Technische Universität München
Institut für Energietechnik MW7

Lehrstuhl und Laboratorium
für Hydraulische Maschinen und Anlagen

Simulation inkompressibler Strömungen mit unstrukturierten Gittern

Rolf Reinelt

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

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr.-Ing., Dr.-Ing.habil. R. Friedrich
Prüfer der Dissertation:		
	1.	Univ.-Prof. Dr.-Ing., Dr.-Ing.habil. R. Schilling
	2.	Univ.-Prof. Dr.rer.nat. Chr. Zenger

Die Dissertation wurde am 06.09.2001 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 05.03.2002 angenommen.

Vorwort

Die vorliegende Dissertation entstand in den Jahren 1998-2001 während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Hydraulische Maschinen und Anlagen der Technischen Universität München.

Mein Dank gilt Herrn **Univ.-Prof. Dr.-Ing. habil. R. Schilling** für die Betreuung der Arbeit und die Schaffung der notwendigen Rahmenbedingungen, ohne die diese Dissertation nicht möglich gewesen wäre. Außerdem danke ich für die anregenden und fruchtbaren Diskussionen.

Herrn Univ.-Prof. Dr. C. Zenger danke ich nicht nur für die Übernahme des Koreferates, sondern auch für die fachliche Unterstützung. Ebenso bedanke ich mich bei Herrn a. o. Prof. Dr.-Ing. Dr.-Ing. habil. R. Friedrich dafür, daß er den Vorsitz der Prüfungskommission übernommen hat.

Bedanken möchte ich mich auch bei allen meinen Kollegen, insbesondere bei **Dipl.-Ing. B. Szilagyi**.

Abschließend möchte ich mich ganz besonders bei meiner **Familie** und meiner Frau **Heidrun** für deren Unterstützung und Geduld bedanken.

München, 20.3.2002

Rolf Reinelt

Inhaltsverzeichnis

Verwendete Formelzeichen und Abkürzungen **IX**

Zusammenfassung **XIII**

1 Einleitung **1**

1.1 Problemstellung 1

1.2 Stand des Wissens 2

1.2.1 Ausgangsgleichungen 2

1.2.2 Berechnungsmethoden für inkompressible Strömungen 3

1.2.3 Diskretisierungsmethoden 3

1.2.4 Benchmarkergebnisse 4

1.2.5 Implementierungstechniken 5

1.3 Zielsetzung 6

2 Theoretische Grundlagen **7**

2.1 Grundgleichungen in differentieller Form 7

2.2 Grundgleichungen in integraler Form 8

2.3 Kennwerte für Pumpen und Turbinen 8

3 Finite-Volumen-Diskretisierung **11**

3.1 Aufteilung in Kontrollvolumen 11

3.2 Zellenzentriertes FV-Verfahren 12

3.3 Flächenvektoren und Volumenberechnung 13

3.4 Berechnung von Interfacewerten 14

3.5 Berechnung der Gradienten 15

3.6 Berechnung der Massenströme 16

3.7 Diskretisierung des konvektiven Terms 16

3.8 Diskretisierung des diffusiven Terms 17

3.9 Koeffizienten der Impulsgleichung 18
3.10 Druckkorrekturgleichung 19
3.11 Randbedingungen 21

4 Lösungsverfahren **23**
4.1 Eingitterverfahren 23
4.2 Mehrgitterverfahren 25
4.2.1 Mehrgittergleichungen 26
4.2.2 Mehrgitterzyklus 28

5 Implementierung **31**
5.1 Designentscheidungen 32
5.2 Klassenstruktur 32

6 Gittergenerierung **35**
6.1 Klassifizierung der Gittertypen 35
6.2 Anforderungen an Finite-Volumen-Gitter 37
6.3 Gittergenerierung für Turbinen 38
6.3.1 Unstrukturierte Hexaedergitter 39
6.3.2 Konvertierung von blockstrukturierten Gittern 46
6.3.3 Hybridgitter 47

7 Validierung durch Testrechnungen **49**
7.1 Ausgebildete 2D Kanalströmung 50
7.2 Driven Cavity 52
7.3 Driven Cavity mit Symmetrierändern 55
7.4 Driven Cavity mit schrägen Wänden 58
7.5 Driven Cavity mit Hybridgitter 59
7.6 Testfall mit logarithmisch-spiraligen Schaufeln 62
7.6.1 Exakte Lösung 62
7.6.2 Leitgitter mit logarithmisch-spiraligen Schaufeln 64
7.6.3 Rotierendes Schaufelgitter mit logarithmisch-spiraligen Schaufeln 67
7.7 Gostelow-Testfall 69
7.8 Francis-Turbine NQ30 71
7.8.1 Einfluß der Viskosität 71

7.8.2 Einfluß der Diskretisierung der konvektiven Terme 74
7.8.3 Einfluß des Abbruchkriteriums 75
7.8.4 Einfluß der Gitterfeinheit 77
7.8.5 Vergleich mit TASCflow-Ergebnissen 78
7.9 Francis-Turbine NQ50 86
7.9.1 Einfluß der Gitterfeinheit 86
7.9.2 Einfluß der Re-Zahl 87
7.10 Francis-Turbine NQ100 91
7.10.1 Einfluß der Gitterfeinheit 91
7.10.2 Einfluß der Viskosität 93
7.11 Zusammenfassung der Ergebnisse der Testrechnungen 96

8 Bewertung und Ausblick **99**

A Transformation in das rotierende System **101**

B Periodische Randbedingungen **105**

Verwendete Formelzeichen und Abkürzungen

Abkürzungen

2D	zweidimensional
3D	dreidimensional
ADI	Alternating Direction Implicit
CDS	Central Differencing Scheme
CFD	Computational Fluid Dynamics
CG	Conjugate Gradients
CS	Correction Scheme
DNS	Direkte Numerische Simulation
E3D	Dreidimensionale Euler Strömungsrechnung
FD	finite Differenzen
FAS	Full Approximation Scheme
FE	finite(s) Element(e)
FEM	Finite-Elemente-Methode (speziell FEM-Strömungslöser von MÜLLER [34])
FMG	Full Multigrid
FT	Francis-Turbine
FV	finite Volumen
FVM	Finite-Volumen-Methode (speziell FV-Strömungslöser dieser Arbeit)
LES	Large Eddy Simulation
MG	Multigrid-Verfahren
NS	Navier-Stokes
Q3D	Quasi-dreidimensionale Strömungsrechnung
QUICK	Quadratic Upwind Interpolation for Convective Kinematics Scheme
RSM	Reynolds Stress Modell
S2M	Mittelschnittverfahren in der Meridianebene
SG	Single-Grid-Verfahren
SIP	Strongly Implicit Procedure
SIMPLE	Semi Implicit Methode for Pressure Linked Equations
SOR	Successive Overrelaxation
UDS	Upwind Differencing Scheme
TF	TASCflow, ein Navier-Stokes Strömungslöser der Firma AEA
TU	Technische Universität

Lateinische Zeichen

A	Matrixeintrag
B	Drucktermmatrix
E	(East) Index von Größen in der Mitte der Nachbarzelle
F	Fluß
H	Förderhöhe
L_2^p	Fehlerquadratnorm des Drucks
L_2^v	Fehlerquadratnorm der Geschwindigkeit
N	Dimension des Problems; Anzahl der Zellen
P	Index von Größen in der Mitte des Kontrollvolumens
$\mathcal{P}$	Prolongationsoperator
Q	Quellterm; Volumenstrom
$R = r/r_a$	dimensionsloser Radius
$\mathcal{R}$	Restriktionsoperator
Re	Reynolds-Zahl
S	Fläche; Matrix der konvektiven und diffusiven Terme
V	Volumen
c	Absolutgeschwindigkeit
e	(East) Index einer Zellgrenzfläche; Einheitsvektor
g	Erdbeschleunigung
$\dot{m}$	Massenstrom
n_c	Anzahl der Kinderzellen einer Grobgitterzelle
n_f	Anzahl der Begrenzungsflächen einer Zelle
n_e^v	Anzahl der Eckpunkte einer Begrenzungsfläche
n^v	Anzahl der Eckpunkte einer Zelle
n_q	spezifische Drehzahl
p	Druck
r	Radius
t	Zeit
u	Umfangsgeschwindigkeit; erste Geschwindigkeitskomponente
w	Relativgeschwindigkeit
x	x - Koordinate
y	y - Koordinate
z	z - Koordinate

Griechische Zeichen

α	Faktor der Busemann-Lösung
α_p	Unterrelaxationsfaktor für Druckkorrekturgleichung
α_u	Unterrelaxationsfaktor für Impulsgleichung
γ	Faktor zur stufenlosen Auswahl zwischen UDS und CDS; Teilungswinkel
$\delta_{i,j}$	Kronecker-Delta
ε	Abbruchkriterium
ε_{ijk}	Epsilontensor
η	2. Problemkoordinate
η_h	hydraulischer Wirkungsgrad
λ_e	Interpolationsfaktor
μ	Dynamische Viskosität
ν	Kinematische Viskosität; 1. Problemkoordinate
ξ	lokale Koordinate
Π_Δ	Druckkopplungsterm
ρ	Dichte
ϕ	Platzhalter für Unbekannte
φ	Winkel
Ψ	Druckzahl
ω	Winkelgeschwindigkeit

Tiefgestellte Zeichen

$_a$	außen
$_{abs}$	absolut
$_b$	(boundary) Randwert
$_c$	(coarse) Grobgitterwert
$_e$	an der Zellfläche e
$_{exact}$	exakter Wert
$_f$	(fine) Feingitterwert
$_{max}$	Maximalwert
$_{min}$	Minimalwert
$_n$	Normalkomponente; an der Zellfläche n
$_{nb}$	benachbart
$_{num}$	numerisch
$_{out}$	Ausströmrand
$_p$	Druck
$_{ref}$	Referenz
$_{rel}$	relativ
$_t$	total
$_{th}$	theoretisch
$_x$	x-Komponente
$_y$	y-Komponente
$_z$	z-Komponente

Hochgestellte Zeichen

^{c}	konvektiv
^{d}	diffusiv
^{expl}	explizit
^{impl}	implizit
^{m}	Iterationszähler
^{mod}	modifiziert
^{n}	Iterationszähler
^{num}	numerisch
^{old}	vom vorherigen Zeitschritt
^{rel}	im Relativsystem
^{T}	transponiert
*	vorläufig
$'$	Korrektur

Kopfnoten

$\overline{\varphi}$	gemittelter Werte von φ
$\underline{\varphi}$	Vektor φ
$\tilde{\varphi}$	restringierter Feingitterwert; Größen im rotierenden System
$\hat{\varphi}$	φ im gedrehten Koordinatensystem

Zusammenfassung

Im Rahmen dieser Arbeit wird ein neu entwickeltes dreidimensionales Navier-Stokes Verfahren für laminare Strömungen in beliebigen Geometrien vorgestellt.

Durch die unstrukturierte Gitterbehandlung im Programm können sowohl strukturierte, blockstrukturierte und unstrukturierte Gitter als auch Hybridgitter verarbeitet werden. Diese Flexibilität kann für eine erheblich vereinfachte oder auch vollständig automatisierte Gittergenerierung genutzt werden.

Das Verfahren baut auf dem objektorientierten Framework eines am Lehrstuhl entwickelten FEM-Verfahrens auf, so daß die Gitterverwaltung und die linearen Löser wiederverwendet[1] werden konnten. Durch die Implementierung in C++ sind zum einen Erweiterungen oder auch der Austausch des Diskretisierungsverfahrens besonders einfach und außerdem konnten die komplizierten Datenstrukturen für unstrukturierte Gitter einfach und effizient umgesetzt werden. Die mit der Diskretisierung und dem nichtlinearen Lösungsverfahren (SIMPLE-Algorithmus) zusammenhängenden Teile mußten dagegen neu entwickelt werden.

Die verwendete Finite-Volumen-Diskretisierung ist erheblich genauer als die vorher verwendete FEM-Diskretisierung, so daß auf recht groben Gittern gerechnet werden kann. Die guten Konvergenzeigenschaften ergeben sich insbesondere durch Einsatz eines nichtlinearen Mehrgitterverfahrens.

Das Navier-Stokes Verfahren soll in einem Optimierungssystem als Quasi-Euler-Verfahren eingesetzt werden. Testrechnungen für Francis-Turbinen bei drei verschiedenen spezifischen Drehzahlen zeigen aber auch, daß die Ergebnisse von Euler und Navier-Stokes Rechnungen charakteristische Unterschiede aufweisen, die aber in einem Optimierungssystem durch entsprechende Kalibrierung berücksichtigt werden können. Erhebliche Rechenzeitvorteile im Vergleich zu einem kommerziellen Navier-Stokes Code ergeben sich durch das gute Konvergenzverhalten, die Nichtberücksichtigung des Turbulenzmodells und durch den Einsatz erheblich gröberer Gitter.

[1] Der Begriff Wiederverwendung bedeutet bei objektorientierten Programmiersprachen, daß Klassen aufgrund der Trennung von Schnittstelle und Implementierung in einem neuen Zusammenhang verwendet werden können.

Kapitel 1

Einleitung

1.1 Problemstellung

Hydraulische Strömungsmaschinen, insbesondere Francis- und Kaplan-Turbinen sowie Kreiselpumpen haben einen hohen Reifegrad erreicht. Bei hydraulischen Wirkungsgraden bis zu 96 % sind technologische Durchbrüche nicht mehr zu erwarten.

Bei den Wasserturbinen mittlerer und hoher Leistung werden die Turbinen entsprechend der Spezifikation des Kunden entwickelt. Das Angebot der Herstellers muß wenige Wochen nach Angebotsanforderung vorliegen. Zentrale Punkte des Angebotes sind der Preis und der Wirkungsgradverlauf der Turbine. Sind die angebotenen Wirkungsgrade zu niedrig, so wird der Hersteller den Auftrag nicht erhalten. Andererseits sind hohe Konventionalstrafen fällig, falls die gelieferte Turbine die Garantiewerte des Angebotes nicht erreicht. Somit müssen schon in der Angebotsphase unter großen Zeitdruck der voraussichtliche Preis und Wirkungsgradverlauf sehr genau bestimmt werden.

Die Bemühungen gehen deshalb dahin, neben dem rechnergestützten Zugriff auf das Wissen von bereits entwickelten Turbinen Optimierungsprogramme einzusetzen, die durch Aufruf von Strömungs- und Festigkeitsberechnungsprogrammen selbständig optimale Turbinengeometrien erzeugen können.

Charakteristisch für Optimierungsprogramme ist aber, daß die Anzahl der Zielfunktionsaufrufe exponentiell mit der Anzahl der Optimierungsparameter wächst. 1000 Zielfunktionsaufrufe sind hierbei eher eine untere Schranke. Wenn dann jeder Zielfunktionsaufruf eine Navier-Stokes Rechnung zur Folge hat, die eine Rechenzeit in der Größenordnung von einer Stunde benötigt, werden sehr große Rechenzeiten erreicht, die den zeitlichen Rahmen der Angebotsphase sprengen.

Ein Ansatz, der dieses Problem lösen könnte und am Lehrstuhl für Hydraulische Maschinen der TU München untersucht wird, ist, daß neben einer Hierarchie der Optimierungsparameter auch eine Hierarchie von Strömungsberechnungsprogrammen eingesetzt wird. Eine solche Hierarchie ist im Sinne zunehmender Genauigkeit aber auch zunehmender Rechenzeit:

1. S2M Mittelschnittverfahren

2. Q3D Stromfunktionsverfahren
3. „schnelle“Eulerverfahren
4. turbulente Navier-Stokes-Verfahren

Im Optimierungskontext haben somit die einfachen, aber schnellen Strömungsrechenprogramme durchaus ihre Berechtigung.

1.2 Stand des Wissens

1.2.1 Ausgangsgleichungen

Numerische Verfahren zur Berechnung von Strömungen (CFD) lassen sich nach einer Reihe von Kriterien klassifizieren. Je nach Ausgangsgleichungen erhält man eine Hierarchie von Berechnungsverfahren[1]:

1. Navier-Stokes-Gleichungen
   (a) DNS (direkte numerische Simulation)
   (b) LES (large eddy simulation)
   (c) Reynoldsgemittelte NS-Gleichungen
2. Euler-Gleichungen
3. Potentialgleichung

Die meisten CFD-Verfahren setzen die Navier-Stokes-Gleichungen oder deren Vereinfachungen voraus. Turbulente Strömungen, die bei technischen Anwendungen die Regel sind, lassen sich mit DNS, LES, Reynolds-Stress-Modellen (RSM) oder Wirbelviskositätsmodellen (Eddy viscosity models) berechnen. Eine gute Übersicht findet sich bei Perić [12]. All diese Verfahren enthalten als Grundlage die originären Navier-Stokes-Gleichungen.

Eine weitere Vereinfachung ergibt sich, wenn bei hohen Reynolds-Zahlen die Grenzschichteffekte vernachlässigt werden und die Viskosität zu Null gesetzt wird. Man erhält dann die Euler-Gleichungen. Wird schließlich noch vorausgesetzt, daß die Strömung rotationsfrei ist, so ist die daraus folgende Strömung eine Potentialströmung.

Recht weit verbreitet sind im Bereich der Turbomaschinen die sogenannten Q3D-Verfahren. Nach einer Modellvorstellung von Wu [66] wird dabei die Berechnung der tatsächlich dreidimensionalen Strömung durch eine Turbomaschine durch Überlagerung von zweidimensionalen Strömungen jeweils in den Meridian- und Gitterebenen sowie in den dazu normal stehenden Flächen, d.h. den S1-, S2- und S3-Flächen,

[1] Diese Hierarchie ist nicht vollständig. So setzen z.B. die Lattice-Boltzmann-Verfahren, siehe z.B. Filippova und Hänel [13], auf anderen Gleichungen auf.

approximiert. Diese Vorgehensweise kann auf Potential-, Euler- oder Navier-Stokes-Gleichungen angewendet werden. In Verbindung mit Potentialgleichungen in Stromfunktionsformulierung ergeben sich unter Vernachlässigung der Strömung in den S3-Ebenen extrem schnelle und robuste Verfahren, siehe z.B. SCHILLING [45] [46] [47], WATZELT [65] und FERNÁNDEZ [11]. RITZINGER konnte jedoch zeigen [43], daß bei der Berücksichtigung der Reibung mit den Navier-Stokes-Gleichungen sowohl in der Formulierung mit Stromfunktion-Wirbelstärke als auch in c_x, c_y, p Variablen selbst die vereinfachte Q3D-Methode keine Rechenzeitvorteile gegenüber einer echten 3D-Methode hat und außerdem die Sekundärströmungseffekte nicht beschrieben werden können. Im Optimierungskontext werden deshalb am Lehrstuhl für Hydraulische Maschinen der TU München ein Mittelschnittverfahren und ein Q3D-Verfahren auf Basis der Stromfunktionsformulierung eingesetzt.

1.2.2 Berechnungsmethoden für inkompressible Strömungen

Für inkompressible Strömungen kann $\rho = const.$ gesetzt werden, womit sich die Gleichungen formal vereinfachen. Allerdings entsteht insofern ein Problem, daß die Geschwindigkeitskomponenten primär den Impulsgleichungen zugeordnet sind, während der Druck in der Kontinuitätsgleichung nicht auftaucht. Es gibt im wesentlichen zwei Lösungsansätze, dieses Problem zu lösen.

Beim Pseudokompressibilitäts-Verfahren nach CHORIN [9] wird der Kontinuitätsgleichung ein Term $\frac{1}{\beta}\frac{\partial p}{\partial t}$ hinzugefügt, der bei einer stationären Endlösung verschwindet. Dadurch wird das gesamte System hyperbolisch, und es können die für kompressible Strömungen entwickelten Verfahren angewendet werden. Eine Beschreibung des Verfahrens findet sich auch bei KWAK ET AL. [29]. Anwendungen im Bereich der Turbomaschinen finden sich bei RIEDEL [42] sowie bei WARFIELD und LAKSHMINARAYANA [32].

Bei den Druckkorrektur-Verfahren wird unter gewissen vereinfachenden Annahmen aus der Kontinuitätsgleichung eine Gleichung für eine Druckkorrektur hergeleitet. Das bekannteste dieser Verfahren ist der SIMPLE-Algorithmus, siehe CARETTO ET AL. [7]. Diese Verfahren sind recht verbreitet, da PATANKAR den SIMPLE-Algorithmus recht anschaulich und ausführlich in seinem Lehrbuch [36] dargestellt hat. Anwendungen im Bereich der Turbomaschinen finden sich auch bei RITZINGER [43] und BADER [2].

1.2.3 Diskretisierungsmethoden

Bei Potentialströmungen ist die beschreibende Gleichung linear, so daß vorhandene Lösungen durch Superposition zu neuen Lösungen kombiniert werden können. Dies führt dann zur großen Klasse der sogenannten Panelverfahren, siehe z.B. [33]. Bei nichtlinearen Gleichungen jedoch, wie dies bei den Euler- und Navier-Stokes-Gleichungen der Fall ist, muß eine diskrete Feldlösung berechnet werden. Dafür gibt es im wesentlichen drei verschiedene Methoden, die der

- Finiten-Differenzen,

- Finiten-Volumen und
- Finiten-Elemente.

Finite-Differenzen-Verfahren sind auf orthogonalen Gittern recht einfach zu programmieren und auch leicht auf höhere Genauigkeit zu erweitern. Bei nichtorthogonalen Gittern müssen entsprechende Koordinatentransformationen angewendet werden, was zu verringerter Genauigkeit führen kann. Bei unstrukturierten Gittern sind die Finite-Differenzen-Verfahren nicht anwendbar.

Die Finite-Volumen-Verfahren benötigen auf nichtorthogonalen Gittern keine Koordinatentransformation und sind deshalb auf beliebige Kontrollvolumen anwendbar, da keine Ansatzfunktionen notwendig sind. Die meisten kommerziellen CFD-Codes sind Finite-Volumen-Verfahren, wie z.B. TASCflow [56] oder Fluent [16], was für die Robustheit der FV-Verfahren spricht. Anwendungen im Bereich der Turbomaschinen finden sich z.B. bei RITZINGER [43] und BADER [2].

Bei Finite-Elemente-Verfahren, wie z.B. dem kommerziellen CFD-Code FIDAP [15], werden die Differentialgleichungen zunächst mit Testfunktionen multipliziert und erst dann integriert. Der Finite-Element-Ansatz leitet sich also von Variationsprinzipien ab. Nachteilig kann sich auswirken, daß die Aufstellung der Koeffizientenmatrix unter Umständen recht aufwendig ist. Außerdem können die Ansatzfunktionen nicht für beliebige Zelltypen formuliert werden. Anwendungen im Bereich der Turbomaschinen finden sich z.B. bei MÜLLER [34].

1.2.4 Benchmarkergebnisse

Eine relativ objektive Methode zur Beurteilung der Leistungsfähigkeit verschiedener Verfahren sind Benchmark-Tests. Dabei werden ein oder mehrere Testfälle mit genauen Spezifikationen definiert. Außerdem wird genau beschrieben, in welcher Form die Ergebnisse präsentiert werden sollen. Im Rahmen des DFG Sonderforschungsbereiches *Flow Simulation on High Performance Computers* wurden laminare Zylinderumströmungen von verschiedenen Forschergruppen berechnet. Unter anderem wurde die stationäre zweidimensionale laminare Zylinderumströmung bei Re=20 berechnet, siehe TUREK [60]. Für diesen Testfall sind bei Turek 15 verschiedene Ergebnisse von verschiedenen Verfahren aufgelistet. Angegeben sind Zahlenwerte für folgende Kennwerte:

- Widerstandsbeiwert c_D,
- Auftriebsbeiwert c_L,
- Länge des Rezirkulationsgebietes L_a und
- Druckdifferenz zwischen Vorder- und Hinterpunkt ΔP.

Für diese 4 Kennwerte sind jeweils eine obere und eine untere Schranke für die *richtigen* Werte angegeben. In Abbildung 1.1 sind Verfahren, die alle 4 Kennwerte innerhalb der Schranke berechnen konnten, mit einem grünen Kreis gekennzeichnet.

Die anderen Verfahren sind mit einem roten Kreis gekennzeichnet. Auf der Ordinate ist die Anzahl der Unbekannten des gröbsten Gitters aufgetragen, bei dem alle 4 Kennwerte richtig bzw. des Gitters, bei dem die Werte am besten waren.

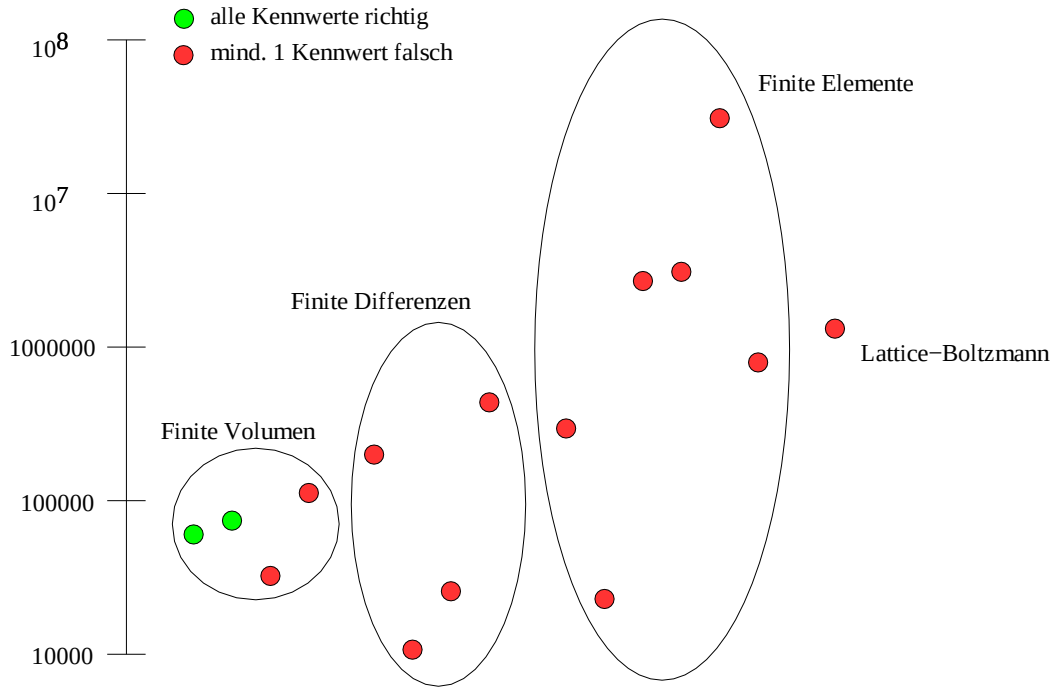

Abbildung 1.1: Vergleich verschiedener Verfahren bzgl. Anzahl der Unbekannten für 2D Zylinderumströmung bei Re=20 in logarithmischer Auftragung

Zunächst fällt auf, daß die meisten Verfahren die 4 Kennwerte nicht genau genug berechnen konnten. Bei den zwei Verfahren, die die richtigen Werte vorhersagten, wurde die gleiche Diskretisierung benutzt, nämlich ein zellenzentriertes Finite-Volumen-Verfahren mit CDS für die konvektiven Terme. Keines der Verfahren mit Finite-Differenzen- bzw. Finite-Elemente-Diskretisierung konnte alle Kennwerte genau genug berechnen.

1.2.5 Implementierungstechniken

Neben der eigentlichen Numerik, d.h. Diskretisierungs- und Lösungsverfahren, wird auch die Frage der Implementierungstechnik zunehmend wichtig, da die Komplexität und der Umfang der Software einen Grad erreicht haben, bei dem ein einzelner Entwickler nicht ausreicht. Bastian et al. [4] entwickelten das Konzept UG, das ein flexibles Softwaretool für die numerische Lösung von partiellen Differentialgleichungen darstellt. UG behandelt die Gitter unstrukturiert und verfügt über Methoden für adaptive Gitter, Multigrid-Löser und Parallelisierung. Der Code ist in C geschrieben. Der Entwicklungsaufwand betrug nach Angaben der Entwickler mehrere Mannjahrzehnte. Konkrete Applikationen umfassen mehrere Strömungs- und Festigkeitsprogramme. Ein ähnliches Konzept von Gloth et al. [18], genannt MOUSE, beschränkt sich auf zweidimensionale Navier-Stokes-Verfahren, benutzt aber objek-

torientierte Konzepte, die in C++ implementiert sind. Ebenfalls in C++ implementiert ist das Framework CONSIST von MÜLLER [34]. Dessen Implementierung eines Euler-Verfahrens für Turbomaschinen auf Basis des finiten Elementes von RANNACHER und TUREK [39] erfordert aber eine relativ hohe Zahl von Gitterpunkten im Vergleich zu Finite-Volumen-Verfahren.

1.3 Zielsetzung

Ziel der vorliegenden Arbeit ist die Entwicklung eines Navier-Stokes-Verfahrens mit Finite-Volumen-Diskretisierung. Das Verfahren soll eine freie Wahl der Gittertypen, wie z.B. blockstrukturierte Gitter, unstrukturierte Gitter, Hybridgitter, ermöglichen. Für den Einsatz als Quasi-Euler-Verfahren in einem Optimierungssystem ist das Konvergenzverhalten und die Lösungsgenauigkeit des Verfahrens zu untersuchen. Bei der Implementierung in C++ ist besonderer Wert auf die spätere Erweiterbarkeit zu legen.

Kapitel 2

Theoretische Grundlagen

In diesem Kapitel sollen die Grundgleichungen zusammengestellt werden, auf denen das numerische Verfahren aufbaut.

2.1 Grundgleichungen in differentieller Form

Es werden die 3D Navier-Stokes-Gleichungen für inkompressible Fluide betrachtet, wobei folgende Einschränkungen gemacht werden:

1. konstante Stoffwerte $\rho, \mu = const.$
2. laminare Strömung.

Unter den getroffenen Voraussetzungen lauten die Kontinuitätsgleichung und die Impulsgleichungen wie folgt:

$$\rho \nabla \cdot \underline{c} = 0, \tag{2.1}$$

$$\rho \frac{D\underline{c}}{Dt} + \nabla p = \mu (\nabla \cdot \nabla) \underline{c} \, . \tag{2.2}$$

Im Anhang A ist die Transformation in das rotierende System beschrieben. Es soll hier eine Formulierung benutzt werden, die von KROLL [28] angegeben wurde, und deshalb als Kroll-Formulierung bezeichnet werden soll.

$$\rho \left(\underline{w} \cdot \nabla \right) \underline{c} + \nabla p = \mu \left(\nabla \cdot \nabla \right) \underline{c} - \rho \underline{\omega} \times \underline{c} \tag{2.3}$$

mit $\underline{w} = \underline{c} - \underline{\omega} \times \underline{x}$. Auf die im Anhang benutzte Tilde als Kennzeichnung von Größen im rotierenden System wurde hier verzichtet, da für $\omega \neq 0$ alle Betrachtungen im rotierenden System erfolgen. Für $\omega = 0$ sind rotierendes und festes Koordinatensystem identisch. Die Impulsgleichungen beschreiben somit die Absolutgeschwindigkeit im rotierenden System. Als zusätzliche Einschränkung wurde dabei vorausgesetzt, daß die Strömung im rotierenden System stationär ist.

Die Kontinuitätsgleichung wird durch die Transformation nicht verändert. Sie gilt auch für die Relativgeschwindigkeit

$$\rho \nabla \cdot \underline{w} = 0 \tag{2.4}$$

2.2 Grundgleichungen in integraler Form

Da das Diskretisierungsverfahren auf den Erhaltungsgleichungen in integraler Form aufsetzt, sollen die Gleichungen (2.1) und (2.3) entsprechend umgeformt werden. Integration von Gleichung (2.1) über ein beliebiges Kontrollvolumen V ergibt

$$\rho \int \nabla \cdot \underline{c}\, dV = 0 \tag{2.5}$$

Durch Anwendung des Gaußschen Integralsatzes, siehe z.B. [5],

$$\int \nabla \cdot \underline{F}\, dV = \oint \underline{F} \cdot d\underline{S} \tag{2.6}$$

kann Gleichung (2.5) umgeformt werden zu:

$$\rho \oint \underline{c} \cdot d\underline{S} = 0. \tag{2.7}$$

In Indexnotation ist dies

$$\rho \oint c_i\, dS_i = 0. \tag{2.8}$$

Integration von Gleichung (2.3) über das Kontrollvolumen ergibt:

$$\rho \int (\underline{w} \cdot \nabla)\, \underline{c}\, dV + \int \nabla p\, dV = \mu \int (\nabla \cdot \nabla)\, \underline{c}\, dV - \rho \int \underline{\omega} \times \underline{c}\, dV. \tag{2.9}$$

Wendet man wiederum den Gaußschen Integralsatz an, so erhält man:

$$\rho \oint \underline{c}\, \underline{w} \cdot d\underline{S} + \oint p\, d\underline{S} = \mu \oint \nabla \underline{c} \cdot d\underline{S} - \rho \int \underline{\omega} \times \underline{c}\, dV. \tag{2.10}$$

In Indexnotation ergibt sich:

$$\rho \oint c_i w_j\, dS_j + \oint p\, dS_i = \mu \oint \frac{\partial c_i}{\partial x_j} dS_j - \rho \int \epsilon_{ijk} \omega_j c_k\, dV. \tag{2.11}$$

2.3 Kennwerte für Pumpen und Turbinen

Bei Pumpen und Turbinen wird die Referenzgeschwindigkeit mit der Winkelgeschwindigkeit gebildet.

$$u_{ref} = \omega r_{ref} \tag{2.12}$$

Die Energieumsetzung wird beschrieben durch die Totaldruckzahl Ψ_t, d.h. Totaldruckaufbau bei Pumpen bzw. Totaldruckabbau bei Turbinen Ψ_t und durch die theoretische Totaldruckzahl $\Psi_{t_{th}}$, die sich aus der Dralländerung errechnet. Die Totaldruckzahl Ψ_t berechnet sich aus

$$\Psi_t = \frac{\overline{p}_{t,out} - \overline{p}_{t,in}}{\rho/2 u_{ref}^2}, \tag{2.13}$$

mit dem impulsgemittelten Totaldruck

$$\overline{p}_t = \frac{\int \rho p_t \underline{c} \cdot d\underline{S}}{\int \rho \underline{c} \cdot d\underline{S}} = \frac{\int \rho p_t \underline{c} \cdot d\underline{S}}{\dot{m}}. \tag{2.14}$$

Die theoretische Totaldruckzahl $\Psi_{t_{th}}$ berechnet sich wie folgt:

$$\Psi_{t_{th}} = \frac{gH_{th}}{1/2u_{ref}^2}, \tag{2.15}$$

wobei sich die theoretische Förderhöhe in der folgenden Weise berechnen läßt.

$$H_{th} = \frac{\omega}{g}\frac{\int \rho r c_u \underline{c} \cdot d\underline{S}}{\int \rho \underline{c} \cdot d\underline{S}} = \frac{\omega}{g\dot{m}} \int \rho r c_u \underline{c} \cdot d\underline{S} \tag{2.16}$$

Der hydraulische Wirkungsgrad ergibt sich für Pumpe und Turbine in der nachstehenden Form:

$$\text{Pumpen:} \quad \eta_h = \frac{\Psi_t}{\Psi_{t_{th}}}, \tag{2.17}$$

$$\text{Turbinen:} \quad \eta_h = \frac{\Psi_{t_{th}}}{\Psi_t}. \tag{2.18}$$

Kapitel 3

Finite-Volumen-Diskretisierung

3.1 Aufteilung in Kontrollvolumen

Bei Finite-Volumen-Verfahren wird zunächst das Rechengebiet in Zellen aufgeteilt. Diese finiten Volumen geben dem Verfahren seinen Namen. Ein großer Vorteil der gewählten Diskretisierung ist dabei, daß an die Aufteilung nur wenige Anforderungen gestellt werden müssen. Als zwingende Forderungen müssen lediglich folgende erfüllt sein:

- Das ganze Rechengebiet muß ausgefüllt werden.
- Die Zellen dürfen nicht überlappen.

Um zu verdeutlichen, welche Freiheiten diese zwei Forderungen bieten, soll das folgende zweidimensionale Gitter betrachtet werden, siehe Abbildung 3.1.

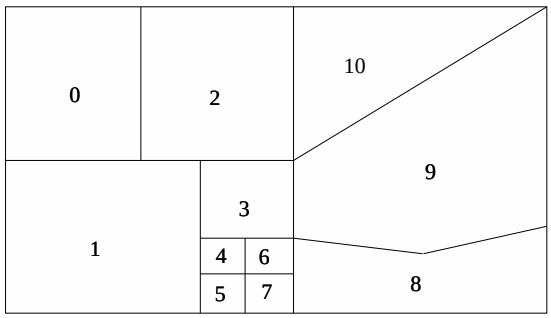

Abbildung 3.1: Allgemeines 2D-Gitter für Finite-Volumen-Verfahren

Dieses Gitter zeichnet sich durch folgende Eigenschaften aus:

- Es enthält ein Dreieck (Zelle 10), 8 Vierecke, ein Fünfeck (Zelle 8) und ein Sechseck (Zelle 9).

- Eine nichtkonvexe Zelle (Zelle 8) zerstört nicht notwendigerweise das Gesamtverfahren, obwohl sich solche Zellen natürlich negativ auf die Konvergenzeigenschaften und die Genauigkeit auswirken.
- Die Kanten der Zellen müssen nicht vollständige Kanten der Nachbarzellen sein, siehe z.B. das Interface zwischen Zelle 1 und 2. Man spricht hier von *Nonmatching Interfaces*.
- Lokale Verfeinerungen sind möglich. Zum Beispiel kann Zelle 1 für die Diskretisierung als 7-Eck behandelt werden.
- Das Gitter ist unstrukturiert, d.h. es gibt keine ausgezeichnete i- bzw. j-Richtung.
- Die gemachten Aussagen lassen sich entsprechend auf den dreidimensionalen Fall übertragen.

Trotz der vielen Freiheiten der Diskretisierungsmethode bezüglich der Gittertopologie müssen aber folgende Einschränkungen beachtet werden:

- Die Gitterqualität hat natürlich Auswirkungen auf Genauigkeit und Konvergenzeigenschaften.
- Die Datenstruktur der konkreten Implementierung muß diese Topologie auch unterstützen.

Im Vorgriff auf Kapitel 5 sei hier schon festgestellt, daß die Implementierung des 3D-Codes der vorliegenden Arbeit nur folgendes zuläßt:

- Unstrukturierte Gitter,
- matching Interfaces,
- keine lokale Verfeinerung und die
- folgenden Zelltypen: Tetraeder, Prismen und Hexaeder.

3.2 Zellenzentriertes FV-Verfahren

Bei der Diskretisierung von Differentialgleichungen mit FV-Verfahren müssen die Variablen, d.h. c_x, c_y, p bei den zweidimensionalen inkompressiblen Navier-Stokes-Gleichungen, nicht dem selben Ort im Gitter zugeordnet sein. Sind die Variablen alle dem gleichen Ort, z.B. der Zellmitte zugeordnet, so spricht man im Englischen vom *Colocated Arrangement*, andernfalls vom *Staggered Arrangement*. Der große Vorteil des Staggered Arrangement ist die gute Kopplung von Druck und Geschwindigkeit, ohne die sich Schwingungen im Feld entwickeln können. Das bekannteste Beispiele ist das Verfahren von HARLOW und WELSH [21], bei dem der Druck der Zellmitte, die u-Geschwindigkeit den senkrechten Kanten und die v-Geschwindigkeit den waagerechten Kanten zugeordnet sind. Dieses Verfahren ist aber nur schwer auf

nichtorthogonale Gitter zu erweitern. Dies gelingt zwar beim Verfahren von HIRT ET AL. [23], bei dem der Druck der Zellmitte und beide Geschwindigkeitskomponenten den Zelleckpunkten zugeordnet sind. Bei diesem Verfahren können jedoch auch Schwingungen im Druck- oder Geschwindigkeitsfeld auftreten.

RHIE und CHOW [40] entwickelten einen einfachen Korrekturterm, der die Entkopplung von Druck und Geschwindigkeit beseitigt, siehe auch Gleichung 3.20. Damit konnten sich die Verfahren mit Colocated Arrangement bei kommerziellen Codes durchsetzen. Die Vorteile des Colocated Arrangement sind u.a. eine einfachere Programmstruktur und die leichte Erweiterbarkeit auf nichtorthogonale Gitter. Außerdem können bei Mehrgitterverfahren die gleichen Prolongations- und Restriktionsoperatoren für alle Variablen benutzt werden.

Aus den genannten Gründen wird in dieser Arbeit ein Colocated Verfahren benutzt. Damit sind die Kontrollvolumen identisch mit den Zellen. Als primäre Variablen werden c_x, c_y, c_z und p benutzt. Diese sind den Zellmittelpunkten zugeordnet und im Sinne von Mittelwerten über das Kontrollvolumen zu verstehen. Als sekundäre Variablen werden für jede Zelle noch die Druckkorrektur p' und die Gradienten ∇c_x, ∇c_y, ∇c_z, ∇p, $\nabla p'$ gespeichert.

Die grundlegende Approximation besteht nun darin, die exakten Integralgleichungen (2.7) und (2.10) durch die Summe über die Begrenzungsflächen des Kontrollvolumens zu ersetzen. Aus der Kontinuitätsgleichung (2.8) wird somit

$$\rho \sum_{e=1}^{n_f} c_{i,e} S_{i,e} = 0 \tag{3.1}$$

und aus den Impulsgleichungen (2.11)

$$\rho \sum_{e=1}^{n_f} \left(c_{i,e}(w_{j,e} S_{j,e}) - \mu \left(\frac{\partial c_i}{\partial x_j} \right)_e S_{j,e} \right) = - \left[\frac{\partial p}{\partial x_i} + \rho \epsilon_{ijk} \omega_j c_k \right] V. \tag{3.2}$$

Führt man die Abkürzungen

$$Q_i^p = -\frac{\partial p}{\partial x_i} V, \tag{3.3}$$

$$Q_i^\omega = -\rho \epsilon_{ijk} \omega_j c_k V, \tag{3.4}$$

$$F_{i,e}^c = c_{i,e}(\rho w_{j,e} S_{j,e}) = c_{i,e} \dot{m}_e \tag{3.5}$$

und

$$F_{i,e}^d = \mu \left(\frac{\partial c_i}{\partial x_j} \right)_e S_{j,e} \tag{3.6}$$

ein, so vereinfachen sich die Impulsgleichungen zu:

$$\sum_{e=1}^{n_f} \left(F_{i,e}^c - F_{i,e}^d \right) = Q_i^p + Q_i^\omega. \tag{3.7}$$

3.3 Flächenvektoren und Volumenberechnung

In den diskretisierten Gleichungen (3.1) und (3.7) treten das Volumen V und der Flächennormalenvektor $\underline{S}_e$ auf. Wenn allgemeine Begrenzungsflächen und Kontrollvolumen zugelassen sind, können diese Größen wie folgt bestimmt werden. Der

Flächenvektor ergibt sich aus:

$$\underline{S}_e = \frac{1}{2} \sum_{i=3}^{n_e^v} \left[(\underline{x}_{i-1} - \underline{x}_1) \times (\underline{x}_i - \underline{x}_1) \right]. \tag{3.8}$$

Dabei sind die Eckpunkte der Fläche von 1 bis n_e^v numeriert, siehe Abbildung 3.2.

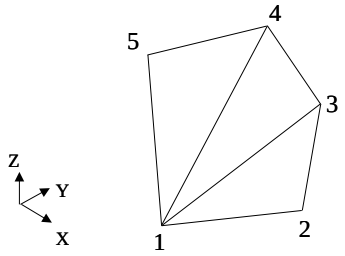

Abbildung 3.2: Berechnung des Flächenvektors für ein n-Eck

Das Volumen kann mit Hilfe des Gaußschen Integralsatzes in einfacher Weise berechnet werden:

$$\int_V \nabla \underline{x} \, dV = \oint_S \underline{x} \cdot d\underline{S} \Rightarrow V = \frac{1}{3} \sum_{i=1}^{n_f} \underline{x}_{e,i} \cdot \underline{S}_e. \tag{3.9}$$

Desweiteren werden die Mittelpunkte der Flächen $\underline{x}_e$ und der Kontrollvolumina $\underline{x}_P$ benötigt. Numerische Experimente haben gezeigt, daß zu deren Berechnung eine einfache Mittelwertbildung der Eckpunkte ausreicht.

$$\underline{x}_e = \frac{1}{n_e^v} \sum_{i=1}^{n_e^v} \underline{x}_i, \tag{3.10}$$

$$\underline{x}_P = \frac{1}{n^v} \sum_{i=1}^{n^v} \underline{x}_i. \tag{3.11}$$

Dabei ist n_e^v die Anzahl der Eckpunkte der Fläche und n^v die Anzahl der Eckpunkte des Volumens.

3.4 Berechnung von Interfacewerten

Die primären Variablen sind den Zellmittelpunkten zugeordnet. Für die Fluß- und Gradientenberechnungen werden jedoch Werte in den Flächenmittelpunkten benötigt. Diese werden durch Interpolation bestimmt. Zur Berechnung eines Interfacewertes ϕ_e wird zunächst der Wert $\phi_{e'}$ am Ort $\underline{x}_{e'}$ bestimmt, siehe Abbildung 3.3.

$$\lambda_e = \frac{|\underline{x}_e - \underline{x}_P|}{|\underline{x}_e - \underline{x}_P| + |\underline{x}_E - \underline{x}_e|}, \tag{3.12}$$

$$\phi_{e'} = \lambda_e \phi_E + (1 - \lambda_e) \phi_P. \tag{3.13}$$

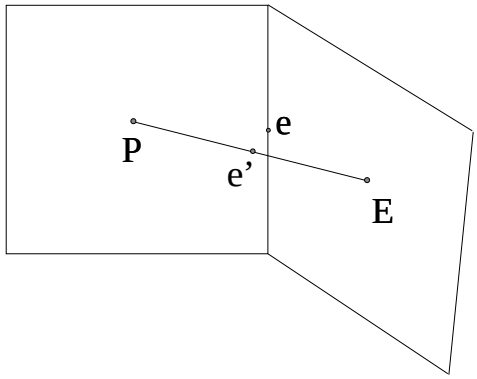

Abbildung 3.3: Bezeichnungen für Gradientenkorrektur von Interfacewerten

Diese Approximation führt zu lokaler und globaler Genauigkeit zweiter Ordnung. Allerdings wird damit die Symmetrie der zugrundeliegenden Differentialoperatoren bzw. -gleichungen für $\lambda_e \neq 1/2$ zerstört. Die daraus resultierenden Nachteile werden von Verstappen und Veldman [63] diskutiert. Die Approximation wird hier aber trotzdem verwendet, da sie sich bei komplizierten Geometrien mit entsprechend schlechten Gittern in der Praxis bewährt hat.

Für nicht-orthogonale Gitter ist der Punkt

$$\underline{x}_{e'} = \lambda_e \underline{x}_E + (1 - \lambda_e)\underline{x}_P \tag{3.14}$$

in der Regel verschieden von $\underline{x}_e$. Die Genauigkeit kann durch eine explizite Gradientenkorrektur erhöht werden. Dazu berechnet man den Gradienten bei $\underline{x}_{e'}$ wie folgt:

$$(\nabla\phi)^{old}_{e'} = \lambda_e (\nabla\phi)^{old}_E + (1 - \lambda_e)(\nabla\phi)^{old}_P \tag{3.15}$$

und korrigiert $\phi_{e'}$ mit

$$\phi_e = \phi_{e'} + (\nabla\phi)^{old}_{e'} \cdot (\underline{x}_e - \underline{x}_{e'}). \tag{3.16}$$

Für p' wird keine Gradientenkorrektur verwendet und im Mehrgitterverfahren wird die Gradientenkorrektur nur auf dem feinsten Gitter benutzt.

3.5 Berechnung der Gradienten

Die Berechnung der Gradienten in den Zellmittelpunkten erfolgt mit Hilfe des Oberflächenintegrals:

$$(\nabla\phi)_P = \frac{1}{V} \sum_{e=1}^{n_f} \phi_e \underline{S}_e. \tag{3.17}$$

Bei der Lösung der Impulsgleichung wird der Druckterm als Quellterm auf die rechte Seite geschrieben. Dabei wird das Volumenintegral des Gradienten benutzt. Da der Gradient aber wie beschrieben mit Hilfe des Oberflächenintegrals bestimmt wird, ist dies äquivalent zur konservativen Behandlung des Druckterms als Oberflächenintegral.

3.6 Berechnung der Massenströme

Die Massenströme treten im konvektiven Term und im Quellterm der Druckkorrekturgleichung auf. Der Massenstrom $\dot{m}_e$ wird in beiden Fällen explizit behandelt.

$$\dot{m}_e = \int_{S_e} \rho \underline{w} \cdot d\underline{S}, \tag{3.18}$$

$$= \rho \underline{w}_e \cdot \underline{S}_e - \overline{\left(\frac{\rho}{A_P^u}\right)}_e S_e^2 \left[p_E - p_P - \frac{(\nabla p)_E + (\nabla p)_P}{2} \cdot (\underline{x}_E - \underline{x}_P) \right], \tag{3.19}$$

$$= \rho \underline{w}_e \cdot \underline{S}_e - \Pi_\Delta. \tag{3.20}$$

Der Korrekturterm Π_Δ bewirkt, daß sich der Druck und die Geschwindigkeit nicht entkoppeln können. Dies ist eine Voraussetzung, um ein Colocated Verfahren benutzen zu können. Der Anteil in eckigen Klammern wird klein für eine glatte Druckverteilung und der Gesamtterm wird bei feiner werdendem Gitter mit h^2 kleiner.

3.7 Diskretisierung des konvektiven Terms

Der konvektive Term ist numerisch am schwierigsten zu behandeln, was zum einen daran liegt, daß der Term nichtlinear ist und zum anderen, daß für große Reynolds-Zahlen der elliptische Charakter der Impulsgleichung immer schwächer wird. Die einfachste Lösung ist, ein Upwind Verfahren zu benutzen (Upwind Differencing Scheme, UDS). Allerdings ist dieses nur 1. Ordnung genau, was zu stark fehlerbehafteten Lösungen führt.

Patankar [36] schlägt vor, eine Approximation der exakten Lösung der eindimensionalen Konvektions-Diffusionsgleichung im numerischen Verfahren zu benutzen. Perić [12] weißt aber darauf hin, daß bei technischen Strömungen der konvektive Term im wesentlichen mit dem Druckterm im Gleichgewicht steht. Perić schlägt deshalb vor, zentrale Differenzen (Central Differencing Scheme, CDS) zu benutzen. Nach Erfahrungen von Skoda [50] sind bei der Berechnung von turbulenten Strömungen mit den k, ε-Gleichungen höherwertige Verfahren vorteilhaft. In Frage kommen dafür z.B. das QUICK-Verfahren von Leonard [30] oder das SMART-Verfahren von Gaskell und Lau [17] kombiniert mit dem Skew-Upwind-Verfahren von Raithby [38]. Allerdings ist die Erweiterung dieser Verfahren auf unstrukturierte Gitter nicht trivial.

Die in dieser Arbeit benutzte Strategie zur Behandlungen der konvektiven Terme besteht somit aus folgenden Ansätzen:

- Die Nichtlinearität wird dadurch entfernt, daß der Massenstromterm $\rho w_{j,e} S_{j,e}$ vom vorherigen Iterationsschritt genommen wird.
- Durch einen Blending Faktor γ kann stufenlos zwischen UDS und CDS gewählt werden. Bei laminaren Strömungen kann in der Regel $\gamma = 1.0$ und bei Euler-Strömungen $\gamma = 0.8$ gewählt werden.
- Die Diagonaldominanz der Koeffizientenmatrix wird durch die Anwendung von Deferred Correction (nachgeführte Korrektur) sichergestellt. Das Deferred Correction-Verfahren wurde zuerst von Khosla und Rubin [27] vorgeschlagen.

Man definiert somit

$$c_{i,e}^{UDS} = \begin{cases} c_{i,P} & : \quad \dot{m}_e \geq 0 \\ c_{i,E} & : \quad \dot{m}_e < 0 \end{cases}, \tag{3.21}$$

und

$$c_{i,e}^{CDS} = \lambda_e c_{i,E} + (1 - \lambda_e) c_{i,P}. \tag{3.22}$$

Dann gilt:

$$F_{i,e}^{c} = \dot{m}_e c_{i,e}^{UDS} + \gamma \dot{m}_e \left(c_{i,e}^{CDS} - c_{i,e}^{UDS} \right)^{m-1}. \tag{3.23}$$

γ ist der Blending-Faktor. $\gamma = 0$ bedeutet UDS mit erster Ordnung Genauigkeit, während für $\gamma = 1$ CDS mit zweiter Ordnung Genauigkeit ausgewählt wird.

3.8 Diskretisierung des diffusiven Terms

Der diffusive Term

$$F_{i,e}^{d} = \mu \left(\frac{\partial c_i}{\partial x_j} \right)_e S_{j,e} \tag{3.24}$$

wird zunächst in die folgende Form umgeschrieben:

$$F_{i,e}^{d} = \mu \left(\frac{\partial c_i}{\partial x_j} n_j \right)_e S_e = \mu (\nabla c_i \cdot \underline{n})_e S_e. \tag{3.25}$$

Allerdings läßt sich $(\frac{\partial c_i}{\partial x_j})_e$ nicht unmittelbar implizit darstellen. Deshalb wird Gleichung 3.25 als explizite Korrektur verwendet und für den impliziten Teil eine weitere Approximation benutzt. Mit der Identität

$$\nabla \phi = \frac{\partial \phi}{\partial x} \underline{i} + \frac{\partial \phi}{\partial y} \underline{j} + \frac{\partial \phi}{\partial z} \underline{k} = \frac{\partial \phi}{\partial n} \underline{n} + \frac{\partial \phi}{\partial t_1} \underline{t}_1 + \frac{\partial \phi}{\partial t_2} \underline{t}_2 \tag{3.26}$$

erhält man:

$$F_{i,e}^{d} = \mu \left(\frac{\partial c_i}{\partial n} \right)_e S_e. \tag{3.27}$$

Diese Form ist zwar nicht einfacher als Gleichung 3.25, jedoch kann daraus für den impliziten Teil eine sehr gute weitere Approximation abgeleitet werden. Ist nämlich ξ die Koordinatenrichtung von P nach E, siehe Abbildung 3.4,

so wird:

$$F_{i,e}^{d,impl} = \mu S_e \left(\frac{\partial c_i}{\partial \xi} \right)_e = \mu S_e \frac{c_{i,E} - c_{i,P}}{L_{P,E}} \tag{3.28}$$

Naheliegend wäre nun, den folgenden Deferred-Correction Ansatz zu machen:

$$F_{i,e}^{d} = F_{i,e}^{d,impl} + \left[F_{i,e}^{d,expl} - F_{i,e}^{d,impl} \right]^{old} \tag{3.29}$$

mit

$$F_{i,e}^{d,expl} = \mu \left[\lambda_e (\nabla c_i)_E + (1 - \lambda_e)(\nabla c_i)_P \right]^{old} \cdot \underline{n}_e S_e. \tag{3.30}$$

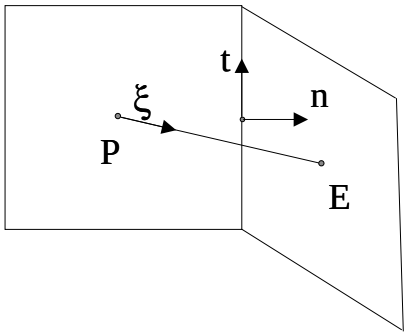

Abbildung 3.4: Bezeichnungen für Approximation des Diffusionsterms

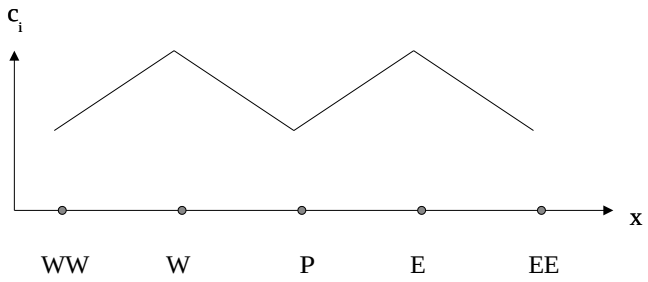

Abbildung 3.5: Oszillierendes Geschwindigkeitsfeld

Bei diesem Ansatz können sich aber oszillierende Lösungen entwickeln, da sich z.B. im eindimensionalen Fall mit $\Delta x = const.$ für Gleichung (3.30) ergibt:

$$F_{i,e}^{d,expl} = \mu \left[\frac{1}{2} \frac{c_{i,EE} - c_{i,P}}{2\Delta x} + \frac{1}{2} \frac{c_{i,E} - c_{i,W}}{2\Delta x} \right]^{old} S_e. \tag{3.31}$$

Dieser Term ist Null für das in Abbildung 3.5 dargestellte Geschwindigkeitsfeld.

Muzaferija [35] schlug eine ebenso einfache wie effektive Korrektur vor:

$$F_{i,e}^{d} = F_{i,e}^{d,impl} + \mu S_e \overline{(\nabla c_i)}_e^{old} \cdot (\underline{n} - \underline{i}_\xi) \tag{3.32}$$

3.9 Koeffizienten der Impulsgleichung

Die Impulsgleichung soll nun auf folgende Form gebracht werden:

$$A_P c_{i,P} + \sum_{nb} A_{nb} c_{i,nb} = Q_P^i, \tag{3.33}$$

wobei gilt:

$$Q_P^i = Q_i^p + Q_i^\omega + Q_i^c + Q_i^d. \tag{3.34}$$

Für den Druckquellterm erhält man:

$$Q_i^p = - \left(\frac{\partial p}{\partial x_i} \right)_P V, \tag{3.35}$$

und für den Rotationsquellterm:

$$Q_1^\omega = +\rho\omega c_y V, \tag{3.36}$$

$$Q_2^\omega = -\rho\omega c_x V, \tag{3.37}$$

$$Q_3^\omega = 0. \tag{3.38}$$

Für die konvektiven Anteil mit UDS ergibt sich

$$A_E^c = \left\{ \begin{array}{rcl} \dot{m}_e & : & \dot{m}_e < 0 \\ 0 & : & \dot{m}_e \geq 0 \end{array} \right. = min(\dot{m}_e, 0), \tag{3.39}$$

$$A_P^c \quad += \left\{ \begin{array}{rcl} \dot{m}_e & : & \dot{m}_e \geq 0 \\ 0 & : & \dot{m}_e < 0 \end{array} \right. = max(\dot{m}_e, 0). \tag{3.40}$$

Dabei bedeutet der Ausdruck $+=$, daß der Term auf der rechten Seite auf der linken Seite hinzuaddiert wird. In A_P^c werden also die entsprechenden Terme von den Nachbarn aufsummiert. Für den Quellterm ergibt sich ein Anteil aus dem CDS wie folgt:

$$\begin{aligned} Q_i^c &= -\sum \gamma \dot{m}_e (c_{i,e}^{CDS} - c_{i,e}^{UDS}), \qquad (3.41) \\ &= \sum \left(-\gamma \dot{m}_e \left(\lambda_e c_{i,E} + (1-\lambda_e) c_{i,P}\right) - min(\dot{m}_e, 0) c_{i,E} - max(\dot{m}_e, 0) c_{i,P}\right). \end{aligned}$$

Hier ist zu beachten, daß wegen der Kontinuitätsgleichung A_P^c auch wie folgt berechnet werden kann:

$$A_P^c = -\sum A_{nb}^c. \tag{3.42}$$

Für den diffusiven Anteil ergibt sich:

$$A_E^d = \frac{-\mu S_e}{L_{P,E}} \tag{3.43}$$

und

$$A_P^d = -\sum A_{nb}^d. \tag{3.44}$$

Insgesamt erhält man:

$$A_E = A_E^c + A_E^d = min(\dot{m}_e, 0) - \frac{\mu S_e}{L_{P,E}} \tag{3.45}$$

und

$$A_P = -\sum A_{nb}. \tag{3.46}$$

3.10 Druckkorrekturgleichung

Wie in den vorigen Abschnitten gezeigt, werden die Impulsgleichungen als Bestimmungsgleichungen für die Geschwindigkeitskomponenten behandelt. Der Druck muß deshalb mit Hilfe der Kontinuitätsgleichung bestimmt werden, in der er aber nicht auftritt. Dieses Problem wird gelöst, indem aus den Impulsgleichungen und der Kontinuitätsgleichung eine Druckkorrekturgleichung hergeleitet wird.

Gleichung 3.33 läßt sich in folgender Form schreiben:

$$A_P c_{i,P}^{m*} + \sum_{nb} A_{nb} c_{i,nb}^{m*} = Q_i^* - \left(\frac{\partial p^{m-1}}{\partial x_i}\right)_P V. \tag{3.47}$$

Dabei ist m ein Iterationszähler. Die Geschwindigkeiten haben den Index $m*$, da sie die Kontinuitätsgleichung noch nicht erfüllen und entsprechend korrigiert werden müssen. Man führt nun die folgenden Größen ein

$$c_i^m = c_i^{m*} + c_i' \tag{3.48}$$

und

$$p^m = p^{m-1} + p' \tag{3.49}$$

Dann soll gelten:

$$A_P c_{i,P}^m + \sum_{nb} A_{nb} c_{i,nb}^m = Q_i^* - \left(\frac{\partial p^m}{\partial x_i}\right)_P V. \tag{3.50}$$

Daraus folgt mit Gleichung 3.47:

$$A_P c_{i,P}' + \sum_{nb} A_{nb} c_{i,nb}' = -\left(\frac{\partial p'}{\partial x_i}\right)_P V. \tag{3.51}$$

Beim SIMPLE-Algorithmus vernachlässigt man nun den zweiten Term und erhält:

$$c_{i,P}' = -\frac{1}{A_P}\left(\frac{\partial p'}{\partial x_i}\right)_P V. \tag{3.52}$$

In gleicher Weise ergeben sich die Geschwindigkeitskorrekturen an den Zellflächen:

$$c_{i,e}' = \overline{\left(\frac{1}{A_P}\right)_e}\left(\frac{\partial p'}{\partial x_i}\right)_e V_e = \overline{\left(\frac{1}{A_P}\right)_e}(p_E' - p_P')S_{i,e}. \tag{3.53}$$

Setzt man dies in die Kontinuitätsgleichung ein, so erhält man:

$$A_P^p p_P' + \sum_l A_l^p p_l' = -\Delta \dot{m}_P^* \tag{3.54}$$

mit

$$A_E^p = -\left(\frac{\rho S^2}{A_P^u}\right)_e \tag{3.55}$$

und

$$A_P^p = -\sum_l A_l^p. \tag{3.56}$$

Am Ausströmrand setzt man:

$$p_{out} = 0 = const \Rightarrow p_{out}' = 0. \tag{3.57}$$

An allen anderen Rändern gilt:

$$\frac{\partial p'}{\partial n} = 0 \Rightarrow p_E' = p_P' \Rightarrow A_E^p = 0. \tag{3.58}$$

Wegen der gemachten Vereinfachung beim Übergang von Gleichung 3.51 zu 3.52 müssen die Impulsgleichungen und die Druckkorrekturgleichung mit den Faktoren α_u und α_p unterrelaxiert werden.

3.11 Randbedingungen

Da die Randbedingungen erheblichen Einfluß auf die Genauigkeit der Lösung und das Konvergenzverhalten haben, werden sie im folgenden näher erläutert.

Am Einströmrand werden die Geschwindigkeiten vorgeschrieben. Der Druck wird aus der Zelle auf dem Rand extrapoliert mit:

$$p_b = (\nabla p)_P * (\underline{x}_b - \underline{x}_P). \tag{3.59}$$

Am Ausströmrand wird $p = 0$ vorgeschrieben, und die Geschwindigkeiten werden extrapoliert mit:

$$c_{i,b} = (\nabla c_i)_P * (\underline{x}_b - \underline{x}_P). \tag{3.60}$$

An Wänden gilt die Haftbedingung, d.h. bei stehenden Wänden ist $\underline{c} = 0$ und an rotierenden Wänden ist $\underline{c} = \underline{\omega} \times \underline{x}$. Der Druck wird wie beim Einströmrand extrapoliert. Außerdem können im Programm noch Wände ohne Haftbedingung gesetzt werden. Diese sogenannten Euler-Wände werden wie Symmetrieränder behandelt.

Bei Symmetrierändern darf die Relativgeschwindigkeit nur tangential zum Rand strömen. Dies wird erreicht, indem von der Relativgeschwindigkeit der Zelle der Anteil normal zum Rand abgezogen wird.

$$\underline{w}_b = \underline{w}_P - (\underline{w}_P * \underline{n}) * \underline{n} \tag{3.61}$$

Der Druck wird nullter Ordnung extrapoliert.

$$p_b = p_P \tag{3.62}$$

Die periodischen Ränder werden in der Druckkorrekturgleichung voll implizit berücksichtigt. Für die Geschwindigkeiten werden die Beziehungen von Anhang B benutzt.

Kapitel 4

Lösungsverfahren

Als Basislösungsverfahren wird der SIMPLE-Algorithmus verwendet. Dieses Verfahren wird in Kapitel 4.1 beschrieben. Es zeigt sich, daß die Effizienz des SIMPLE-Algorithmus und seiner Derivate um so schlechter wird, je feiner das Gitter wird. Deshalb wird, wenn eine entsprechende Hierarchie von Gitterebenen zur Verfügung steht, ein Mehrgitterverfahren eingesetzt. Das Mehrgitterverfahren benutzt seinerseits den SIMPLE-Algorithmus als Glätter und wird in Kapitel 4.2 beschrieben.

4.1 Eingitterverfahren

Zunächst sei vorangestellt, daß das hier als Eingitterverfahren bezeichnetet Verfahren ebenfalls eine Hierarchie von Gitterebenen nutzt, allerdings nur, um eine gute Startlösung zu erhalten. Dazu wird zunächst auf dem gröbsten Gitter eine Lösung erzeugt und dann auf das nächst feinere Gitter prolongiert. Dies wird so oft wiederholt, bis man beim feinsten Gitter angekommen ist. Allerdings wird, im Gegensatz zum Mehrgitterverfahren, nicht mehr auf die groben Gitter zurückgegangen, wenn die Rechnung auf dem feinen Gitter begonnen hat.

Der SIMPLE-Algorithmus wurde erstmals von Caretto et al. veröffentlicht [7]. Von diesem Basisalgorithmus ausgehend wurden verschiedene Varianten bzw. Verbesserungen entwickelt, u.a. SIMPLER von Patankar [36], SIMPLEC von van-Doormal und Raithby [62] sowie PISO von Issa [24]. Den größten Einfluß hatte sicherlich das Buch von Patankar [36], das zur Grundlage vieler CFD-Vorlesungen an Universitäten wurde. Der Lösungsalgorithmus dieser Klasse von Verfahren kann wie folgt zusammengefaßt werden:

1. Wähle eine Anfangslösung für die Geschwindigkeiten und den Druck. In der Regel werden alle Werte zu Null gesetzt.

2. Berechne die Koeffizientenmatrix und die rechten Seiten für die Impulsgleichungen mit den Werten für den Druck und die Geschwindigkeiten aus der letzten Iteration. Löse sequentiell die Impulsgleichungen, um c_i^* zu erhalten.

3. Berechne die Koeffizientenmatrix[1] und die rechte Seite für die Druckkorrekturgleichung. Berechne p'.

4. Berechne aus $\nabla p'$ die Geschwindigkeitskorrekturen c_i' und korrigiere die Geschwindigkeiten und Drücke.

5. Falls die Residuen noch zu groß sind, gehe zu Schritt 2.

Im Code wird dies wie folgt umgesetzt:

```
//#######################################################
void FVNonlinearSolver::simple()
//#######################################################
{
  for(n=0; n<numSteps; n++)
  {
    nOuter++;
    calcGradients('u',level);
    // assemble matrix and rhs for momentum eqn.
    calcS(level,withURFU=true);
    // solve for u and v
    uvwSolver->solve(S,u,su);
    uvwSolver->solve(S,v,sv);
    uvwSolver->solve(S,w,sw);
    updateMassFluxes(level);
    // assemble matrix and rhs for pressure correction eqn.
    calcP(level);
    // solve for p'
    pp = zero;
    pSolver->solve(P,pp,su);
    // calculate grad(p')
    calcGradients('p',level);
    // correct p and u,v
    updateUVWP(level,nOuter);
    updateMassFluxes(level);
  }
}
```

Zur Lösung der linearen Gleichungssysteme stehen eine Reihe von Standardverfahren zur Verfügung. Allerdings können das ADI-Verfahren von PEACEMAN und RACHFORD [37] [10] sowie die SIP (Strongly Implicit Procedure) von STONE [52] und die modifizierte SIP von SCHNEIDER und ZEDAN [48] nicht benutzt werden, da diese Verfahren Eigenschaften der Matrix ausnutzen, die nur bei strukturierten Gittern gegeben sind.

Für allgemeine, dünnbesetzte Matrizen kommen folgende Verfahren in Betracht:

[1] Die Koeffizientenmatrix enthält im wesentlichen nur geometrieabhängige Terme und braucht deshalb unter Umständen nur einmal berechnet werden.

- Das Jacobi-Verfahren ist sehr einfach zu implementieren, hat aber sehr schlechte Konvergenzeigenschaften. Bei massiv parallelisierten Programmen kann es aber sinnvoll sein, da die neue Lösung an einem Punkt nur von alten Werten abhängt. Das Jacobi-Verfahren ist als Klasse `JacobiSolver` implementiert.
- Das Gauß-Seidel-Verfahren konvergiert doppelt so schnell wie das Jacobi-Verfahren, was normalerweise noch zu langsam ist. Als Glätter in einem Mehrgitterverfahren kann es jedoch benutzt werden, da die Glättungseigenschaften sehr gut sind und ein einzelner Iterationsschritt sehr schnell ist. Das Gauß-Seidel-Verfahren ist als Klasse `GaussSeidelSolver` implementiert.
- Das CG-Verfahren von HESTENES und STIEFEL [22] kann für positiv definite, symmetrische Matrizen angewendet werden. Eine wesentliche Steigerung der Konvergenzrate kann mit entsprechenden Präkonditionierern erreicht werden. Das CG-Verfahren ist als Klasse `CGSolver` implementiert. Als Präkonditionierer stehen der `DiagonalPreconditioner` und der `ILUPreconditioner` zur Verfügung.
- Das BiCGStab-Verfahren von VAN DEN VORST [61] ist auf nichtsymmetrische Matrizen anwendbar und als Klasse `BiCGStabSolver` implementiert.

Weitere Verfahren finden sich z.B. in [3]. Die Matrix S der Impulsgleichung ist wegen der Upwind-Diskretisierung im impliziten Teil nicht symmetrisch. Zur Lösung der Impulsgleichungen wird deshalb der `BiCGStabSolver` eingesetzt. Die Matrix P der Druckkorrekturgleichung ist dagegen symmetrisch, so daß der `CGSolver` mit `DiagonalPreconditioner` eingesetzt werden kann.

4.2 Mehrgitterverfahren

Mehrgitterverfahren (Multigrid MG) gehören zu den leistungsfähigsten Verfahren, um die algebraischen Gleichungssysteme zu lösen, die bei der Diskretisierung von Differentialgleichungen entstehen. Die Effizienzvorteile gegenüber den in Kapitel 4.1 erwähnten Verfahren kommen jedoch erst bei hinreichend feinen Gittern zum Tragen. In diesem Zusammenhang können Gitter ab etwa 32^N Gitterpunkten als fein gelten, wobei N die Dimension des Problems ist.

Eine Abschätzungen für die nötige Iterationsanzahl, um den Fehler um einen Faktor 10^p auf einem Gitter mit J^N Gitterpunkten zu reduzieren, liefert Tabelle 4.1.

Methode	Iterationsanzahl
Jacobi-Verfahren	$1/2pJ^2$
Gauss-Seidel-Verfahren	$1/4pJ^2$
SOR, ADI	$1/3pJ$
Multigrid	p

Tabelle 4.1: Abschätzung der Iterationsanzahl für verschiedene Lösungsverfahren

Mehrgitterverfahren sind anwendbar sowohl bei linearen als auch bei nichtlinearen Systemen. Sie funktionieren bei unstrukturierten Gittern mit gleicher Effizienz wie

bei strukturierten Gittern. Die Konvergenzraten sind unabhängig von der Gitterpunktanzahl, was das theoretische Optimum darstellt. Die einzige Einschränkung ist, daß die Probleme rein elliptisch sein müssen. Nichtelliptische Probleme, wie z.B. die Lösung von Transportgleichungen, können jedoch auch behandelt werden, wenn eine ausreichend starke *numerische Reibung* vorhanden ist. Damit ist gemeint, daß auf der Ebene der Gitterpunkte eine hinreichende Kopplung zwischen den Knoten vorhanden sein muß. Die Mehrgitterverfahren versagen, wenn diese Kopplung in einer Richtung zu klein wird.

Im Hinblick auf den geplanten Einsatz des Verfahrens als schnelles *Quasi-Euler*-Verfahren muß somit durch numerische Experimente geklärt werden, wie groß die Reibung, die sich aus dem viskosen und numerischen Anteil ergibt, eingestellt werden muß, um einerseits eine gut konvergierende aber andererseits auch physikalisch sinnvolle Lösung zu gewinnen. Daß diese Vorgehensweise zielführend ist, kann wie folgt begründet werden: Bei Turbulenzmodellen, die auf einem Wirbelviskositätsansatz beruhen, wird eine effektive turbulente Viskosität berechnet, die eine Funktion von Ort und Zeit ist. Bei Navier-Stokes-Verfahren, die ein solches Turbulenzmodell benutzen, wird die laminare Viskosität, die ein Stoffwert ist, durch die effektive turbulente Viskosität ersetzt. Die beim *Quasi-Euler*-Verfahren benutzte globale viskose und numerische Viskosität entspricht somit näherungsweise der effektiven turbulenten Viskosität eines Navier-Stokes-Verfahrens mit Wirbelviskositätsturbulenzmodell.

Es gibt zwei Kategorien von Mehrgitterverfahren:

- Correction Schemes (CS)
- Full Approximation Schemes (FAS)

Die einfacheren Correction Schemes sind anwendbar bei linearen Gleichungen, bei denen sowohl die Differentialgleichungen als auch die diskretisierten Gleichungen sowohl für die Unbekannten als auch für den Fehler bzw. die Korrekturen gelten. Diese Art des Mehrgitterverfahrens kann z.B. angewendet werden, wenn in einem instationären Verfahren die Poissongleichung für den Druck gelöst werden muß.

Die etwas komplizierteren Full Approximation Schemes werden bei nichtlinearen und/oder gekoppelten Differentialgleichungen angewendet, also auch im vorliegenden Fall der stationären NS-Gleichungen. Beim FAS wird auf den Grobgittern mit den Unbekannten gerechnet, bei entsprechend angepaßter rechter Seite. Als Glätter wird in diesem Fall der SIMPLE-Algorithmus eingesetzt, d.h. die Basis des Eingitterverfahrens.

4.2.1 Mehrgittergleichungen

Der Übersichtlichkeit halber sollen die Gleichungen des Mehrgitterverfahrens nur für ein grobes Gitter c (coarse grid) und ein feines Gitter f (fine grid) hergeleitet werden. Die Erweiterung auf mehrere Gitterebenen ist problemlos möglich.

Gesucht ist die Lösung von

$$\underline{\underline{A}}_f \underline{\phi}_f = \underline{b}_f, \tag{4.1}$$

wobei ϕ für c_x, c_y, c_z, p steht. Nach einigen Iterationen des SIMPLE-Algorithmus auf dem feinen Gitter erhält man die approximative Lösung $\underline{\phi}_f^*$, die die Gleichung (4.1) nur bis auf ein Residuum $\underline{r}_f$ erfüllt:

$$\underline{\underline{A}}_f^* \underline{\phi}_f^* = \underline{b}_f^* - \underline{r}_f. \tag{4.2}$$

Subtraktion von Gleichung (4.2) von (4.1) ergibt

$$\underline{\underline{A}}_f \underline{\phi}_f = \underline{b}_f + \underline{\underline{A}}_f^* \underline{\phi}_f^* - \underline{b}_f^* + \underline{r}_f \tag{4.3}$$

Nun ist eine Approximation von Gleichung (4.3) auf dem groben Gitter zu lösen. Die Approximation lautet:

$$\underline{\underline{A}}_c \underline{\phi}_c = \underline{b}_c + \underline{\tilde{\underline{A}}}_c \underline{\tilde{\phi}}_c - \underline{\tilde{b}}_c + \underline{\tilde{r}}_c. \tag{4.4}$$

Im Sinne des Grobgitteriterationsverfahrens ist dies:

$$\underline{\underline{A}}_c^* \underline{\phi}_c^* = \underline{b}_c^* + \underline{r}_c. \tag{4.5}$$

Die Größen mit der Tilde sind restringierte Feingitterwerte. Somit gilt:

$$\underline{\tilde{r}}_c = \sum \underline{r}_f, \tag{4.6}$$

$$\underline{\tilde{\phi}}_c = \mathcal{R} \underline{\phi}_f^*, \tag{4.7}$$

$$\underline{\tilde{b}}_c = \underline{b}_c(\underline{\tilde{\phi}}_c), \tag{4.8}$$

$$\underline{\tilde{\underline{A}}}_c = \underline{\underline{A}}_c(\underline{\tilde{\phi}}_c), \tag{4.9}$$

wobei $\mathcal{R}$ ein Restriktionsoperator ist, der Feingitterwerte auf das grobe Gitter interpoliert. Dabei ist folgendes zu beachten:

1. Der Term $\underline{r}_c = \underline{\tilde{\underline{A}}}_c \underline{\tilde{\phi}}_c - \underline{\tilde{b}}_c + \underline{\tilde{r}}_c$ bleibt während der Grobgitteriterationen unverändert.

2. Die Startwerte $\underline{\phi}_c^*$ werden am Anfang der Grobgitteriterationen mit $\underline{\tilde{\phi}}_c$ initialisiert. Daraus ergibt sich initial $\underline{\underline{A}}_c^* = \underline{\tilde{\underline{A}}}_c$ und $\underline{b}_c^* = \underline{\tilde{b}}_c$. Wenn nun $\underline{r}_f = 0$ ist, weil die Feingitterlösung auskonvergiert ist, so ist $\underline{\tilde{r}}_c = 0$ und die restringierte Lösung $\underline{\tilde{\phi}}_c$ erfüllt Gleichung (4.4) identisch.

3. Wenn $\underline{\tilde{r}}_c \neq 0$ ist, so wird sich ein $\underline{\phi}_c \neq \underline{\tilde{\phi}}_c$ einstellen und die Differenz $\underline{\phi}_c' = \underline{\phi}_c - \underline{\tilde{\phi}}_c$ ist die Korrektur, die die Feingitterlösung verbessert.

Besondere Beachtung brauchen noch die Behandlung des Drucks und der Massenströme. Die Kontinuitätsgleichung und damit die Druckkorrekturgleichung sind linear. Der SIMPLE-Algorithmus arbeitet mit der Druckkorrektur p'. Der Druck muß deshalb nicht restringiert werden, weil direkt mit der Druckkorrektur p_c^* auf dem Grobgitter gerechnet werden kann. Initial wird $p_c^* = 0$ gesetzt.

Die initialen Massenströme auf dem Grobgitter werden konservativ aus den Feingittermassenströmen berechnet.

$$\tilde{\dot{m}}_c = \sum \dot{m}_f \tag{4.10}$$

Während der Grobgitteriterationen berechnet sich dann der Massenstrom aus:

$$\dot{m}^*_{c,e} = \tilde{\dot{m}}_{c,e} + \rho^*_e \left(\underline{w}^*_e - \underline{\tilde{w}}_e - \Pi_\Delta\right) S_e. \tag{4.11}$$

Hat man die Grobgitterlösung durch eine festgelegte Anzahl von Iterationen des SIMPLE-Algorithmus berechnet, so werden die Korrekturen vom Grobgitter auf das Feingitter prolongiert:

$$\underline{\phi}'_f = \mathcal{P}\underline{\phi}'_c, \tag{4.12}$$

$$p'_f = \mathcal{P}p_c. \tag{4.13}$$

Für die Restriktion $\mathcal{R}$ hat sich der arithmetische Mittelwert bewährt,

$$\tilde{\phi}_c = \frac{1}{n_c}\sum_{i=1}^{n_c} \phi_{f,i}, \tag{4.14}$$

wobei n_c die Anzahl der Kinder der Grobgitterzelle ist. Die Prolongation $\mathcal{P}$ benutzt die schon beschriebene Gradientenkorrektur:

$$\phi'_f = \phi'_c + (\nabla\phi'_c) \cdot (\underline{x}_f - \underline{x}_c). \tag{4.15}$$

4.2.2 Mehrgitterzyklus

In einem Mehrgitterverfahren sind grundsätzlich unterschiedliche Strategien möglich, in welcher Reihenfolge wieviele Glättungsschritte auf welcher Ebene (Level) ausgeführt werden. Bei der sogenannten Full Multigrid (FMG) Strategie wird die Startlösung auf Level n dadurch erzeugt, daß auf Level n-1 mit dem MG-Verfahren eine auskonvergierte Lösung erzeugt wird, die dann auf den Level n prolongiert wird. Bei dieser rekursiven Definition bezeichnet Level 0 das Grobgitter. Die FMG-Strategie ergibt in der Regel optimales Konvergenzverhalten.

Ist die Startlösung auf dem feinsten Gitter vorhanden, so sind verschiedene Abarbeitungsfolgen auf den verschiedenen Leveln möglich, z.B. V- und W-Zyklen. Die Konvergenzeigenschaften der verschiedenen Strategien sind jedoch weitgehend ähnlich und dem konkreten Anwendungsfall durch numerische Experimente anzupassen. Im Code ist ein FMG mit V-Zyklus implementiert, siehe Abbildung 4.1.

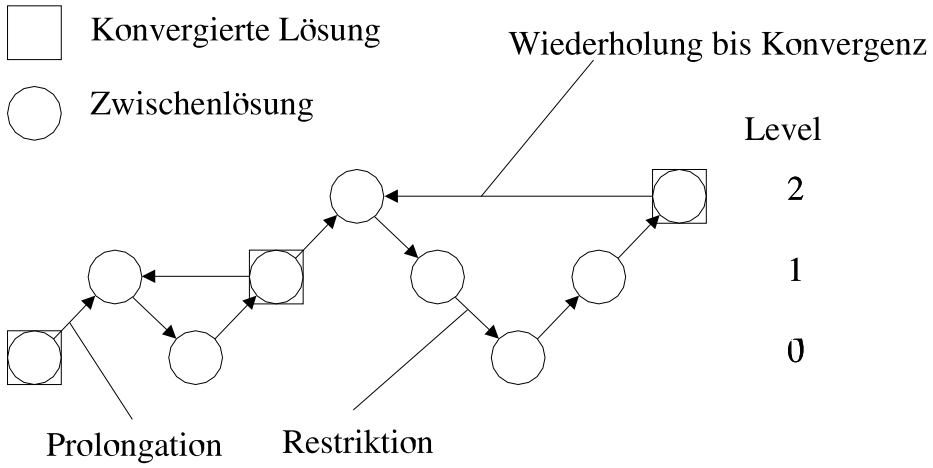

Abbildung 4.1: Schematische Darstellung des Mehrgitterzyklus mit FMG und V-Zyklus

Kapitel 5

Implementierung

Die Implementierung, d.h. die programmtechnische Umsetzung von numerischen Verfahren, wird in wissenschaftlichen Veröffentlichungen in der Regel nicht behandelt. Es wird vorausgesetzt, daß die Implementierung sowohl korrekt als auch effizient ist. Andere Kriterien spielen keine Rolle. So wird z.B. in den Dissertationen von Riedel [42], Bader [2], Schuster [49] und Ritzinger [43] die Implementierung lediglich von Riedel erwähnt, nämlich daß die Programmierung mit C und Fortran bei Verwendung der Parallelisierungssoftware PVM einen kostengünstigen Einsatz auf PCs zuläßt.

In Zukunft wird aber die Implementierung immer wichtiger werden, da die Software für numerische Verfahren immer komplexer wird. Dies liegt zum einen daran, daß die künftigen Strömungslöser in zunehmendem Maße mit unstrukturierten Gittern arbeiten werden, und zum anderen an der wachsenden Zahl der zu modellierenden physikalischen Effekte. Im Bereich der hydraulischen Maschinen sind dies zum Beispiel instationäre Strömungen, bewegte Gitter, verschiedene Turbulenzmodelle, Kavitation etc. Solch komplexe Softwaresysteme können dann nicht mehr von einer einzelnen Person entwickelt werden.

Für die Implementierung spielt die Auswahl der Programmiersprache eine zentrale Rolle, wobei folgende Punkte beachtet werden müssen:

1. Laufzeiteffizienz der kompilierten Programme.
2. Portabilität der Quellprogramme.
3. Eignung für große Softwareprojekte, d.h. Teamarbeit.
4. Abstraktionsniveau der Modellierung.

In der Vergangenheit wurde ein Großteil der numerischen Software in Fortran implementiert, da

- Fortran sehr weit verbreitet ist und somit viele in dieser Sprache programmierte Programme und Bibliotheken vorhanden sind,
- Fortran-Compiler sehr laufzeiteffiziente Programme erzeugen und

- Fortran recht gut portabel ist.

In dem objektorientierten Framework von MÜLLER [34] wurde dagegen C++ als Programmiersprache gewählt. SZILAGYI [53] konnte zeigen, daß für den bekannten Linpack-Benchmark die Implementierungen für Fortran, C und C++ sowohl unter Linux als auch unter Windows NT auf PCs praktisch gleiches Laufzeitverhalten haben. Die Portabilität von C++ ist gleich gut wie die von Fortran.[1] Entscheidende Vorteile gegenüber Fortran hat C++ aber durch die objektorientierten Aspekte der Sprache. Mit Klassen, Datenkapselung, Vererbung und Polymorphie eignet sich C++ auch sehr gut für sehr große und komplexe Softwareprojekte.

5.1 Designentscheidungen

Bei der Implementierung des Finite-Volumen-Codes wurde bei einigen Aspekten der Rahmen des Konzeptes von MÜLLER [34] bewußt verlassen. Die Gründe dafür waren:

- Die Pressure-Schur-Complement-Schreibweise, also $\underline{\underline{S}}\,\underline{u} + \underline{\underline{B}}\,\underline{p} = \underline{q}$ und $\underline{\underline{B}}^T\,\underline{u} = 0$ ist nur bei symmetrieerhaltenden Diskretisierungen gültig, siehe auch Verstappen und Veldman [63]. Da in der vorliegenden Arbeit die Diskretisierung nicht symmetrieerhaltend ist, siehe Kapitel 3.4, kann die Matrix der Druckkorrekturgleichung nicht als $\underline{\underline{B}}^T\underline{\underline{S}}^{-1}\underline{\underline{B}}$ dargestellt werden. Folglich mußten die Lösungsverfahren für das nichtlineare algebraische Gleichungssystem neu implementiert werden.

- Beim Konzept von [34] führt die doppelte Speicherung der Unbekannten in Nodes und Unbekanntenvektor zu unerwünschter Redundanz und erhöhtem Speicherplatzbedarf.

- Die angestrebte größtmögliche Allgemeingültigkeit des Programms von [34] führte zu einer komplizierten Implementierung mit einer großen Zahl von Klassen.

- Da bei der Finite-Volumen-Diskretisierung alle Variablen (c_x, c_y, c_z, p, p') der Zellmitte zugeordnet werden, sind besonders einfache Speichertechniken möglich.

5.2 Klassenstruktur

Die Klassen, die durch die Implementierung des Finite-Volumen-Verfahrens neu hinzugekommen sind, beginnen mit den Buchstaben `FV`. Die für die FEM Diskretisierung

[1] C++ wird nicht nur auf PCs, Workstations und Superrechnern eingesetzt, sondern z.B. auch bei Embedded Systems, was aber im Zusammenhang mit Strömungsberechnungsprogrammen kein Vorteil ist.

spezifischen Klassen und Dateien beginnen mit den Buchstaben `NVS`.[2] Eine Ausnahme bilden die von der Klasse `NVSData` abgeleiteten Klassen, die für das Bereitstellen von Einströmrandbedingungen und einer eventuell vorhandenen exakten Lösung zuständig sind. Folgende Klassen sind im wesentlichen neu hinzugekommen:

- `FVNet` erbt von der Klasse `Net`. Eine wesentliche Erweiterung der Klasse `FVNet` ist, daß alle Zellflächen (Faces) in Listen verwaltet werden, so daß Schleifen über die Faces ebenso effizient sind wie Schleifen über die Zellen.

- `FVNonlinearSolver` erbt von der Klasse `NonlinearSolver`. Im Gegensatz zur Klasse `NVSNonlinearSolver` übernimmt die Klasse `FVNonlinearSolver` auch die Assemblierung der globalen Matrizen und rechten Seiten. Dadurch werden die im FEM-Code benutzten Klassen für die Assemblierung überflüssig.

- `FVNonlinearMGSolver` erbt von der Klasse `FVNonlinearSolver` und übernimmt die Abarbeitung des Multigrid V-Zyklus. Als Glätter wird die Methode `simple()` der Klasse `FVNonlinearSolver` benutzt.

- `FVSimulator` erbt von `Simulator`. Die Unbekanntenvektoren und die Koeffizientenmatrizen sind Datenelemente der Klasse `FVSimulator`, wodurch die Klassen `LinearEquationSystem` und `NonlinearEquationSystem` überflüssig werden.

- Die Klasse `UniformFVSimulator` erbt von der Klasse `FVSimulator`. In der Methode `UniformFVSimulator::simulate()` ist die Abfolge von Verfeinerung und Lösung implementiert.

- Die Klasse `FVSpaceQuad` erbt von `SpaceQuad` und wird für Faces im Gebietsinneren benutzt. Die Klasse hat Datenelemente für den Massenstrom und für das Mehrgitterverfahren außerdem für den sogenannten *frozen massflux*.

- Die Klassen `FVInflowQuad`, `FVOutflowQuad`, `FVBoundaryQuad`, `FVEulerQuad` und `FVPeriodicQuad` erben von der Klasse `FVSpaceQuad` und sind im wesentlichen für die Implementierung der Randbedingungen zuständig.

- Die Klasse `FVHexa` erbt von der Klasse `Hexa`. Durch Speicherung einer ID und eines Zeigers auf die Lösungsvektoren kann bei minimalen Speicherbedarf trotzdem auf alle Strömungswerte zugegriffen werden. Die Methoden zum Berechnen der lokalen Koeffizienten der Matrizen sind in der Klasse `Cell` implementiert, da diese allgemeingültig sind auch für Prismen und Tetraeder. Würde der FEM-Anteil aus dem Code entfernt werden, so könnte die gesamte Funktionalität der Klasse `FVHexa` in die Klasse `Cell` verlagert werden.

Eine Übersicht der mit den Netzkomponenten zusammenhängenden Klassen gibt Abbildung 5.1.

Aufgrund der für das FVM-Verfahren getroffenen Designentscheidungen gibt es für folgende Klassen des FEM-Codes keine entsprechenden Klassen im FVM-Code.

[2] Die Buchstaben NVS stehen für **N**a**v**ier-**S**tokes.

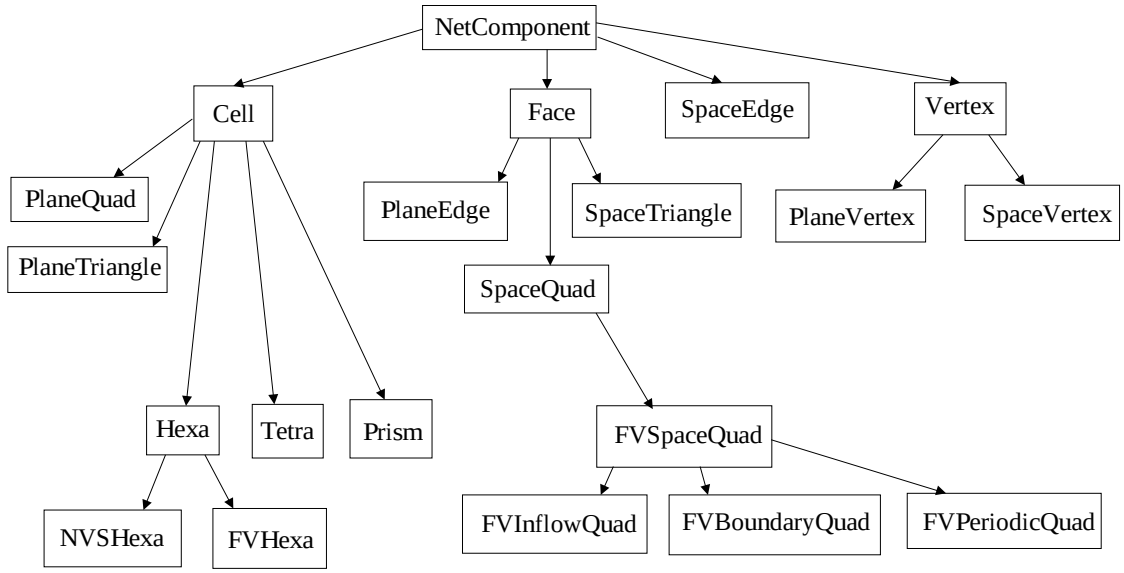

Abbildung 5.1: Klassenansicht der wesentlichen Netzkomponenten

- `Node` und abgeleitete Klassen: Die Klasse `FVHexa` kann direkt auf den Unbekanntenvektor zugreifen.
- `SingleLevelNode` und abgeleitete Klassen: Die Unbekannten werden nur einmal im Unbekanntenvektor gespeichert.
- `NonlinearEquationSystem` und `LinearEquationSystem`: Die Unbekanntenvektoren sind in einem Objekt der Klasse `FVSimulator` verankert. Die während des Lösens benötigten Koeffizientenmatrizen werden in der Methode `FVNonlinearSolver::solve()` temporär angelegt.
- Klassen `Assembler` und `CRSAssembler`: Die Assemblierung erfolgt jetzt in den Methoden `FVNonlinearSolver::calcS()` und `FVNonlinearSolver::calcP()`.

Somit wurde eine erhebliche Reduzierung der Anzahl der Klassen erreicht, wodurch die Einarbeitung eines neuen Bearbeiters vereinfacht wird. Durch Verwendung einer Quellcodeverwaltung und einer Testfalldatenbank ist sichergestellt, daß auch nach einer Modifikation bzw. Erweiterung des Programms alle Testfälle weiterhin funktionieren.

Kapitel 6

Gittergenerierung

Im Prozeß der numerischen Simulation von Strömungsvorgängen ist die Gittergenerierung mindestens so wichtig wie die numerische Lösung der Strömungsgleichungen, da zum einen die Gitter starken Einfluß auf Konvergenzverhalten und Lösungsgenauigkeit der Strömungslöser haben und zum anderen der größte Teil der Arbeitszeit des Berechnungsingenieurs für die Gittergenerierung benötigt wird.

Es gibt eine außerordentlich große Zahl von Veröffentlichungen zum Thema Gittergenerierung. Es sei deshalb hier nur auf die Standardwerke von Thompson et al. [58], Arcilla et al. [1], Carey [8] und Thompson et al. [57] verwiesen. Roberts hat im Internet unter der Adresse

`http://www-users.informatik.rwth-aachen.de/~roberts/software.html`

eine sehr ausführliche Liste von kommerziellen und freien Gittergeneratoren veröffentlicht. Die Liste läßt sich nach verschiedenen Kriterien durchsuchen, und sie ist ein exzellenter Startpunkt für eine Online-Recherche.

6.1 Klassifizierung der Gittertypen

Da es keine physikalischen Gesetze der Gittergenerierung gibt, ist der Gittergenerierungsprozeß immer auch von der Erfahrung und Intuition des Anwenders geprägt. Jede Methode, die zu einem brauchbaren Gitter führt, ist eine zulässige Methode. Insofern ist die folgende Einteilung der Gittertypen weder vollständig noch die einzig denkbare. Dennoch wird damit ein Großteil der gebräuchlichen Gittertypen erfaßt:

1. Strukturierte Gitter
   Die Gitterpunkte werden in einer dreidimensionalen Matrix[1] angeordnet. Nachbarpunkte können durch entsprechende Zugriffe in der Matrix ermittelt werden. Als Generierungsmethoden werden in der Regel folgende benutzt:

   - Algebraische Methoden, die sehr einfach und sehr schnell sind.

[1] Bei 3D Problemen.

- Elliptische Methoden, die eine Kontrolle von Gitterabstand und -winkel an den Rändern ermöglichen. Die elliptischen Methoden sind relativ rechenintensiv und deshalb langsam.
- Hyperbolische Methoden, die sich nur für Umströmungen eignen, wenn die genaue Lage des Fernfeldrandes nicht wichtig ist.

Strukturierte Gitter können nur für sehr einfache Topologien erzeugt und für Strömungsrechnungen eingesetzt werden.

2. Blockstrukturierte Gitter
   Das gesamte Rechengebiet wird unstrukturiert mit Blöcken ausgefüllt, wobei jeder Block aus einem strukturierten Gitter besteht. Die Generierungsmethoden entsprechen denen der strukturierten Gitter. Mit blockstrukturierten Gittern können auch sehr komplizierte Topologien vernetzt werden, wobei ein manueller Aufwand von 2 Wochen für z.B. eine Flugzeugkonfiguration jedoch nicht ungewöhnlich ist.

3. Unstrukturierte Gitter
   Die Zellen sind beliebig angeordnet, wobei jede Zelle Informationen über ihre Nachbarzellen abspeichern muß. Die Zellen selbst können in einem Vektor oder in einer Liste gespeichert sein. Unstrukturierte Gitter können Dreiecke bzw. Tetraeder enthalten, aber auch Vierecke bzw. Hexaeder. Als Generierungsmethoden für Triangulierungen gibt es

   - Methoden, die auf der Delaunay-Triangulierung beruhen, sowie
   - die Advancing Front Methode.

   Beide Methoden können sehr gut automatisiert werden, so daß der manuelle Aufwand sehr klein ist.

4. Adaptive Gitter
   Bei adaptiven Gittern steuert die temporäre Lösung der Strömungsgleichungen den Gitterverfeinerungs- bzw. -vergröberungsprozeß. Diese Verfahren werden insbesondere bei transsonischen Strömungen eingesetzt, weil dort Stöße mit sehr feinen Gittern aufgelöst werden müssen, wobei deren Lage sich erst durch die Lösung ergibt. Die Grundgitter können strukturiert, blockstrukturiert oder unstrukturiert sein.

Von 1 nach 4 wird die Gittergenerierung flexibler, d.h. es können kompliziertere Geometrien vernetzt werden. Dafür steigt der Aufwand für die Implementierung der Strömungslöser von 1 nach 4. Außerdem steigt der Rechenaufwand der Strömungslöser von 1 nach 3, da zum einen die sehr effizienten Gleichungslöser, wie z.B. SIP und ADI nur für strukturierte Gitter anwendbar sind und zum anderen bei unstrukturierten Gittern die Ermittlung der Nachbarzellen etwas mehr Zeit in Anspruch nimmt.

Die künftige Entwicklung der Gittergenerierung wird durch drei wesentliche Kriterien geprägt:

- Die Rechenleistung der Computer steigt bei im wesentlichen stabilen Preisen weiterhin exponentiell an. Verantwortlich dafür sind immer schneller werdende Prozessoren und die Erhöhung der Knotenzahl bei Parallelrechnern.

- Größere Komplexität der Geometrien durch die zunehmende Berücksichtigung von Details in den Geometrien.

- Durch weitere Verbreitung von CFD-Methoden werden Gitter in zunehmenden Maße auch von Nichtspezialisten erzeugt.

Der Trend geht deshalb zu einer Vereinfachung der Gittergenerierung, auch wenn dafür aufwendigere Strömungslöser gebraucht werden.

6.2 Anforderungen an Finite-Volumen-Gitter

Die Finite-Volumen-Diskretisierung kann bei fast beliebigen Gittern eingesetzt werden, siehe Kapitel 3.1 mit Abbildung 3.1. Die Genauigkeit bei endlichen Gitterauflösungen hängt aber stark von der Zellenform ab. Der Winkel zwischen Flächennormalenvektor $\underline{S}_e$ und der Verbindung der Zellmittelpunkte $\underline{x}_E - \underline{x}_P$ sollte so klein wie möglich sein. In diesem Sinne sind übrigens gleichschenklige Dreiecke als orthogonales Gitter zu bezeichnen. Bei der Approximation der Integrale mit der Mittelpunktregel und der Verwendung von zentralen Differenzen (CDS) findet im Falle von Vierecken bei gegenüberliegenden Flächen eine Auslöschung von Fehlertermen statt, so daß Vierecke genauer sind als Dreiecke.

Die Zellen sollten außerdem in Strömungsrichtung ausgerichtet sein, da sonst eine zusätzliche, d.h. numerische, Diffusion entsteht. Diese Forderung kann unter günstigen Umständen mit Vierecken erfüllt werden. Schließlich müssen Grenzschichten senkrecht zur Kontur sehr fein aufgelöst werden, während in Strömungsrichtung erheblich weniger Punkte benötigt werden. Dafür eignen sich Vierecke erheblich besser als Dreiecke. Im 3D-Fall eignen sich dementsprechend Hexaeder und Prismen erheblich besser zur Grenzschichtvernetzung als Tetraeder.

Für die Gittergenerierung ergeben sich die folgenden Richtlinien:

- Minimierung der Winkel zwischen Flächennormalenvektor $\underline{S}_e$ und dem Verbindungsvektor der Zellmittelpunkte $\underline{x}_E - \underline{x}_P$,

- Minimierung der Abstände $\underline{x}_e - \underline{x}_{e'}$, siehe Abbildung 3.3,

- Minimierung der Größenunterschiede zwischen benachbarten Zellen,

- Minimierung der Tetraederanzahl,

- Ausrichtung der Zellen nach der Strömungsrichtung,

- Auflösen der Grenzschichten mit Hexaedern bzw. Prismen.

Um eine quantitative Aussage über die Gitterqualität zu machen, werden folgende drei Kennziffern berechnet:

$$c_{g,1} = 1 - \sqrt{\frac{1}{n_f}\sum_{e=1}^{n_f}\left(1 - \frac{\underline{S}_e * (\underline{x}_E - \underline{x}_P)}{|\underline{S}_e| * |\underline{x}_E - \underline{x}_P|}\right)^2}, \tag{6.1}$$

$$c_{g,2} = 1 - \sqrt{\frac{1}{n_f}\sum_{e=1}^{n_f}\left(1 - \frac{(\underline{x}_e - \underline{x}_P) * (\underline{x}_E - \underline{x}_P)}{|\underline{x}_e - \underline{x}_P| * |\underline{x}_E - \underline{x}_P|}\right)^2}, \tag{6.2}$$

$$c_{g,3} = 1 - \sqrt{\frac{1}{n_f}\sum_{e=1}^{n_f}\left(1 - \frac{V_{min}}{V_{max}}\right)^2}, \tag{6.3}$$

wobei die Summation über die inneren Flächen läuft und V_{min} und V_{max} das kleinere und das größere Volumen der zugeordneten zwei Zellen sind. Es ergibt sich $c_g = 1$ für ein perfektes Gitter und $c_g \to 0$ für sehr schlechte Gitter.

6.3 Gittergenerierung für Turbinen

Eine Francis-Turbine besteht aus den Elementen Spirale, Traverse, Leitrad, Laufrad und Saugrohr. Die eigentliche Energieumsetzung findet im Laufrad statt. Es ist deshalb sinnvoll, sich bei der numerischen Strömungsberechnung in einem ersten Schritt auf das Laufrad zu beschränken. Es wird dann eine umfangsgemittelte stationäre Zuströmung vorgegeben, so daß die Strömung durch die einzelnen Kanäle des Laufrades periodisch ist. Deshalb ist es möglich, sich bei der Laufradberechnung auf einen repräsentativen Schaufelkanal mit entsprechenden periodischen Randbedingungen zu beschränken, was den numerischen Aufwand erheblich senkt.

Die Topologie des Strömungsgebietes ist relativ einfach. Legt man das Rechengebiet zwischen zwei Schaufeln, so ist das Rechengebiet topologisch ein Kanal mit periodischen Rändern im Zulauf und im Nachlauf.

Die bislang am Lehrstuhl für Hydraulische Maschinen der TU München eingesetzten H-Gitter haben sich sehr gut bewährt, siehe z.B. Richter [41], Ritzinger [43] oder Riedel [42]. Dabei werden in der Meridianebene algebraische Gitter erzeugt, die zu geometrischen Stromlinien führen. Diese Stromlinien wiederum führen zu Rotationsflächen, auf denen die sogenannten Blade-to-Blade Gitter mit elliptischen Methoden erzeugt werden. Gitterorthogonalität und -abstand auf den Schaufeln werden dabei mit *Forcing Functions* kontrolliert [51]. Die Blade-to-Blade Gitter werden schließlich übereinander gestapelt, um das dreidimensionale Gitter zu erzeugen. Das Blade-to-Blade Gitter auf der Nabe einer Francis-Turbine ist in Abbildung 6.1 dargestellt.

Eine detaillierte Beschreibung dieser Methode für Turbomaschinen findet sich bei Schilling [47]. Das Problem dieser Methode ist jedoch, daß bei starken Geometrieänderungen in der Regel eine manuelle Anpassung von Einstellparametern nötig ist, was in einem Optimierungssystem natürlich unerwünscht ist.

Es wurde deshalb ein Programm entwickelt, das für den Einsatz in dem FEM-Code von [34] unstrukturierte Hexaedergitter generiert. Aus Gründen, die im folgenden Kapitel 6.3.1 dargestellt werden, erwies sich dieser Weg allerdings als nicht gangbar.

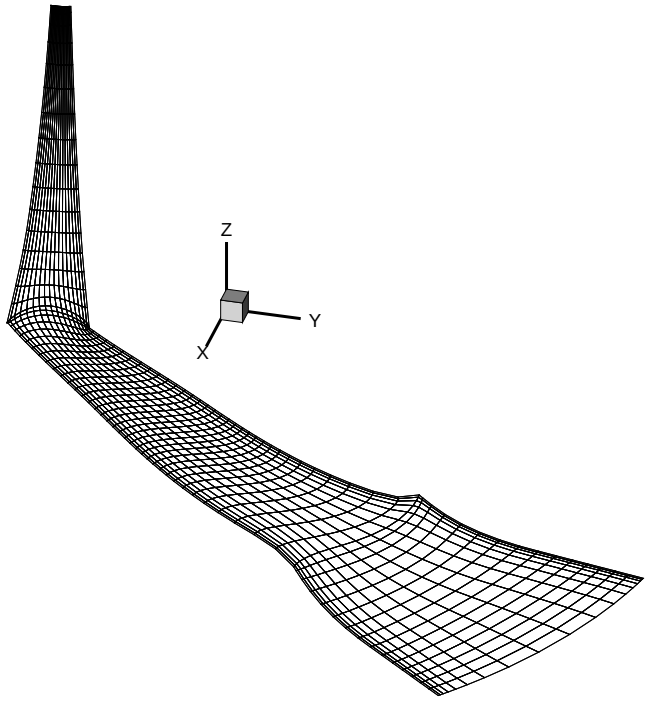

Abbildung 6.1: Ein H-Gitter auf der Nabe für Testfall FT100

Für den vorliegenden Code wird deshalb zur Zeit ein blockstrukturierter Gittergenerator benutzt. Die blockstrukturierten Gitter werden mit einem von THUM, REINELT, SZILAGYI und EINZINGER [59] entwickelten Konverter in mehrgitterfähige unstrukturierte Gitter konvertiert, siehe Kapitel 6.3.2.

Langfristig ist jedoch eine Hybridgittergenerierung anzustreben, siehe auch KALLINDERIS [25], die in Kapitel 6.3.3 diskutiert wird.

6.3.1 Unstrukturierte Hexaedergitter

Die im folgenden beschriebene Gittergenerierung war für den Einsatz im FEM-Programm von MÜLLER [34] vorgesehen und sollte folgende Anforderungen erfüllen:

1. Zellform Hexaeder.

2. Matching Interfaces.

3. Sehr geringe Anzahl von Zellen für Grobgitter, um gute Konvergenzeigenschaften des Mehrgitterverfahrens zu erreichen.

4. Möglichkeit der Gitterverfeinerung im Strömungslöser, um adaptive Verfeinerung zu ermöglichen. Deshalb müssen die Randflächen der Zellen die Information über die kontinuierlichen Flächen der Gebietsberandungen, z.B. die Tensorprodukt-B-Splinefläche der Schaufel, speichern. Diese Flächen müssen

als C++ Klassen sowohl im Gittergenerator als auch im Strömungslöser zur Verfügung stehen.

5. Große Robustheit während einer Optimierungsrechnung, also Einstellparameter nur, wenn diese zum Beispiel als Funktion der spezifischen Drehzahl n_q vorgegeben werden können.

6. Vernetzung für Francis-Turbinenlaufräder.

7. Programmierung in C++.

8. Unstrukturierte Gitter.

Keiner der freien Gittergeneratoren in der anfangs des Kapitels erwähnten Liste von ROBERTS konnte obige Anforderungsliste erfüllen. Es wurde deshalb eine Eigenentwicklung vorgenommen.

Als Eingabedateien wurde die Datei `des.dat` für die Tensorprodukt-B-Splinefläche der Schaufel und die Datei `meri.dat` für die r-z B-Splinekurven der Nabe (Hub), Deckscheibe (Shroud) sowie Einström- und Ausströmrand gewählt. Weitere Informationen z.B. über die Anzahl der Schaufeln werden jetzt der Datei `fvmOperating.dat` entnommen. Abbildung 6.2 zeigt die Eingabe- und Ausgabedateien des Programms `femgrd12`, wobei die 12 die Anzahl der Zellen im Grobgitter bezeichnet. Die Programme `refiner` und `grid2grid` sind lediglich Konverter, deren Funktion unten erklärt wird.

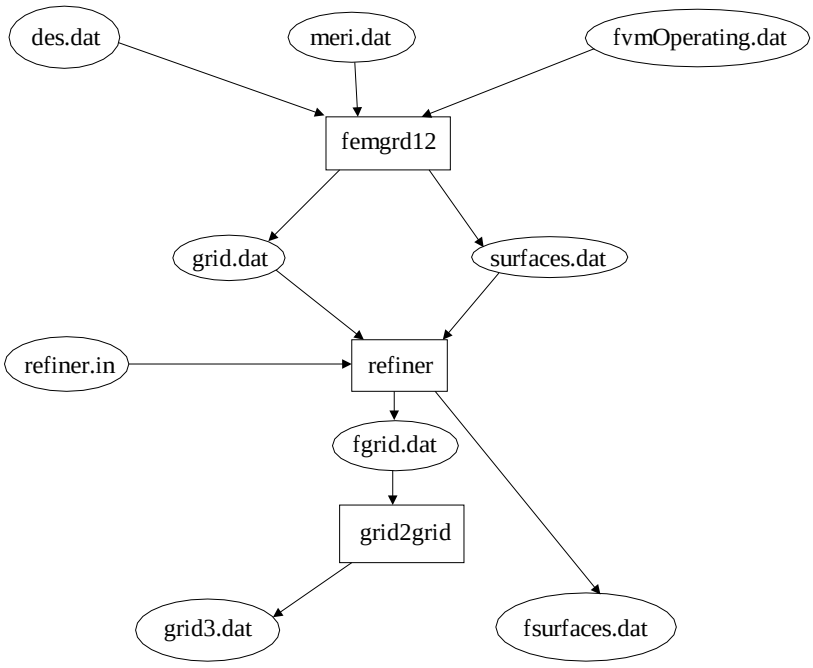

Abbildung 6.2: Programme für unstrukturierte Hexaedergitter

Die Schaufel ist ein Objekt der Klasse `BSplineSurface`. Die zweidimensionalen r-z Kurven der Nabe und Deckscheibe können mit der Klasse `BRotSurface` in dreidimen-

sionale Flächen umgewandelt werden, womit in Abbildung 6.3 schon drei wesentliche Randflächen des Rechengebietes erzeugt sind.

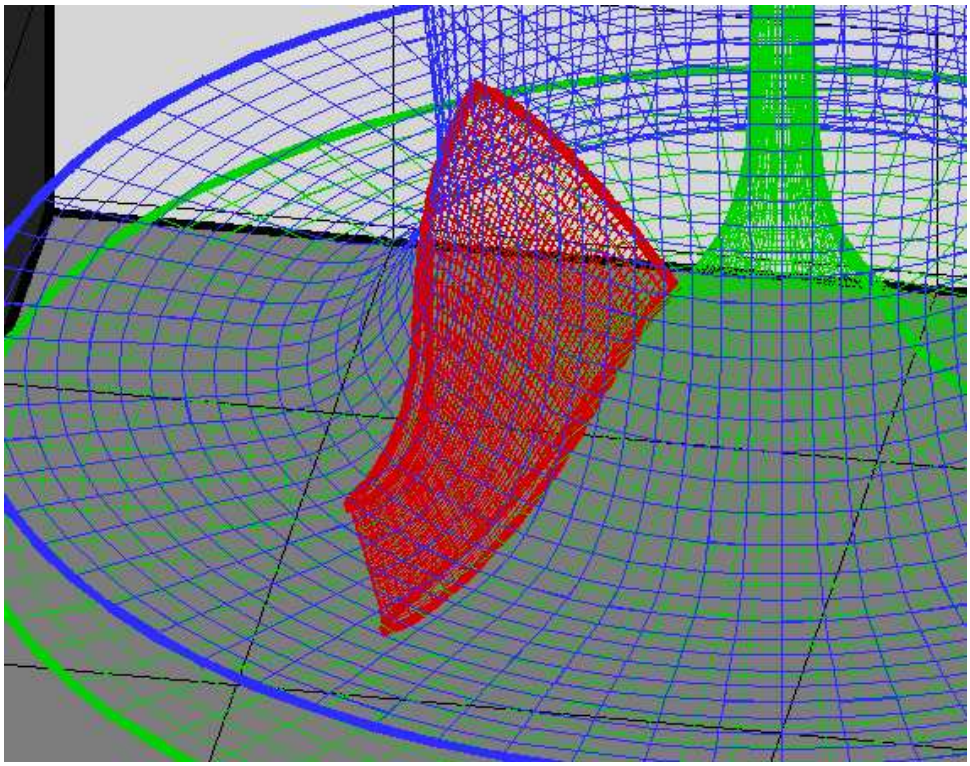

Abbildung 6.3: Ausgangsflächen der Gittergenerierung:Grün = Nabe, Blau = Deckscheibe, Rot = Schaufel

Die Schnittkurven von Schaufel mit Nabe und Deckscheibe sowie die druck- und saugseitige Hinterkantenlinie zeigt Abbildung 6.4. Diese ergeben sich durch elementare Schnittroutinen der Klassen `BSplineSurface` und `BRotSurface`.

Da sich die Geometrie auf Nabe und Deckscheibe nicht grundsätzlich unterscheidet, wird für das Grobgitter nur eine Schicht von Zellen zwischen Nabe und Deckscheibe gelegt. Damit reduziert sich die Gittergenerierung topologisch auf zwei zweidimensionale Gittergenerierungen auf Nabe und Deckscheibe.

Für die Gittertopologie muß nun entschieden werden, ob das Gitter a) zwischen zwei Schaufeln oder b) um eine Schaufel gelegt werden soll. Die zwei Alternativen unterscheiden sich nur wenig. Da die periodischen Flächen in den Impulsgleichungen explizit behandelt werden, sollte Variante a) numerisch etwas besser sein. Bei Variante b) braucht nur eine Schaufelfläche gespeichert werden. Nach einer Vielzahl von Versuchen wurde schließlich die in Abbildung 6.5 dargestellte Anordnung von Zellen ausgewählt.

Obwohl das Gitter unstrukturiert ist, unterliegt es doch einer Reihe von Einschränkungen, von denen insbesondere die periodischen Ränder großen Einfluß haben. So müssen die Ränder A+ und A-, F+ und F-, D+ und D- periodisch sein. Im Nachlauf entstehen sehr lange Zellen (E), da die Nabe dort einen sehr kleinen Radius aufweist. Die Zellen F+ und F- decken fast die gesamte Schaufel ab, wodurch sehr unterschiedliche Zellgrößen auf Druck- und Saugseite der Schaufel entstehen.

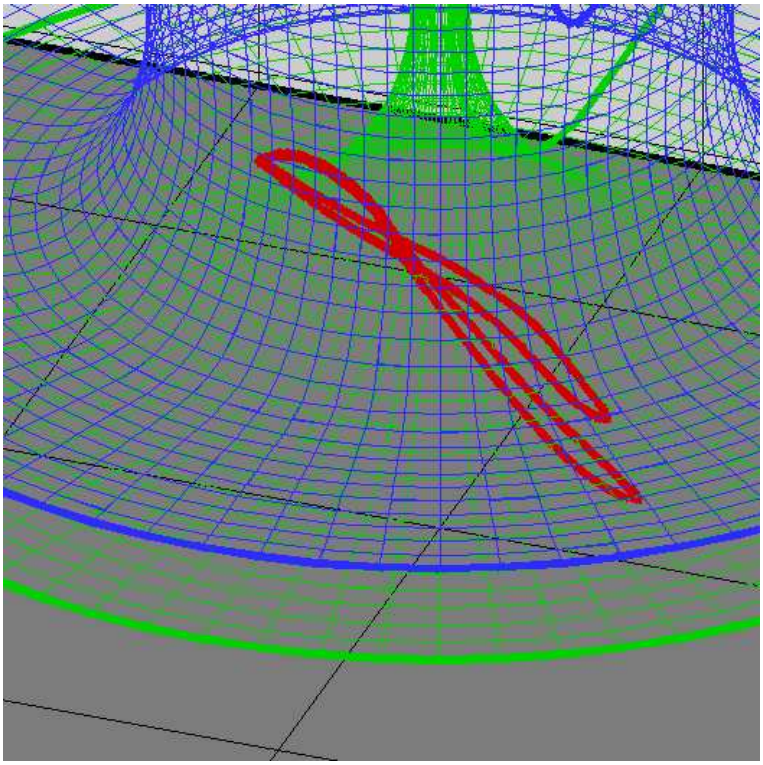

Abbildung 6.4: Schnittlinien der Schaufel mit Nabe und Deckscheibe

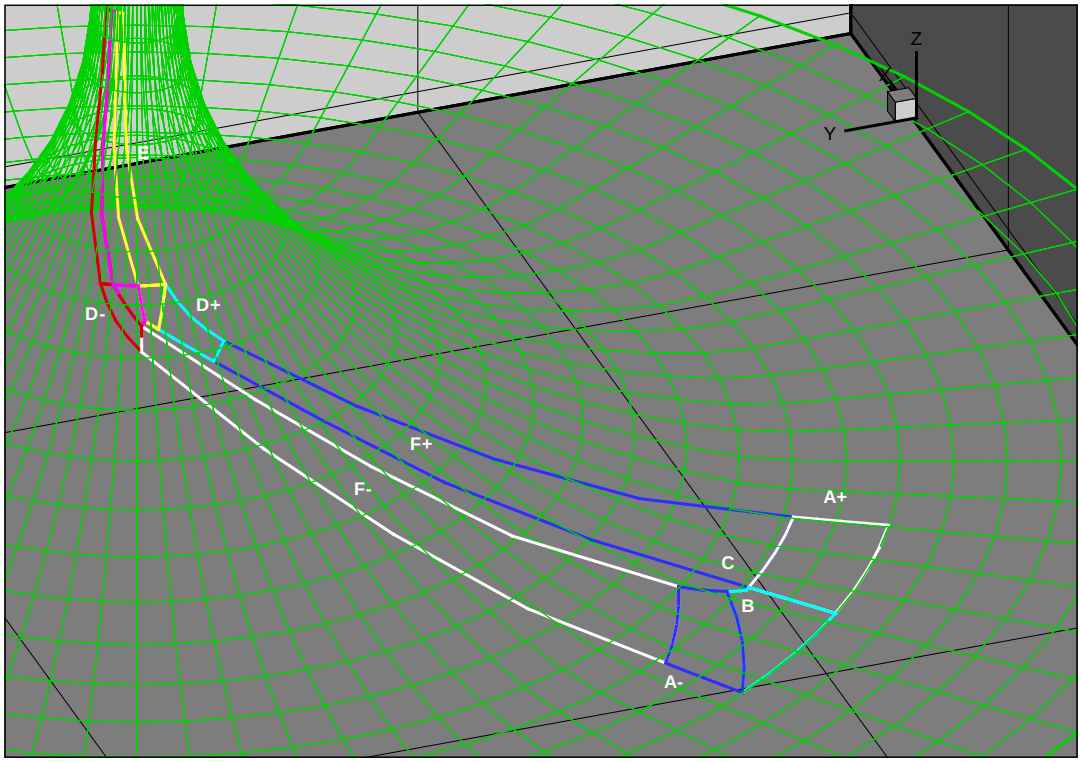

Abbildung 6.5: Anordnung der 12 Grobgitterzellen auf der Nabe

Das Gitter auf der Deckscheibe hat den gleichen Aufbau wie auf der Nabe. Allerdings ist der Nachlauf hier viel kürzer als auf der Nabe und gleichzeitig der Radius viel

größer, so daß sehr breite Zellen entstehen. Dargestellt in Abbildung 6.6 sind auch die 3 Zellflächen des Ausströmrandes.

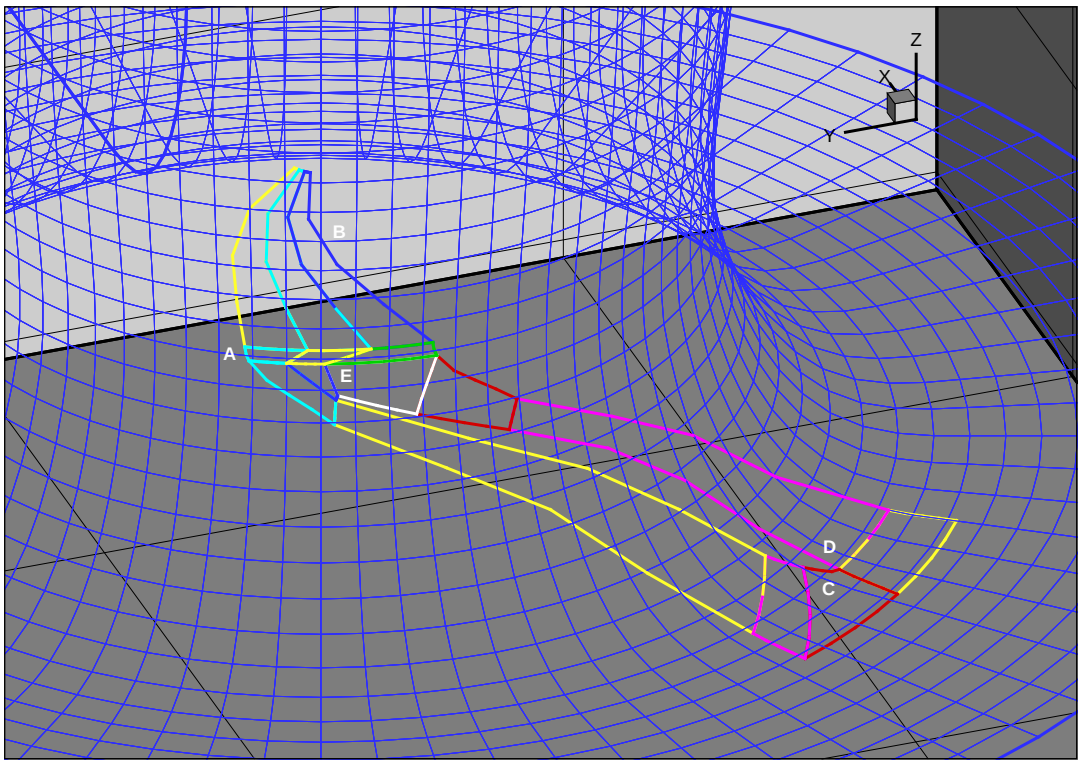

Abbildung 6.6: Anordnung der 12 Grobgitterzellen auf der Deckscheibe sowie der 3 Ausströmflächen

Die erheblichen Größenunterschiede der Zellen, insbesondere an der Vorder- und Hinterkante der Schaufel sollten nach dem Pflichtenheft von Seite 39 durch lokale Verfeinerung ausgeglichen werden. Die lokale Verfeinerung wurde jedoch im FEM-Strömungslöser von [34] nur vorbereitet, aber nicht implementiert.

Es wurde deshalb das Programm `refiner` geschrieben, daß eine selektive Unterteilung der Grobgitterzellen zuläßt, wobei allerdings die Bedingungen für Periodizität und Matching Interfaces beachtet werden müssen. So können in Abbildung 6.5 die Zellen F+ und F- entlang der Schaufel in eine gleiche Zahl von Zellen unterteilt werden. Eine Unterteilung der Zelle D+ entlang der Schaufel würde aber wegen der Periodizität eine entsprechende Unterteilung der Zelle D- zur Folge haben und wegen der Bedingung der Matching Interfaces somit auch eine Unterteilung der Zellen D+, F+ und A+ quer zur Schaufel, was hier aber unerwünscht wäre.

Abbildung 6.7 zeigt das Gitter auf Nabe, Schaufel und Ausströmrand nach der selektiven Unterteilung mit dem `refiner`. Von Nabe zu Deckscheibe wurde in 2 Zellschichten unterteilt und die Zellen F+ und F- in Abbildung 6.5 wurden in 4 Zellen unterteilt.

Das Programm `grid2grid` konvertiert die Datei `fgrid.dat` schließlich in ein Format, bei dem grundsätzlich auch Tetraeder möglich sind. Abbildung 6.8 zeigt schließlich das von Level 0 auf Level 3 verfeinerte Gitter mit den Isolinien des Drucks der Lösung. Das Level-3-Gitter hat 21504 Zellen. Abbildung 6.9 zeigt das Gitter auf der

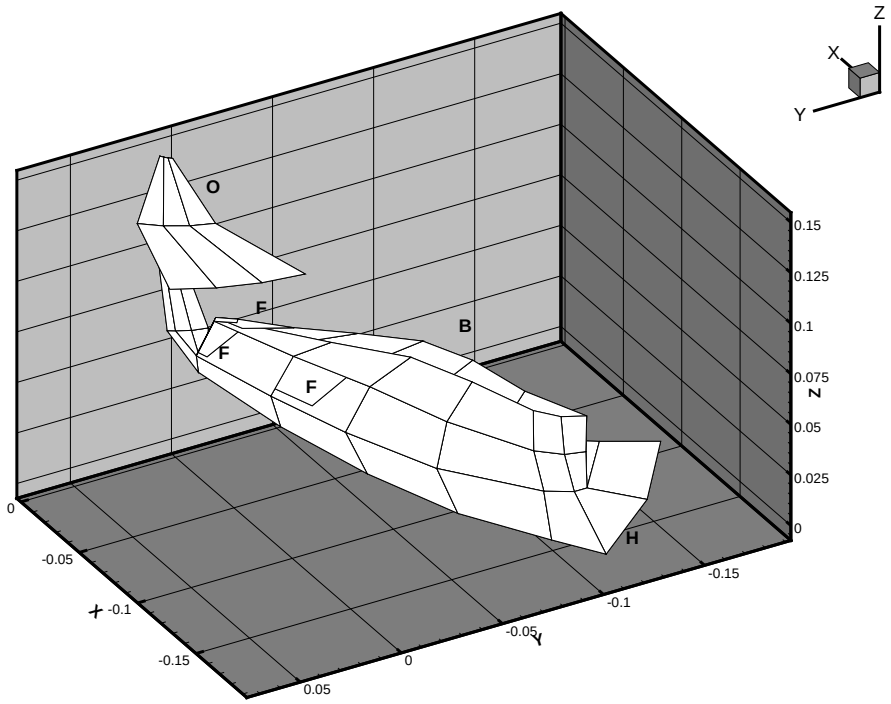

Abbildung 6.7: Selektiv verfeinertes Gitter auf der Nabe und der Schaufel sowie dem Ausströmrand; F=Fehler im Visualisierungsprogramm Tecplot

Nabe. Man erkennt, daß die Schaufelkontur bei der Verfeinerung korrekt nachgeführt wurde. Man erkennt aber auch, daß die Gitterauflösung an der Vorderkante (LE) auf der Saugseite (SS) sehr schlecht ist und die Auflösung des hinteren Teils der Schaufel auf Druck- und Saugseite stark unterschiedlich ist.

Die wesentlichen Nachteile dieser Gittergenerierungsmethode sind somit:

- Stark eingeschränkte Flexibilität trotz unstrukturierter Gitter.
- Große Unterschiede in der Gitterfeinheit, die bei globaler Verfeinerung nicht verschwinden.
- Gittergenerierung nicht hinreichend allgemeingültig für eine Optimierung, da sowohl bei den Grobgittern als auch bei den Feingittern entartete Zellen entstehen können.
- Die Zellen des Grobgitters entsprechen den Blöcken eines blockstrukturierten Gitters. Allerdings werden die Zellen uniform verfeinert, während die Zellen in den einzelnen Blöcken unterschiedliche Größen haben können.

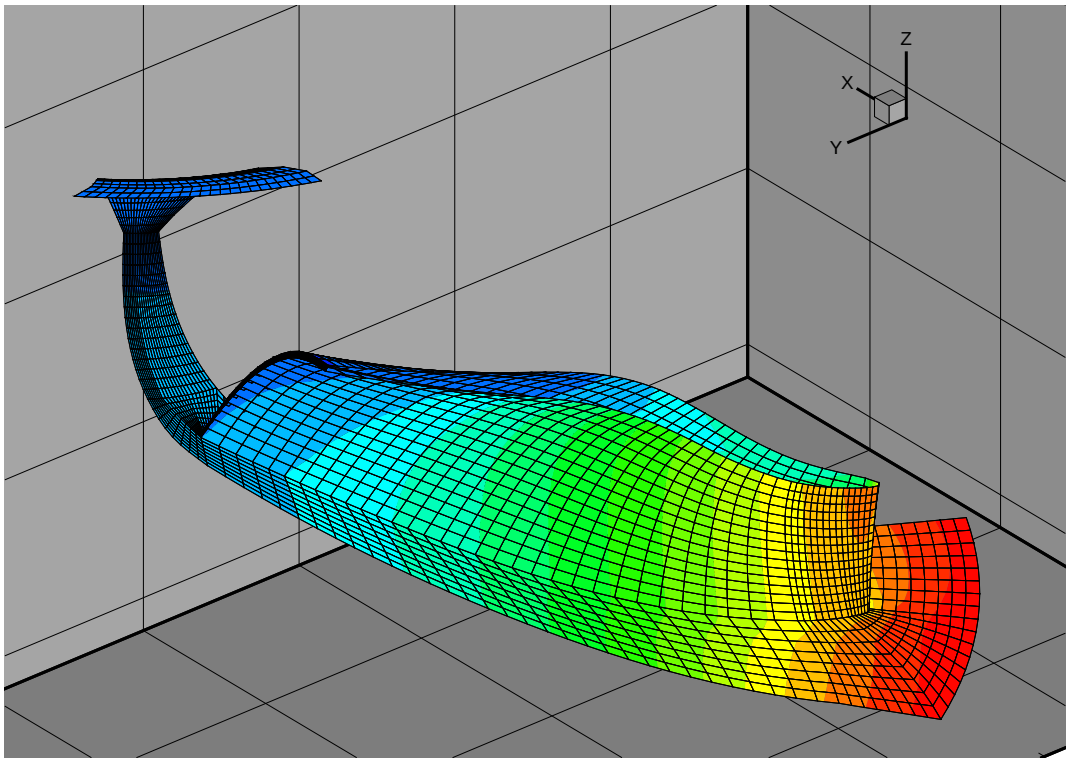

Abbildung 6.8: Level-3-Gitter auf der Nabe und der Schaufel sowie dem Ausströmrand mit Druckisolinien

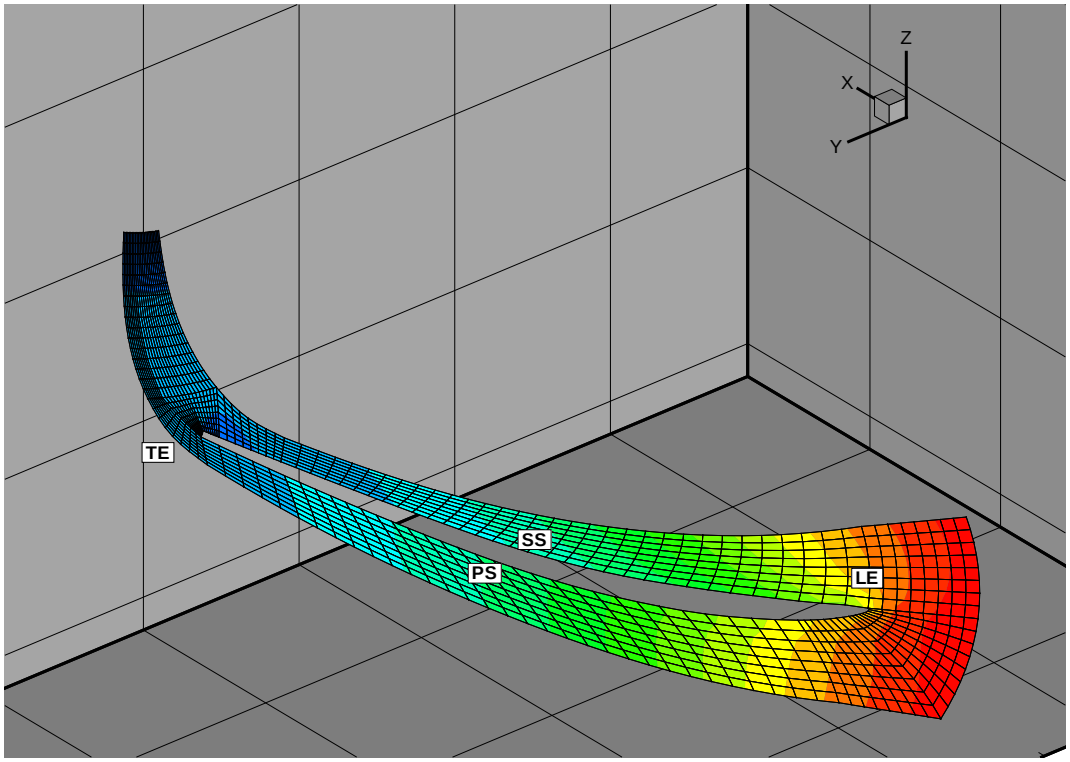

Abbildung 6.9: Gitter auf der Nabe nach 3 Verfeinerungen

6.3.2 Konvertierung von blockstrukturierten Gittern

Als Standardgittergenerator wird gegenwärtig ein blockstrukturierter Gittergenerator der Firma Voith eingesetzt, der eine bessere Kontrolle des y^+-Kriteriums und allgemein der Zellqualität rund um die Schaufel erlaubt. Dazu wird der Grenzschichtbereich um die Schaufel mit einem O-Gitter algebraisch vernetzt. Danach wird der Bereich zwischen den Schaufeln mit einem H-Gitter vernetzt. Diese Gitterart wurde für die FT-Testfälle in Kapitel 7.8 bis 7.10 verwendet. Beispiele für solche Gitter zeigen die Abbildungen 6.10 und 6.11. Man erkennt, daß der Bereich schlechter Zellen am äußeren Rand des O-Gitters liegt, was die Lösungsgenauigkeit auf der Schaufel verbessert.

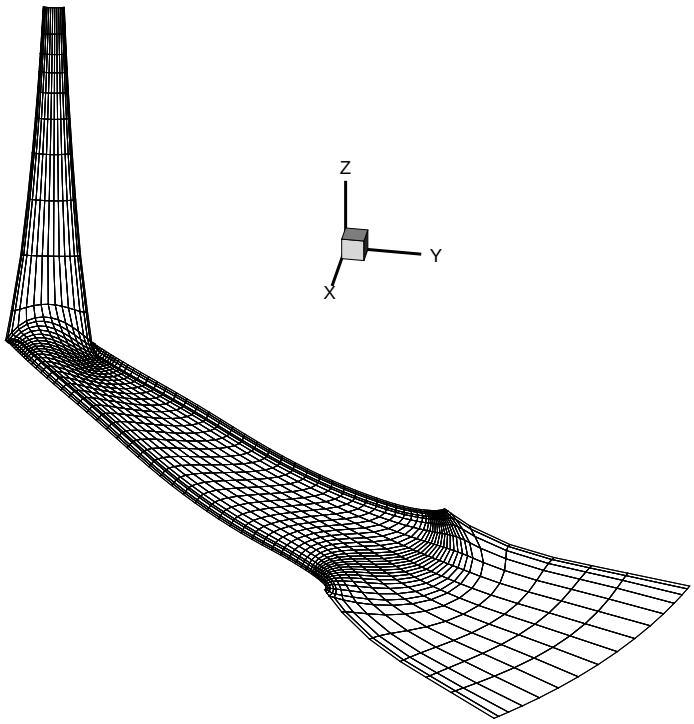

Abbildung 6.10: Ein HO-Gitter auf der Nabe für Testfall FT100

Weitere Verbesserungen der Gitterqualität können im Prinzip durch die Einführung immer weiterer Blöcke erreicht werden, wodurch aber die Gittergenerierung immer komplizierter wird. Derzeit wird ein Gittergenerator mit 13 Blöcken verwendet.

Um die H-Gitter oder die blockstrukturierten HO-Gitter in einem unstrukturierten Verfahren wie dem vorliegenden zu verwenden, müssen die Gitter konvertiert werden. Das von Thum, Reinelt, Szilagyi und Einzinger [59] entwickelte Programm `mb2fem` konvertiert blockstrukturierte Gitter in mehrgitterfähige unstrukturierte Gitter. Dabei können sowohl die Eingabedateien des kommerziellen Strömungslösers TASCflow als auch Dateien in einem offenen Format gelesen werden. Wichtig ist, daß der Konverter bei der Umwandlung eine Mehrgitterhierarchie aufbaut. Die jeweilige Anzahl von Gitterpunkten in den Blöcken ist dabei beliebig. Potenzen von 2 führen a priori zu den besten Gittern, sind aber nicht notwendig, siehe auch Kapitel 7.3.

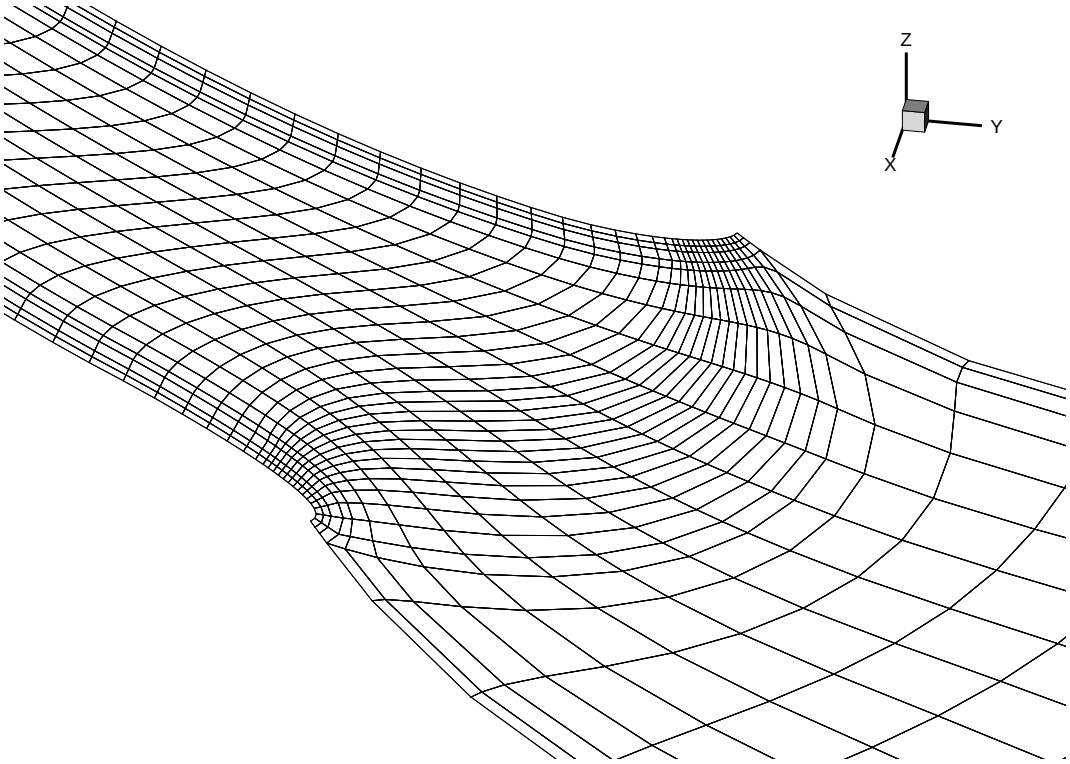

Abbildung 6.11: Detailansicht des HO-Gitters auf der Nabe im Bereich der Vorderkante

6.3.3 Hybridgitter

Hybridgitter sind Gitter mit unterschiedlichen Zelltypen. Als Zelltypen kommen Hexaeder, Pyramiden, Prismen und Tetraeder in Betracht. Diese Zellen sollen als reguläre Zellen bezeichnet werden. Allgemeinere Zellen mit m Begrenzungsflächen, die 3 bis n Begrenzungskanten haben[2], sind für die Finite-Volumen-Diskretisierung grundsätzlich kein Problem. Darüber hinaus ist die Implementierung im Programm schon zum größten Teil erfolgt. Zwei Probleme sind jedoch zu beachten:

1. Gittergeneratoren erzeugen in der Regel nur einfache Zelltypen.
2. Die Verfeinerung einer allgemeinen Zelle ist nicht eindeutig. Bei der Verfeinerung von Hexaedern, Prismen und Tetraedern entstehen wiederum Hexaeder, Prismen und Tetraeder. Bei der Verfeinerung von Pyramiden entstehen Pyramiden und Tetraeder. Entsprechende Aussagen sind für allgemeine Zellen nicht möglich.

Bei Hybridgittern mit regulären Zellen ist zu beachten, daß sich bei Matching Interfaces Hexaeder nicht direkt mit Tetraedern koppeln lassen. Es muß entweder auf eine Hexaederschicht eine Lage mit Pyramiden gelegt werden oder die Körperoberflächen werden von Anfang an mit Prismen vernetzt. Da die Implementierung der noch fehlenden Pyramiden im FVM-Code nur sehr wenig Aufwand bedeutet, bestehen bei zukünftigen Änderungen der Gittergenerierungsstrategie keine Einschränkungen von Seiten des Strömungslösers.

[2]Ein Fußball ist z.B. eine Zelle mit 31 Begrenzungsflächen, d.h. 20 Sechsecke und 11 Fünfecke.

Eine optimale Gittergenerierung mit Hybridgittern sollte somit folgende Bedingungen erfüllen:

- Möglichst wenige Zellen, um Verfeinerungen zuzulassen und damit eine gute Konvergenz des Mehrgitterverfahrens zu gewährleisten.
- Konturkrümmungen sollten schon auf dem Grobgitter so weit aufgelöst sein, daß sich bei der Verfeinerung der Zellen die Form nicht mehr wesentlich ändert.
- Hexaeder sollten vorrangig verwendet werden. Wo dies auf den Berandungen nicht (automatisch) möglich ist, sollten Prismen benutzt werden. Pyramiden sollten nur beim Übergang von Hexaedern zu Tetraedern benutzt werden. Tetraeder sollten nur bei schwierigen Geometrien in den Teilen des Strömungsraums benutzt werden, in denen keine guten Hexaeder erzeugt werden können.

Für die Gittergenerierung bei Laufrädern von Francis-Turbinen sollte folgende Strategie gewählt werden:

1. C-Gitter im Grenzschichtbereich um die Schaufel sollen mit Hexaedern erzeugt werden.
2. Periodische Ränder sowie Ein- und Ausströmrand sollen ebenfalls mit einer Schicht von Hexaedern belegt werden.
3. Im verbleibenden Zwischenraum zwischen den Schaufeln sollen so weit wie möglich weitere Hexaeder eingefügt werden.
4. Der restliche Strömungsraum kann dann mit Pyramiden und Tetraedern unter Verwendung der entsprechenden Verfahren automatisch vernetzt werden.

Kapitel 7

Validierung durch Testrechnungen

Die in diesem Kapitel beschriebenen und einige weitere Testfälle sind in einer Testfalldatenbank gespeichert. In einem Verzeichnis `Referenzfaelle` gibt es eine Datei `allesok.bat`, die alle Testfälle erneut durchrechnet und mit gespeicherten alten Lösungen vergleicht. Dieser Mechanismus wurde während der Code-Entwicklung intensiv genutzt. Keine Änderung wurde in die Quellcodeverwaltung aufgenommen, bevor die Testfälle nicht überprüft wurden.

Alle angegebenen Rechenzeiten beziehen sich auf einen PC mit Pentium II Prozessor und 450 MHz Taktfrequenz. Der Sourcecode wurde mit Microsoft Visual C++ 6.0 übersetzt. Die Compilereinstellungen entsprechen den Defaultwerten für *Release*. Als Betriebssystem wurde Windows NT benutzt.

Bei den Testfällen, für die eine analytische Lösung existiert, werden zur Beurteilung der numerischen Lösungen Fehlernormen nach folgenden Vorschriften berechnet:

$$L_2^p = \sqrt{\frac{1}{N}\sum (p_{num} - p_{exact})^2} \tag{7.1}$$

und

$$L_2^v = \sqrt{\frac{1}{N}\sum (c_{x,num} - c_{x,exact})^2 + (c_{y,num} - c_{y,exact})^2 + (c_{z,num} - c_{z,exact})^2}, \tag{7.2}$$

wobei N die Anzahl der Zellen ist und die Summation über alle Zellen erfolgt. Die Ordnung O des Verfahrens läßt sich dann mit den Fehlerreduktionsfaktoren

$$\Delta L_2^v = \frac{L_{2,leveln}^v}{L_{2,leveln-1}^v}, \tag{7.3}$$

$$\Delta L_2^p = \frac{L_{2,leveln}^p}{L_{2,leveln-1}^p}, \tag{7.4}$$

ermitteln zu

$$O = min(\log_2(\Delta L_2^v), \log_2(\Delta L_2^p)). \tag{7.5}$$

7.1 Ausgebildete 2D Kanalströmung

Es wurde die ausgebildete Strömung in einem Kanal der Höhe $H = 1[m]$ berechnet. Die ebene Hagen-Poiseuille-Strömung ist dann analytisch gegeben durch:

$$u(y) = 6\overline{u}y(1-y) \tag{7.6}$$

und

$$\frac{dp}{dx} = -\frac{12\rho}{Re \cdot H} \tag{7.7}$$

mit

$$Re = \frac{\overline{u}H}{\nu}, \tag{7.8}$$

siehe SCHADE/KUNZ [44]. Für die Parameter $\overline{u} = 1[m/s]$ und $\nu = 0.02[m^2/s]$ ergibt sich $Re = 50$. Das äquidistante Gitter hatte 96*32*1 Zellen auf Level 4. Das Grobgitter hatte 6*2*1 Zellen. Es wurde die L_2-Norm der Fehler der Geschwindigkeiten und des Drucks berechnet. Die Ergebnisse sind in Tabelle 7.1 dargestellt.

Level	L_2^v	ΔL_2^v	L_2^p	ΔL_2^p
0	0.000002631		0.20712100	
1	0.026364500		0.07851430	2.64
2	0.009084270	2.90	0.02075720	3.78
3	0.002426780	3.74	0.00530482	3.91
4	0.000616876	3.93	0.00133624	3.97

Tabelle 7.1: Ergebnisse für Kanal mit L_2-Fehlern

Auf Level 0 mit zwei Zellen senkrecht zur Strömungsrichtung ergibt sich in diesem Fall die Geschwindigkeit gleich der exakten Lösung. Da sowohl ΔL_2^v als auch ΔL_2^p gegen 4 laufen, ist das Verfahren zweiter Ordnung genau.

In Abbildung 7.1 sind die Konvergenzverläufe für das Multigridverfahren (MG) und das Eingitterverfahren (SG) mit Prolongation dargestellt. Die sprunghaften Anstiege in den Verläufen sind auf die Prolongation der konvergierten Grobgitterlösungen auf das nächst feinere Gitter zurückzuführen.

Schließlich sind in Tabelle 7.2 die Rechenzeiten für das Multigridverfahren und das Eingitterverfahren angegeben.

Level	Gitter	T_{MG} [s]	T_{SG} [s]	T_{SG}/T_{MG}
0	6* 2*1	1.08	1.12	1.0
1	12* 4*1	2.32	2.37	1.0
2	24* 8*1	4.70	4.43	0.9
3	48*16*1	10.35	18.24	1.8
4	96*32*1	30.81	182.50	5.9

Tabelle 7.2: Rechenzeiten für Kanalströmung

Aus Tabelle 7.2 erkennt man, daß das Mehrgitterverfahren ab der Gitterebene 3, d.h dem 48*16*1 Gitter, schneller ist als das Eingitterverfahren.

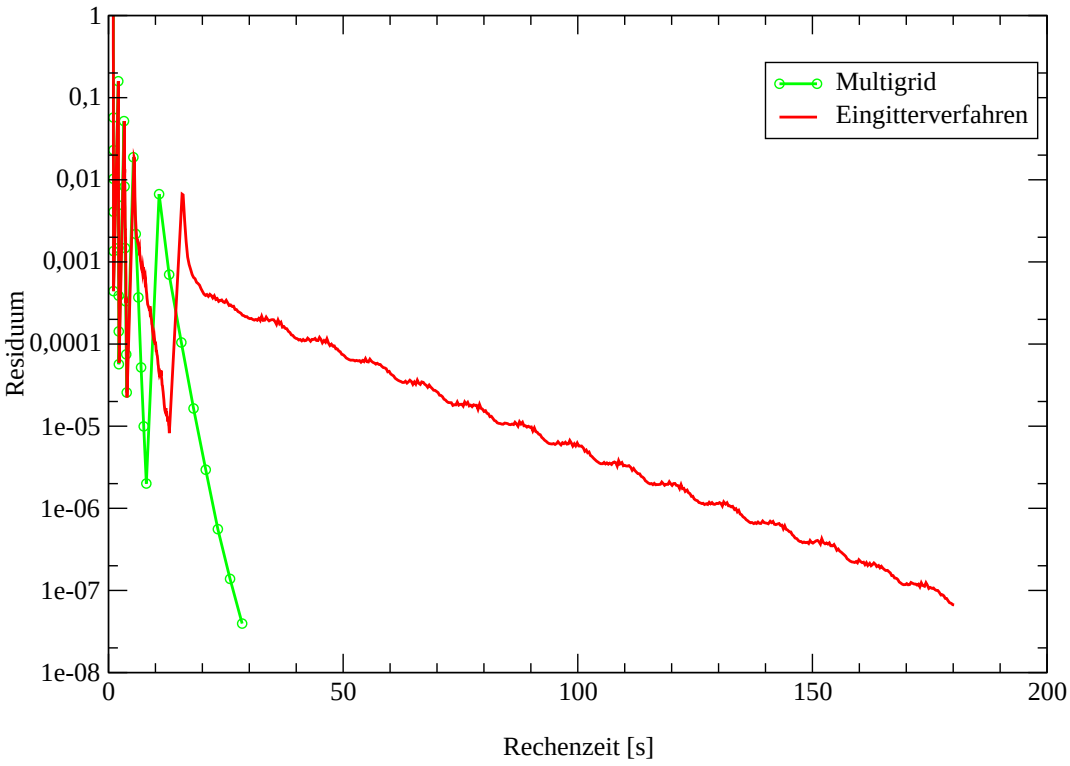

Abbildung 7.1: Konvergenzverlauf für einfache Kanalströmung

7.2 Driven Cavity

Dieser Testfall wird oft bei der Entwicklung von CFD-Codes benutzt, weil die Randbedingungen sehr einfach sind und die Gittergenerierung trivial ist. Allerdings existiert keine exakte Lösung, so daß in der Regel nur mit numerischen Ergebnissen anderer CFD-Codes verglichen wird.

Es wurde mit einem äquidistanten Gitter gerechnet, wobei das Grobgitter 2*2*2 Zellen besitzt. Die Re-Zahl war 100. Verglichen werden die Ergebnisse mit denen von Lilek et al. [31]. Zu beachten ist, daß die Geschwindigkeitsverteilungen von Lilek aus der Veröffentlichung abgelesen wurden, so daß der Vergleich der Genauigkeit nicht sehr gut ist.

Abbildung 7.2 zeigt die u-Geschwindigkeitsverteilung. Die Übereinstimmung mit den Ergebnissen von Lilek ist im Rahmen der Ablesegenauigkeit sehr gut. Die Ergebnisse des FEM-Codes sind etwas schlechter, wobei allerdings zu beachten ist, daß der FEM-Code bei gleichem Gitter etwa die 2.5-fache Anzahl von Unbekannten hat. Diese Aussagen bestätigen sich auch für die Geschwindigkeitsverteilung w(x), siehe Abbildung 7.3.

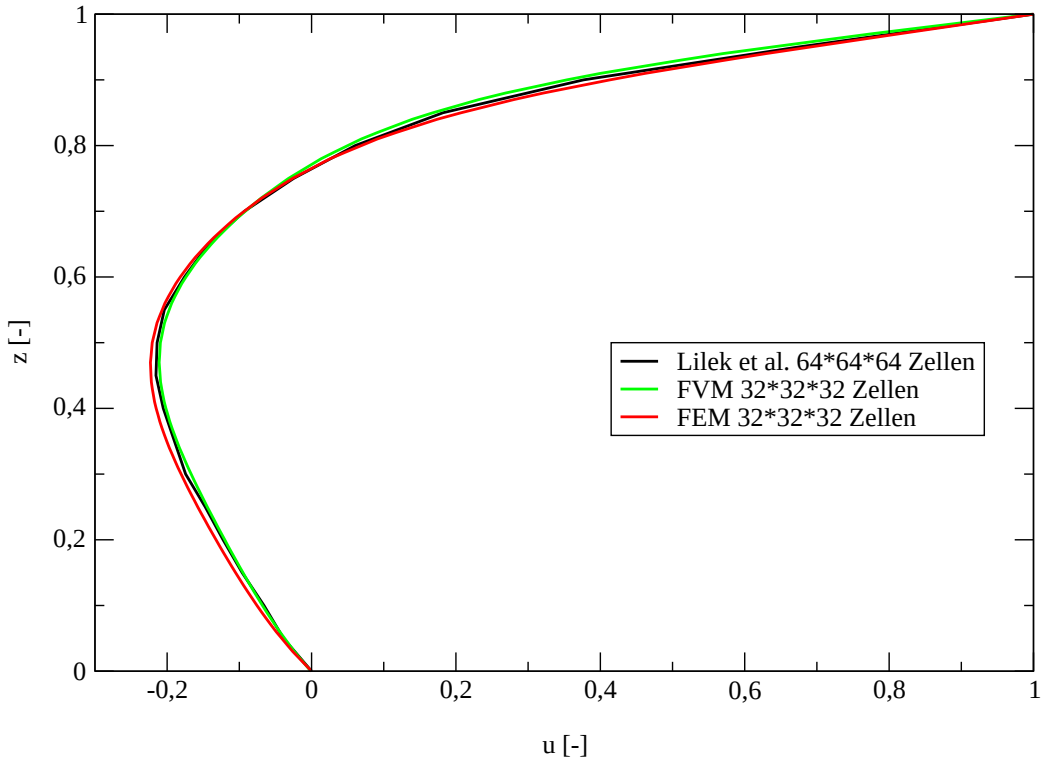

Abbildung 7.2: Verteilung der Geschwindigkeitskomponente u(z) bei x=0.5 und y=0.5

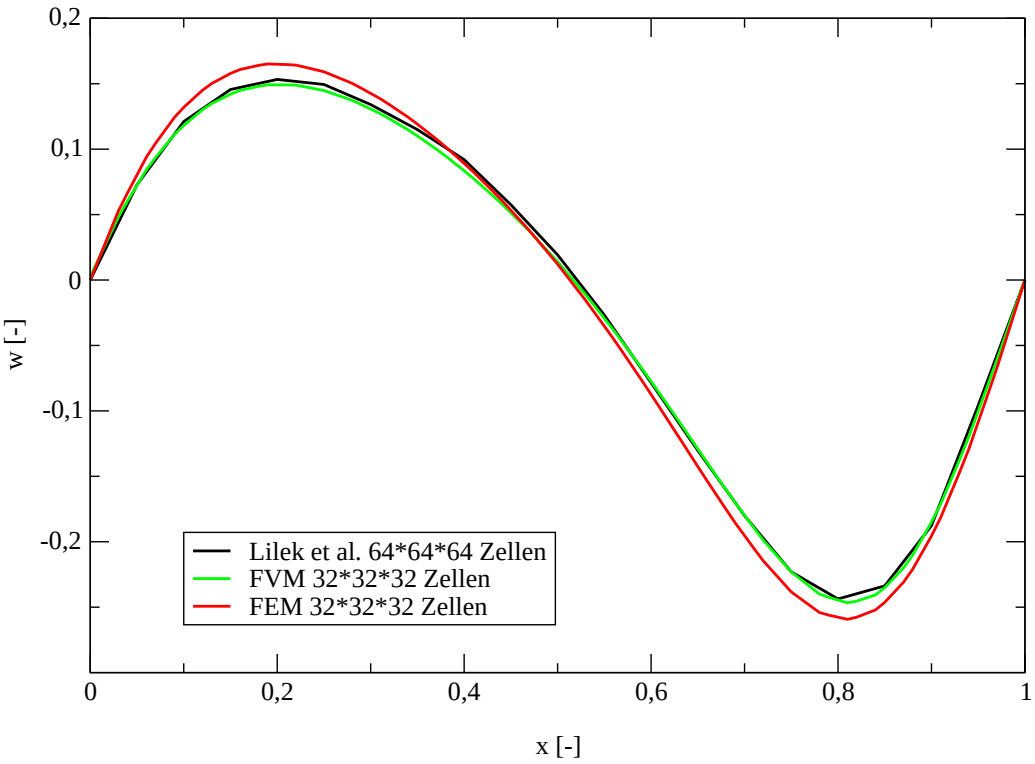

Abbildung 7.3: Verteilung der Geschwindigkeitskomponente w(x) bei y=0.5 und z=0.5

Der Konvergenzverlauf für das Mehrgitterverfahren und das Eingitterverfahren sind in der Abbildung 7.4 dargestellt. Dieser Verlauf ist ebenso wie der von Kapitel 7.1 als typisch für Mehrgitterverfahren zu bezeichnen:

- Das Mehrgitterverfahren hat einen linearen Konvergenzverlauf. Hingegen hat das Eingitterverfahren einen abflachenden Konvergenzverlauf, was darauf zurückzuführen ist, daß am Anfang die hochfrequenten Fehler sehr gut gedämpft werden, während die langwelligen Fehler nur sehr langsam verringert werden. Daraus folgt, daß das Mehrgitterverfahren um so besser wird, je genauer das Konvergenzkriterium gesetzt wird.
- Das Mehrgitterverfahren kann pro V-Zyklus das Residuum um eine Größenordnung verkleinern. Dabei ist allerdings zu berücksichtigen, daß das Mehrgitterverfahren im konkreten Fall 6 Feingitterglättungsschritte pro V-Zyklus macht.
- Je feiner das Gitter, desto besser funktioniert das Mehrgitterverfahren. In diesem Fall ergeben sich erst auf dem 32*32*32 Gitter deutliche Einsparungen gegenüber dem Eingitterverfahren.
- Bei der Re=100 und 32*32*32 Zellen sowie einem absolutem Konvergenzkriterium von 10^{-6}) ergibt sich ein Rechenzeitgewinn von Faktor 4.5.

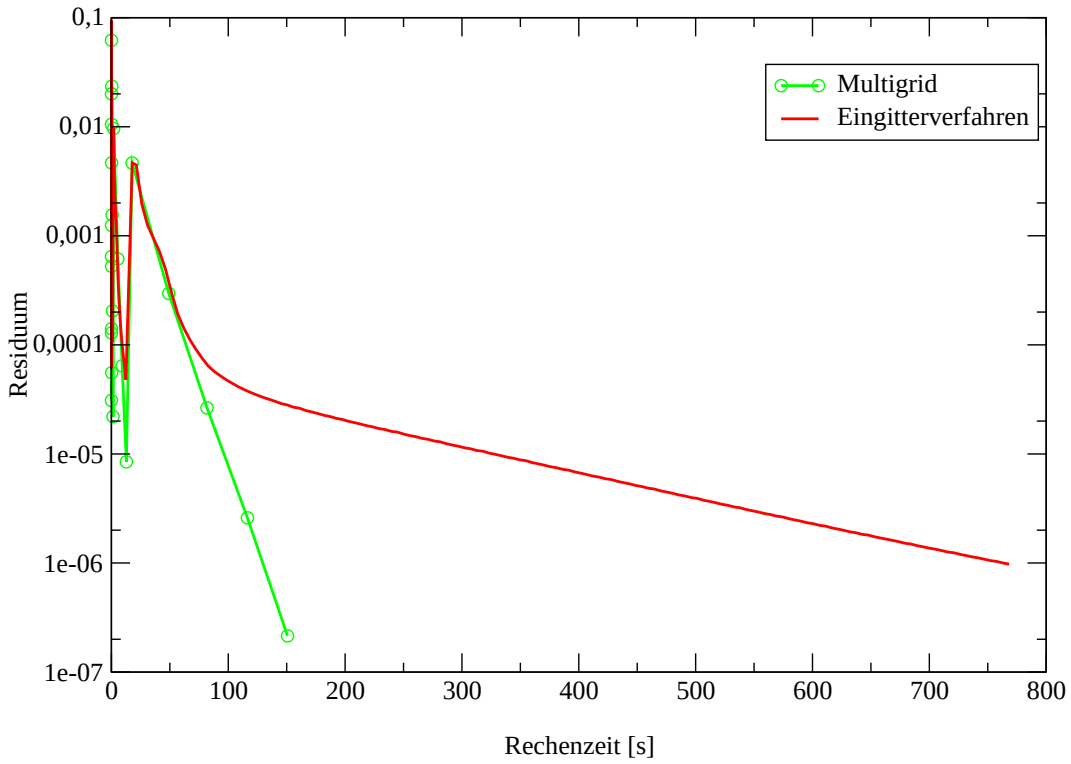

Abbildung 7.4: Konvergenzverlauf für den Driven Cavity Testfall mit 32768 Zellen

7.3 Driven Cavity mit Symmetrieränder

Setzt man bei der Driven Cavity u(z=1)=1 und bei y=0 sowie y=1 eine Symmetrierandbedingung, so erhält man eine zweidimensionale Strömung. Dieser Testfall wurde benutzt, um das Mehrgitterverfahren für Gitter zu testen, die nicht durch gleichmäßige Verfeinerung entstanden sind.

In der Symmetrierichtung wird auch auf dem Feingitter nur eine Zelle benutzt. In Tabelle 7.3 ist aufgeführt, auf welchen Gittern gerechnet wurde. Insgesamt wurden 5 Gitterebenen verwendet, Level 0 bis 4.

Gitter	Zellen	CPU-Zeit [s]	CPU-Zeit pro Zelle [s]
32*1*32	1024	6.05	0.00591
30*1*31	930	5.72	0.00615
33*1*34	1122	7.08	0.00631
40*1*40	1600	9.95	0.00622

Tabelle 7.3: Rechenzeiten für die Driven Cavity mit Symmetrierandbedingungen und Mehrgitterverfahren

Der Tabelle ist zu entnehmen, daß das Gitter mit 32*1*32 Zellen zu den kürzesten Rechenzeiten pro Zelle führt. Aber auch die anderen Gitter sind nicht wesentlich schlechter. Beim schlechtesten Gitter (33*1*34) ist die Rechenzeit pro Zelle nur um 7% schlechter. 33 Zellen sind auch der härteste Testfall für das Mehrgitterverfahren. Wegen

$$2^5 + 1 = 33 \tag{7.9}$$

entstehen bei jedem Vergröberungsschritt eine ungerade Zahl von Zellen. Dies wiederum führt beim gegenwärtigen Stand des Konverters `mb2fem` dazu, daß das Grobgitter eine sehr kleine Restzelle des Feingitters enthält. Die Abbildungen 7.5 und 7.6 zeigen den beschriebenen Effekt.

In Abbildung 7.7 ist die Geschwindigkeitsverteilung $w(x)$ bei $y = 1/2$ und $z = 1/2$ aufgetragen. Die Lösungen auf Level 5 und 6 unterscheiden sich nur noch marginal, so daß die Level-6-Lösung als *exakte* Lösung betrachtet werden kann. Diese Lösung wird in Kapitel 7.5 als Referenzlösung verwendet.

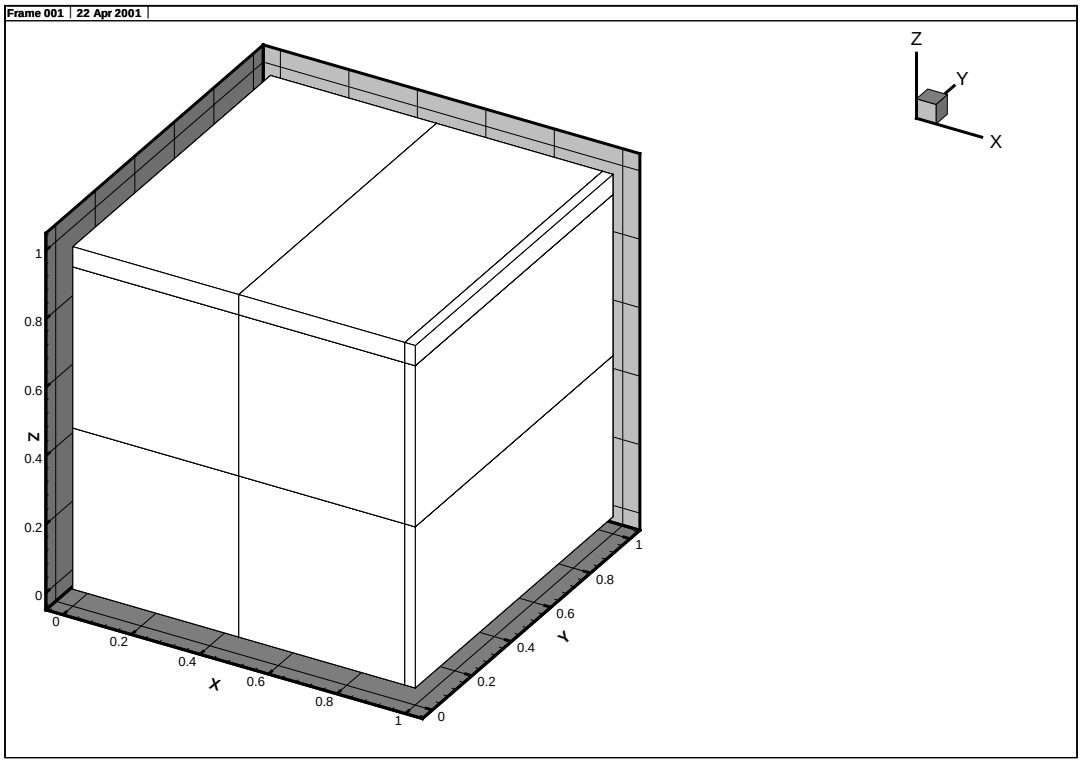

Abbildung 7.5: Gitter für die quasi 2D Driven Cavity Strömung, Level 0

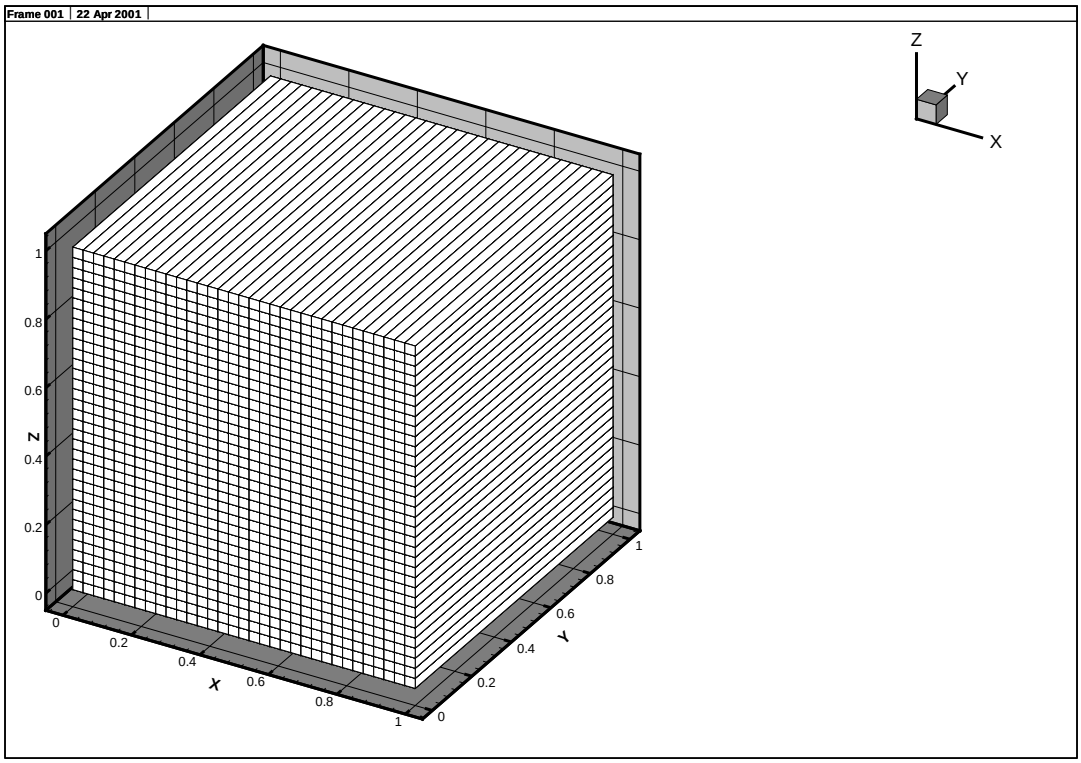

Abbildung 7.6: Gitter für die quasi 2D Driven Cavity Strömung, Level 4, 33*1*34 Zellen

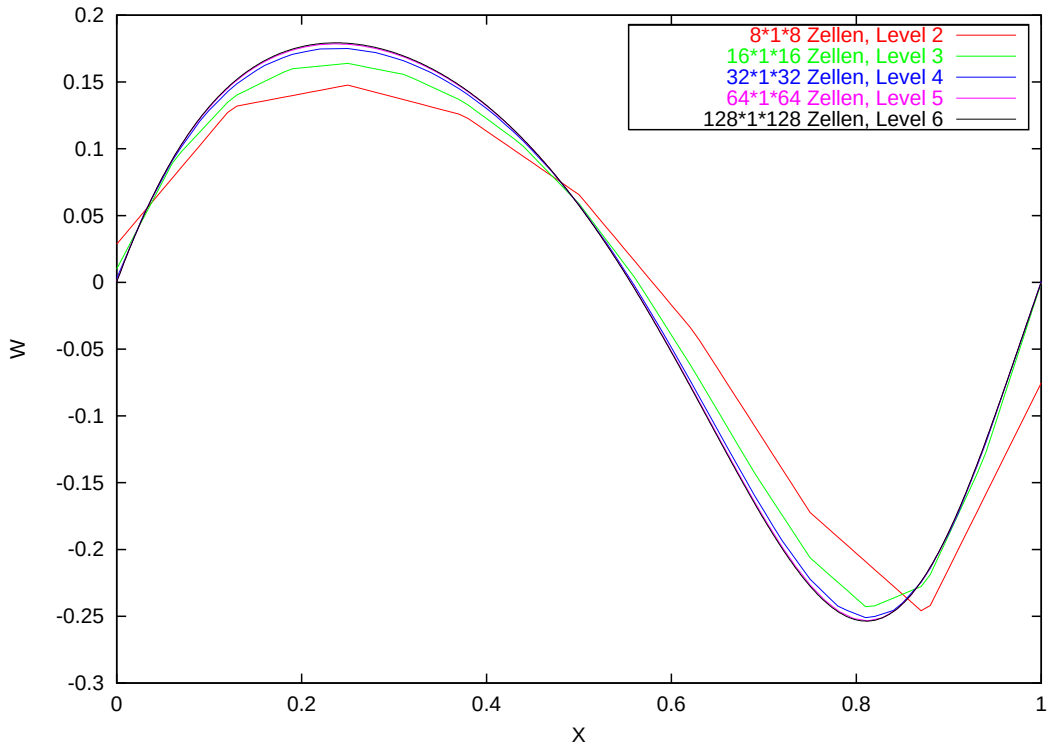

Abbildung 7.7: Geschwindigkeitsverteilung $w(x)$ der Driven Cavity Strömung bei $y = 1/2$ und $z = 1/2$.

7.4 Driven Cavity mit schrägen Wänden

In Kapitel 8.8 beschreibt PERIĆ [12], wie ein Korrekturterm für nicht-orthogonale Gitter die Konvergenzeigenschaften der Druckkorrekturgleichung erheblich verbessern kann. Dieser Term ist auch in diesem Code implementiert. Der Korrekturterm kann über den Parameter `gNpCorMax` in der Datei `fvmControl.dat` ein- und ausgeschaltet werden. Der entsprechende Testfall, siehe Abbildung 8.11 von PERIĆ [12], wurde nachgerechnet. Dabei wurde ein äquidistantes Gitter mit 32*32*1 Zellen benutzt. Abbildung 7.8 zeigt, daß mit zwei Korrekturschritten, d.h. mit einer Anwendung des Korrekturterms, ein viel größerer Bereich des Unterrelaxationsfaktors α_p zu Konvergenz führt, und auch eine erhebliche Einsparung von Iterationen erreicht werden kann.

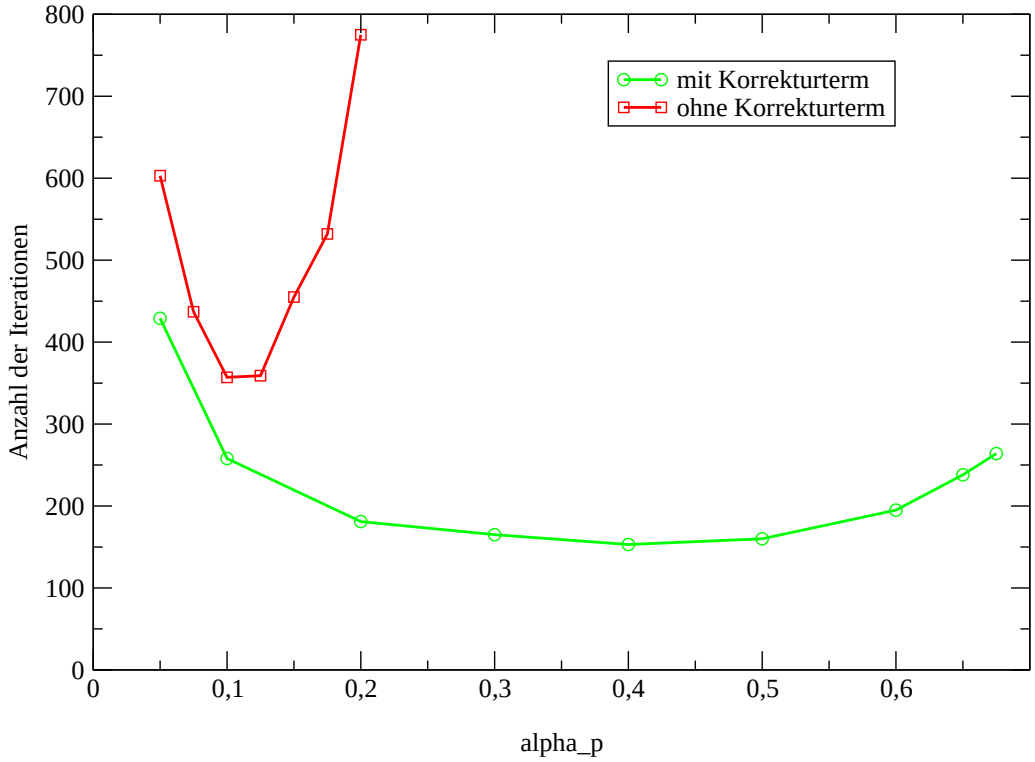

Abbildung 7.8: Anzahl der Iterationen in Abhängigkeit von α_p

Das von Perić angegebene Verhalten wird sehr gut bestätigt. Gewisse Unterschiede zu Abbildung 8.11 von Perić sind unvermeidbar, da der genaue Verlauf der Kurven mindestens von folgenden Parametern abhängt:

- verwendetes Gitter
- Abbruchkriterien für Konvergenz
- Linearer Löser (SIP bei Perić, konjugierte Gradienten hier)

7.5 Driven Cavity mit Hybridgitter

Dieser Testfall entspricht dem von Kapitel 7.3, d.h. die Strömung ist zweidimensional. Im Unterschied zu Kapitel 7.3 wird hier jedoch ein Hybridgitter benutzt, daß aus Tetraedern, Prismen und Hexaedern besteht. Man beachte, daß für diese einfache Geometrie das Hybridgitter keinen Vorteil bietet. Es soll lediglich das Verhalten des Codes getestet werden. Abbildung 7.9 zeigt das Grobgitter.

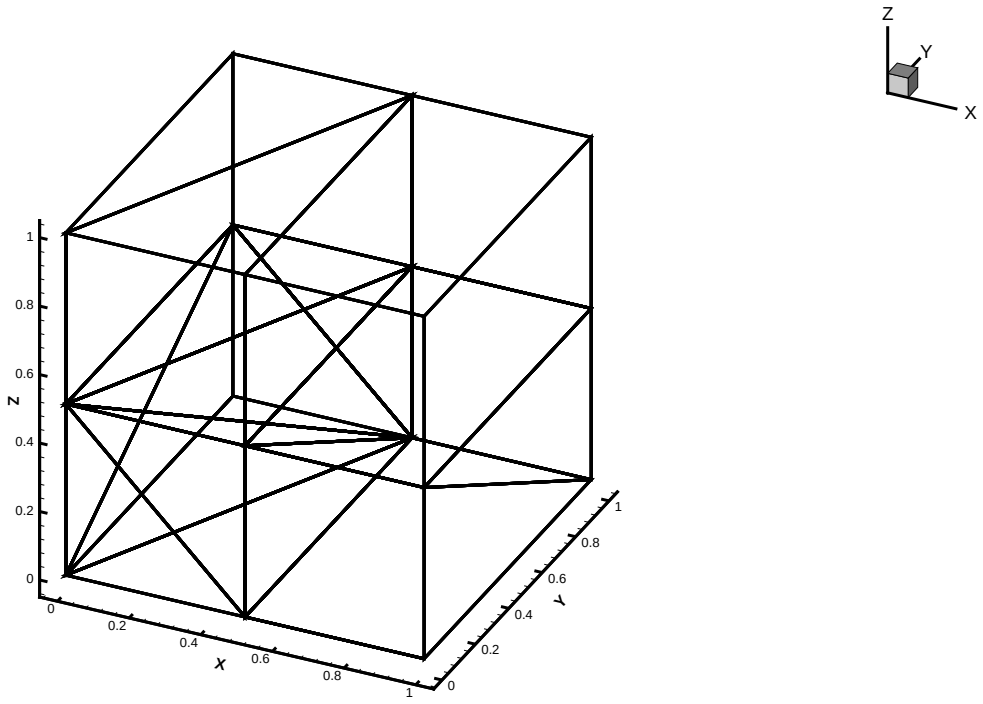

Abbildung 7.9: Hybridgitter Level 0

Das Grobgitter besteht aus einem Hexaeder in der rechten oberen Ecke, aus je zwei Prismen in der linken oberen und der rechten unteren Ecke sowie aus 6 Tetraedern in der linken unteren Ecke. Das Grobgitter besteht also aus 11 Zellen. Es wurde erzeugt, indem manuell eine entsprechende Grobgitterdatei erstellt wurde. Abbildung 7.10 zeigt, daß die Hybridgitterlösung gegen die Lösung des Hexaedergitters konvergiert.

Abbildung 7.11 zeigt den Konvergenzverlauf über der Rechenzeit mit und ohne Mehrgitterverfahren. Das Level 4 Gitter hat dabei 45056 Zellen.

Ebenfalls in Abbildung 7.11 aufgeführt ist der Konvergenzverlauf, wenn man den Korrekturterm nach Gleichung (3.32) nicht benutzt. Die Lösung ist dann allerdings falsch, auch bei Gitterverfeinerung. Man erkennt aber, daß der notwendige Korrekturterm nach Gleichung (3.32) die Konvergenzrate deutlich mindert. Das liegt daran, daß bei diesem Gitter für 3 Interfaces das Skalarprodukt $\underline{n} \cdot \underline{i_\xi} < 0.6$ ist, die Winkel zwischen Flächennormalenvektor und dem Verbindungsvektor von $\underline{x_P}$ nach $\underline{x_E}$ also größer als 53° sind. Zur Verdeutlichung sind in Abbildung 7.12 für den zweidimensionalen Fall drei Dreiecke dargestellt.

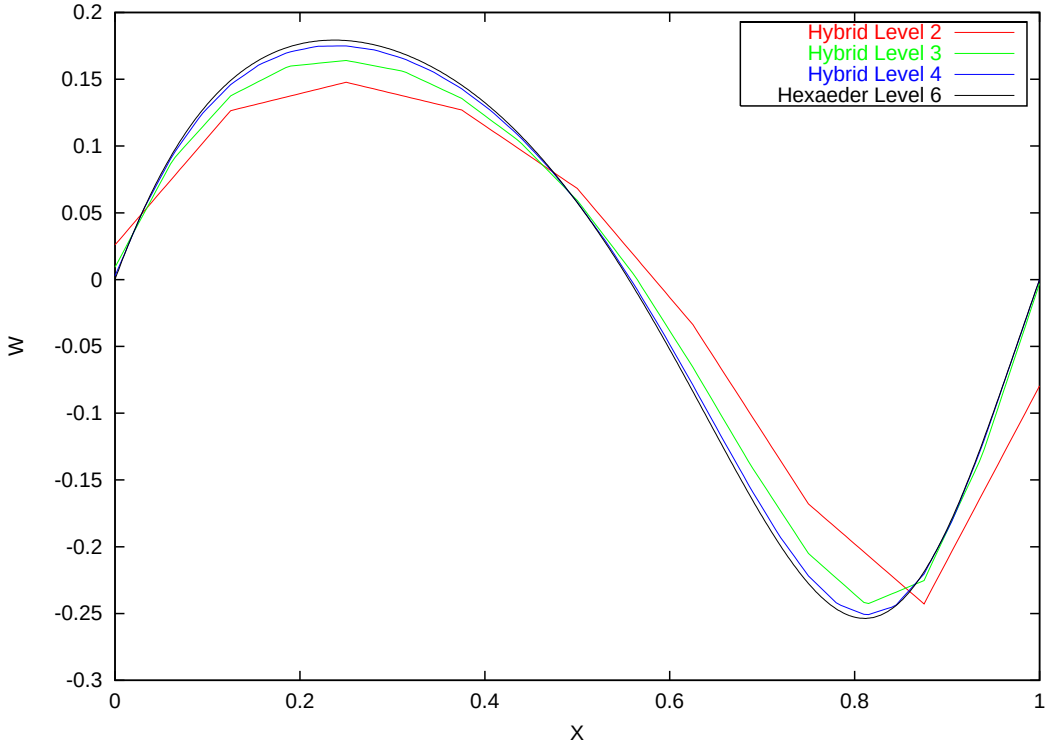

Abbildung 7.10: Geschwindigkeitsverteilung für Hybridgitter

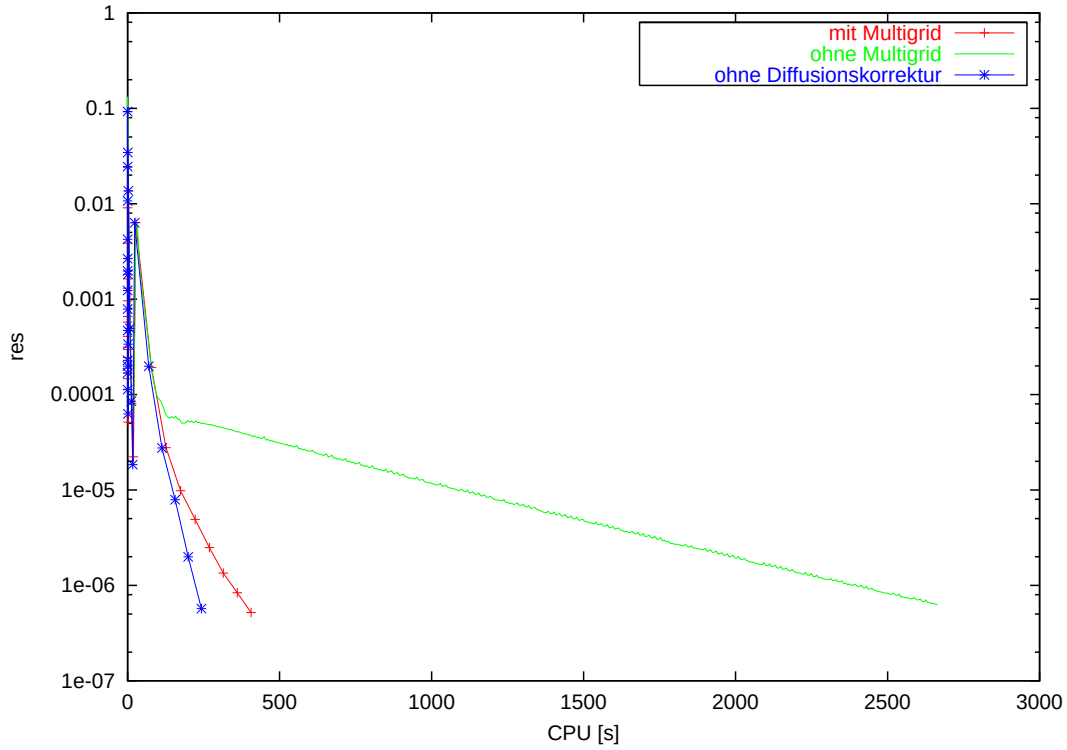

Abbildung 7.11: Konvergenzverlauf für Hybridgitter mit und ohne Mehrgitterverfahren

Beim Interface zwischen den Zellen P und A liegt $\underline{x_e}$ bei $\underline{x_{e'}}$, aber die Vektoren $\underline{n}$ und $\underline{i_\xi}$ zeigen in unterschiedliche Richtungen. Genau umgekehrt ist es beim Interface zwischen den Zellen P und B. Für hohe Genauigkeit und gute Konvergenz sollten die Dreiecke also möglichst gleichschenklig sein. Insgesamt sollte die Zahl der Tetraeder minimiert werden. Diese Forderung kann erfüllt werden, wenn die Tetraeder lediglich als *Kitt* zwischen strukturierten Gitterbereichen eingesetzt werden.

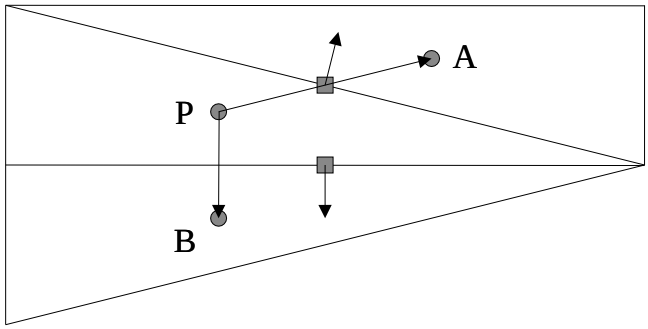

Abbildung 7.12: Geometrische Verhältnisse bei gestreckten Dreiecken

7.6 Testfall mit logarithmisch-spiraligen Schaufeln

Von BUSEMANN [6] wurde bereits 1928 die exakte Lösung der Potentialströmung radialer Kreiselpumpen mit logarithmisch-spiraligen Schaufeln angegeben. Busemanns Lösung führt zu Strömungen, die nicht schaufelkongruent sind und damit die Berechnung von Kennlinien erlauben.

Es soll hier nur die geometrische Transformation, also die *Busemann-Geometrie*, übernommen werden. Für die Strömung wird eine Eulerlösung hergeleitet. Die geometrische Transformation von μ, η Problemkoordinaten in r, φ Polarkoordinaten lautet mit $a = \cot \beta_S$:

$$r = e^{-\mu/a}, \tag{7.10}$$

$$\varphi = \eta + \mu. \tag{7.11}$$

In diesem Kapitel gilt ohne Einschränkung der Allgemeingültigkeit $r_{ref} = 1m$, $c_{ref} = 1\frac{m}{s}$ und $\rho_{ref} = 1\frac{kg}{m^3}$. Also gilt insbesondere $r = \frac{r'}{r_{ref}}$, wobei r' der dimensionsbehaftete Radius ist. Außerdem wird in dieser Arbeit der Schaufelwinkel $\beta_S = 60°$ und die Schaufelanzahl $z = 6$ benutzt. In Abbildung 7.13 sind die resultierenden Schaufelkonturen für $\eta = 0$ und $\eta = \pi/3$ dargestellt.

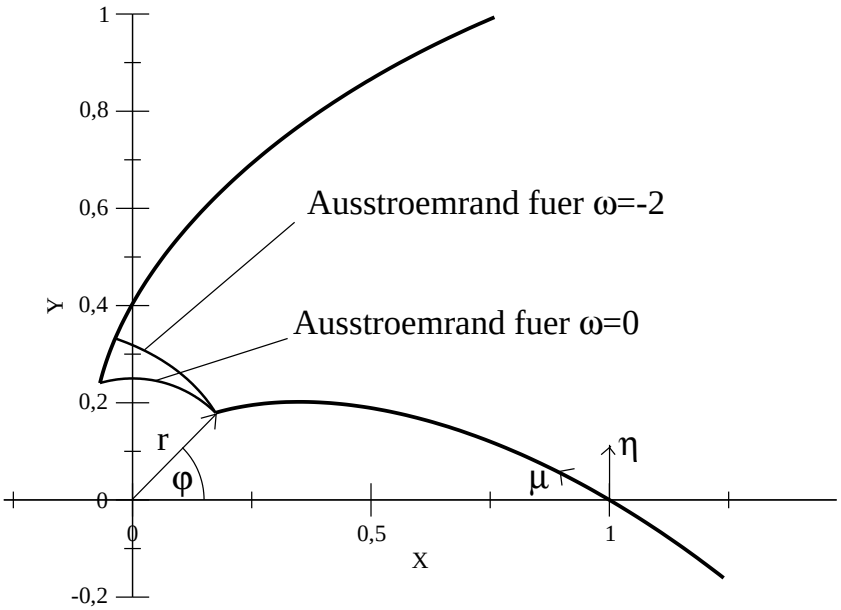

Abbildung 7.13: Koordinatensysteme für Busemann-Geometrie

7.6.1 Exakte Lösung

Für die Geschwindigkeiten wird folgender Ansatz gemacht:

$$c_r = w_r = \frac{q}{r}, \tag{7.12}$$

$$w_\varphi = -a\frac{q}{r}. \tag{7.13}$$

Daraus folgt für die Absolutgeschwindigkeitskomponente

$$c_\varphi = \omega r - a\frac{q}{r}. \tag{7.14}$$

Der Ansatz erfüllt die Kontinuitätsgleichung:

$$\frac{\partial c_r}{\partial r} + \frac{c_r}{r} + \frac{1}{r}\frac{\partial c_\varphi}{\partial \varphi} = \frac{-q}{r^2} + \frac{q}{r^2} + 0 = 0 \quad qed. \tag{7.15}$$

Für den Druck wird folgender Ansatz gemacht:

$$p(r,\varphi) = -2\rho\omega q\,(\varphi + a\cdot \ln r) + \frac{\rho}{2}\omega^2 r^2 - \frac{\rho q^2(1+a^2)}{2r^2} + p_0. \tag{7.16}$$

Der Ansatz wurde so gewählt, daß die Impulsgleichungen erfüllt werden. Für die r-Impulsgleichung gilt:

$$\begin{aligned}
&\rho c_r\frac{\partial c_r}{\partial r} + \rho\frac{c_\varphi}{r}\frac{\partial c_r}{\partial \varphi} - \rho\frac{c_\varphi^2}{r} + \frac{\partial p}{\partial r} = \\
&\frac{-\rho q^2}{r^3} + 0 - \frac{\rho}{r}\left(\omega r - a\frac{q}{r}\right)^2 + \left[\frac{-2a\rho\omega q}{r} + \rho\omega^2 r + \frac{\rho q^2(1+a^2)}{r^3}\right] = \\
&\frac{-\rho q^2}{r^3} - \rho\omega^2 r + \frac{2a\rho\omega q}{r} - \frac{\rho q^2 a^2}{r^3} - \frac{2a\rho\omega q}{r} + \rho\omega^2 r + \frac{\rho q^2(1+a^2)}{r^3} = \\
&\frac{\rho q^2}{r^3}\left(-1 - a^2 + 1 + a^2\right) = 0 \quad qed.
\end{aligned} \tag{7.17}$$

Für die φ-Impulsgleichung gilt:

$$\begin{aligned}
&\rho c_r\frac{\partial c_\varphi}{\partial r} + \rho\frac{c_\varphi}{r}\frac{\partial c_\varphi}{\partial \varphi} + \rho\frac{c_r c_\varphi}{r} + \frac{1}{r}\frac{\partial p}{\partial \varphi} = \\
&\frac{\rho q}{r}\left(\omega + a\frac{q}{r^2}\right) + 0 + \frac{\rho q}{r^2}\left(\omega r - a\frac{q}{r}\right) - \frac{2\rho\omega q}{r} = \\
&\frac{2\rho\omega q}{r} - \frac{2\rho\omega q}{r} = 0 \quad qed.
\end{aligned} \tag{7.18}$$

Damit ist gezeigt, daß der Ansatz (7.12), (7.14) und (7.16) die Euler Gleichungen und die Kontinuitätsgleichung identisch erfüllt. Der Ansatz beschreibt eine bezüglich der Geschwindigkeiten schaufelkongruente Strömung. Allerdings ist der reduzierte Gesamtdruck p_g^* nur entlang einer Stromlinie konstant, aber quer zur Strömung geschichtet, was wie folgt gezeigt werden kann.

Durch elementare Umformungen erhält man

$$|\underline{w}| = w = \frac{q}{r}\sqrt{1+a^2} = \frac{q}{r}\frac{1}{\sin\beta_S}. \tag{7.19}$$

Mit (7.10) und (7.11) kann (7.16) umgeformt werden zu

$$p(r,\eta) = -2\rho\omega q\eta + \frac{\rho}{2}\omega^2 r^2 - \frac{q^2\rho(1+a^2)}{2r^2} + p_0. \tag{7.20}$$

Damit ergibt sich der reduzierte Gesamtdruck mit $p^* = p - \frac{\rho}{2}r^2\omega^2$ zu

$$p_g^* = p^* + \frac{\rho}{2}\left(w_r^2 + w_\varphi^2\right) = p_0 - 2\rho\omega q\eta = f_1(\eta). \tag{7.21}$$

Für den Totaldruck p_t erhält man

$$\begin{aligned} p_t &= p_g^* + \rho\omega(rc_\varphi) \\ &= p_0 - 2\rho\omega q\eta + \rho\omega r\left(\omega r - a\frac{q}{r}\right) \\ &= p_0 - a\rho\omega q - 2\rho\omega q\eta + \rho\omega^2 r^2 = f_2(r,\eta) \end{aligned} \tag{7.22}$$

In Polarkoordinaten ergibt sich

$$p_t(r,\varphi) = p_0 - a\rho\omega q - 2\rho\omega q\left(\varphi - a\cdot\ln r\right) + \rho\omega^2 r^2 \tag{7.23}$$

Die Reynolds-Zahl soll für diesen Testfall gebildet werden als

$$Re = \frac{|w_r| * r}{\nu} = \frac{|q|}{\nu}. \tag{7.24}$$

Alle Rechnungen wurden mit $q = \frac{-\sqrt{3}}{2}$ durchgeführt, um Vergleichbarkeit mit den Ergebnissen von MÜLLER [34] zu gewährleisten.

7.6.2 Leitgitter mit logarithmisch-spiraligen Schaufeln

Zunächst wurde der Fall ohne Rotation ($\omega = 0$) untersucht. Bei einem Teilungswinkel von 30° hat das Grobgitter 8*2*1=16 Zellen. Das Level 4 Feingitter hat dann 128*32*1=4096 Zellen. Es wurden die L_2-Fehlernormen von Geschwindigkeit und Druck berechnet. Die Abhängigkeit der Fehler von der Verfeinerung und der Reynolds-Zahl zeigt Abbildung 7.14. Dabei wurde mit CDS gerechnet.

Für den Druckfehler ist die Abhängigkeit von der Reynolds-Zahl eindeutig. Je höher die Reynoldszahl, desto kleiner der Fehler im Vergleich zur exakten Lösung, wobei der Unterschied zwischen $Re = 500$ und $Re = 8000$ sehr klein ist.

Bei den Geschwindigkeiten wirkt sich offensichtlich aus, daß auf groben Gittern bei Verwendung von CDS Schwingungen auftreten, die von Reibung gedämpft werden. Während auf den groben Gittern von Level 0 bis 3 das Geschwindigkeitsfeld für $Re = 125$ am besten ist, ist auf Level 4 $Re = 500$ am besten.

Zu beachten ist noch, daß mit steigender Reynolds-Zahl die Effizienz des Mehrgitterverfahrens sinkt, siehe Abbildung 7.15.

Die nötige Reibung kann auch numerisch erzeugt werden, indem die CDS-Diskretisierung der konvektiven Terme teilweise durch eine UDS-Diskretisierung ersetzt wird. In Abbildung 7.16 sind die Fehler in Abhängigkeit von Blending-Faktor γ aufgetragen, wobei ohne Reibungsterme gerechnet wurde.

Für den Druckfehler ist $\gamma = 1.0$ (reines CDS) optimal, während für die Geschwindigkeiten $\gamma = 0.8$ am besten ist. Berücksichtigt man allerdings die Rechenzeiten, siehe Abbildung 7.17, so ist $\gamma = 0.8$ eindeutig zu bevorzugen.

Das Verfahren kann also bei Verwendung von Euler-Wänden Euler-Strömungen berechnen, wobei allerdings etwas Reibung in Form von echter Reibung und/oder numerischer Reibung nötig ist, um die Lösung zu glätten und die Konvergenz zu beschleunigen.

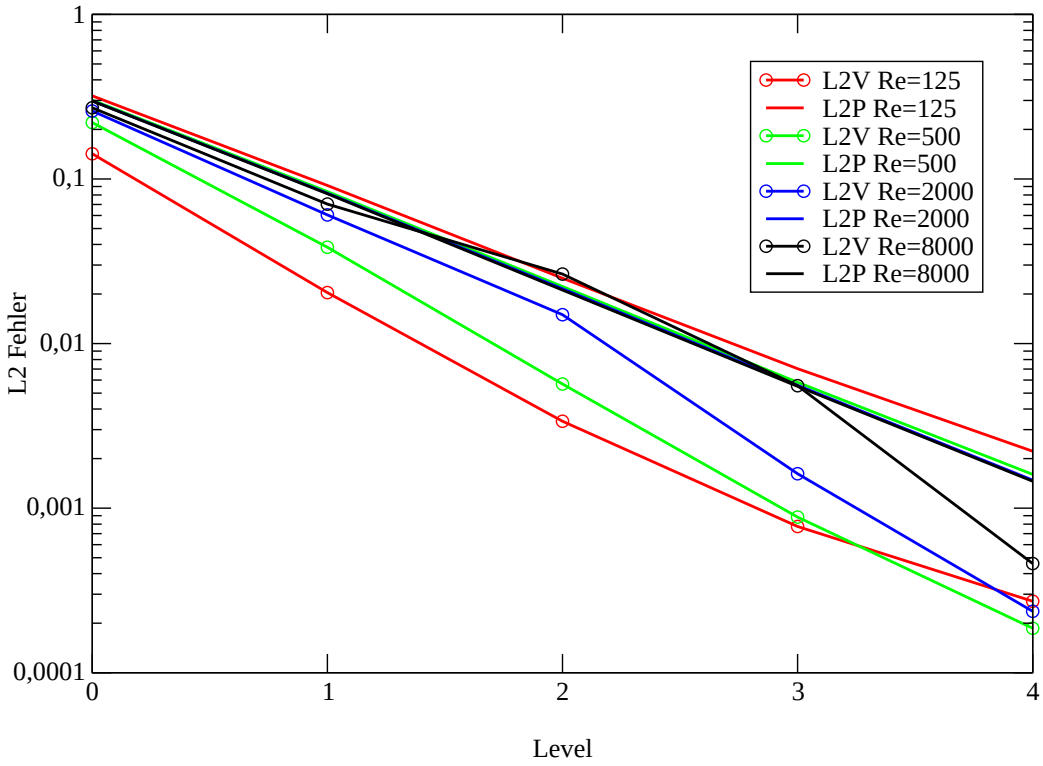

Abbildung 7.14: Fehler in Abhängigkeit von der Viskosität für Leitgitter mit logarithmisch-spiraligen Schaufeln

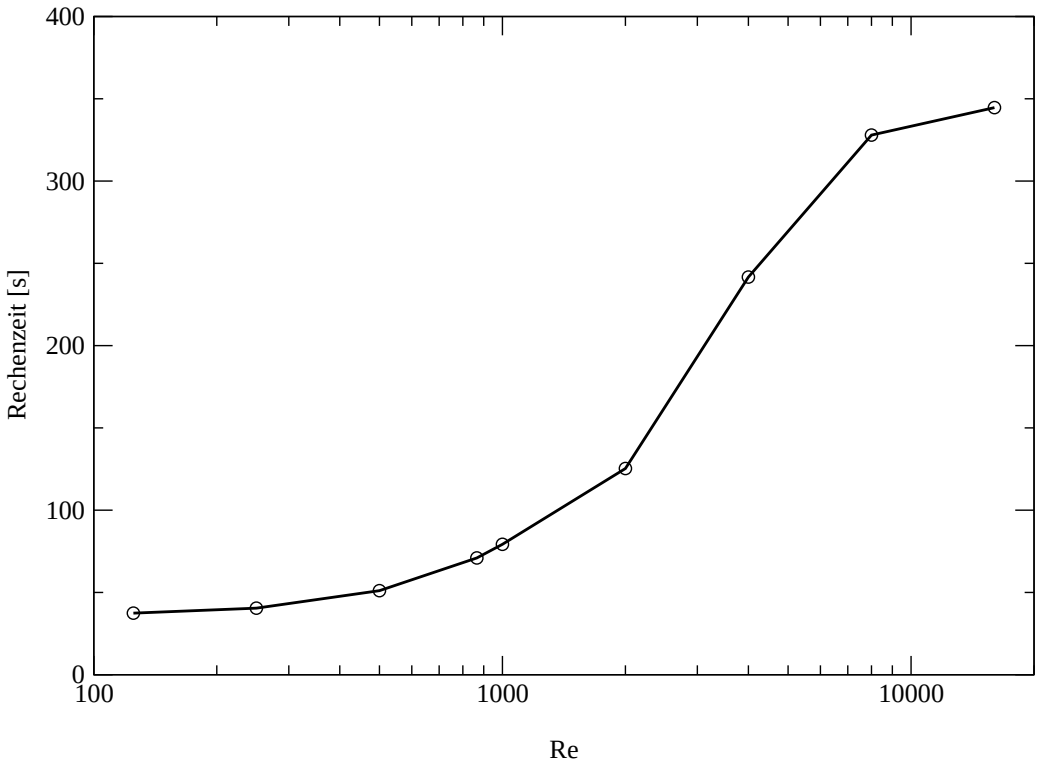

Abbildung 7.15: Rechenzeiten in Abhängigkeit von der Reynolds-Zahl für Leitgitter mit logarithmisch-spiraligen Schaufeln

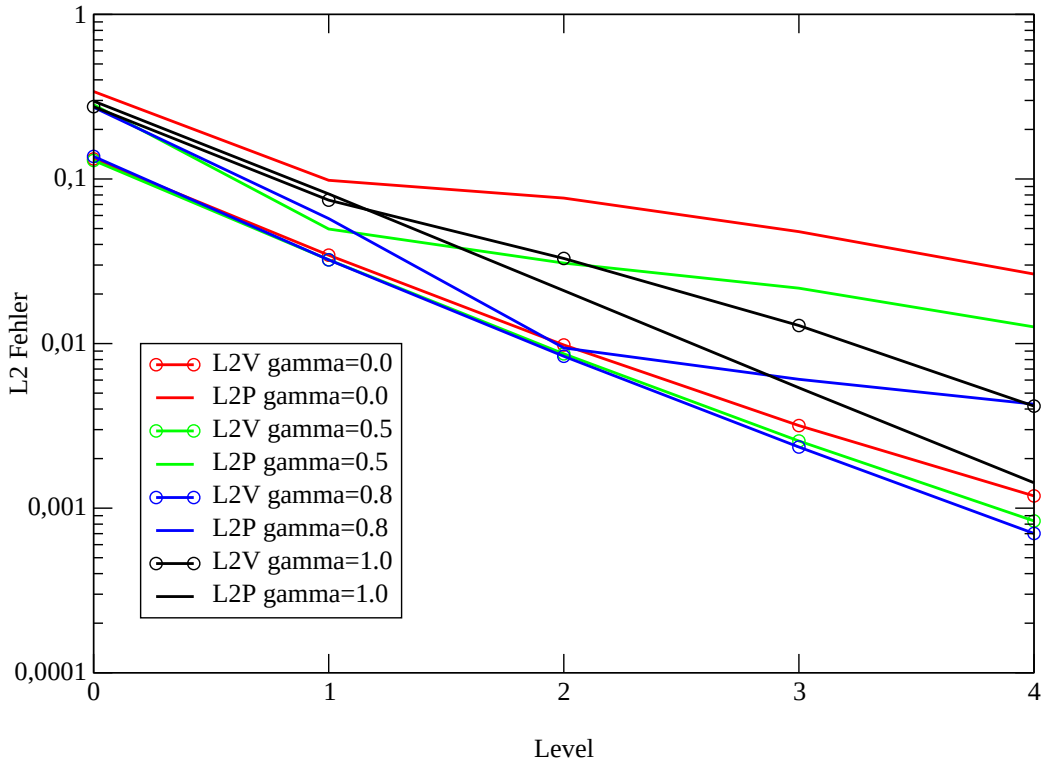

Abbildung 7.16: Fehler in Abhängigkeit vom Blending-Faktor γ für Leitgitter mit logarithmisch-spiraligen Schaufeln

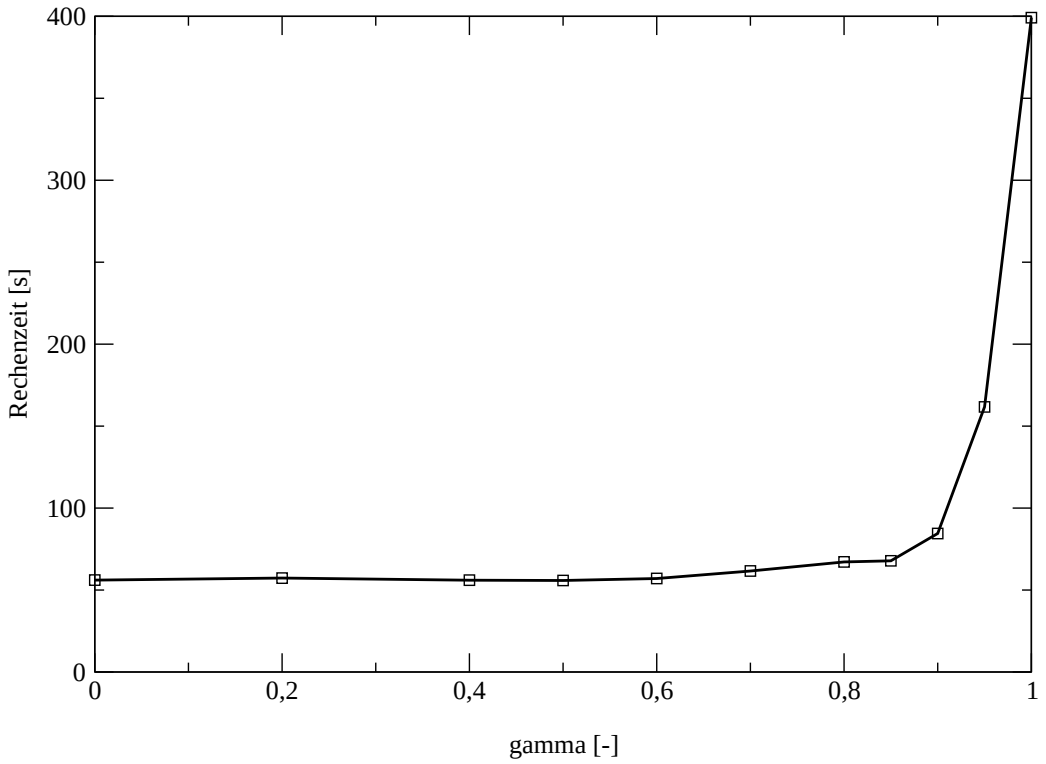

Abbildung 7.17: Rechenzeiten in Abhängigkeit vom Blending-Faktor γ für Leitgitter mit logarithmisch-spiraligen Schaufeln

7.6.3 Rotierendes Schaufelgitter mit logarithmisch-spiraligen Schaufeln

Schließlich wurde der Testfall mit logarithmisch-spiraligen Schaufeln benutzt, um die Implementierung der Rotationsterme zu überprüfen. Da der Finite-Volumen-Code als Randbedingung am Ausströmrand $p = 0$ benutzt, wurde der Ausströmrand so gelegt, daß er auf der Linie $p = 0$ liegt statt auf $r = const$. Es ergibt sich aus Gleichung 7.20

$$r(\eta) = \sqrt{-\frac{\mathcal{P}}{2} + \sqrt{\frac{\mathcal{P}^2}{4} - \mathcal{Q}}} \tag{7.25}$$

mit

$$\mathcal{P} = \frac{2}{\rho\omega^2}\left[p_0 - 2\rho\omega q\eta\right] \tag{7.26}$$

und

$$\mathcal{Q} = \frac{-q^2(1+a^2)}{\omega^2}. \tag{7.27}$$

Das zugehörige Gitter und die Drucklösung zeigt Abbildung 7.18.

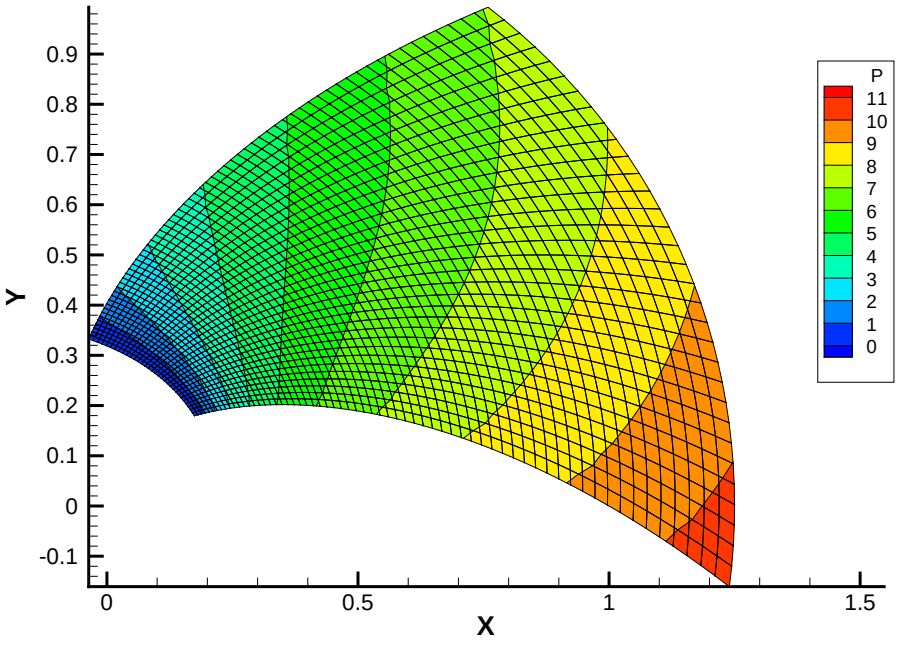

Abbildung 7.18: Druckverteilung und Gitter für Schaufelgitter mit rotierenden logarithmisch-spiraligen Schaufeln

Bei dieser Rechnung wurde, wie in der Arbeit von Müller [34], $\omega = -2$ benutzt. Die Rechnung erfolgte mit $\gamma = 0.8$. Abbildung 7.18 ist also direkt zu vergleichen mit Abbildung 6.26 von Müller.

Die Abbildung 7.19 und die Tabelle 7.4 zeigen, daß auch bei dieser Strömung der Einfluß der Re-Zahl relativ groß ist, wobei bei $Re = 500$ die besten Ergebnisse erzielt werden. Man erkennt, daß das Finite-Volumen-Verfahren 2. Ordnung genau ist. Die Rechenzeiten sind relativ unabhängig von der Re-Zahl, siehe Tabelle 7.5.

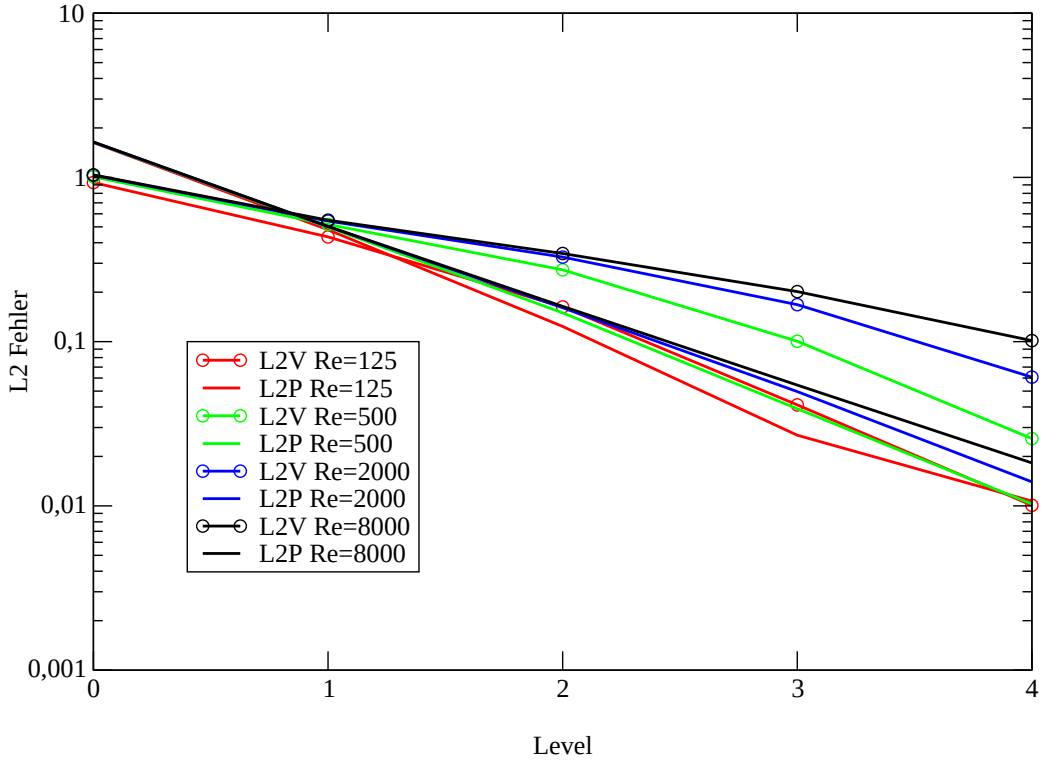

Abbildung 7.19: Fehler in Abhängigkeit von der Viskosität für Schaufelgitter mit rotierenden logarithmisch-spiraligen Schaufeln

Level	Gitter	Zellen	L_2^v	L_2^p
0	4*2*1	8	1.0076400	1.6368000
1	8*4*1	32	0.5171770	0.4972900
2	16*8*1	128	0.2734520	0.1498550
3	32*16*1	512	0.1004870	0.0390267
4	64*32*1	2048	0.0256734	0.0102675

Tabelle 7.4: L2-Fehler bei Re=500 für Schaufelgitter mit rotierenden logarithmisch-spiraligen Schaufeln

Re	Rechenzeit [s]
125	19.48
500	22.86
2000	26.14
8000	28.50

Tabelle 7.5: Rechenzeiten bei verschiedenen Reynolds-Zahlen für Schaufelgitter mit rotierenden logarithmisch-spiraligen Schaufeln

7.7 Gostelow-Testfall

Die exakte Lösung einer zweidimensionalen Potentialgitterströmung wurde 1965 von GOSTELOW angegeben [19]. Nach Untersuchungen von SZILAGYI und MÜLLER [55] enthält aber sowohl die Originalveröffentlichung [19] als auch das Buch von GOSTELOW [20] eine Reihe von Fehlern, so daß die Rechnung erneut durchgeführt werden mußte. Zur Verfügung stehen jetzt O-Gitter, die neben den Gitterpunkten auch die Werte der exakten Lösung enthalten. Die Rechnungen wurden mit einer Zellschicht in der dritten Dimension durchgeführt. Für die Testrechnungen wurde das O-Gitter mit 6000 Zellen benutzt. Der Konverter `mb2fem` erzeugt daraus ein Gitter mit 4 Leveln mit 112, 432, 1530 und 6000 Zellen. Abbildung 7.20 zeigt das Gitter und die berechneten Isolinien für den statischen Druck auf Level 1 mit 432 Zellen.

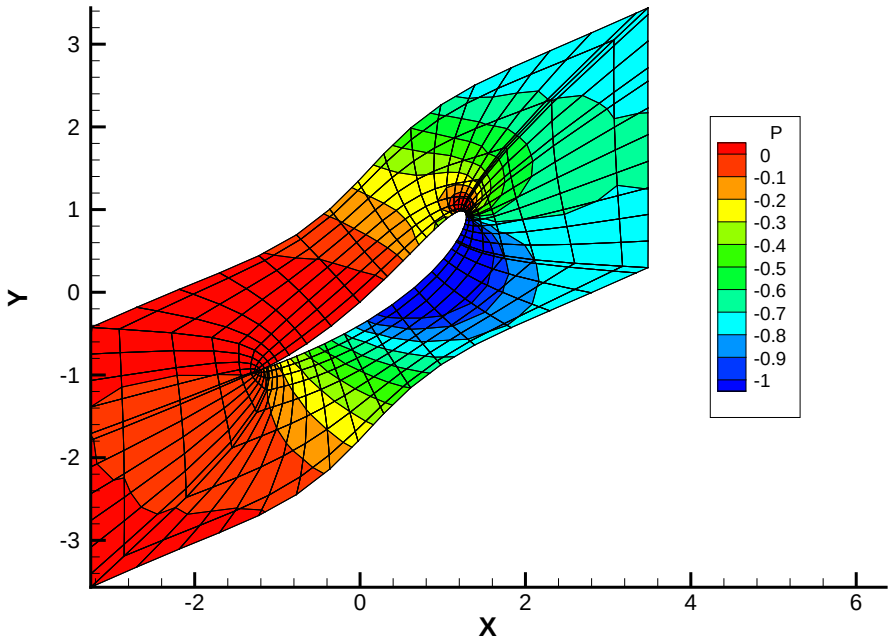

Abbildung 7.20: Drucklösung für Gostelow-Testfall auf Level 1 (432 Zellen)

Man erkennt, daß der Druck am Ausströmrand nicht konstant ist, was aber von der Randbedingung im FVM-Code mit p=0 vorausgesetzt wird. Ein Beweis, daß das Verfahren zweiter Ordnung genau ist, gelang deshalb für diesen Testfall nicht. Allerdings zeigen die Druckverteilungen in Abbildung 7.21, daß der FVM-Code schon auf groben Gittern sehr genau ist.

Gewisse Abweichungen im Bereich des Druckminimums und an der Hinterkante sind durch dissipative Effekte des Verfahrens und der eingestellten molekularen Viskosität zu erklären. Die Reynolds-Zahl betrug

$$Re = \frac{c_\infty L}{\nu} = 533. \tag{7.28}$$

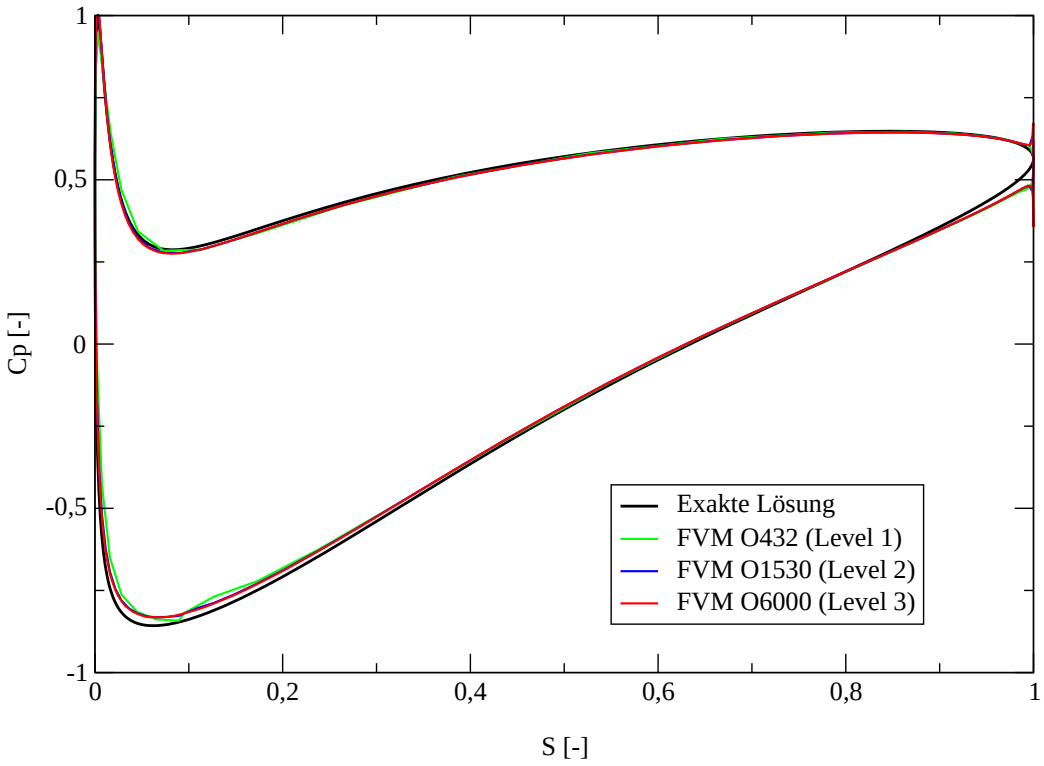

Abbildung 7.21: Druckverteilungen für Gostelow-Testfall

7.8 Francis-Turbine NQ30

Dieser Testfall ist ein Beispiel für eine Francis-Turbine mit niedriger spezifischer Drehzahl $n_q = 30[1/min]$. Der Testfall wurde auch von MÜLLER [34] berechnet. Es wurden zwei Arten von Gittern benutzt. Zum einen die in Kapitel 6.3.1 beschriebenen unstrukturierten Hexaedergitter und zum anderen die in Kapitel 6.3.2 beschriebenen blockstrukturierten HO-Gitter.

Zunächst soll der Einfluß verschiedener Parameter auf die Integralwerte Ψ_t und $\Psi_{t,th}$ sowie die Rechenzeit untersucht werden. Die Parameter sind:

1. Viskosität, d.h. Re-Zahl
2. Diskretisierung der konvektiven Terme
3. Abbruchkriterium
4. Gitterfeinheit

Gerechnet wurde in den Fällen 1 bis 3 auf dem Gitter HO12376 mit 2 Leveln und Mehrgitterverfahren.

7.8.1 Einfluß der Viskosität

Da hier turbulente Strömungen hoher Reynolds-Zahl mit einem Navier-Stokes Verfahren mit Euler-Randbedingungen näherungsweise berechnet werden sollen, kann die molekulare Viskosität in gewissen Grenzen variiert werden. Bei zu großer Viskosität ist eine schlechte Lösungsgenauigkeit und bei zu kleiner Viskosität eine Verschlechterung der Konvergenzrate zu erwarten. Diese theoretisch vorhergesagten Ergebnisse werden durch Testrechnungen bestätigt, siehe Tabelle 7.6 und Abbildung 7.22.

Re	Ψ_t	$\Psi_{t,th}$	η_h	CPU-Zeit [s]
882	1.89925	1.78520	0.93995	570.6
1764	1.88069	1.80021	0.95721	445.3
3526	1.86969	1.81002	0.96809	265.8
6000	1.86657	1.81219	0.97087	175.8
7054	1.86447	1.81577	0.97388	239.4
9404	1.86282	1.81751	0.97567	378.5
14106	1.86179	1.82007	0.97759	600.4

Tabelle 7.6: Abhängigkeit der Integralwerte von der Viskosität

Die Re-Zahl ist für Turbinen definiert als

$$Re = \frac{\omega D^2}{2\nu} = \frac{\omega r_{ref} D_{ref}}{\nu} = \frac{2\omega r_{ref}^2}{\nu}. \tag{7.29}$$

Die reale Re-Zahl der Turbine beträgt $Re = 7.05 * 10^6$. Für eine Re-Zahl im Bereich von 6000 werden sehr gute Rechenzeiten erreicht, wobei sich die Integralwerte nur noch wenig von denen bei sehr hoher Re-Zahl unterscheiden.

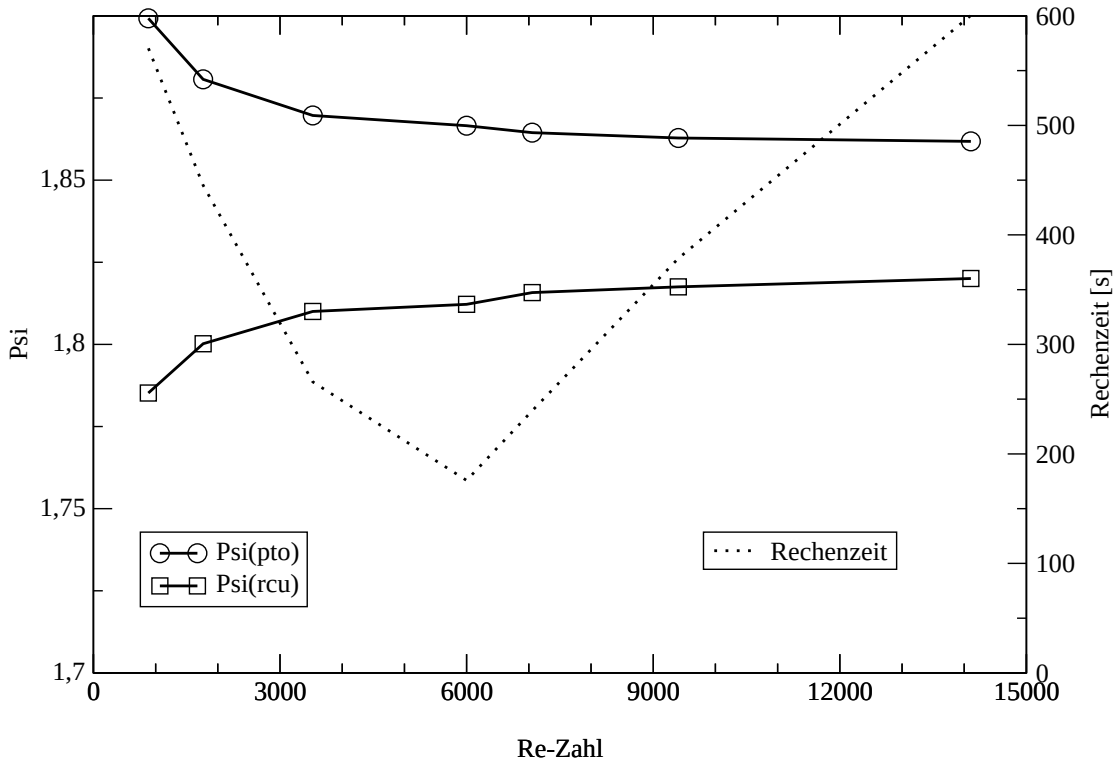

Abbildung 7.22: Einfluß der Re-Zahl auf die Druckzahlen Ψ_t, $\Psi_{t,th}$ und die Rechenzeit für FT30, HO12376 Gitter, $\gamma = 0.8$

Beim Vergleich der Druckverteilungen bei unterschiedlichen Re-Zahlen, siehe Abbildung 7.23, zeigt sich, daß mit zunehmender Viskosität die Saugspitze deutlich kleiner wird, so daß bei hoher Viskosität die Übereinstimmung mit dem TASCflow-Ergebnis am besten ist. Der physikalische Effekt ist zum einen in einer Dämpfung am Ort der Saugspitze und zum anderen in einer lokalen Änderung der Zuströmung an der Schaufelspitze zu sehen.

Für die Drallverteilung hinter der Schaufel, siehe Abbildung 7.24, ist die Übereinstimmung der Euler-Rechnung mit dem TASCflow-Ergebnis ebenfalls bei der größten Viskosität am besten.

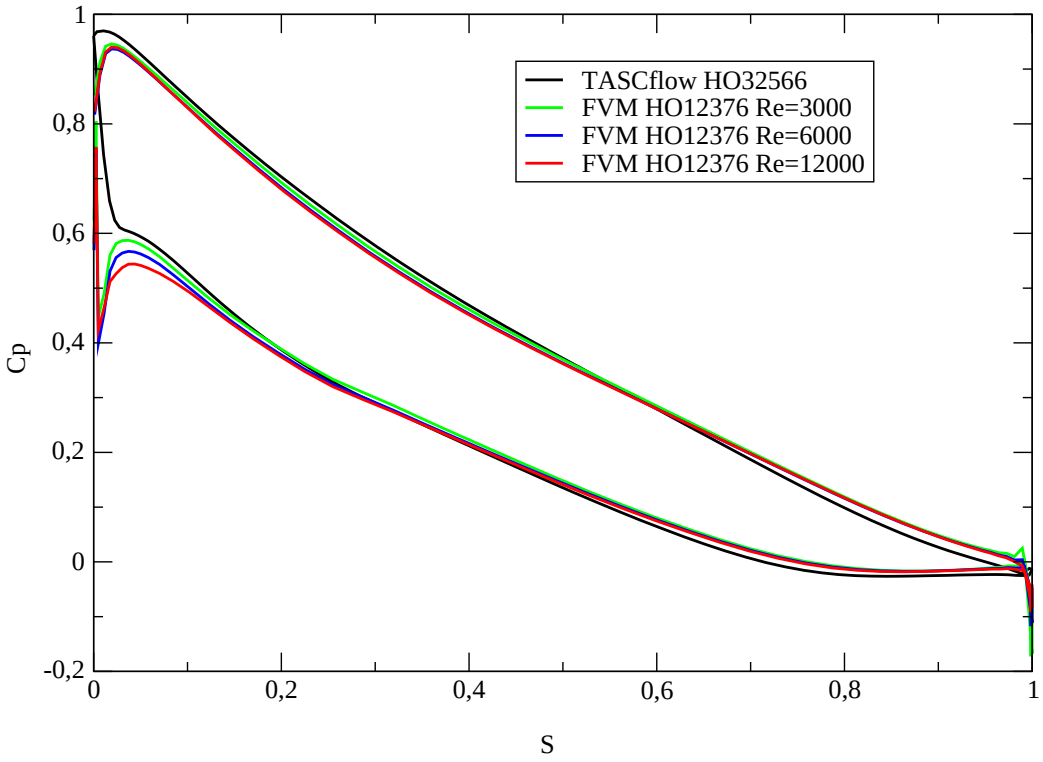

Abbildung 7.23: Einfluß der Viskosität auf die Druckverteilung für FT30, Schaufelmitte

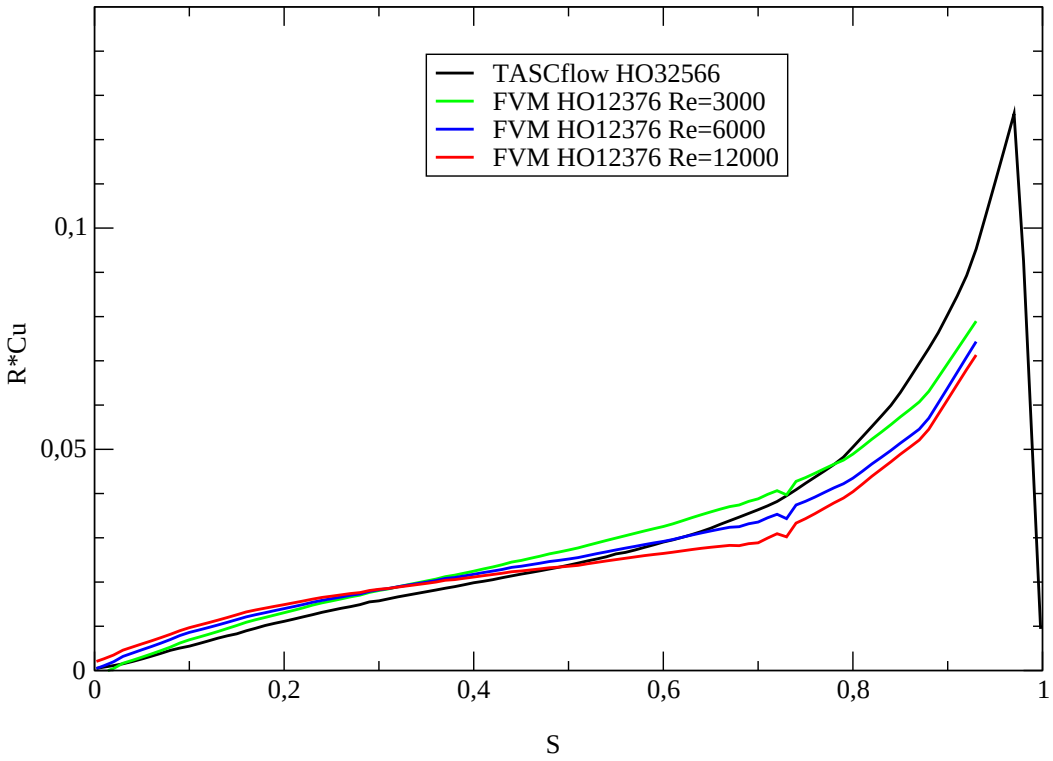

Abbildung 7.24: Einfluß der Viskosität auf die Drallverteilung hinter der Schaufel für FT30

7.8.2 Einfluß der Diskretisierung der konvektiven Terme

Bei den Berechnung laminarer Strömungen, siehe Kapitel 7.1 bis 7.5, konnten zentrale Differenzen für die konvektiven Terme verwendet werden. Die Genauigkeit des Verfahrens war somit von zweiter Ordnung. Bei den reibungsfreien Strömungen der Kapitel 7.6 und 7.7 erwies sich eine Mischung von zentralen Differenzen und reinem Upwind als besser, da bei reinen zentralen Differenzen unerwünschte Schwingungen auftraten. Den Einfluß des Blending-Faktors γ bei $Re = 7054$ zeigt Tabelle 7.7 und Abbildung 7.25.

γ	Ψ_t	$\Psi_{t,th}$	η_h	CPU-Zeit [s]
0.0	1.88030	1.79888	0.95670	175.6
0.5	1.87276	1.80683	0.96479	175.8
0.6	1.87084	1.80867	0.96677	176.9
0.7	1.86921	1.81077	0.96873	193.4
0.8	1.86695	1.81321	0.97121	209.4
0.9	1.86447	1.81577	0.97388	239.4
0.95	1.86277	1.81698	0.97542	323.4
1.0	1.86114	1.81809	0.97687	579.1

Tabelle 7.7: Abhängigkeit der Integralwerte und CPU-Zeit vom Blending-Faktor γ

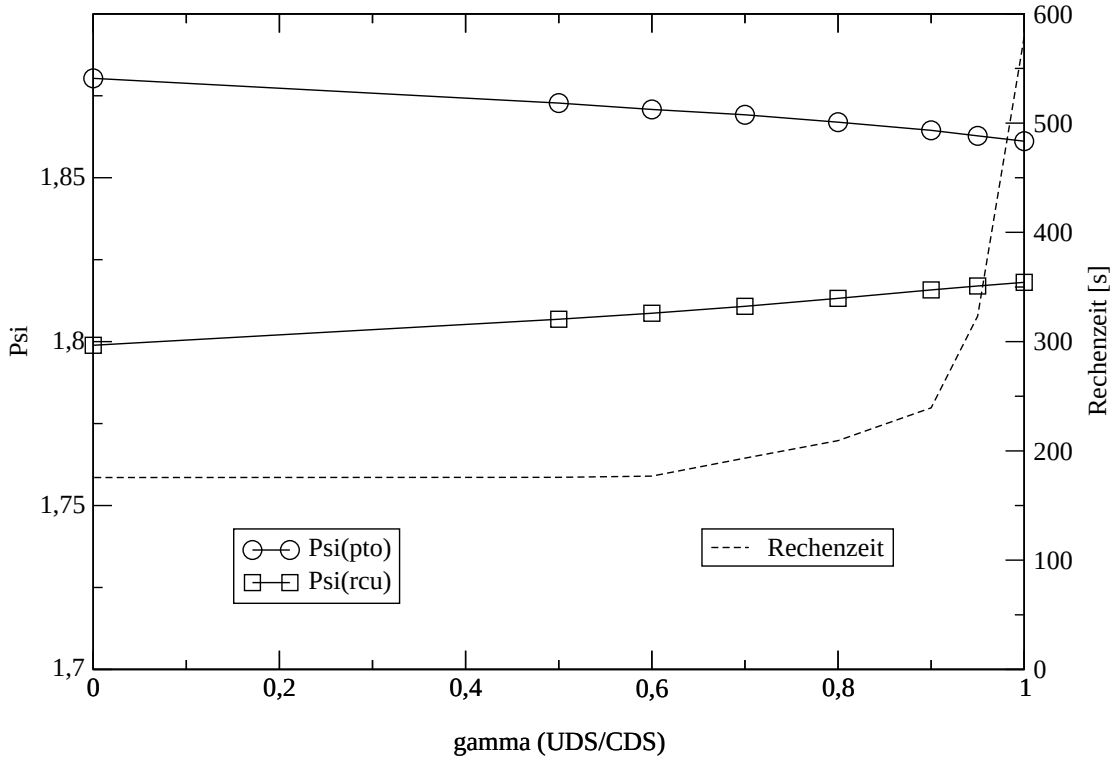

Abbildung 7.25: Einfluß des Blending-Faktors γ auf die Druckzahlen Ψ_t, $\Psi_{t,th}$ und die Rechenzeit für FT30, HO12376 Gitter, Re=7054

Man erkennt, daß mit steigendem γ die Rechenzeit länger wird, während die Genauigkeit besser wird. Dies erkennt man an dem steigenden hydraulischen Wirkungsgrad

η_h, der bei einer exakten Euler Lösung gleich 1 sein müßte. Allerdings bleibt die Rechenzeit für $0 < \gamma < 0.6$ fast konstant, so daß also Werte zwischen 0.6 und 0.9 als am geeignetsten erscheinen.

7.8.3 Einfluß des Abbruchkriteriums

Es soll hier untersucht werden, bei welchem Abbruchkriterium die Rechnung abgebrochen werden kann. Dabei sollen hier Rechnungen von der Vorbelegung mit Null betrachtet werden. Bei Optimierungsrechnungen könnte jedoch die Lösung der letzten Geometriemodifikation als Startlösung benutzt werden. Die dafür optimalen Abbruchkriterien müßten mit konkreten Optimierungsrechnungen gesondert bestimmt werden.

In der Tabelle 7.8 sind die Integralwerte in Abhängigkeit vom absoluten Konvergenzkriterium ε_{abs} aufgeführt. Außerdem sind in der Tabelle die relativen Fehler für Ψ_t und $\Psi_{t,th}$ in Bezug auf die *auskonvergierte* Lösung bei $\varepsilon_{abs} = 2 * 10^{-6}$ aufgeführt. Diese Werte und die Rechenzeiten sind auch in Abbildung 7.26 dargestellt.

ε_{abs}	$\varepsilon_{rel,SG}$	$\varepsilon_{rel,MG}$	Ψ_t	$\Psi_{t,th}$	$\frac{\Delta\Psi_t}{\Psi_t}$	$\frac{\Delta\Psi_{t,th}}{\Psi_{t,th}}$	η_h	CPU-Zeit
[-]	[-]	[-]	[-]	[-]	$[10^{-5}]$	$[10^{-5}]$	[-]	[s]
1.0e-3	3.2e-2	1.8e-1	1.87097	1.79717	239	-897	0.96056	76.5
5.0e-4	1.6e-2	8.9e-2	1.87935	1.80698	688	-356	0.96149	99.8
3.0e-4	9.6e-3	5.4e-2	1.87069	1.81586	224	133	0.97069	121.1
1.5e-4	4.8e-3	2.7e-2	1.86415	1.81626	-126	156	0.97431	138.6
1.0e-4	3.2e-3	1.8e-2	1.86429	1.81464	-119	66	0.97337	156.6
6.8e-5	2.2e-3	1.2e-2	1.86580	1.81281	-38	-35	0.97160	174.6
4.5e-5	1.4e-3	8.0e-3	1.86679	1.81280	15	-35	0.97108	193.6
3.0e-5	9.6e-4	5.4e-3	1.86695	1.81321	23	-13	0.97121	209.4
1.5e-5	4.8e-4	2.7e-3	1.86652	1.81360	1	9	0.97165	237.2
7.5e-6	2.4e-4	1.3e-3	1.86643	1.81345	-4	1	0.97162	274.0
2.0e-6	6.4e-5	3.6e-4	1.86651	1.81344			0.97157	344.2

Tabelle 7.8: Abhängigkeit der Integralwerte vom Abbruchkriterium

Bei $\varepsilon_{abs} = 6.8 * 10^{-5}$ ist der Fehler kleiner als ein halbes Promille. Das initiale Residuum auf dem Feingitter beträgt beim Mehrgitterverfahren $5.6 * 10^{-3}$ und würde auf dem feinsten Gitter bei Nullösung den Wert $3.13 * 10^{-2}$ haben. Das Konvergenzkriterium $\varepsilon_{abs} = 6.8 * 10^{-5}$ entspricht deshalb bei einer Eingitterrechnung ohne Prolongation einem relativen Konvergenzkriterium von $\varepsilon_{rel,SG} = 2.2 * 10^{-3}$ und bei einer Mehrgitterrechnung einem relativen Konvergenzkriterium von $\varepsilon_{rel,MG} = 1.2 * 10^{-2}$. Es genügt also, beim Mehrgitterverfahren das Residuum auf dem Feingitter um 2 Größenordnungen zu reduzieren!

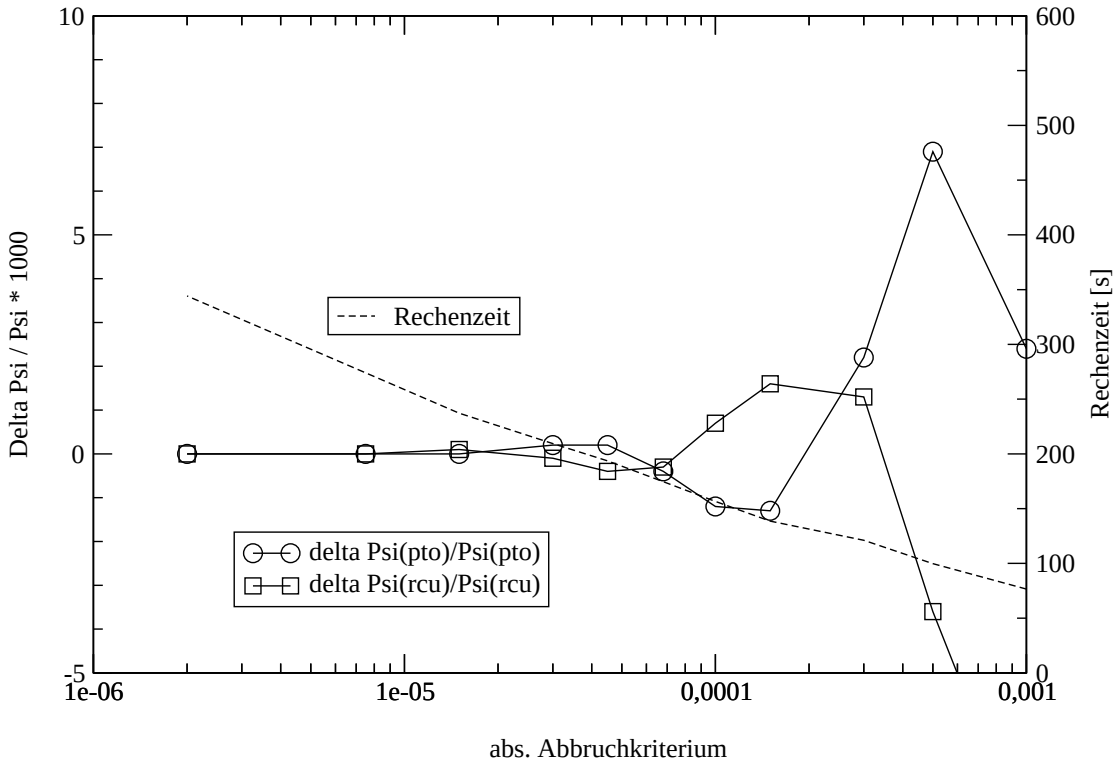

Abbildung 7.26: Einfluß des absoluten Abbruchkriteriums auf die Druckzahlen und die Rechenzeit für FT30, HO12376 Gitter, Re=7054

7.8.4 Einfluß der Gitterfeinheit

Es soll im folgenden der Einfluß der Gitterfeinheit bei den HO-Gittern untersucht werden. Zum Zeitpunkt der Untersuchung konnte keine systematische Gittervariation durchgeführt werden, da der Gittergenerator der Firma Voith aus lizenzrechtlichen Gründen nicht einsetzbar war und der entsprechende Gittergenerator des Lehrstuhls noch in der Entwicklung war. Es wurden deshalb zwei Gitter benutzt, die schon früher beim Test des FEM-Codes generiert worden waren, nämlich die Gitter HO12376 und HO32566. Außerdem wurde das Ergebnis des Gitters HO4730 ausgewertet, das durch den Konverter `mb2fem` aus dem Gitter HO32566 durch Vergröberung entsteht.

Die entsprechenden Integralwerte sind Tabelle 7.9 und Abbildung 7.27 zu entnehmen.

Code	Anzahl der Zellen	Ψ_t	$\Psi_{t,th}$	η_h	CPU-Zeit [s]
FVM	4730	1.865	1.809	0.970	73.5
	12376	1.867	1.813	0.971	186.5
	32566	1.862	1.806	0.970	425.0
TASCflow	12376	1.867	1.707	0.914	639.0
	32566	1.864	1.744	0.935	1465.0

Tabelle 7.9: Abhängigkeit der Integralwerte von der Zellenanzahl

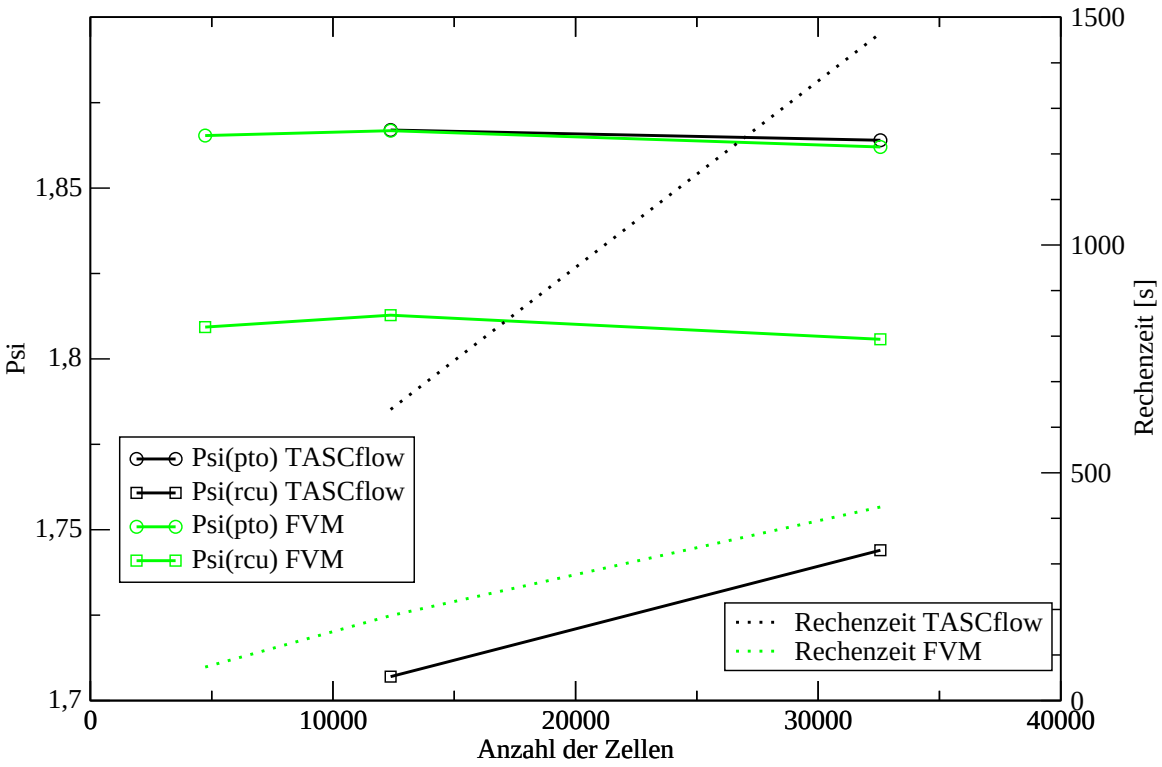

Abbildung 7.27: Einfluß der Zellenanzahl auf die Druckzahlen und die Rechenzeit für FT30

Bei den TASCflow-Lösungen unterscheidet sich die theoretischen Totaldruckzahlen $\Psi_{t,th}$ für die verschiedenen Gitter deutlich. Bei der Euler-Rechnung ist dies nicht der Fall. Die Totaldruckzahlen Ψ_t sind bei beiden Verfahren nur wenig von Gitter

abhängig. Gitter für Euler-Rechnung können somit deutlich gröber als bei Navier-Stokes-Rechnungen sein.

7.8.5 Vergleich mit TASCflow-Ergebnissen

Die Strömungslösung soll nun anhand von Druckverteilungen auf der Schaufel und von Drallverteilungen am Ausströmrand mit Ergebnissen von TASCflow verglichen werden. Dabei wurden die folgenden Parameter zugrunde gelegt:

- $Re = 6000$,
- $\gamma = 0.8$ und
- $\varepsilon_{rel,MG} = 8 * 10^{-3}$.

Der Konvergenzverlauf der Rechnung auf dem HO12376 Gitter ist in Abbildung 7.28 dargestellt.

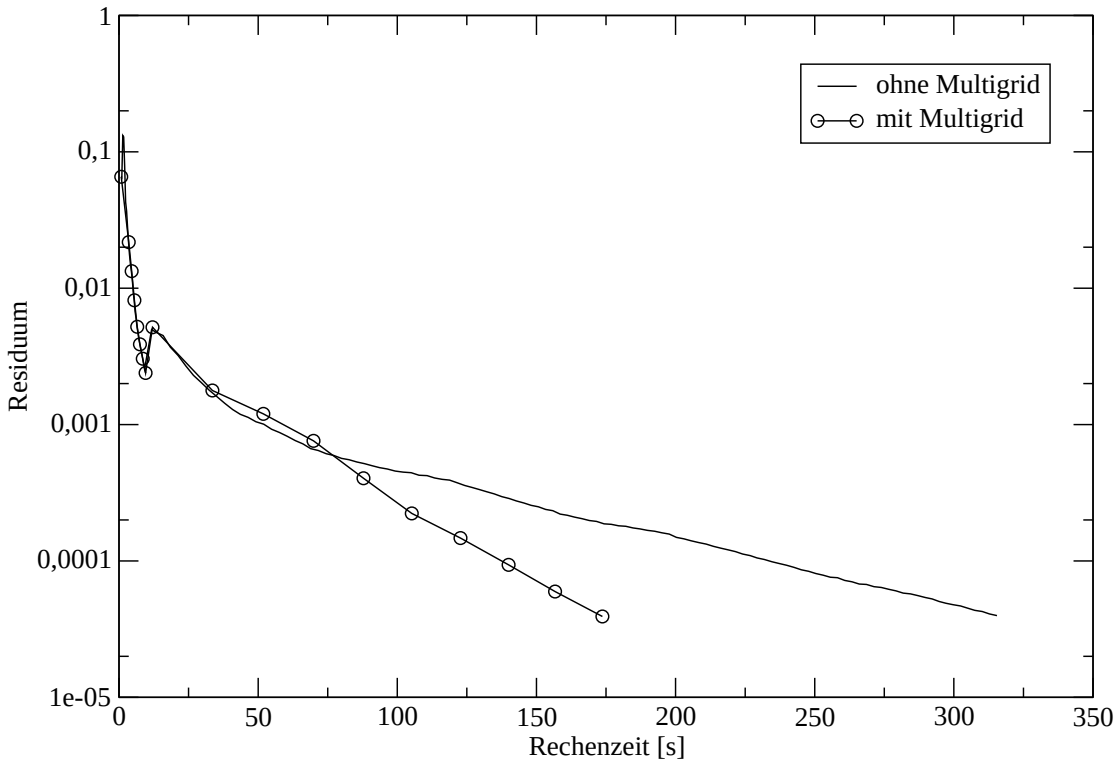

Abbildung 7.28: Konvergenzverlauf für FT30 auf HO12376 Gitter

In Abbildung 7.29 ist die Druckverteilung etwa in Schaufelmitte[1] für die TASCflow-rechnung und die drei verschiedenen HO-Gitter dargestellt. Abbildung 7.30 zeigt die entsprechenden Ergebnisse bei Verwendung der Finite-Element-Diskretisierung. Abbildung 7.31 zeigt eine Ausschnittsvergrößerung der Druckverteilungen des Finite-Volumen-Codes nahe der Vorderkante.

[1]Genau ist dies die 9. Strombahn des TASCflow Gitters bzw. $\eta = 0.44$ beim Postprocessing des unstrukturierten Codes.

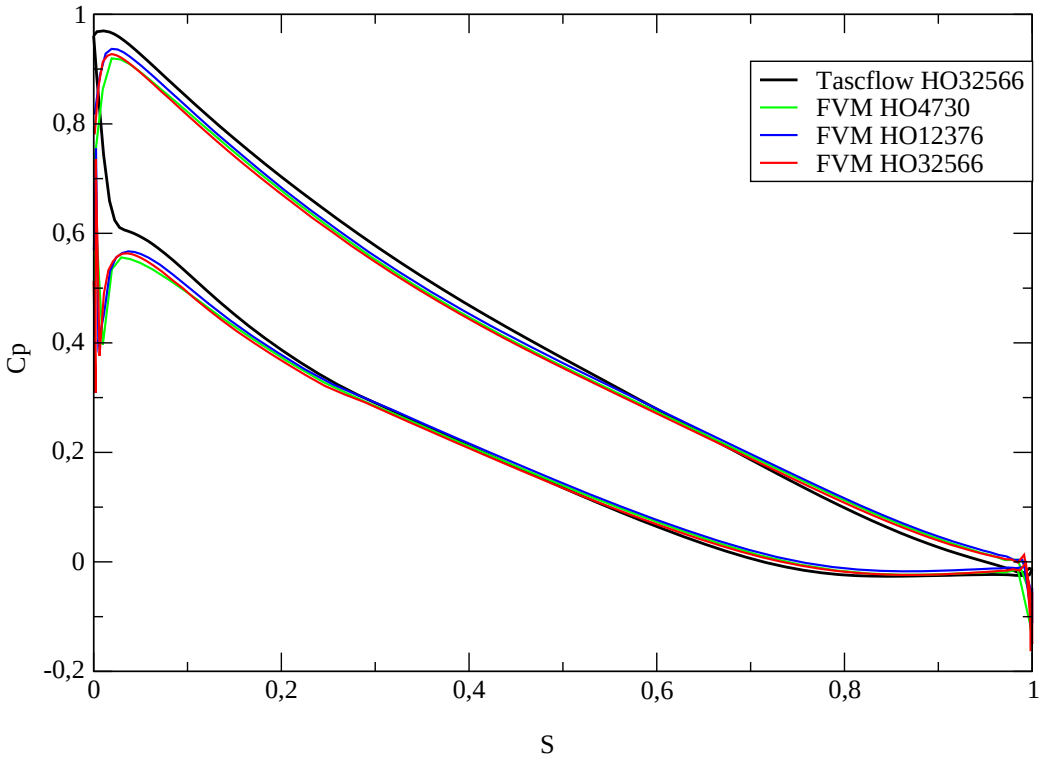

Abbildung 7.29: Druckverteilungen mit Finite-Volumen-Diskretisierung für FT30 bei Verwendung blockstrukturierter Gitter

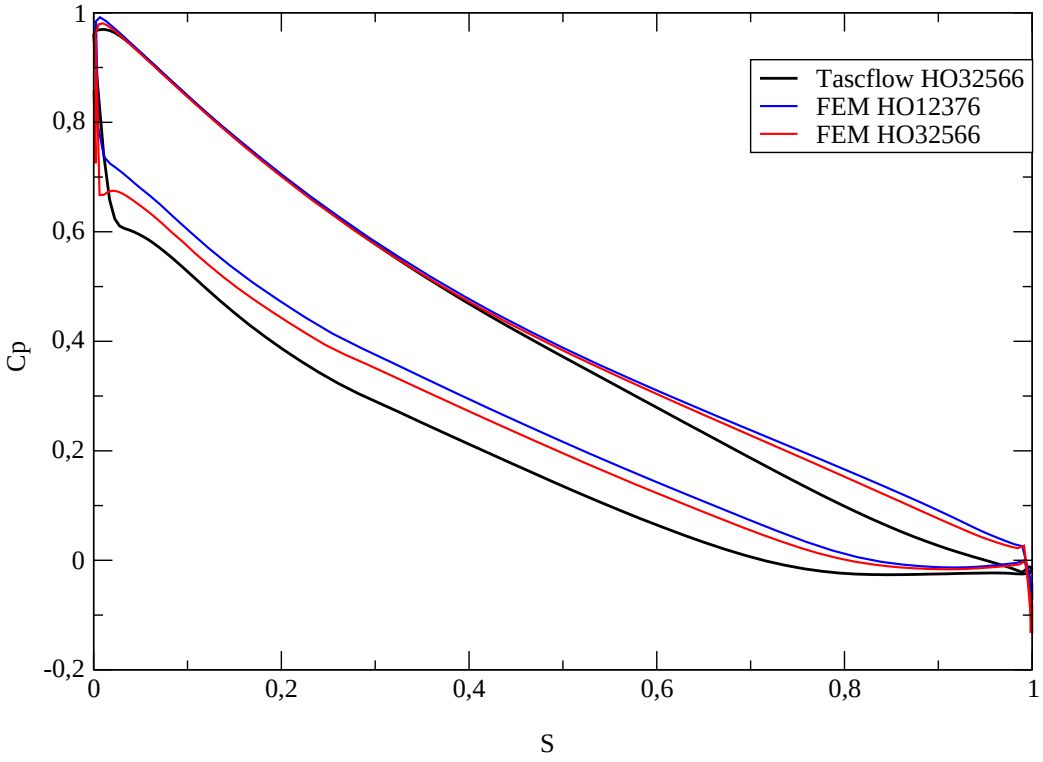

Abbildung 7.30: Druckverteilungen mit Finite-Elemente-Diskretisierung für FT30 bei Verwendung blockstrukturierter Gitter

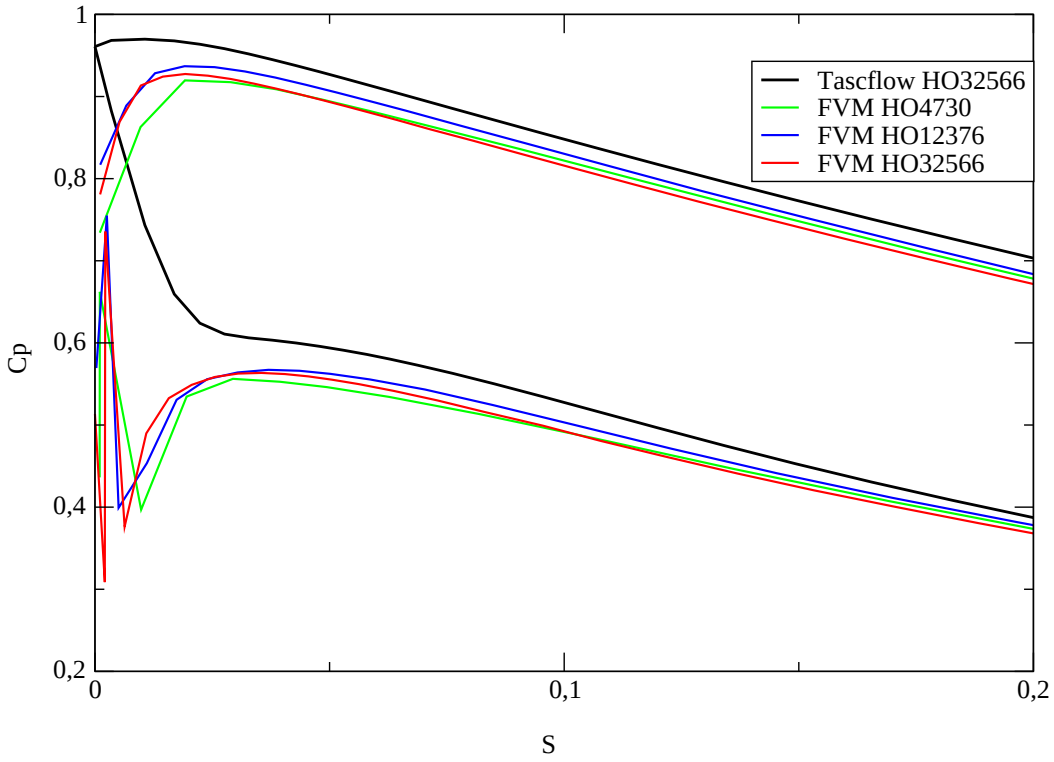

Abbildung 7.31: Druckverteilungen an der Vorderkante der Schaufel mit Finite-Volumen-Diskretisierung für FT30 bei Verwendung blockstrukturierter Gitter

In Abbildung 7.32 sind entsprechende Druckverteilungen für die unstrukturierten Hexaedergitter von Kapitel 6.3.1 dargestellt.

Für diese Druckverteilungen ist folgendes festzustellen:

- Die Unterschiede der verschiedenen Gitterfeinheiten sind für das unstrukturierte FV-Verfahren auf den HO-Gittern klein, siehe Abbildung 7.29.
- Das unstrukturierte FV-Verfahren berechnet auf den HO-Gittern eine Saugspitze, die bei der TASCflow-Rechnung und den FEM-Code Rechnungen nicht auftaucht. In den FEM-Rechnungen kann eine Saugspitze aufgrund der mangelhaften Genauigkeit nicht berechnet werden. Die TASCflow-Rechnung wurde dagegen mit Einstellungen[2] durchgeführt, die zu ähnlicher Genauigkeit wie beim vorliegenden FV-Code bei $\gamma = 0.8$ führen soll. Hier kann der Unterschied dadurch erklärt werden, daß zum einen das HO32566 Gitter im Nasenbereich für eine Navier-Stokes-Rechnung sehr grob ist und zum anderen unterschiedliche physikalische Gleichungen gelöst werden.
- Das FVM-Verfahren ist erheblich genauer als das FEM-Verfahren, siehe Abbildung 7.32 oder Abbildung 7.29 im Vergleich zu Abbildung 7.30.
- Unmittelbar an der Nase (0.5% der Sehnenlänge), siehe Abbildung 7.31, zeigen die Druckverteilungen des FV-Euler-Verfahrens starke Schwankungen. Für eine Optimierung müßte dieser Bereich unter Umständen ausgeblendet werden.

[2]ISKEW=3 ist eine Einstellung für robustes Konvergenzverhalten bei einer Lösungsgenauigkeit von *fast* zweiter Ordnung.

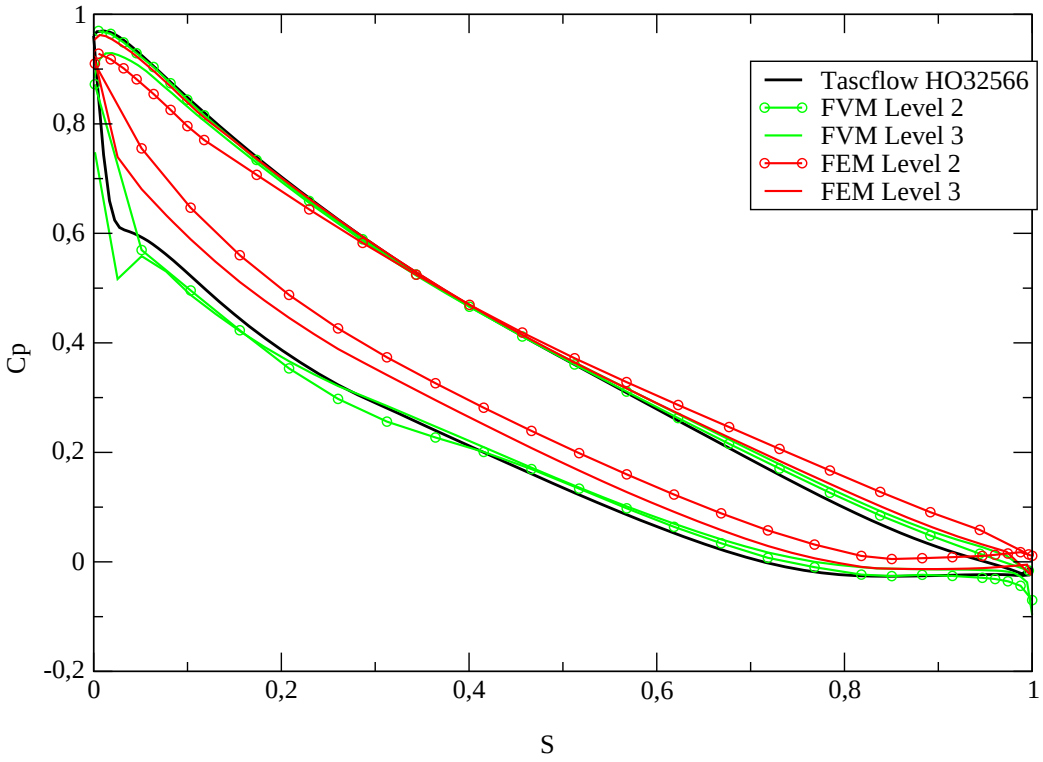

Abbildung 7.32: Druckverteilungen für FT30 mit unstrukturierten Hexaedergittern für die Level 2 und 3

Ebenso gibt es Extremwerte an der Hinterkante, siehe Abbildung 7.29, die aber bei einem Eulerverfahren zu erwarten sind.

Das sehr gute Konvergenzverhalten des FV-Codes bezüglich Gitterverfeinerung kann auch quantifiziert werden, wenn ein Integralmaß der Form

$$c_{p,I} = \int_{DS} \left| c_p - c_p^{TF} \right| \, dS + \int_{SS} \left| c_p - c_p^{TF} \right| \, dS \tag{7.30}$$

definiert wird. Dabei ist c_p^{TF} der c_p-Wert der TASCflow-Rechnung auf dem HO32566 Gitter und c_p der c_p-Wert der Vergleichsrechnung. Das entsprechende Verhalten von TASCflow, FV-Code und FEM-Code ist in Abbildung 7.33 dargestellt. Während TASCflow und FEM-Code einen deutlichen Gittereinfluß erkennen lassen, ist dieser beim FV-Code klein.

In Abbildung 7.34 ist zu erkennen, daß durch Hinzufügen numerischer Viskosität, d.h. Verkleinern des Blendingfaktors $\gamma = 0.8$ auf $\gamma = 0.0$, die Saugspitze abgeschwächt wird und insgesamt die Übereinstimmung von TASCflow und FV-Euler-Verfahren bzgl. der Druckverteilung im Nasenbereich verbessert wird.

Schließlich zeigen die Abbildungen 7.35 und 7.36 die Druckverteilungen an der Nabe und an der Deckscheibe.

Die gemachten Aussagen treffen auch für diese Druckverteilungen zu. Die tendenziell etwas größeren Unterschiede im Vergleich zu TASCflow sind sicherlich auf die Nichtberücksichtigung der Grenzschichten auf Nabe und Deckscheibe im FV-Eulercode zurückzuführen.

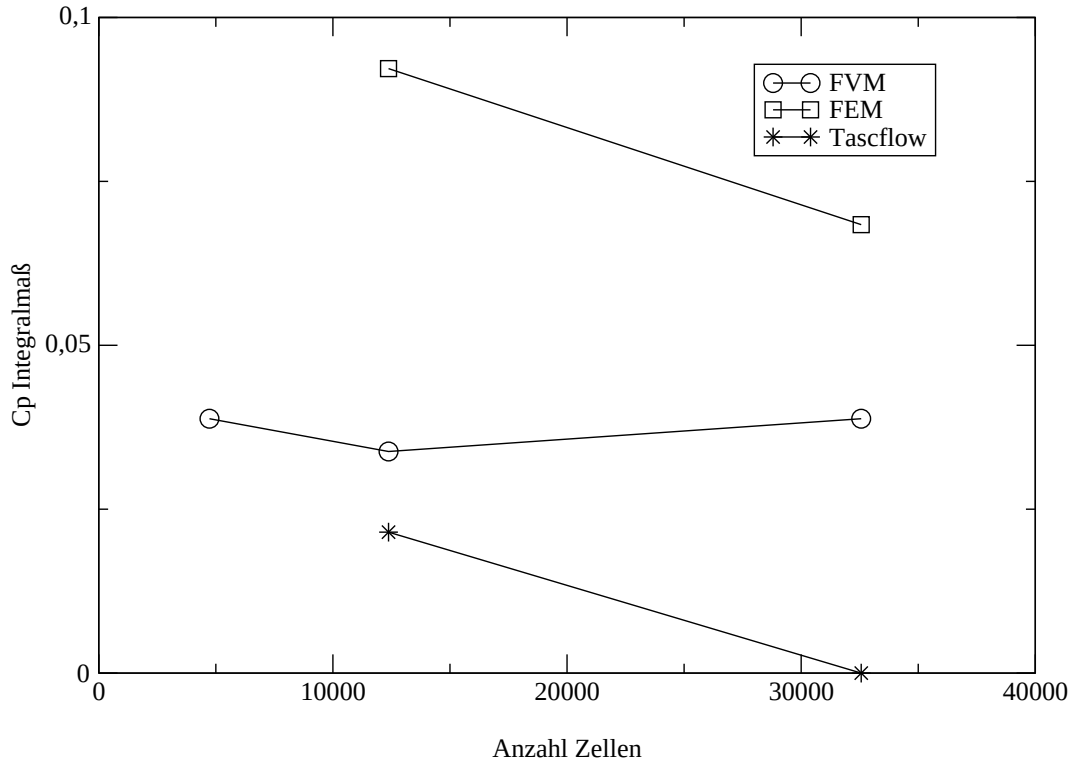

Abbildung 7.33: Integrales Maß der Abweichung der FVM- und FEM-Lösungen von der TASCflow HO32566 Druckverteilung

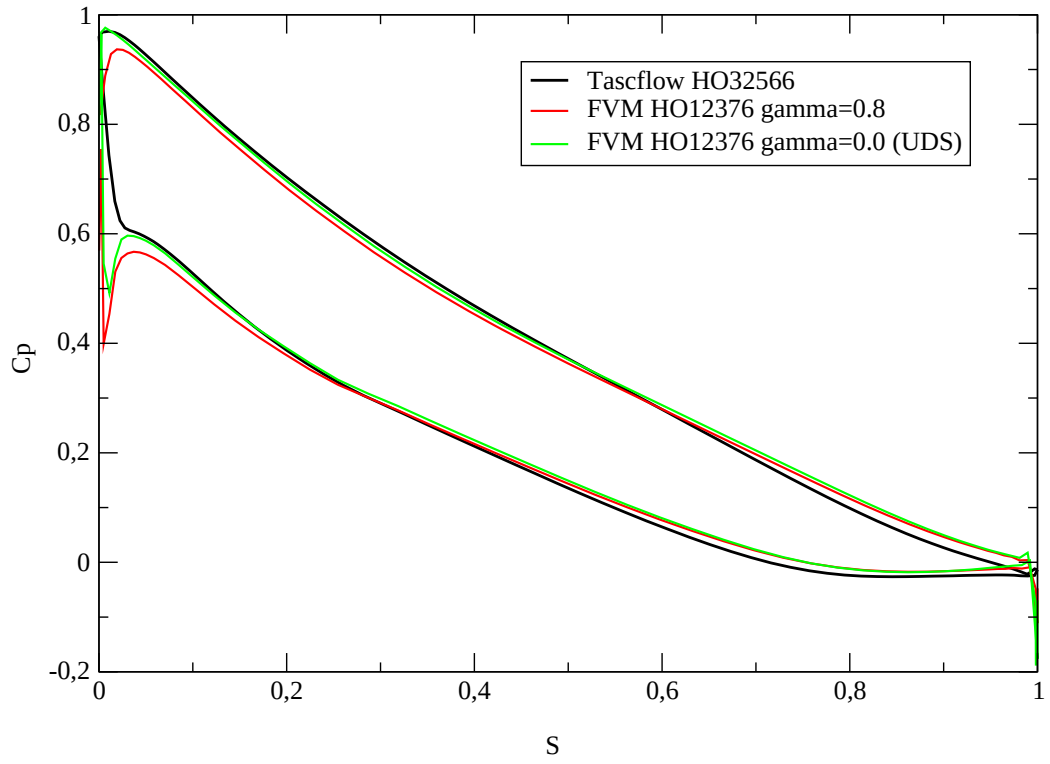

Abbildung 7.34: Druckverteilungen bei unterschiedlicher Approximation des konvektiven Terms in der Mitte der Schaufel bei FT30

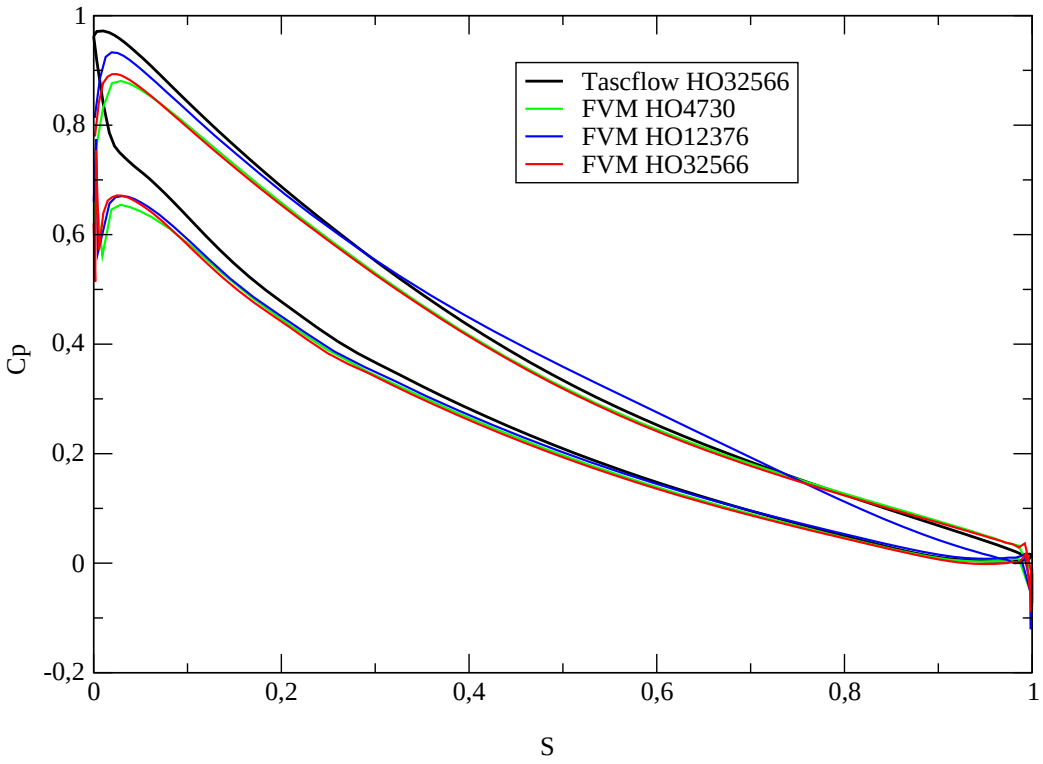

Abbildung 7.35: Druckverteilungen an der Nabe bei FT30 für unterschiedliche Gitter

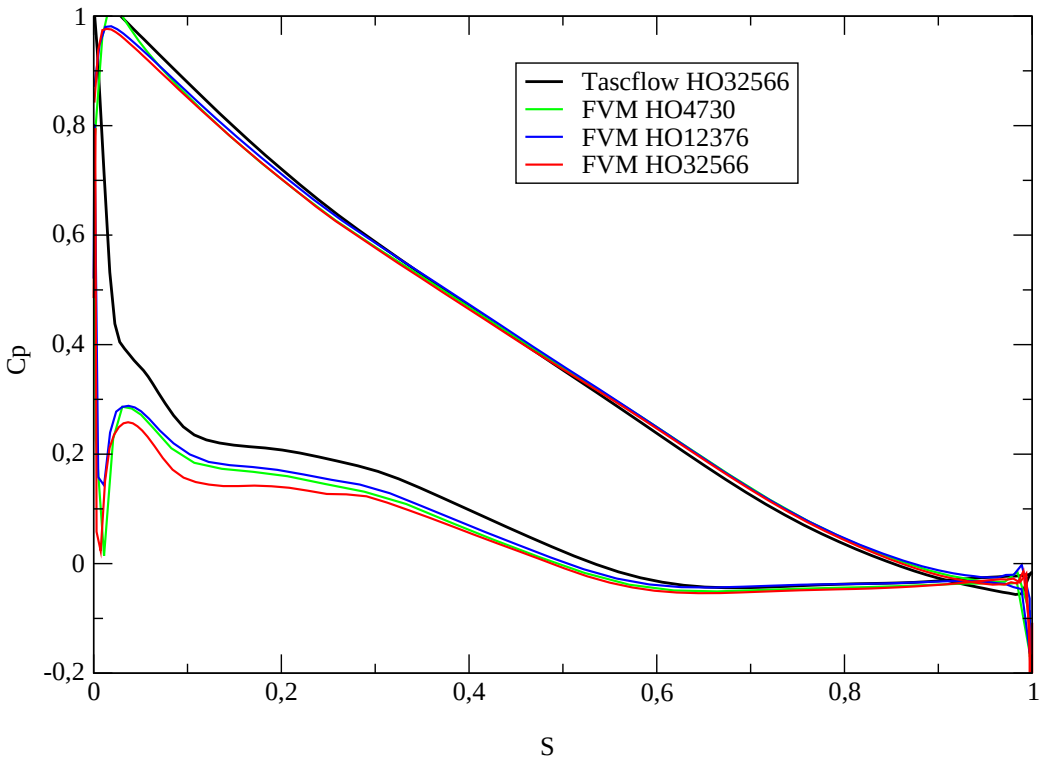

Abbildung 7.36: Druckverteilungen an der Deckscheibe bei FT30 für unterschiedliche Gitter

Schließlich sollen noch die Drallverteilungen am Ausströmrand verglichen werden. Diese sind von besonderem Interesse, da die Saugrohrströmung wesentlich von der Drallverteilung bestimmt wird. Die Drallverteilungen sind in Abbildung 7.37 dargestellt.

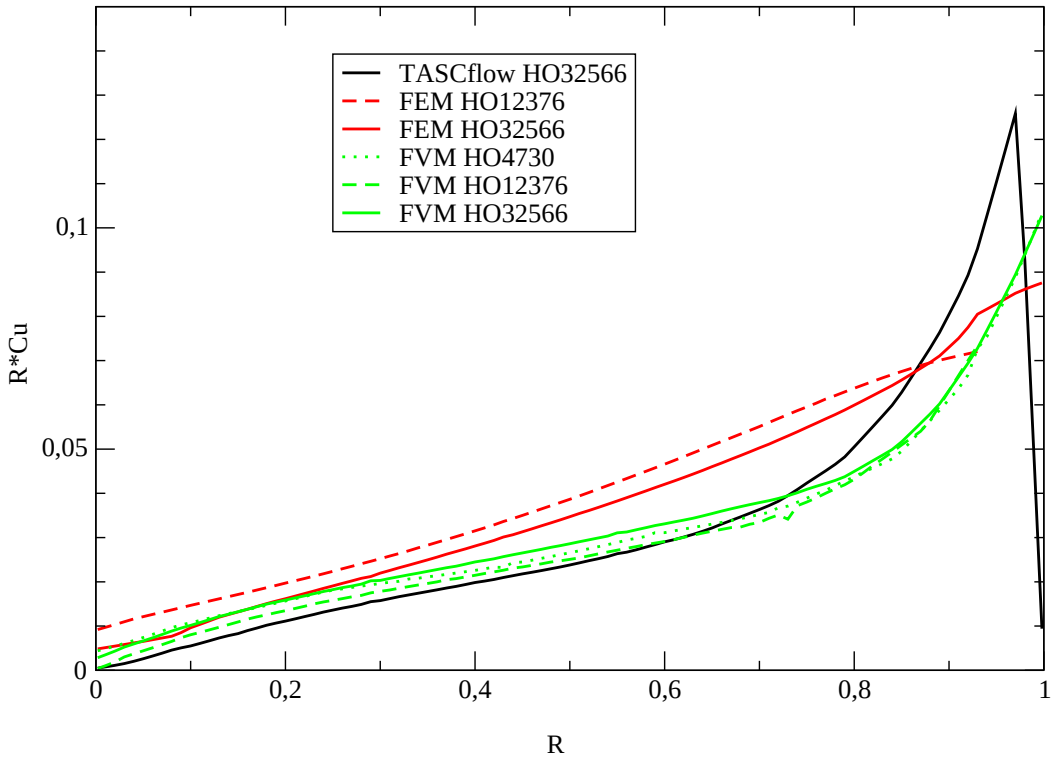

Abbildung 7.37: Radiale Drallverteilung im Nachlauf bei FT30 für FVM- und FEM-Verfahren und unterschiedliche Gitter

Die für die Druckverteilung gemachten Aussagen gelten auch für die Drallverteilung am Laufradaustritt der Turbinen. Die Ergebnisse des FVM-Codes stimmen besser mit dem TASCflow-Ergebnis überein als die FEM-Code Ergebnisse. Der Gittereinfluß ist beim FVM-Code relativ gering. Unterschiede zwischen FVM-Code und TASCflow lassen sich, zumindest teilweise, durch die unterschiedliche physikalische Modellierung erklären.

Für die Verteilung des Dralls $R * c_u$ kann ebenfalls ein Integralmaß eingeführt werden. Der entsprechende Verlauf für FV-Code und FEM-Code ist in Abbildung 7.38 dargestellt.

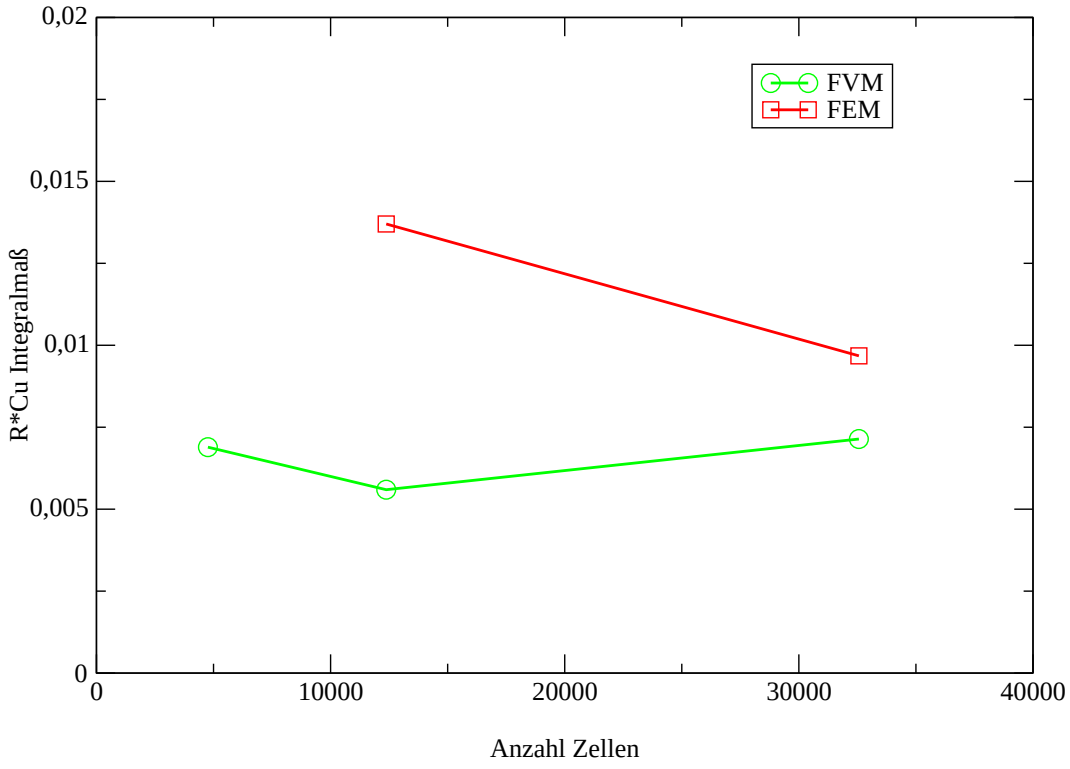

Abbildung 7.38: Integrales Maß der Abweichung von der TASCflow HO32566 Drallverteilung bei FT30 in Abhängigkeit von der Anzahl der Zellen

7.9 Francis-Turbine NQ50

Dieser Testfall ist ein Beispiel für eine Francis-Turbine mit mittlerer Drehzahl. Der Einfluß des Abbruchkriteriums und der Diskretisierung der konvektiven Terme wurde nach den umfangreichen Untersuchungen beim FT30 nicht gesondert untersucht.

7.9.1 Einfluß der Gitterfeinheit

Es sollen hier Ergebnisse auf drei verschieden feinen HO-Gittern bei einer Re-Zahl von $Re = 1792$ präsentiert werden. In Tabelle 7.10 und in Abbildung 7.39 ist die Abhängigkeit der Integralwerte von der Gitterfeinheit dargestellt.

Code	Anzahl der Zellen	Ψ_t	$\Psi_{t,th}$	η_h	CPU-Zeit [s]
FVM	4730	1.437	1.369	0.953	59.2
	12376	1.442	1.374	0.953	314.4
	32566	1.443	1.374	0.952	1682.4
TASCflow	12376	1.436	1.370	0.962	1429.0
	32566	1.431	1.386	0.968	3007.0

Tabelle 7.10: Abhängigkeit der Integralwerte von der Zellenanzahl für FT50

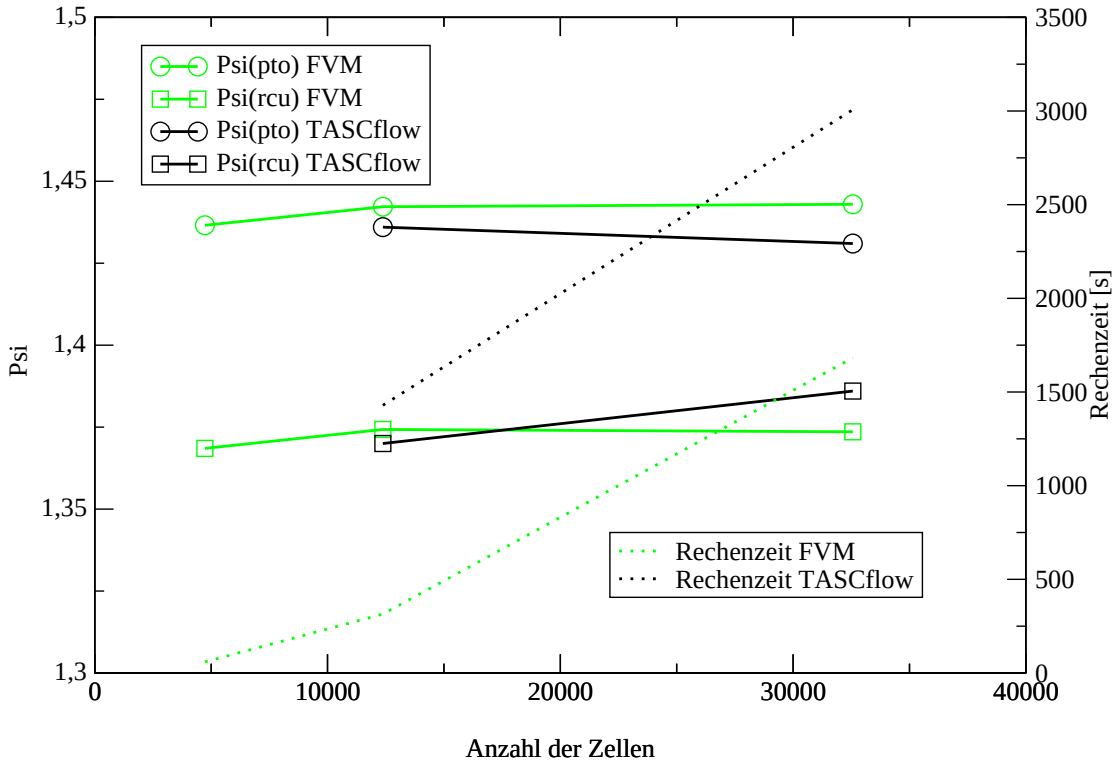

Abbildung 7.39: Einfluß der Anzahl der Zellen auf die Druckzahlen und die Rechenzeit für FT50

Der Einfluß der Gitterfeinheit ist sehr klein, so daß im Optimierungskontext recht grobe Gitter eingesetzt werden könnten. Dies gilt auch für die Druckverteilungen

auf der Schaufel und die Drallverteilungen im Nachlauf, siehe Abbildungen 7.40 und 7.41. Die Abweichungen zum TASCflow-Ergebnis sind zum Teil von der molekularen Viskosität des Euler-Verfahrens abhängig, siehe das folgende Kapitel 7.9.2.

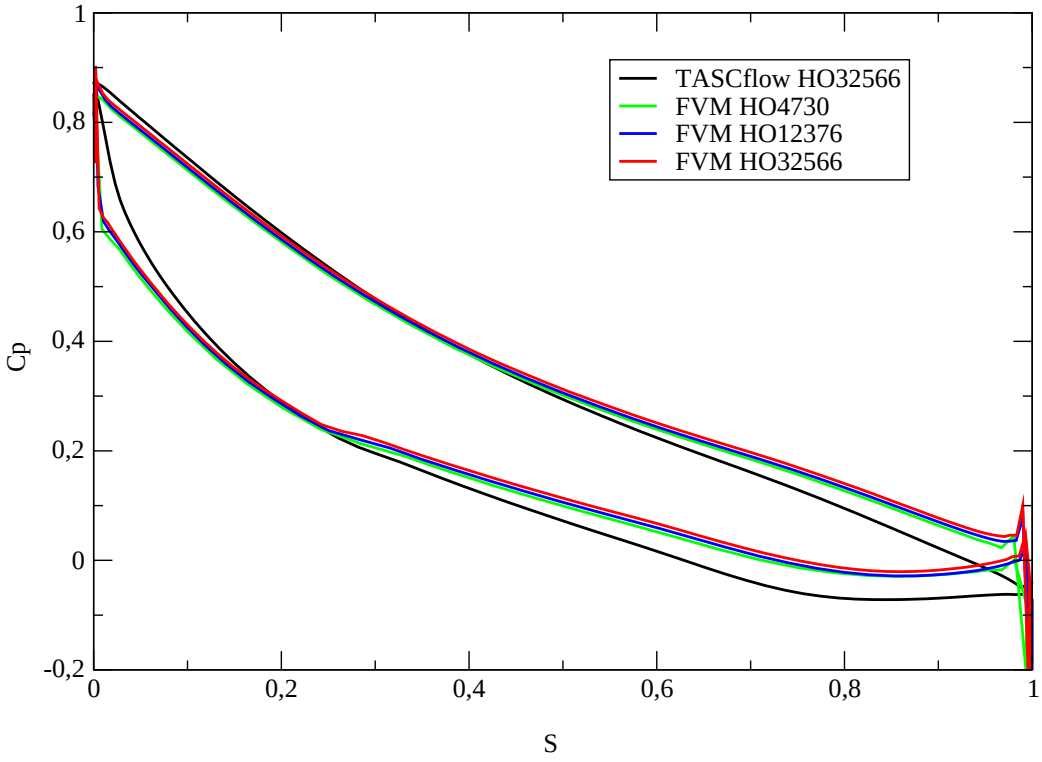

Abbildung 7.40: Vergleich von Druckverteilungen in der Schaufelmitte bei FT50 für verschiedene Gitter

7.9.2 Einfluß der Re-Zahl

Der Einfluß der molekularen Viskosität auf die Integralwerte ist in Tabelle 7.11 und Abbildung 7.42 dargestellt.

Re-Zahl	Ψ_t	$\Psi_{t,th}$	η_h	CPU-Zeit [s]
500	1.490	1.333	0.894	421
1000	1.457	1.359	0.933	380
1500	1.446	1.370	0.948	318
1792	1.442	1.374	0.953	299
2000	1.440	1.376	0.956	282
3000	1.435	1.383	0.964	249
3500	1.434	1.385	0.966	251

Tabelle 7.11: Abhängigkeit der Integralwerte von der Viskosität für FT50

Die reale Re-Zahl der Turbine beträgt $Re = 7.17 * 10^6$. Für $Re \geq 4000$ divergierte das Verfahren. Die Re-Zahl mit der besten Konvergenz, $Re = 3000$, liegt also relativ nah an der Stabilitätsgrenze. Ähnlich wie beim FT30-Testfall, verschlechtert sich die Konvergenzrate bei zu großer Viskosität.

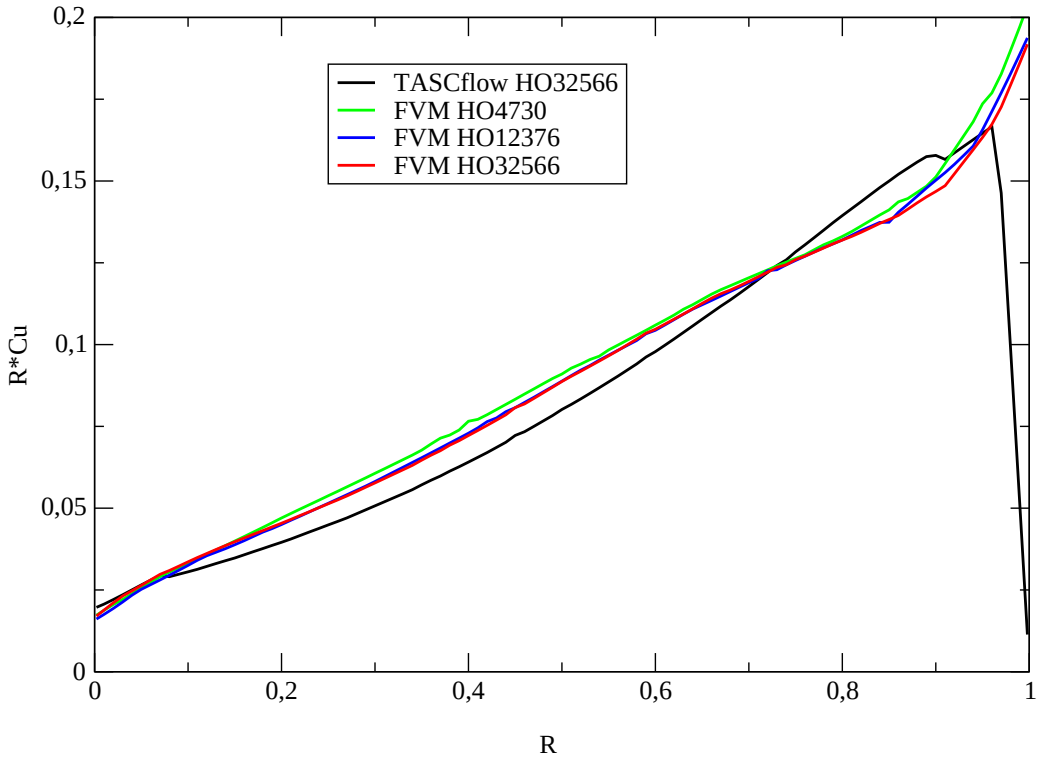

Abbildung 7.41: Vergleich von Drallverteilungen hinter der Schaufel bei FT50 für verschiedene Gitter

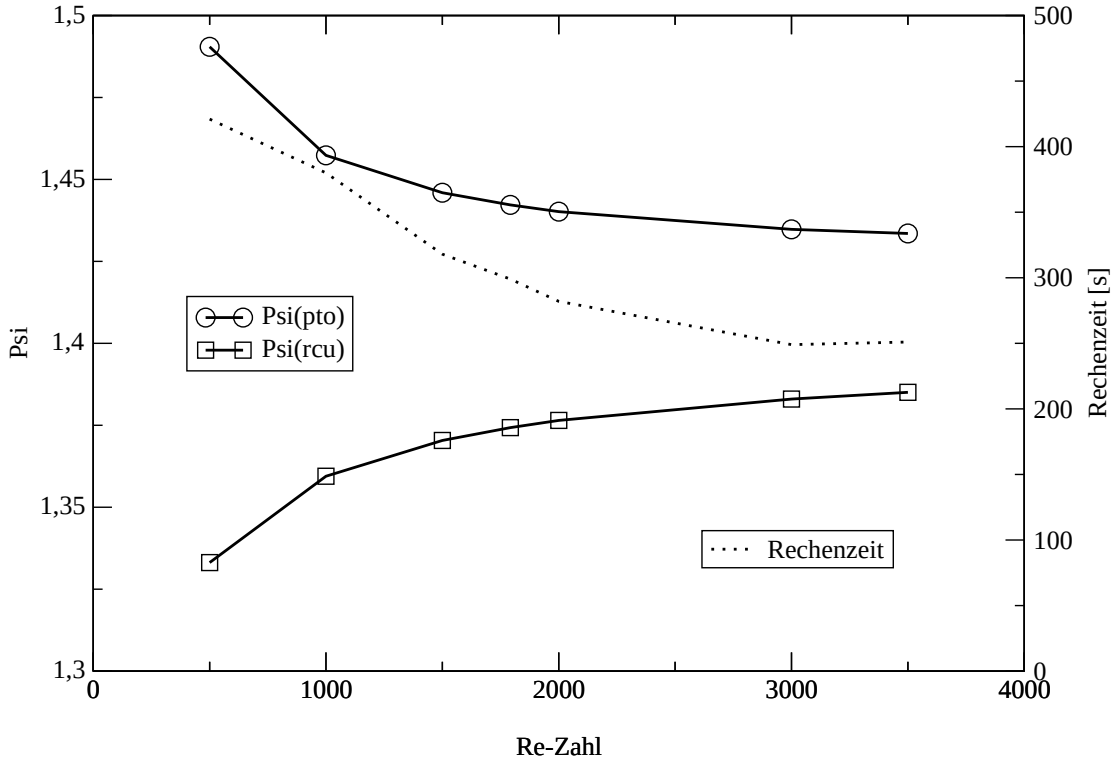

Abbildung 7.42: Einfluß der Viskosität für FT50 auf Druckzahlen und Rechenzeit, HO12376 Gitter

Die Abbildungen 7.43 und 7.44 zeigen den Einfluß der Viskosität auf die Druckverteilung auf der Schaufel und die Drallverteilung im Nachlauf. Die Übereinstimmung zwischen TASCflow und dem FV-Verfahren ist für die Druckverteilung nicht so gut wie beim FT30. Die Drallverteilung wird dagegen gut approximiert.

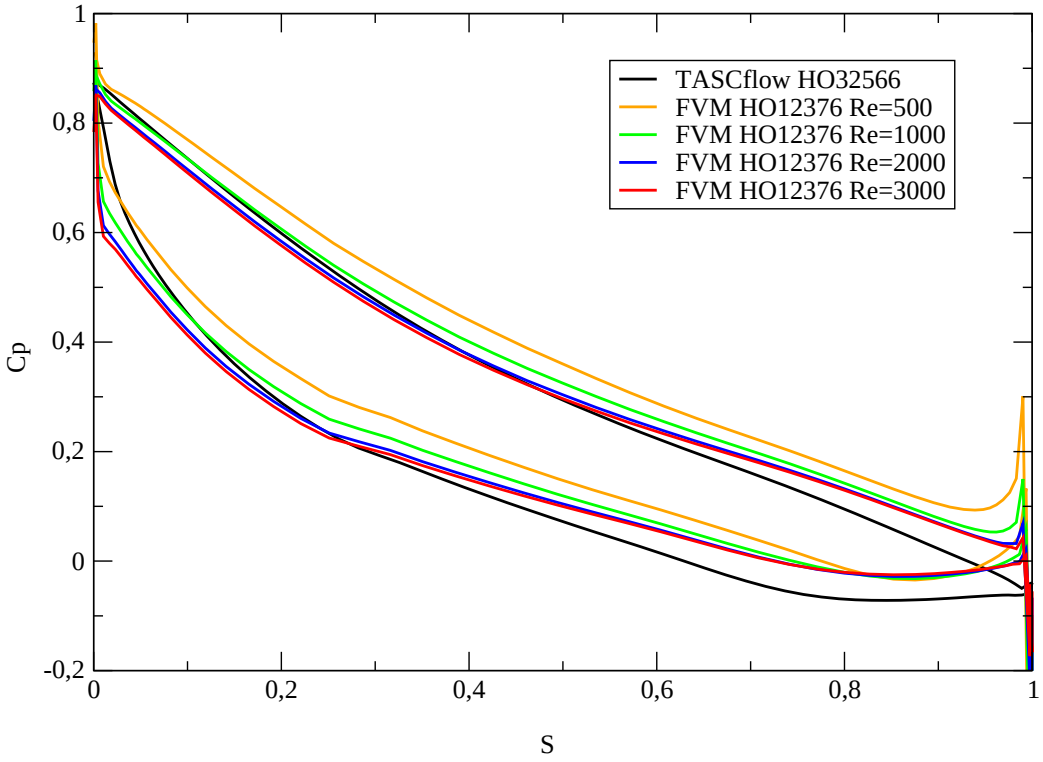

Abbildung 7.43: Einfluß der Viskosität auf die Druckverteilung für FT50, Schaufelmitte

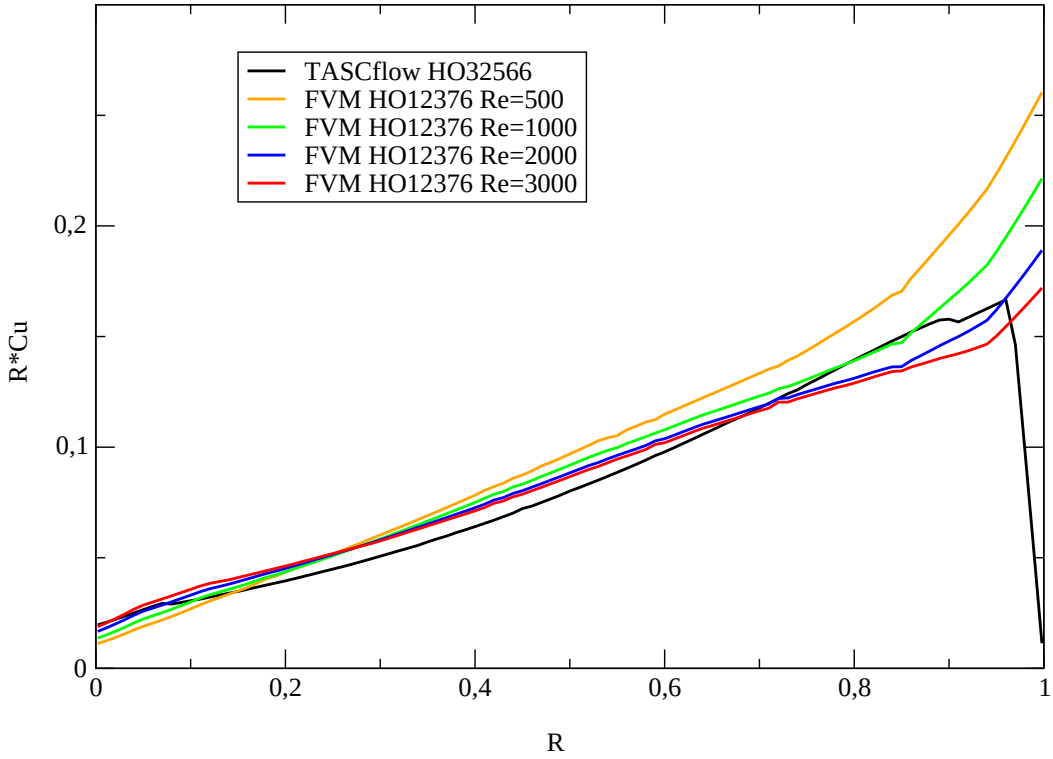

Abbildung 7.44: Einfluß der Viskosität auf die Drallverteilung hinter der Schaufel für FT50

7.10 Francis-Turbine NQ100

Dieser Testfall ist ein Beispiel für eine Francis-Turbine mit hoher spezifischer Drehzahl $n_q = 100[1/min]$. Es wurden die gleichen Untersuchungen wie bei der FT50-Turbine durchgeführt.

7.10.1 Einfluß der Gitterfeinheit

Für diesen Testfall standen HO-Gitter mit 10620, 14131, 21870 und 54740 Zellen zur Verfügung. Mit Ausnahme des 10620-Zellen Gitters wurden auch die jeweiligen Grobgitter untersucht, so daß in Tabelle 7.12 und Abbildung 7.45 insgesamt Ergebnisse von sieben verschiedenen Gittern dargestellt sind. Die Euler-Rechnungen wurden für $Re = 2867$ durchgeführt.

Code	Anzahl der Zellen	Ψ_t	$\Psi_{t,th}$	η_h	CPU-Zeit [s]
FVM	2282	1.246	1.122	0.900	48
	3296	1.206	1.116	0.925	83
	7540	1.190	1.101	0.925	229
	10620	1.192	1.106	0.927	334
	14131	1.186	1.102	0.929	353
	21870	1.179	1.100	0.933	969
	54740	1.172	1.095	0.934	2688
TASCflow	10620	1.282	1.111	0.867	673
	14131	1.279	1.099	0.859	799
	21870	1.262	1.066	0.845	1142
	54740	1.218	1.097	0.900	2798

Tabelle 7.12: Abhängigkeit der Integralwerte von der Zellenanzahl für FT100

Die Euler-Ergebnisse sind bereits ab 7540 Zellen sehr stabil, während sich die Werte bei den TASCflow-Rechnungen auch bei hoher Zellenanzahl noch ändern. Im Optimierungskontext muß also beim Einsatz von TASCflow darauf geachtet werden, die Anzahl der Zellen während der Optimierung konstant zu halten. Die Abbildung 7.46 zeigt die Druckverteilung in der Schaufelmitte.

Die Druckverteilungen des Euler-Verfahrens zeigen eine nur geringe Abhängigkeit von der Gitterauflösung. Die Übereinstimmung mit dem TASCflow-Ergebnis ist, mit Ausnahme des vorderen Bereichs der Saugseite der Schaufel, sehr gut. Dieses Verhalten wurde auch schon beim FT30-Testfall beobachtet. Wiederum können, zumindest teilweise, die Unterschiede durch den Einfluß der Viskosität im Euler-Verfahren erklärt werden, siehe auch das folgende Kapitel 7.10.2. Abbildung 7.47 zeigt den Einfluß der Gitterfeinheit auf die Drallverteilung hinter der Schaufel, wobei ein eindeutiger Trend in diesem Fall nicht erkennbar ist. Jedenfalls scheint das HO3296-Gitter zu grob zu sein.

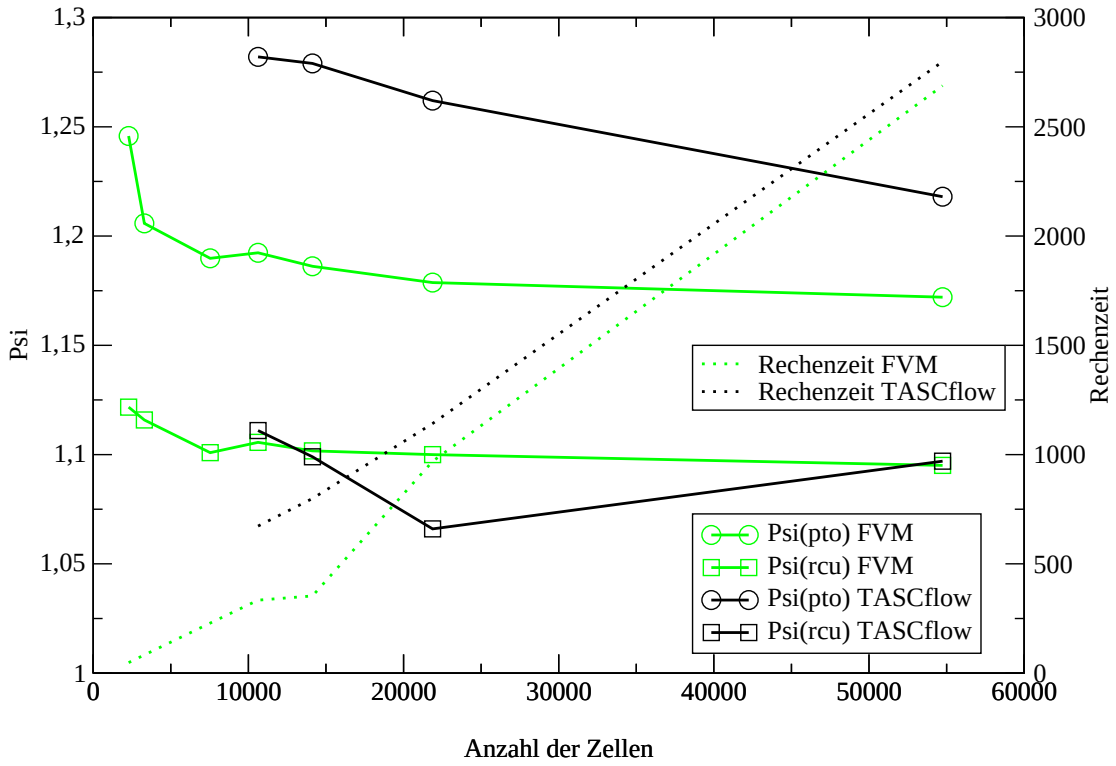

Abbildung 7.45: Einfluß der Anzahl der Zellen auf die Druckzahlen und die Rechenzeit für FT100

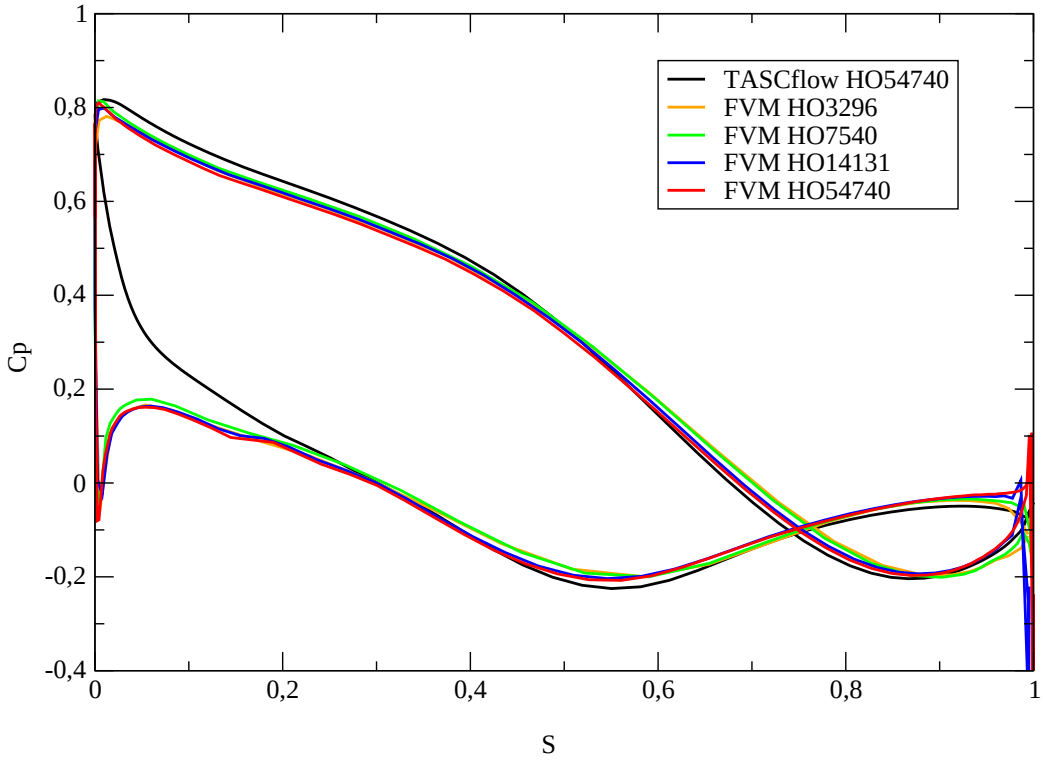

Abbildung 7.46: Vergleich von Druckverteilungen in der Schaufelmitte bei FT100 für verschieden feine HO-Netze

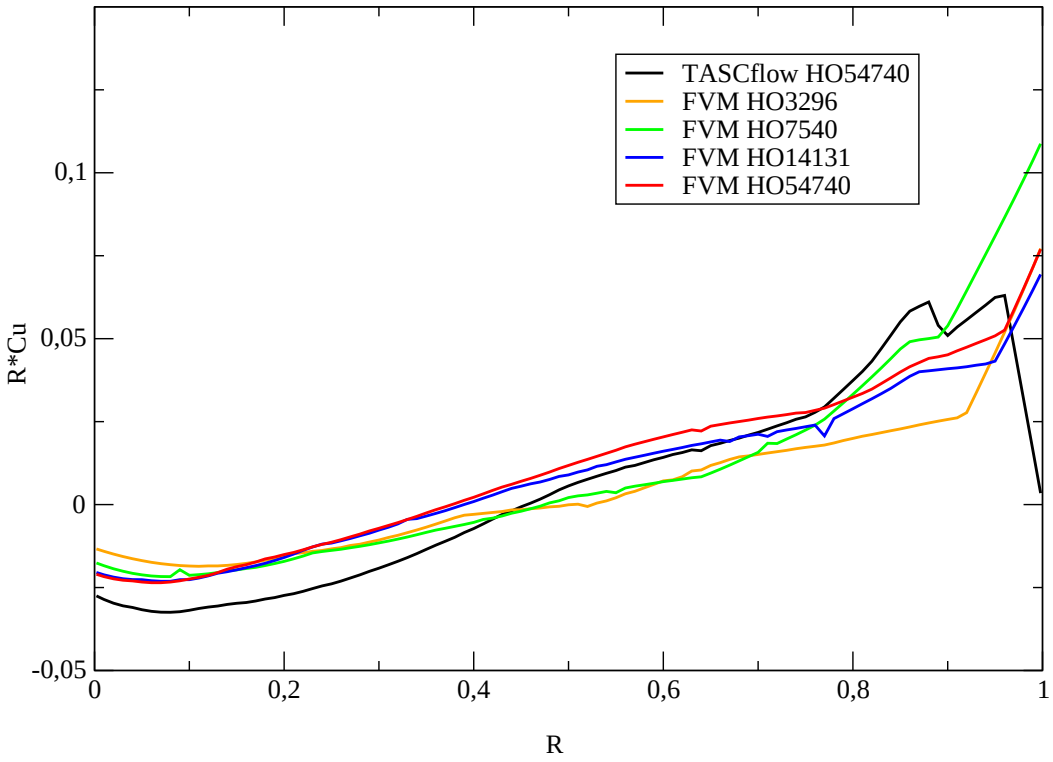

Abbildung 7.47: Vergleich der Drallverteilungen hinter der Schaufel bei FT100 für verschieden feine HO-Netze

7.10.2 Einfluß der Viskosität

Der Einfluß der (laminaren) Viskosität auf die Integralwerte ist in Tabelle 7.13 und Abbildung 7.48 dargestellt.

Re-Zahl	Ψ_t	$\Psi_{t,th}$	η_h	CPU-Zeit [s]
1000	1.209	1.091	0.902	350
1500	1.200	1.098	0.915	270
2000	1.198	1.100	0.918	229
2867	1.190	1.101	0.925	229
3000	1.188	1.100	0.926	222
6000	1.178	1.100	0.934	221
9000	1.175	1.100	0.936	212

Tabelle 7.13: Abhängigkeit der Integralwerte von der Re-Zahl für FT100, HO7540 Gitter

Die reale Re-Zahl der Turbine beträgt $Re = 7.17 * 10^6$. Für $Re \geq 9000$ divergierte das Verfahren. Die Re-Zahl mit der besten Konvergenz, $Re = 9000$, liegt also an der Stabilitätsgrenze. Allerdings sind die Konvergenzraten im Bereich $2000 \leq Re \leq 9000$ fast konstant, so daß $Re = 4000$ eine gute Wahl ist. Die Abbildung 7.49 zeigt den Einfluß der Re-Zahl auf die Druckverteilung auf der Schaufel.

Es zeigt sich, daß die Unterschiede zwischen TASCflow und FVM-Code im Bereich der Saugspitze mit kleiner werdender Re-Zahl kleiner werden. Dieses Verhalten wurde

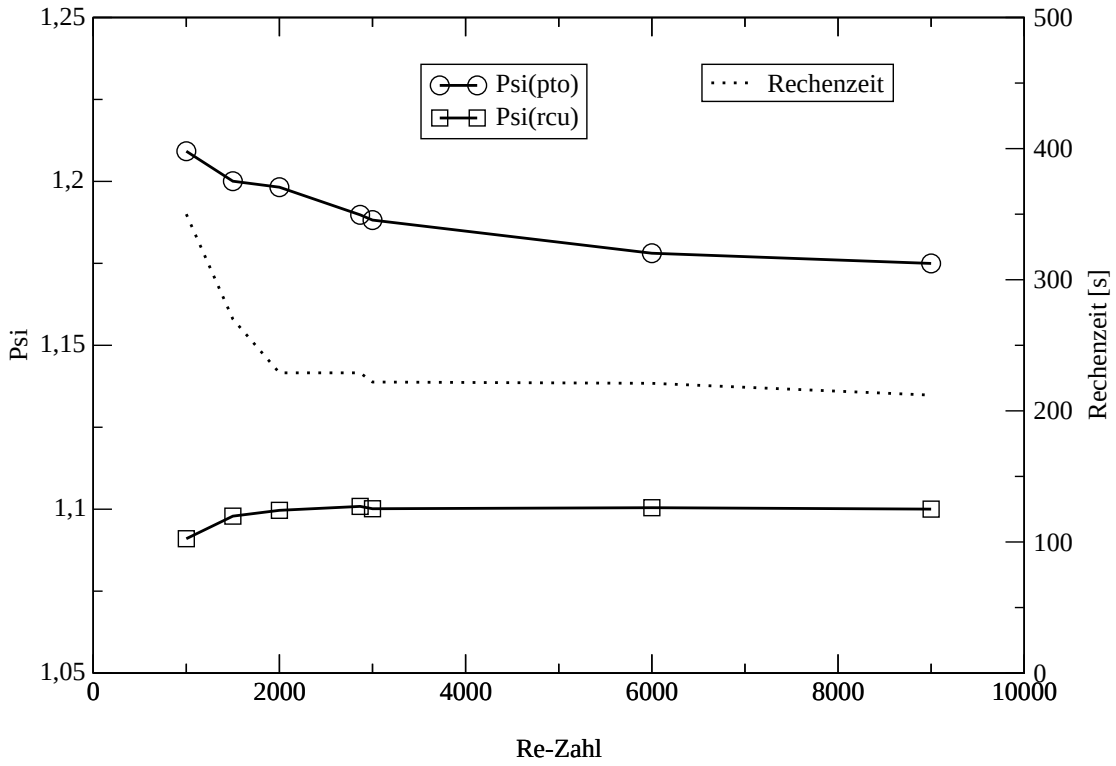

Abbildung 7.48: Einfluß der Re-Zahl auf die Druckzahlen und die Rechenzeit für FT100, HO7540 Gitter

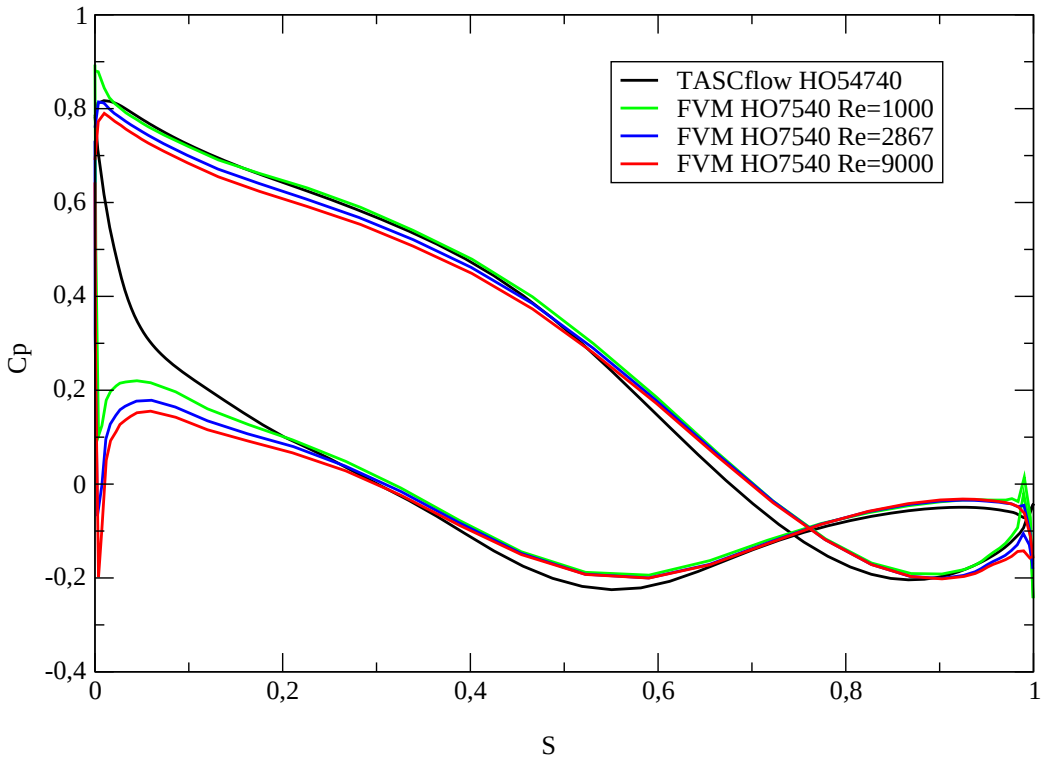

Abbildung 7.49: Einfluß der Re-Zahl auf die Druckverteilung für FT100, Schaufelmitte

auch beim FT30-Testfall beobachtet.

Abbildung 7.50 zeigt den Einfluß der Viskosität auf die Drallverteilung im Nachlauf. Wiederum ist die Übereinstimmung zwischen TASCflow- und Euler-Ergebnis am besten, wenn beim Euler-Verfahren eine kleine Re-Zahl eingestellt wird.

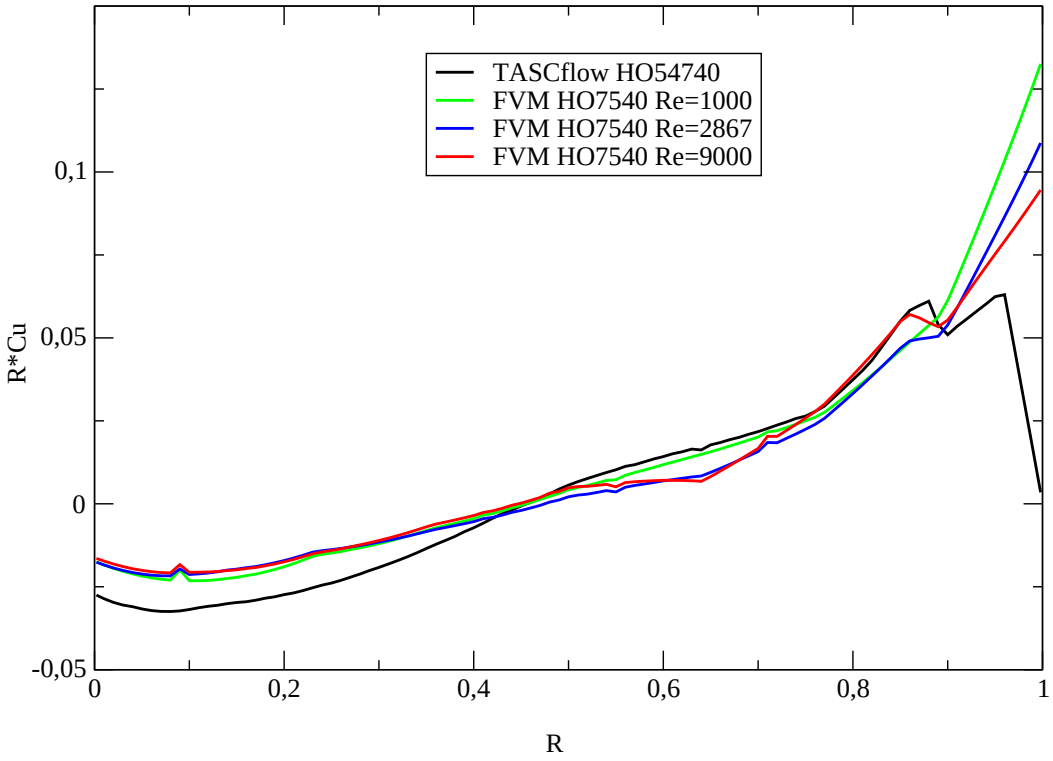

Abbildung 7.50: Einfluß der Re-Zahl auf die Drallverteilung hinter der Schaufel für FT100

7.11 Zusammenfassung der Ergebnisse der Testrechnungen

Bei den laminaren Testfällen für Kanalströmung und Driven Cavity wurden eine sehr gute Genauigkeit und optimale Konvergenzraten erzielt. Dies gilt auch für den Fall des Hybridgitters, siehe Kapitel 7.5.

Für die Testfälle mit logarithmisch-spiraligen Schaufeln und den Gostelow-Testfall, für die exakte Potentiallösungen vorliegen, wurden ebenfalls sehr gute Ergebnisse erzielt. Wie theoretisch zu erwarten, ist dabei die Übereinstimmung des FVM-Verfahrens mit den exakten Lösungen um so besser, je kleiner die Viskosität gewählt wird.

Bei den Testrechnungen für drei verschiedene Francis-Turbinen konnten folgende Rechenzeiten erzielt werden, siehe Tabelle 7.14.

Testfall	Anzahl der Zellen	Rechenzeit [s]	
		FVM	TASCflow
FT30	4730	73	-
	(geschätzt) 7000	(*) 106	-
	12376	186	639
	32566	425	(*) 1465
FT50	4730	59	-
	(geschätzt) 7000	(*) 135	-
	12376	314	1429
	32566	1682	(*) 3007
FT100	3296	83	-
	7540	(*) 229	-
	14131	353	799
	54740	2688	(*) 2798

Tabelle 7.14: Vergleich von Rechenzeiten für verschiedene Francis-Turbinen, (*) typische bzw. empfohlene Gitterauflösung

Im Vergleich mit den TASCflow-Ergebnissen konnten folgende Eigenschaften des Euler-Verfahrens festgestellt werden.

- Der Einfluß der Gitterauflösung ist beim FVM-Code gering. Für alle drei Testfälle ist eine Gitterauflösung von ca. 7000 Zellen ausreichend. Dabei ist zu beachten, daß die verwendeten HO-Gitter für Navier-Stokes Rechnungen optimiert waren. Für ein Euler-Verfahren müßte das Gitter z.B. an Nabe und Deckscheibe nicht verdichtet werden.

- Der Einfluß der Viskosität auf die Euler-Ergebnisse ist relativ groß. Unterschiede zu den TASCflow-Ergebnissen können zum Teil durch eine größere Viskosität im Euler-Code vermindert werden, wobei sich aber bei sehr kleiner Re-Zahl das Konvergenzverhalten verschlechtert.

- Die Integralwerte stimmen in der Regel recht gut mit den TASCflow-Werten überein.
- Wird die Feingitterstartlösung aus der Prolongation einer Grobgitterlösung berechnet, so genügt es, das Feingitterresiduum um 2 Größenordnungen zu reduzieren.
- Der optimale Wert des Blending-Faktors γ der Diskretisierung der konvektiven Terme liegt im Bereich von $0.7 \leq \gamma \leq 0.8$.
- Die Konvergenzraten des Mehrgitterverfahrens sind in der Regel schlechter als bei den laminaren Testfällen von Kapitel 7.1 bis 7.5.

Ein Einsatz des vorliegenden FVM-Codes als Euler-Verfahren muß also je nach Einsatzgebiet beurteilt werden. Für Nachrechnungsaufgaben sind die Rechenzeiten von Navier-Stokes Verfahren auf modernen PC's kein Problem mehr.

Im Optimierungskontext kann dagegen die Eigenschaft des Euler-Verfahrens, auf sehr groben Gittern rechnen zu können, zu erheblich schnelleren Zielfunktionsauswertungen führen. Dabei muß allerdings eine problemangepaßte Kalibrierung mittels eines Navier-Stokes Verfahrens erfolgen.

Der Beweis, daß Unterschiede zu TASCflow-Ergebnissen auf die unterschiedliche physikalische Modellierung zurückzuführen sind, kann erst erbracht werden, wenn in den FVM-Code ein $k\varepsilon$-Modell implementiert wird.

Kapitel 8

Bewertung und Ausblick

Im Rahmen dieser Arbeit wurde ein Navier-Stokes Verfahren mit FV-Diskretisierung für unstrukturierte Gitter entwickelt. Primär soll das Programm als Euler-Code in einem Optimierungssystem eingesetzt werden.

Aufbauend auf dem objektorientierten Framework von Müller [34] konnten wesentliche Bausteine/Module, z.B. Gitterverwaltung, lineare Löser, wiederverwendet werden. Die mit der Diskretisierung und dem nichtlinearen Lösungsverfahren (SIMPLE-Algorithmus) zusammenhängenden Bausteine mußten dagegen neu programmiert werden.

Durch die erheblich bessere Genauigkeit der FVM-Diskretisierung im Vergleich zur FEM-Diskretisierung von Turek/Rannacher [39] können verläßliche Lösungen auf relativ groben Gittern erzielt werden, was zu sehr kurzen Rechenzeiten führt. Es wurden detaillierte Vergleiche von Navier-Stokes und Euler Lösungen durchgeführt. Es zeigte sich, daß charakteristische Unterschiede bestehen, die durch Gitterverfeinerung nicht verschwinden. Wie sich diese Unterschiede im Optimierungskontext bei entsprechender Kalibrierung auswirken, muß noch untersucht werden.

Die gegenwärtig verwendeten blockstrukturierten HO-Gitter nutzen das Potential der unstrukturierten Gitterbehandlung im Code nicht aus. Hier könnte die Verwendung von Hybridgittern zu sehr robusten Gittergeneratoren führen.

Besonderer Wert wurde bei der (objektorientierten) Programmierung auf die spätere Erweiterbarkeit bzw. Modifizierbarkeit gelegt. Gegenwärtig wird das Programm am Lehrstuhl für Hydraulische Maschinen der TU München um ein Standard $k\varepsilon$-Modell erweitert. Der Aufwand dafür wird auf etwa 1 bis 2 Mannmonate geschätzt. Dabei ist aber noch nicht die Anpassung des Mehrgitterverfahrens berücksichtigt, da die übliche Verwendung von Wandfunktionen eine Anpassung erforderlich macht, für die noch theoretische Arbeit erforderlich ist. Eine Erweiterung für instationäre Strömungen, kompressible Strömungen und adaptive Gitter sollte dagegen keine grundsätzlichen Schwierigkeiten bereiten. Ebenso ist eine Erweiterung für Stufenrechnungen anzustreben.

Bei der Parallelisierung könnte ausgenutzt werden, daß die Aufteilung des Gebietes (Domain Decomposition) sich nicht an den Blöcken eines blockstrukturierten Gitters orientieren muß. Statt dessen kann die Aufteilung entsprechend der Anzahl der Prozessoren, gewichtet mit deren relativer Leistungsfähigkeit, erfolgen. Entsprechende

Programme, die die Aufteilung automatisieren und dabei die Anzahl der Interfaces zwischen den Gebieten minimieren, sind frei verfügbar, siehe z.B. *Jostle*[1] von WALSHAW ET AL. [64] oder *Metis*[2] von KARYPIS ET AL. [26].

Desweiteren könnte die Sensitivitätsanalyse von SZILAGYI [54] an die FV-Diskretisierung angepaßt werden. Eine Festigkeitsrechnung als zweite Komponente zur Untersuchung von Fluid-Struktur-Wechselwirkungen könnte analog zur FEM-Diskretisierung des Strömungslösers erfolgen.

Zusammenfassend läßt sich sagen, daß der Code bereits in seiner jetzigen Form ein sehr effizientes und flexibles Werkzeug zur Berechnung von Euler-Strömungen ist. Gleichermaßen wichtig ist aber auch, daß der Code die gemeinsame Basis für die unterschiedlichsten Erweiterungen sein könnte.

[1] http://www.gre.ac.uk/jostle
[2] http://www-users.cs.umn.edu/ karypis/metis/

Anhang A

Transformation in das rotierende System

Die Kontinuitätsgleichung lautet

$$\nabla \underline{c} = 0 \tag{A.1}$$

Die Impulsgleichungen mit Reibung lauten

$$\frac{D\underline{c}}{Dt} + \nabla p = \nu(\nabla \cdot \nabla)\underline{c}, \tag{A.2}$$

wobei $p = P/\rho$. Die Koordinaten im festem Koordinatensystem $\underline{x}$ und die Koordinaten im rotierenden System $\underline{\tilde{x}}$ sind über eine Transformationsmatrix $\underline{\underline{Q}}$ verknüpft.

$$\underline{x} = \underline{\underline{Q}}\,\underline{\tilde{x}} \tag{A.3}$$

Wenn die z-Achse die Drehachse ist, so ist die Transformationsmatrix gegeben durch

$$\underline{\underline{Q}} = \begin{pmatrix} \cos\omega t & -\sin\omega t & 0 \\ \sin\omega t & \cos\omega t & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{A.4}$$

$\underline{\underline{Q}}$ ist eine orthogonale Matrix, da $\underline{\underline{Q}}$ eine orthogonale Transformation beschreibt. Es gilt für orthogonale Matrizen

$$\underline{\underline{Q}}^T = \underline{\underline{Q}}^{-1} \tag{A.5}$$

d.h.

$$\underline{\underline{Q}}\,\underline{\underline{Q}}^T = \underline{\underline{I}} \tag{A.6}$$

In Indexschreibweise ist dies

$$Q_{i,j}Q_{i,k} = \delta_{j,k} \tag{A.7}$$

Für orthogonale Matrizen gilt außerdem noch folgender Zusammenhang

$$\left(\underline{\underline{Q}}\,\underline{a}\right) \cdot \left(\underline{\underline{Q}}\,\underline{b}\right) = \underline{a} \cdot \underline{b} \tag{A.8}$$

was sich wie folgt beweisen läßt:

$$\begin{aligned}
\left(\underline{\underline{Q}}\,\underline{a}\right)\cdot\left(\underline{\underline{Q}}\,\underline{b}\right) &= \sum_{i=1}^{n}\left(\sum_{j=1}^{n} Q_{i,j}a_j\right)\cdot\left(\sum_{k=1}^{n} Q_{i,k}b_k\right) && \text{(A.9)}\\
&= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} Q_{i,j}a_j Q_{i,k}b_k && \text{(A.10)}\\
&= \sum_{j=1}^{n}\sum_{k=1}^{n}\left(\sum_{i=1}^{n} Q_{i,j}Q_{i,k}\right)a_j b_k && \text{(A.11)}\\
&= \sum_{j=1}^{n}\sum_{k=1}^{n}\delta_{j,k}a_j b_k && \text{(A.12)}\\
&= \sum_{j=1}^{n} a_j b_j && \text{(A.13)}\\
&= \underline{a}\cdot\underline{b} && \text{(A.14)}
\end{aligned}$$

Deshalb gelten noch folgende Zusammenhänge

$$\underline{\tilde{x}} = \underline{\underline{Q}}^T\,\underline{x} \tag{A.15}$$

$$\underline{\tilde{c}}(\underline{\tilde{x}}) = \underline{\underline{Q}}^T\,\underline{c}(\underline{x},t) \tag{A.16}$$

$$\tilde{p}(\underline{\tilde{x}}) = p(\underline{x},t) \tag{A.17}$$

$$\underline{c}(\underline{x},t) = \underline{\underline{Q}}\,\underline{\tilde{c}}(\underline{\tilde{x}}) \tag{A.18}$$

Außerdem gilt

$$\nabla = \underline{\underline{Q}}\tilde{\nabla} \tag{A.19}$$

und

$$\tilde{\nabla} = \underline{\underline{Q}}^T\nabla \tag{A.20}$$

Mit diesen Vorarbeiten kann die Transformation der Kontinuitätsgleichung und der Impulsgleichungen wie folgt vorgenommen werden. Einsetzen von Gleichung (A.18), (A.19) und (A.17) in (A.2) ergibt

$$\frac{D}{Dt}\left(\underline{\underline{Q}}\,\underline{\tilde{c}}(\underline{\tilde{x}})\right) + \nabla\tilde{p} = \nu\left(\underline{\underline{Q}}\tilde{\nabla}\cdot\underline{\underline{Q}}\tilde{\nabla}\right)\underline{\underline{Q}}\underline{\tilde{c}} \tag{A.21}$$

Die rechte Seite läßt sich sofort mit Gleichung (A.8) vereinfachen zu

$$\frac{D}{Dt}\left(\underline{\underline{Q}}\,\underline{\tilde{c}}(\underline{\tilde{x}})\right) + \nabla\tilde{p} = \nu\left(\tilde{\nabla}\cdot\tilde{\nabla}\right)\underline{\underline{Q}}\underline{\tilde{c}} \tag{A.22}$$

Mit der Produktregel erhält man

$$\frac{D}{Dt}\left(\underline{\underline{Q}}\right)\underline{\tilde{c}}(\underline{\tilde{x}}) + \underline{\underline{Q}}\,\frac{D}{Dt}\left(\underline{\tilde{c}}(\underline{\tilde{x}})\right) + \nabla\tilde{p} = \nu\left(\tilde{\nabla}\cdot\tilde{\nabla}\right)\underline{\underline{Q}}\underline{\tilde{c}} \tag{A.23}$$

Die Matrix $\underline{\underline{Q}}$ hängt nur von der Zeit ab. Also gilt

$$\frac{D}{Dt}\left(\underline{\underline{Q}}\right) = \underline{\underline{\dot{Q}}} \tag{A.24}$$

Der Operator D/Dt ist definiert als

$$\frac{D\phi}{Dt} = \frac{\partial \phi}{\partial t} + \left(\frac{\partial \underline{x}}{\partial t} \cdot \nabla \right) \phi = \frac{\partial \phi}{\partial t} + \left(\frac{\partial \tilde{\underline{x}}}{\partial t} \cdot \tilde{\nabla} \right) \phi \tag{A.25}$$

Der Term $\frac{\partial \tilde{\underline{x}}}{\partial t}$ ist nicht etwa gleich $\tilde{\underline{c}}$, sondern ergibt sich aus

$$\frac{\partial \tilde{\underline{x}}}{\partial t} = \frac{\partial \underline{\underline{Q}}^T \cdot \underline{x}}{\partial t} = \dot{\underline{\underline{Q}}}^T \cdot \underline{x} + \underline{\underline{Q}}^T \cdot \dot{\underline{x}} \tag{A.26}$$

Man zeigt leicht, daß

$$\dot{\underline{\underline{Q}}}^T \cdot \underline{x} = -\underline{\omega} \times \tilde{\underline{x}} \tag{A.27}$$

Außerdem ist die Relativgeschwindigkeit definiert als

$$\tilde{\underline{w}} = \tilde{\underline{c}} - \underline{\omega} \times \tilde{\underline{x}} \tag{A.28}$$

Damit folgt für Gleichung (A.25)

$$\frac{D\phi}{Dt} = \frac{\partial \phi}{\partial t} + \left(\tilde{\underline{w}} \cdot \tilde{\nabla} \right) \phi \tag{A.29}$$

Also folgt für Gleichung (A.23)

$$\dot{\underline{\underline{Q}}}\, \tilde{\underline{c}}(\tilde{\underline{x}}) + \underline{\underline{Q}} \cdot \left(\frac{\partial \tilde{\underline{c}}(\tilde{\underline{x}})}{\partial t} + \left(\tilde{\underline{w}} \cdot \tilde{\nabla} \right) \tilde{\underline{c}}(\tilde{\underline{x}}) \right) + \nabla \tilde{p} = \nu \left(\tilde{\nabla} \cdot \tilde{\nabla} \right) \underline{\underline{Q}} \tilde{\underline{c}} \tag{A.30}$$

Ist die Strömung im rotierenden System stationär, was im Rahmen dieser Arbeit immer der Fall ist, so entfällt der instationäre Term und man erhält

$$\dot{\underline{\underline{Q}}}\, \tilde{\underline{c}}(\tilde{\underline{x}}) + \underline{\underline{Q}} \cdot \left(\left(\tilde{\underline{w}} \cdot \tilde{\nabla} \right) \tilde{\underline{c}}(\tilde{\underline{x}}) \right) + \nabla \tilde{p} = \nu \left(\tilde{\nabla} \cdot \tilde{\nabla} \right) \underline{\underline{Q}} \tilde{\underline{c}} \tag{A.31}$$

Nach Multiplikation von links mit $\underline{\underline{Q}}^T$ ergibt sich mit

$$\underline{\underline{Q}}^T\, \dot{\underline{\underline{Q}}}\, \tilde{\underline{c}}(\tilde{\underline{x}}) = \underline{\omega} \times \tilde{\underline{c}} \tag{A.32}$$

folgende endgültige Form der Impulsgleichung:

$$\underline{\omega} \times \tilde{\underline{c}} + \left(\tilde{\underline{w}} \cdot \tilde{\nabla} \right) \tilde{\underline{c}} + \tilde{\nabla} \tilde{p} = \nu \left(\tilde{\nabla} \cdot \tilde{\nabla} \right) \tilde{\underline{c}} \tag{A.33}$$

Diese Form der Impulsgleichung soll nach Kroll [28] als Kroll-Formulierung bezeichnet werden. Auf dieser Gleichung setzt auch das numerische Verfahren auf. Eine alternative Formulierung erhält man, wenn man die Zusammenhänge

$$\underline{\omega} \times \tilde{\underline{c}} = \underline{\omega} \times \tilde{\underline{w}} + \underline{\omega} \times \tilde{\underline{u}} = \underline{\omega} \times \tilde{\underline{w}} + \underline{\omega} \times (\underline{\omega} \times \tilde{\underline{x}}) \tag{A.34}$$

und

$$\begin{aligned} \left(\tilde{\underline{w}} \cdot \tilde{\nabla} \right) \tilde{\underline{c}} &= \left(\tilde{\underline{w}} \cdot \tilde{\nabla} \right) \tilde{\underline{w}} + \left(\tilde{\underline{w}} \cdot \tilde{\nabla} \right) \tilde{\underline{u}} && \text{(A.35)} \\ &= \left(\tilde{\underline{w}} \cdot \tilde{\nabla} \right) \tilde{\underline{w}} + \left(\tilde{\underline{w}} \cdot \tilde{\nabla} \right) (\underline{\omega} \times \tilde{\underline{x}}) && \text{(A.36)} \\ &= \left(\tilde{\underline{w}} \cdot \tilde{\nabla} \right) \tilde{\underline{w}} + (\underline{\omega} \times \tilde{\underline{w}}) && \text{(A.37)} \end{aligned}$$

einsetzt in Gleichung (A.33).

$$\left(\tilde{\underline{w}} \cdot \tilde{\nabla} \right) \tilde{\underline{w}} + \underline{\omega} \times (\underline{\omega} \times \tilde{\underline{x}}) + 2(\underline{\omega} \times \tilde{\underline{w}}) + \tilde{\nabla} \tilde{p} = \nu \left(\tilde{\nabla} \cdot \tilde{\nabla} \right) \tilde{\underline{c}} \tag{A.38}$$

bzw.

$$\left(\tilde{\underline{w}} \cdot \tilde{\nabla} \right) \tilde{\underline{w}} + \underline{\omega} \times (\underline{\omega} \times \tilde{\underline{x}}) + 2(\underline{\omega} \times \tilde{\underline{w}}) + \tilde{\nabla} \tilde{p} = \nu \left(\tilde{\nabla} \cdot \tilde{\nabla} \right) \tilde{\underline{w}} \tag{A.39}$$

Diese Gleichung wird z.B. bei Fister [14] angegeben. Der Term $\underline{\omega} \times (\underline{\omega} \times \tilde{\underline{x}})$ ist die Zentrifugalbeschleunigung und der Term $2(\underline{\omega} \times \tilde{\underline{w}})$ ist die Coriolisbeschleunigung.

Anhang B

Periodische Randbedingungen

Die periodischen Randbedingungen für rotationssymmetrische Ränder können wie folgt hergeleitet werden. Sei γ der Teilungswinkel und $\underline{\hat{x}}$ der Ortsvektor im gedrehten Koordinatensystem. Die Transformationsmatrix ist gegeben als

$$\underline{\underline{Q}} = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{B.1}$$

Wiederum ist $\underline{\underline{Q}}$ eine orthogonale Matrix, siehe Kapitel A. Dann gilt

$$\underline{x} = \underline{\underline{Q}}\,\underline{\hat{x}} \tag{B.2}$$

$$\underline{c} = \underline{\underline{Q}}\,\underline{\hat{c}} \tag{B.3}$$

$$\nabla = \underline{\underline{Q}}\,\hat{\nabla} \tag{B.4}$$

Der Druck muß als skalare Größe nicht transformiert werden.

$$p_S = p_P \tag{B.5}$$

P und S sind die periodischen Punkte. P um den Winkel γ gedreht ergibt den Punkt S. Der Druckgradient muß allerdings transformiert werden. Die Periodizitätsbedingung lautet

$$(\hat{\nabla} p)_S = (\nabla p)_P \tag{B.6}$$

Gesucht ist $(\nabla p)_S$, was sich ergibt aus

$$(\nabla p)_S = \underline{\underline{Q}}(\hat{\nabla} p)_S = \underline{\underline{Q}}(\nabla p)_P \tag{B.7}$$

Für die Geschwindigkeiten gilt

$$\underline{\hat{c}}_S = \underline{c}_P \tag{B.8}$$

Daraus folgt

$$\underline{c}_S = \underline{\underline{Q}}\underline{\hat{c}}_S = \underline{\underline{Q}}\underline{c}_P \tag{B.9}$$

Für die Geschwindigkeitsgradienten gilt

$$(\hat{\nabla}\underline{\hat{c}}^T)_S = (\nabla \underline{c}^T)_P \tag{B.10}$$

Daraus folgt

$$(\nabla \underline{c}^T)_S = \left(\underline{\underline{Q}}\hat{\nabla}(\underline{\underline{Q}}\hat{\underline{c}})^T\right)_S = \underline{\underline{Q}}(\hat{\nabla}\hat{\underline{c}}^T)_S\underline{\underline{Q}}^T = \underline{\underline{Q}}(\nabla \underline{c}^T)_P\underline{\underline{Q}}^T \tag{B.11}$$

In Indexschreibweise ist dies

$$\frac{\partial c_j}{\partial x_i} = \sum_k \sum_l Q_{ik} Q_{jl} \frac{\partial c_l}{\partial x_k} \tag{B.12}$$

bzw.

$$\frac{\partial c_j}{\partial x_i} = \sum Q_{ik} \sum_l Q_{jl} \frac{\partial c_l}{\partial x_k} \tag{B.13}$$

Diese Gleichung ist ausgeschrieben auch bei RITZINGER [43] angegeben. Berücksichtigt man, daß die Matrix $\underline{\underline{Q}}$ viermal die Null und einmal die Eins als Eintrag hat, so können einige überflüssige Multiplikationen und Additionen eingespart werden.

$$\left(\frac{\partial c_x}{\partial x}\right)_S = C2\left(\frac{\partial c_x}{\partial x}\right)_P - CS\left(\left(\frac{\partial c_y}{\partial x}\right)_P + \left(\frac{\partial c_x}{\partial y}\right)_P\right) + S2\left(\frac{\partial c_y}{\partial y}\right)_P \tag{B.14}$$

$$\left(\frac{\partial c_x}{\partial y}\right)_S = C2\left(\frac{\partial c_x}{\partial y}\right)_P + CS\left(\left(\frac{\partial c_x}{\partial x}\right)_P - \left(\frac{\partial c_y}{\partial y}\right)_P\right) - S2\left(\frac{\partial c_y}{\partial x}\right)_P \tag{B.15}$$

$$\left(\frac{\partial c_x}{\partial z}\right)_S = C\left(\frac{\partial c_x}{\partial z}\right)_P - S\left(\frac{\partial c_y}{\partial z}\right)_P \tag{B.16}$$

$$\left(\frac{\partial c_y}{\partial x}\right)_S = C2\left(\frac{\partial c_y}{\partial x}\right)_P + CS\left(\left(\frac{\partial c_x}{\partial x}\right)_P - \left(\frac{\partial c_y}{\partial y}\right)_P\right) - S2\left(\frac{\partial c_x}{\partial y}\right)_P \tag{B.17}$$

$$\left(\frac{\partial c_y}{\partial y}\right)_S = C2\left(\frac{\partial c_y}{\partial y}\right)_P + CS\left(\left(\frac{\partial c_y}{\partial x}\right)_P + \left(\frac{\partial c_x}{\partial y}\right)_P\right) + S2\left(\frac{\partial c_x}{\partial x}\right)_P \tag{B.18}$$

$$\left(\frac{\partial c_y}{\partial z}\right)_S = C\left(\frac{\partial c_y}{\partial z}\right)_P + S\left(\frac{\partial c_x}{\partial z}\right)_P \tag{B.19}$$

$$\left(\frac{\partial c_z}{\partial x}\right)_S = C\left(\frac{\partial c_z}{\partial x}\right)_P - S\left(\frac{\partial c_z}{\partial y}\right)_P \tag{B.20}$$

$$\left(\frac{\partial c_z}{\partial y}\right)_S = C\left(\frac{\partial c_z}{\partial y}\right)_P + S\left(\frac{\partial c_z}{\partial x}\right)_P \tag{B.21}$$

$$\left(\frac{\partial c_z}{\partial z}\right)_S = \left(\frac{\partial c_z}{\partial z}\right)_P \tag{B.22}$$

mit $C = \cos\gamma$; $S = \sin\gamma$, $C2 = C^2$, $S2 = S^2$ und $CS = C * S$. Gleichung (B.13) führt auf 108 Multiplikationen und 72 Additionen, was in 756 Takten abgearbeitet ist[1]. Die Gleichungen (B.14) bis (B.22) benötigen 20 Multiplikationen und 16 Additionen, also nur 148 Takte.

[1]Die modernen PC-Prozessoren brauchen 3 Takte für eine Addition bzw. Subtraktion von doppelt genauen Realzahlen und 5 Takte für eine Multiplikation. Eine Division benötigt 32 Takte und ein sin() 95 Takte.

Literaturverzeichnis

[1] A.S. Arcilla, J. Häuser, P.R. Eiseman, and J.F. Thompson, editors. *Numerical grid generation in computational fluid dynamics and related fields*, North-Holland, Amsterdam, 1991.

[2] Robert Bader. *Simulation kompressibler und inkompressibler Strömungen in Turbomachinen.* Dissertation, Lehrstuhl und Laboratorium für Hydraulische Maschinen und Anlagen, Technische Universität München, 2000.

[3] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods.* SIAM, 2. edition, 1994.

[4] P. Bastian, K. Birken, K. Johannsen, S. Lang, N. Neu, H. Rentz-Reichert, and C. Wieners. Ug - a flexible software toolbox for solving partial differential equations. *Computing and Visualization in Science*, 01(01):027–040, 1997.

[5] I. N. Bronstein, K. A. Semendjajew, G. Musiol, and H. Mühlig. *Taschenbuch der Mathematik.* Verlag Harri Deutsch, Thun, Frankfurt am Main, 1995.

[6] A. Busemann. Das Förderverhältnis radialer Kreiselpumpen mit logarithmisch-spiraligen Schaufeln. *Z. f. angew. Math. Mech.*, (8, 5), 1928.

[7] L.S. Caretto, A.D. Gosman, S.V. Patankar, and D.B. Spalding. Two calculation procedures for steady, three-dimensional flows with recirculation. In *Proc. Third Int. Conf. Numer. Methods Fluid Dyn.*, Paris, 1972.

[8] G.F. Carey. *Computational Grids: Generation, Adaptation, and Solution Strategies.* Taylor & Francis, 1997.

[9] A. J. Chorin. A numerical method for solving incompressible viscous flow problems. *J. Comp. Phys.*, 2:12–26, 1967.

[10] J. Douglas and H.H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Trans. Amer. Math. Soc.*, 82:421–439, 1956.

[11] A. Fernández. *Strömungstechnische Optimierung von Beschaufelungen hydraulischer Maschinen.* Dissertation, Lehrstuhl und Laboratorium für Hydraulische Maschinen und Anlagen, Technische Universität München, 1997.

[12] J. H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics.* Springer Verlag, Berlin Heidelberg, 1996.

[13] O. Filippova and D. Hänel. Computation of viscous flow by lattice-gas methods. In *Proc. of 6th Int. Symp. on Comp. Fluid Dynamics*, Lake Tahoe, USA, Sept. 3-6 1995.

[14] W. Fister. *Fluidenergiemaschinen: Physikalische Voraussetzungen*, volume 1. Springer, Berlin, 1984.

[15] Fluent Inc., Centerra Resource Park, 10 Cavendish Court, Lebanon, NH 03766, USA. *Fidap 8 Theory Manual*, December 1998.

[16] Fluent Inc., Centerra Resource Park, 10 Cavendish Court, Lebanon, NH 03766, USA. *Fluent 6 User's Guide Volume 5*, December 2001.

[17] P.H. Gaskell and A.K.C. Lau. Curvature compensated convective transport: Smart, a new boundedness preserving transport algorithm. *International Journal Numerical Methods in Fluids*, 8:617–641, 1988.

[18] O. Gloth, D. Hänel, and R. Vilsmeier. An object oriented framework for finite volume applications. In *Proceedings of the International Workshop on Modern Software Tools for Scientific Computing*, Oslo, 1998.

[19] J. P. Gostelow. Potential flow through cascades - a comparison between exact and approximate solutions. Technical Report CP No. 807, ARC, 1965.

[20] J. P. Gostelow. *Cascade Aerodynamics.* Pergamon Pr., 1. edition, 1984.

[21] F. H. Harlow and J. E. Welsh. Numerical calculation of the time dependent viscous incompressible flow with free surface. *Phys. Fluids*, 8:2182–2189, 1965.

[22] M.R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Nat. Bur. Standards, J. of Res.*, 49:409–436, 1952.

[23] C.W. Hirt, A.A. Amsden, and J.L. Cook. An arbitrary lagrangian–eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14:227–253, 1974.

[24] R.I. Issa. Solution of implicitly discretized fluid flow equations by operator-splitting. *Journal of Computational Physics*, 62:40–65, 1986.

[25] Y. Kallinderis, A. Khawaja, and H. McMorris. Hybrid prismatic/tetrahedral grid generation for complex geometries. AIAA-Paper 95-0211, Reno, NV, January 1995.

[26] G. Karypis and V. Kumar. A parallel algorithm for multilevel graph partitioning and sparse matrix ordering. *Journal of Parallel and Distributed Computing*, 48:71–85, 1998.

[27] P.K. Khosla and S.G. Rubin. A diagonal dominant second-order accurate implicit scheme. *Computers & Fluids*, 2:207–209, 1974.

[28] N. Kroll. Berechnung von Strömungsfeldern um Propeller und Rotoren im Schwebeflug durch die Lösung der Euler-Gleichungen. Forschungsbericht 89-37, Deutsche Forschungsanstalt für Luft- und Raumfahrt, Institut für Entwurfsaerodynamik, Braunschweig, Juni 1989.

[29] D. Kwak, J.L.C. Chang, S.P. S.P. Shanks, and S.R. Chakravarthy. A three-dimensional incompressible navier-stokes flow solver using primitive variables. *AIAA Journal*, 24:390–396, 1986.

[30] B.P. Leonard. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Comp. Meth. Appl. Mech. Eng.*, 19:59–98, 1987.

[31] Z. Lilek, S. Muzaferija, and M. Perić. Efficiency and accuracy aspects of a full-multigrid simple algorithm for three-dimensional flows. *Numerical Heat Transfer, Part B*, 31:23–42, 1997.

[32] M. M. Warfield and B. Lakshminarayana. Calculation of three dimensional locally elliptic flow with a zonal equation method. *AIAA Journal*, 29:684–685, 1987.

[33] J. Moran. *An Introduction to Theoretical and Computational Aerodynamics.* John Wiley & Sons, 1984.

[34] A. Müller. *Objektorientierte Strukturen für Adaptive Multilevelverfahren zur Strömungssimulation.* Dissertation, Lehrstuhl und Laboratorium für Hydraulische Maschinen und Anlagen, Technische Universität München, 2000.

[35] S. Muzaferija. *Adaptive finite volume method for flow predictions using unstructured meshes and multigrid approach.* Phd thesis, University of London, 1994.

[36] S. V. Patankar. *Numerical Heat Transfer and Fluid Flow.* Hemisphere Publishing Corporation, Washington, London, New York, 1. edition, 1980.

[37] D.W. Peaceman and H.H. Rachford. The numerical solution of parabolic and elliptic differential equations. *J. Soc. Ind. Appl. Math.*, 3:28–41, 1955.

[38] G.D. Raithby. Skew upstream differencing schemes for problems involving fluid flow. *Computer Methods in applied mechanics and engineering*, 9:75–103, 1976.

[39] R. Rannacher and S. Turek. A simple nonconforming quadrilateral stokes element. *Numer. Part. Diff. Equ.*, 8:97–111, 1992.

[40] C.M. Rhie and W.L. Chow. A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation. *AIAA Journal*, 21:1525–1532, 1983.

[41] Rainer Richter. *3D-Echtzeit-Entwurf von Beschaufelungen hydraulischer Strömungsmaschinen auf Multiprozessorsystemen.* Dissertation, Lehrstuhl und Laboratorium für Hydraulische Maschinen und Anlagen, Technische Universität München, 1999.

[42] Norbert Riedel. *Rotor-Stator Wechselwirkung in hydraulischen Maschinen.* Dissertation, Lehrstuhl und Laboratorium für Hydraulische Maschinen und Anlagen, Technische Universität München, 1997.

[43] Stephan Ritzinger. *Simulation realer Laufradströmungen.* Dissertation, Technische Universität München, Lehrstuhl und Laboratorium für Hydraulische Maschinen und Anlagen, 1997.

[44] Heinz Schade and Ewald Kunz. *Strömungslehre.* Walter de Gruyter, 1980.

[45] R. Schilling. Numerical calculation of the q3d incompressible, inviscid flow in turbomachines. In *Proceedings of the* 11^{th} *IAHR Symposium*, Amsterdam, 1982.

[46] R. Schilling. Cfd aided design of hydraulic machinery bladings. In B. Velensek, editor, *CFD'91 Intensive Course on Computational Fluid Dynamics*, Ljubljana, 1991.

[47] R. Schilling. Rechnergestützte Entwicklung von Strömungsmaschinen. Vorlesungsbegleitende Unterlagen, 1996.

[48] G. E. Schneider and M. Zedan. A modified strongly implicit procedure for the numerical solution of field problems. *Numer. Heat Transfer*, 4:1 – 19, 1981.

[49] Martin Schuster. *Simulation gehäuseloser, hydraulischer Strömungsmachinen.* Dissertation, Lehrstuhl und Laboratorium für Hydraulische Maschinen und Anlagen, Technische Universität München, 2000.

[50] R. Skoda. Private Mitteilungen. Lehrstuhl für Hydraulische Maschinen und Anlagen, TU München, 2001.

[51] Steger and Sorenson. Automatic mesh-point clustering near a boundary in grid generation with elliptic partial differential equations. *Journal of Computational Physics*, 33(3):405–410, 1979.

[52] H.L. Stone. Iterative solution of implicit approximations of multidimensional partial differential equations. *SIAM J. Numer. Anal.*, 5(3):530–558, 1968.

[53] B. Szilagyi. Private Mitteilungen. Lehrstuhl für Hydraulische Maschinen und Anlagen, TU München, 2001.

[54] B. Szilagyi. *Sensitivitätsanalyse.* Diss., Lehrstuhl für Hydraulische Maschinen und Anlagen, TU München, 2001.

[55] B. Szilagyi and A. Müller. Herleitung der Gleichungen für den Gostelow-Testfall. Private Mitteilungen, Lehrstuhl für Hydraulische Maschinen und Anlagen, TU München, 2000.

[56] TASCflow. *User Documentation.* ASC, Waterloo Ontario, Canada, Version 2.4 edition, 1995.

[57] J.F. Thompson, B.K. Soni, and N.P. Weatherill. *Handbook of Grid Generation.* CRC Press, London, 1999.

[58] J.F. Thompson, Z.U.A. Warsi, and C.W. Mastin. Numerical grid generation, foundations and applications, 1985.

[59] S. Thum, R. Reinelt, B. Szilagyi, and J. Einzinger. mb2fem - Konvertierung von blockstrukturierten Gittern in unstrukturierte Gitter. Interner Bericht, TU München, Lehrstuhl für Hydraulische Maschinen und Anlagen, 2001.

[60] S. Turek. *Efficient Solvers for Incompressible Flow Problems. An Algorithmic and Computational Approach.* Springer, 1999.

[61] H.A. Van den Vorst. Bi-cgstab: a fast and smoothly converging variant of bi-cg for the solution of non-symmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 13:631–644, 1992.

[62] J.P. Van Doormal and G.D. Raithby. Enhancements of the simple method for predicting incompressible fluid flows. *Journal of Numerical Heat Transfer*, 7:147–163, 1984.

[63] R.W.C.P. Verstappen and A.E.P. Veldman. Symmetry-preserving discretization of turbulent channel flow. In M. Breuer, F. Durst, and C. Zenger, editors, *High Performance Scientific and Engineering Computing, Proceedings of the 3rd Internatinal FORTWIHR Conference on HPSEC*, pages 107–114, Erlangen, Germany, March 12-14 2001.

[64] C. Walshaw, M. Cross, and M. G. Everett. Parallel Dynamic Graph Partitioning for Adaptive Unstructured Meshes. *J. Parallel Distrib. Comput.*, 47(2):102–108, 1997. (originally published as Univ. Greenwich Tech. Rep. 97/IM/20).

[65] Christian Watzelt. *Echtzeit-Entwurf radialer Beschaufelungen auf einem Multiprozessorsystem.* Dissertation, Lehrstuhl und Laboratorium für Hydraulische Maschinen und Anlagen, Technische Universität München, 1995.

[66] C. H. Wu. A general theory of the 3d flow in subsonic and supersonic turbomachines of axial, radial and mixed flow type. Technical Report TN 2604, NACA, 1952.