- [28] Hughes, T.J.R., Franca, L.P., Balestra, M.: *A new finite element formulation for computational fluid mechanics: V. Circumventing the Babuska–Brezzi condition: A stable Petrov–Galerkin formulation of the Stokes problem accomodating equal order interpolation*, Comp. Meth. Appl. Mech. Eng., 59, 85–99 (1986)
- [29] Koseff, J.R., Street, R.L.: *The Lid–Driven Cavity–Flow: A Synthesis of Qualitative and Quantitativ Observations*, Journal of Fluids Engineering, 106, 390–398, (1984).
- [30] Lambert, J.D.: *Computational Methods in Ordinary Differential Equations*, John Wiley 1973
- [31] Langer, A.: *Zur Konstruktion diskret divergenzfreier Basen für nicht–konforme Finite–Elemente–Diskretisierungen des Stokes–Problems im R^3* , Diplomarbeit, Dortmund 1994
- [32] Lin, Qun: *Error resolution of Finite Element Method* Abstract, Beijing, 1993
- [33] Lube, G.: *Stabilized Galerkin Finite Element Methods for Convection Dominated and Incompressible Flow Problems*, Banach Center Publications, Volume 29, Warschau 1994
- [34] Müller, S., Prohl, A., Rannacher, R., Turek, S.: *Implicit time–discretization of the nonstationary incompressible Navier–Stokes equations*, Proc. 10th GAMM–Seminar, Kiel, January 14–16, 1994 (G. Wittum, W.Hackbusch, eds), Vieweg

- [35] Morley, L.S.D.: *The triangular equilibrium problem in the solution of plate bending problems*, Aero. Quart., 19, 149–169 (1968)
- [36] Ohmori, K., Ushijima, T.: *A Technique of Upstream Type Applied to a Linear Nonconforming Finite Element Approximation of Convective Diffusion Equations*, R.A.I.R.O. Numer. Anal., 18, 309–332 (1984)
- [37] Pironneau, O.: *On the transport–diffusion algorithm and its applications to the Navier–Stokes equations*, Numer. Math., 38, 309–332 (1982)
- [38] Rannacher, R.: *Numerical Analysis of Nonstationary Fluid Flow (A survey)*, Technischer Report Nr. 492, SFB 123, Universität Heidelberg, November 1988
- [39] Rannacher, R., Turek, S.: *A simple nonconforming quadrilateral Stokes element*, Numer. Meth. Part. Diff. Equ., 8, 97–111 (1992)
- [40] Rannacher, R., Schreiber, P., Turek, S.: *Numerische Modellierung von Gasbrennern*, Beitrag zum Proceedingsband des Statusseminars des Verbundprojektes des BMBF 1995, Springer Verlag (1996)
- [41] Schäfer, M., Turek, S.: *Benchmark Computations of Laminar Flow around a Cylinder*, Preprint 96–03 SFB 359, Universität Heidelberg, Januar 1996
- [42] Schieweck, F., Grambow, W., Risch, U.: *Experiences with the Multigrid Method applied to high Reynolds Number, steady, incompressible flow*, WZ TU Magdeburg, 6/90, Februar 1990
- [43] Schieweck, F., Tobiska, L.: *An error estimate of optimal order for a nonconforming upwind finite element discretization of the Navier–Stokes equations*, Preprint Nr.13, TU Magdeburg, 1994
- [44] Schreiber, R., Keller, H.B.: *Driven Cavity Flows by Efficient Numerical Techniques*, J. of Comp. Phys., 49, 310–333 (1983)
- [45] Schreiber, P.: *Eine nichtkonforme Finite Elemente Mehrgittermethode zur Lösung der biharmonischen Gleichung*, Diploma Thesis, Heidelberg (1992)
- [46] Schreiber, P., Turek, S.: *An efficient finite element solver for the nonstationary incompressible Navier–Stokes equations in two and three dimensions*, Proc. Workshop “Numerical Methods for the Navier–Stokes Equations“, Heidelberg, Oct. 25–28, 1993, Vieweg

- [47] Temam, R.: *Theory and Numerical Analysis of the Navier–Stokes Equations*, North–Holland 1977
- [48] Turek, S.: *Ein robustes und effizientes Mehrgitterverfahren zur Lösung der instationären, inkompressiblen 2D Navier–Stokes–Gleichungen mit diskret divergenzfreien finiten Elementen*, Dissertation, Heidelberg 1991
- [49] Turek, S.: *On discrete projection methods for the incompressible Navier–Stokes equations: An algorithmical approach*, Preprint 94–70 SFB 359, Heidelberg 1994
- [50] Turek, S.: *FEATFLOW*, Preprint 95–85 SFB 359, Heidelberg 1995
- [51] Tobiska, L.: *Full and weighted Upwind Finite Element Methods*, In: *Splines in Numerical Analysis*, ed. von J.W. Schmidt und H. Späth, Internationales Seminar ISAM 89 in Weißig 24.–28. April 1989, Akademie Verlag Berlin, 1989
- [52] Tobiska, L., Schieweck, F.: *A nonconforming finite element method of upstream type applied to the stationary Navier–Stokes equation*, *MMAN*, 23, 627–647 (1989)
- [53] Van Dyke, M.: *An Album of Fluid Motion*, The Parabolic Press, Stanford, California, 1982
- [54] Vanka, S.P.: *Implicit Multigrid Solutions of Navier–Stokes Equations in Primitive Variables*, *J. of Comp. Phys.*, 65, 138–158 (1985)
- [55] Van Kan, J.: *A second–order accurate pressure–correction scheme for viscous incompressible flow*, *SIAM J. Sci. Stat. Comp.*, 7, 870–891 (1986)
- [56] Wittum, G.: *The use of fast solvers in computational fluid dynamics*, in: “Numerical Methods in Fluid Mechanic“ (P. Wesseling, eds.), Vieweg, Braunschweig 1990