

TECHNISCHE UNIVERSITÄT MÜNCHEN  
INSTITUT FÜR ENERGIETECHNIK MW7

LEHRSTUHL FÜR FLUIDMECHANIK

**Ein modulares System zur  
Nachrechnung und Optimierung  
hydraulischer Turbomaschinen radialer Bauweise**

Carsten W. Schmalhorst

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. W. H. Polifke, Ph.D. (CCNY)

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. habil. Dr. h.c. R. Schilling
2. Univ.-Prof. Dr.-Ing. H. E. Siekmann, em.  
Technische Universität Berlin

Die Dissertation wurde am 26. Oktober 2010 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 16. Dezember 2010 angenommen.



*Für Sonja und Justus*





# Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Fluidmechanik, Fakultät Maschinenwesen der Technischen Universität München.

Mein besonderer Dank gilt Prof. Dr.-Ing. habil. Dr. h.c. Rudolf Schilling, der mir diese Arbeit erst ermöglichte. Ich danke ihm für seine Anregungen, die kritischen Fragen, die lehrreichen Diskussionen und für die gewährten Freiräume, eigene Ideen auch über längere Zeiträume verfolgen zu können.

Ich danke Prof. Dr.-Ing. Helmut Siekmann für die bereitwillige Übernahme des Korreferats und sein freundliches Interesse an meiner Arbeit. Für die Übernahme des eher mühseligen Vorsitzes der Prüfungskommission möchte ich Herrn Prof. Wolfgang Polifke, Ph.D., herzlich danken.

Bedanken möchte ich mich bei den Kollegen aus dem Forschungsfonds Pumpen, insbesondere den Teilnehmer des User-Clubs „Integrated Design System“, für die Einblicke in die alltäglichen Fragestellungen beim Entwurf von Turbomaschinen und die Anregungen, wie Software bei deren Beantwortung helfen kann.

Herrn Dr.-Ing. Jochen Fritz, KSB AG, danke ich für die Bereitstellung der Pumpengeometrie mitsamt Messdaten, sowie den Herren Dipl.-Ing. Steffen Schmidt und Dipl.-Ing. Philip Spenner der KSB AG für die bereitwillige Beantwortung der anfallenden Fragen dazu.

Bedanken möchte ich mich bei meinen Kollegen am Lehrstuhl für die gute Zusammenarbeit, die vielen Anregungen und die freundschaftliche, ja familiäre Arbeitsatmosphäre.

Meinen Eltern danke ich für die fortgesetzte Unterstützung meines Werdegangs.

Mein herzlichster Dank gilt meiner Frau Sonja, die mich in vielfältigster Weise von Beginn meines Studiums an unterstützte.

München, im Februar 2011

Carsten Schmalhorst



# Zusammenfassung

In der vorliegenden Arbeit wird ein Programmsystem zur Nachrechnung und Optimierung von radialen Turbomaschinen vorgestellt und seine Leistungsfähigkeit anhand von praxisbezogenen Beispielen aufgezeigt. Dabei wird für die schnelle Voroptimierung von Turbomaschinen das Quasi-Navier-Stokes-Verfahren (QNS) betrachtet und mit dem reynoldsgemittelten Navier-Stokes-Verfahren (RANS) verglichen. Es wird nachgewiesen, dass das QNS-Verfahren bei deutlich geringerer Laufzeit ähnliche Ergebnisänderungen bei gegebenen Geometrieänderungen aufweist wie das RANS-Verfahren. Das QNS-Verfahren wird für die Optimierung zweier radialer Pumpenlaufräder genutzt, indem es mit einem frei verfügbaren Genetischen Algorithmus gekoppelt wird. Die Optimierung wird vollautomatisch ohne Benutzereingaben durchgeführt und hat eine deutlich verbesserte Laufschaufelgeometrie als Ergebnis.

Außerdem kann das CFD-Experiment weitestgehend automatisiert werden, wobei sowohl die Vernetzung, die Modellierung der Wandreibung sowie die Konfiguration des CFD-Solver NS3D betrachtet werden. Das automatisierte CFD-Experiment wird genutzt, um für eine gegebene Reihe von Radialpumpen  $10 \text{ min}^{-1} \leq n_q \leq 90 \text{ min}^{-1}$  Kennlinien zu ermitteln. Die ermittelten Kennlinien zeigen für  $n_q \geq 30 \text{ min}^{-1}$  einen plausiblen Verlauf, für kleinere  $n_q$  ergeben sich jedoch noch zu klärende Tendenzen, die der experimentellen Erfahrung widersprechen.

Für eine Francis-Turbine mit der Schnellläufigkeit  $n_q = 44 \text{ min}^{-1}$  wird das Programmsystem genutzt, um ein Kennfeld basierend auf Einzelkanalrechnungen zu ermitteln. Dieses Kennfeld wird mit einem experimentell ermittelten Kennfeld einer Turbine gleicher Schnellläufigkeit verglichen.

Mit der Ermittlung des Axialschubs für eine Radialpumpe mit dem  $n_q = 53 \text{ min}^{-1}$  kann die Leistungsfähigkeit des Systems bei komplexeren geometrischen Modellen nachgewiesen werden. Dabei werden die Radseitenräume, Spalte, Bohrungen, Spiralgehäuse und Zu- und Ablauf im System modelliert, simuliert und ausgewertet. Die in der Simulation ermittelten integralen Größen weisen die gleichen Richtungstendenzen der Messung auf.



# Inhaltsverzeichnis

<b>Verwendete Formelzeichen und Abkürzungen</b>	<b>XIX</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Aufgabenstellung . . . . .	1
1.2 Stand der Technik . . . . .	2
1.3 Aufbau der Arbeit . . . . .	3
<b>2 Grundlagen</b>	<b>5</b>
2.1 Strömungssimulation . . . . .	5
2.1.1 Numerische Verfahren . . . . .	5
2.1.2 Reynoldsgemittelte Navier-Stokes-Gleichungen . . . . .	6
2.1.3 Turbulenzmodelle . . . . .	7
2.1.4 Quasi-Navier-Stokes-Verfahren (QNS) . . . . .	10
2.1.5 CFD-Solver NS3D . . . . .	11
2.2 Programmiertechnik . . . . .	12
2.2.1 Modularisierung komplexer Programmsysteme . . . . .	12
2.2.2 Frameworks . . . . .	14
2.2.3 Universelle Schnittstellendefinition . . . . .	16
2.3 Geometriedarstellung . . . . .	17
2.3.1 Kubische Splines . . . . .	17
2.3.2 B-Splines . . . . .	20
2.4 Geometriemodellierung . . . . .	21
2.4.1 Rotationssymmetrische Teile . . . . .	22
2.4.2 Schaufelflächen . . . . .	24
2.4.3 Andere strömungsführende Teile . . . . .	25
2.5 Entwurfsverfahren . . . . .	25
2.5.1 Dimensionierung, Layout . . . . .	26

2.5.2	Erstentwurf . . . . .	27
2.5.3	Optimierung . . . . .	28
2.6	Vernetzung . . . . .	28
2.6.1	Verfügbare Interfaces . . . . .	30
2.6.2	Hierarchische Netzgenerierung . . . . .	31
2.6.3	Partitionierung . . . . .	32
2.6.4	Schaufelbereich . . . . .	32
2.6.5	Rotationssymmetrische Bereiche . . . . .	36
2.6.6	Entlastungsbohrung . . . . .	37
2.6.7	Spiralteilstücke . . . . .	37
2.6.8	Spirale . . . . .	40
2.7	Auswertung . . . . .	42
2.7.1	Integrale Größen . . . . .	42
2.7.2	Lokale Größen . . . . .	44
<b>3</b>	<b>Echtzeitverhalten</b>	<b>49</b>
3.1	Anwendung des QNS-Verfahrens . . . . .	49
3.2	Integrale Größen . . . . .	50
3.3	Lokale Größen . . . . .	52
3.4	Verhalten bei Änderungen . . . . .	57
3.5	Laufzeitverhalten . . . . .	62
3.6	Beurteilung des QNS-Verfahrens . . . . .	62
<b>4</b>	<b>Automatisierung des CFD-Experiments</b>	<b>63</b>
4.1	Effiziente Shared Memory Parallelisierung . . . . .	64
4.1.1	Abschätzung der Anzahl möglicher Lösungen . . . . .	64
4.1.2	Zielfunktion . . . . .	65
4.1.3	Heuristische Aufteilung . . . . .	67
4.1.4	Laufzeitverhalten praktischer Beispiele . . . . .	68
4.2	Radialpumpen . . . . .	74
4.2.1	Partitionierung des Strömungsbereichs . . . . .	74
4.2.2	Vernetzungsparameter . . . . .	76
4.2.3	Konfiguration des Solvers . . . . .	83
4.3	Francis-Turbine . . . . .	89
4.3.1	Partitionierung des Strömungsbereichs . . . . .	90

---

4.3.2	Vernetzungsparameter . . . . .	90
4.3.3	Konfiguration des Solvers . . . . .	93
<b>5</b>	<b>Anwendung</b>	<b>95</b>
5.1	Einzelkanalsimulationen . . . . .	95
5.1.1	Pumpenreihe . . . . .	95
5.1.2	Turbinenkennfeld . . . . .	102
5.2	Automatisierte Optimierung . . . . .	105
5.2.1	Genetische Algorithmen . . . . .	105
5.2.2	Ergebnisse . . . . .	109
5.3	Axialschub einer Radialpumpe . . . . .	119
5.3.1	Einzelkanalrechnung . . . . .	120
5.3.2	Charakteristiken . . . . .	129
5.3.3	Axialkraft . . . . .	133
5.3.4	Frozen-Rotor-Rechnung . . . . .	138
5.3.5	Bewertung . . . . .	142
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>145</b>
	<b>Literaturverzeichnis</b>	<b>147</b>





# Abbildungsverzeichnis

2.1	Geschwindigkeitsprofil einer turbulenten Grenzschicht an einer hydraulisch glatten Wand . . . . .	10
2.2	Kubische Splines: Parametrisierungen . . . . .	19
2.3	NURBS-Beispiel . . . . .	22
2.4	Meridiankontur bestehend aus Hub- und Shroudkontur als NURBS am Beispiel einer Modellturbine . . . . .	23
2.5	Flächenverlauf einer Meridiankontur und daraus ermittelter $\overline{c_m}$ -Verteilung am Beispiel einer Modellturbine . . . . .	24
2.6	Entwurfsprozess . . . . .	26
2.7	Bauartenauswahl von Turbinen nach Voith, vgl. RAABE [42] . . . . .	27
2.8	Multi-Level-CFD: Integrated Design Process . . . . .	29
2.9	Beispiel für Smart-Wall Interface, s. HERRMANN [28] . . . . .	31
2.10	Hierarchische Vernetzung . . . . .	32
2.11	Vernetzung: Partitionierung . . . . .	33
2.12	Offene Laufräder: Einblockstrategie . . . . .	36
2.13	Offene Laufräder: Multi-Block-Strategie . . . . .	36
2.14	Beispiel Entlastungsbohrung . . . . .	38
2.15	zweidimensionales Basisnetz der Entlastungsbohrung . . . . .	38
2.16	Spiralteilstück . . . . .	39
2.17	Schritte der Spiralvernetzung . . . . .	41
2.18	Flächendefinitionen für Postprocessing einer Strömungsmaschine am Beispiel eines offenen Pumpenlaufrads . . . . .	45
3.1	Abweichung der integralen Größen $\Psi_t$ , $\Psi_{t,th}$ und $\eta$ von der pauschalen Turbulenzkonstante $C_t$ . . . . .	51
3.2	Abhängigkeit integraler Werte von $C_t$ , korrigiert für hohe $n_q$ . . . . .	52
3.3	Geschwindigkeitsverteilung QNS, RANS . . . . .	53
3.4	Druckverteilung QNS, RANS, $n_q = 30 \text{ min}^{-1}$ . . . . .	54

3.5	Druckverteilung QNS, RANS, $n_q = 50 \text{ min}^{-1}$ . . . . .	55
3.6	Druckverteilung QNS, RANS, $n_q = 70 \text{ min}^{-1}$ . . . . .	56
3.7	Systemantworten für $\Delta\beta_{s1}$ . . . . .	58
3.8	Systemantworten für $\Delta\beta_{s2}$ . . . . .	59
3.9	Systemantworten für $\Delta\varphi$ . . . . .	60
3.10	Systemantworten für $\Delta l$ . . . . .	61
4.1	Anzahl der möglichen Kombinationen $M$ . . . . .	66
4.2	Effizienz für die Parallelisierung von 41 Blöcken . . . . .	70
4.3	Speedup der Shared-Memory-Parallelisierung . . . . .	71
4.4	Effizienz der Shared-Memory-Parallelisierung . . . . .	72
4.5	Aufteilung des Rechengebiets einer Radialpumpe . . . . .	76
4.6	Übersicht möglicher Wandabstände . . . . .	78
4.7	$y^+$ -Klassen für Hub-Shroud-Vernetzung . . . . .	79
4.8	$s(t_1)$ -Gültigkeitsbereich . . . . .	79
4.9	Sollwerte für $n_k$ und $b$ . . . . .	80
4.10	$y^+$ -Klassen bei der Schaufelvernetzung . . . . .	81
4.11	Integrale Größen bei unterschiedlicher Schaufelvernetzung als Funktion von $\overline{y^+}$ . . . . .	82
4.12	$\overline{y^+}$ als Funktion von $y_p$ für verschiedene $n_q$ . . . . .	83
4.13	$\lambda_{\text{opt}}(n_q)$ als Funktion von $n_j$ . . . . .	84
4.14	Konvergenzverhalten der Parametervariationen . . . . .	85
4.15	Solververhalten für $n_q \leq 25 \text{ min}^{-1}$ . . . . .	86
4.16	Solververhalten für $n_q \geq 30 \text{ min}^{-1}$ . . . . .	87
4.17	Stabilitätsgrenze des Konvergenzverhaltens und Empfehlung für $\alpha_{\text{uvw}}$ . . . . .	88
4.18	Stabilität des Konvergenzverhaltens am Beispiel der integralen Werte über $n_q$ . . . . .	89
4.19	Aufteilung des Rechengebiets einer Francisturbine . . . . .	90
4.20	Verlauf des wandnächsten $y_p$ für $\overline{y^+} \approx 60$ über $n_q$ . . . . .	91
4.21	Klassen der $y^+$ -Werte im Hub-Shroud-Netz über $n_q$ . . . . .	92
5.1	Verlauf von $\overline{y^+}$ . . . . .	97
5.2	Charakteristiken der Pumpenreihe für $n_q \geq 40 \text{ min}^{-1}$ . . . . .	99
5.3	Charakteristiken der Pumpenreihe für $n_q \leq 30 \text{ min}^{-1}$ . . . . .	100
5.4	Wirkungsgradvorhersage der Pumpenreihe mit CFX und NS3D . . . . .	101
5.5	$y^+$ -Verteilung für die Pumpenreihe auf der Schaufel . . . . .	101

5.6	Kennfeld der Simulation und Iso- $\eta$ -Linien der Messung, FT44. . . . .	104
5.7	Konvergenzverhalten genetischer Algorithmus . . . . .	108
5.8	Systemantworten der Generationen 1, 4 und 7 für die Radialpumpe $n_q = 70 \text{ min}^{-1}$ . . . . .	110
5.9	Radialpumpe $n_q = 70 \text{ min}^{-1}$ , Wirkungsgradverlauf und Totaldruckerhöhung original, optimiert . . . . .	111
5.10	Konforme Abbilder, Schaufelwinkel der Radialpumpe $n_q = 70 \text{ min}^{-1}$ . .	112
5.11	Druckverteilungen der Radialpumpe $n_q = 70 \text{ min}^{-1}$ . . . . .	113
5.12	Systemantworten der Generationen 1, 4 und 7 für die Radialpumpe $n_q = 90 \text{ min}^{-1}$ . . . . .	115
5.13	Radialpumpe $n_q = 90 \text{ min}^{-1}$ , Wirkungsgradverlauf und Totaldruckerhöhung original, optimiert . . . . .	115
5.14	Konforme Abbilder, Schaufelwinkel der Radialpumpe $n_q = 90 \text{ min}^{-1}$ . .	116
5.15	Druckverteilungen der Radialpumpe $n_q = 90 \text{ min}^{-1}$ . . . . .	117
5.16	Meridianansicht des Rechennetzes der Einzelkanalrechnung ohne RSR, Bilanzierungsebenen . . . . .	120
5.17	Kennlinien der Einzelkanalrechnung ohne RSR für die Radialpumpe mit $n_q = 53 \text{ min}^{-1}$ . . . . .	121
5.18	Totaldruckverlauf über dem relativen Volumenstrom $q$ der Einzelkanalrechnung an den Bilanzierungseben 0, 1 sowie am Einlaß und am Auslaß (links), Totaldruckdifferenzen (rechts) . . . . .	122
5.19	Dimensionsloser Drall $\Psi_{rcu}$ an der Bilanzierungsebene 0 in Abhängigkeit vom relativen Volumenstrom $q$ . . . . .	123
5.20	Wirkungsgrade Einzelkanalrechnung . . . . .	123
5.21	Meridianansicht der Rechengebiete inkl. Radseitenräumen . . . . .	125
5.22	Laufradauslaß als schmale radiale Düse . . . . .	125
5.23	Laufradauslaß als breite radiale Düse . . . . .	126
5.24	Laufradauslaß als Spiralteil, kombiniert periodisch/tangentialer Auslaß	127
5.25	Bilanzierungsebenen Impeller und Laufschaufelkanal . . . . .	129
5.26	Axialschub: Konvergenzverhalten . . . . .	130
5.27	Integrale Werte der Varianten I bis IV . . . . .	132
5.28	Integrale Werte der Varianten V bis VII . . . . .	133
5.29	Axialschubbeiwerte der Einzelkanalrechnungen . . . . .	135
5.30	Spaltvolumenströme der Radseitenräume . . . . .	136
5.31	Teillastwirbel $\Psi_{rcu}$ . . . . .	137
5.32	Umfangsgemittelte Größen . . . . .	137
5.33	Vernetzung für die Frozen-Rotor-Simulation . . . . .	139

5.34	Achsschnitt der Vernetzung für die Frozen-Rotor-Simulation . . . . .	139
5.35	Charakteristiken aus der Frozen-Rotor-Simulation über dem relativen Volumenstrom $q$ , Unterschiede zwischen MINMOD und UDS . . . . .	140
5.36	Charakteristiken aus der Frozen-Rotor-Simulation über dem relativen Volumenstrom $q$ , Frozen Rotor (FR) und Einzelkanalrechnung Variante VI (VI) . . . . .	141

# Tabellenverzeichnis

3.1	Laufzeitverhalten QNS und RANS . . . . .	62
4.1	Parallelisierte Rechengitter . . . . .	69
4.2	Verwendete Hardware . . . . .	70
4.3	Geforderte Standardwerte des Pumpenexperiments . . . . .	74
4.4	Auslegungsdaten der Pumpenreihe nach KRÄMER [32] . . . . .	75
4.5	Auslegungsdaten der Turbinenreihe . . . . .	89
5.1	Netzdimensionen der Pumpenreihe . . . . .	96
5.2	Auslegungsdaten der simulierten Turbine . . . . .	102
5.3	Daten für die Radialpumpe $n_q = 70 \text{ min}^{-1}$ . . . . .	109
5.4	Daten für die Radialpumpe $n_q = 90 \text{ min}^{-1}$ . . . . .	114
5.5	Entwurfsdaten zur Nachrechnung des Axialschubs der Radialpumpe mit $n_{q,\text{opt}} = 53 \text{ min}^{-1}$ . . . . .	119
5.6	Integrale Daten der Netzvariationen für die Volumenzahl $\phi = 0.115$ . .	128
5.7	Netzvarianten für die Einzelkanalrechnung . . . . .	131
5.8	Unterschiede der integralen Werte der Variante VI bei $\phi = 0.1253$ in Abhängigkeit des Lösungswegs . . . . .	138



# Verwendete Formelzeichen und Abkürzungen

**Abkürzungen**

CAD	Computer Aided Design
CAE	Computer Aided Engineering
CFD	Computational Fluid Dynamics
CSA	Computational Structural Analysis
DNS	Direkte numerische Simulation
DS	Druckseite
FLM	Lehrstuhl für Fluidmechanik der Technischen Universität München
GPL	GNU Public License
GUI	Graphical User Interface
IDE	Integrated Development Environment
IDS	Integrated Design System
JVM	Java Virtual Machine
KNN	Künstliches Neuronales Netz
LCL	Turbulenzmodell nach LIEN, CHEN UND LESCHZINER [37]
LES	Large Eddy Simulation
MINMOD	Verfahren nach HARTEN [27]
MPI	Message Passing Interface
NPSH	net positive suction height
NS3D	Navier-Stokes 3-dimensional
NUBS	Non-Uniform B-Spline
NURBS	Non-Uniform Rational B-Spline
Q3D	Quasi-dreidimensionale Strömungsrechnung
QNS	Quasi-Navier-Stokes Strömungsrechnung
RANS	Reynolds-gemittelte Navier-Stokes-Gleichungen
RFK	Radformkennzahl
RSR	Radseitenraum
RTD	Real-Time Design System
SS	Saugseite
UDS	Upwind Differencing Scheme
VDMA	Verband Deutscher Maschinen- und Anlagenbau e.V.



**Lateinische Zeichen**

$A$	$[m^2]$	Fläche
$b$		Verzerrungsparameter
$b_{ij}$		Anisotropietensor der Reynolds-Spannungen
$b_0$	$[m]$	Laufraudeintrittsbreite (Turbine)
$b_2$	$[m]$	Laufradaustrittsbreite (Pumpe)
$C$		Konstante
$c_{ax}$		Axialschubbeiwert
$c_p$		Druckbeiwert
$C_t$		pauschale Turbulenzkonstante im QNS-Verfahren
$C_{\epsilon,1}, C_{\epsilon,2}$		Modellkonstanten der $k$ - $\epsilon$ -Modelle
$C_\mu$		Modellkonstante
$c$	$[m]$	Laufvariable
$\vec{c}, c_i$	$[m/s]$	Geschwindigkeitsvektor
$D$	$[m]$	Durchmesser
$E$		Effizienz der Parallelisierung
$H, h$	$[m]$	Höhe, Fallhöhe, Förderhöhe
$k$	$[m^2/s^2]$	turbulente kinetische Energie
$k$		Anzahl Prozesse
$L$	$[W]$	Leistung
$L_t$	$[m]$	turbulentes Längenmaß
$l$	$[m]$	Länge, Koordinate der konformen Abbildung
$\dot{m}$	$[m^3/s]$	dichtebezogener Massenfluß
$m$	$[m]$	Laufvariable in Meridianrichtung
$M$	$[Nm]$	Moment
$N$		Anzahl Knotenpunkte oder Zellen
$\vec{n}, n_i$		Normalenvektor
$n_I$		Anzahl Iterationen
$n$		Koordinate in Normalenrichtung; lineare Iteration
$n_q$	$[1/min]$	spezifische Drehzahl, Radformkennzahl
$p$	$[N/m^2]$	statischer Druck
$p_t$	$[N/m^2]$	Totaldruck
$p_c, p_m$		Wahrscheinlichkeit
$P$	$[-]$	dimensionsloser Druck
$P$		Kontrollpunkt
$P_k$	$[m^2/s^3]$	Produktion von $k$

$q$		relativer Volumenstrom
$Q$	$[m^3/s]$	Volumenstrom
$\vec{r}, r_i$	$[m]$	Ortsvektor
$r$		Residuum
$r$	$[m]$	Radius
$Re$		Reynolds-Zahl
$Re_T$		turbulente Reynolds-Zahl
$s$	$[m]$	Strecke
$S$		Speedup
$S_{ij}$	$[1/s]$	Deformationstensor
$t$	$[s]$	Zeit
$t$		normierter Parameterwert
$T$	$[s]$	Zeitintervall
$Tu$		Turbulenzgrad
$u$		allgemeine Koordinate parametrischer Flächen
$u_\tau$	$[m/s]$	Schubspannungsgeschwindigkeit
$\overline{c'_i c'_j}$	$[m^2/s^2]$	Reynolds-Spannungstensor
$V$	$[m^3]$	Volumen
$v$		allgemeine Koordinate parametrischer Flächen
$v$	$[m/s]$	wandnormale Geschwindigkeitskomponente; Geschwindigkeitskomponente in $y$ -Richtung
$\vec{w}, w_i$	$[m/s]$	Geschwindigkeitsvektor
$\vec{x}, x_i, x$	$[m]$	Ortskoordinate
$y$	$[m]$	Ortskoordinate; Wandabstand
$y_p$	$[m]$	Wandabstand
$Y$	$[-]$	spezifische Schaufelarbeit
$y^+$		dimensionsloser Wandabstand
$z$	$[m]$	Ortskoordinate
$Z$	$[-]$	Schaufelzahl

## Griechische Zeichen

$\alpha$	[°]	Strömungswinkel
$\alpha_{uvw}$		
$\alpha_p$		Unterrelaxationsfaktoren
$\alpha_{\text{upd}}$		
$\beta$		Blendingfaktor
$\beta_s, \beta_{s1}, \beta_{s2}$	°	Staffelungswinkel; Schaufelwinkel an Vorderkante, Hinterkante
$\Gamma$		Zirkulation
$\gamma$	[°]	Staffelungswinkel; Öffnungswinkel
$\Delta A$	[m <sup>2</sup> ]	diskrete Fläche
$\Delta V$	[m <sup>3</sup> ]	diskretes Volumen
$\delta$		allgemeiner Differentialoperator
$\delta$		relative Abweichung
$\delta_{ij}$		Kronecker-Symbol
$\varepsilon$	[m <sup>2</sup> /s <sup>3</sup> ]	Dissipationsrate
$\eta$		Wirkungsgrad
$\eta_T$	[m]	Kolmogorov-Länge
$\kappa$		von-Karman-Konstante
$\lambda$		Verzerrungsfaktor des Schaufelnetz
$\mu$	[kg/(m s)]	dynamische Viskosität
$\nu$	[m <sup>2</sup> /s]	kinematische Viskosität
$\nu_{\text{eff}}$	[m <sup>2</sup> /s]	effektive kinematische Viskosität
$\omega$	[1/s]	Winkelgeschwindigkeit bei Rotation um die z-Achse
$\rho$	[kg/m <sup>3</sup> ]	Dichte
$\tau_w$	[N/m <sup>2</sup> ]	Wandschubspannung
$\Phi$		allgemeine Variable
$\varphi$		Winkel der Zylinderkoordinate
$\phi$		dimensionsloser Volumenstrom
$\vec{\Phi}$		Vektor einer allgemeinen Variable
$\bar{\Phi}$		Mittelwert eine allgemeinen Variable, Median einer allgemeinen Variable
$\Psi$		Druckzahl
$\sigma$		Thomazahl

**Tiefgestellte Zeichen**

<i>0</i>	Referenz
<i>a</i>	Axiale Richtung
<i>anl</i>	Anlage
<i>ausl</i>	Auslegung
<i>ax</i>	axial
<i>b</i>	barometrisch
<i>const</i>	konstanter Wert
<i>D</i>	Durchmesser
<i>erf</i>	erforderlich
<i>exp</i>	Experiment
<i>h</i>	hydraulisch
<i>i</i>	Zählindex
<i>in</i>	Einlaß
<i>j</i>	Zählindex
<i>k</i>	Zählindex
<i>krit</i>	kritisch
<i>La</i>	Laufgrad
<i>lam</i>	laminar
<i>lin</i>	linear
<i>loc</i>	lokal
<i>max</i>	Maximum
<i>mech</i>	mechanisch
<i>min</i>	Minimum
<i>opt</i>	Optimum
<i>out</i>	Auslaß
<i>Pu</i>	Pumpe
<i>r</i>	radiale Richtung
<i>ref</i>	Referenz-
<i>soll</i>	Sollwert
<i>sp</i>	Spalt
<i>sys</i>	System-
<i>t</i>	total
<i>th</i>	theoretisch

$Tu$	Turbine
$turb$	turbulent
$v$	Verlust
$va$	Vapour, Dampf
$x$	$x$ -Richtung
$y$	$y$ -Richtung
$z$	$z$ -Richtung
$\phi$	auf die allgemeine Variable $\phi$ bezogen
$\varphi$	Umfangsrichtung

### Hochgestellte Zeichen

$(l)$	$l$ -te Ableitung
$T$	transponierte
$+$	dimensionslose turbulente Größen

### Sonstige

$\Delta$	Differenz
$\frac{\partial}{\partial t}$	Partielle Ableitung nach der Zeit
$\frac{d}{dt}$	Ableitung nach der Zeit
$\mathcal{O}$	Größenordnung; Ordnung
$\times$	Kreuzprodukt
$-$	Mittelwert, Median
$'$	Schwankungsgröße



# Kapitel 1

## Einleitung

Turbomaschinen sind ein wesentlicher Bestandteil der Energie- und Anlagentechnik in der Industrie. Die Energiegewinnung durch hydraulische Kraftmaschinen wird immer bedeutender, da diese von regenerativen, natürlichen Energiequellen angetrieben werden und somit einen Beitrag zur Vermeidung von  $CO_2$ -Emissionen leisten können. Hydraulische Arbeitsmaschinen werden in der Industrie und in der Versorgung der privaten Haushalte für zahlreiche Aufgaben eingesetzt. Die Deutsche Netzagentur (dena) hat Pumpensysteme in ihrer *Initiative EnergieEffizienz* als Querschnittstechnologie aufgenommen, da sie hier ein erhebliches Energiesparpotential sieht [5].

Eine hydraulische Turbomaschine muß möglichst genau auf die jeweilige Aufgabenstellung abgestimmt werden, damit sie wirtschaftlich und ohne Verschwendung natürlicher Ressourcen sinnvoll betrieben werden kann. Vor dem Hintergrund ökonomischer Zwänge wird ein Werkzeug benötigt, mit dem diese Abstimmung bei möglichst geringem Aufwand geschehen kann. Bei der Entwicklung von Turbomaschinen jeder Art hat sich die numerische Simulation der Strömung, Computational Fluid Dynamics (CFD), als Hilfsmittel der Entwicklung etabliert. Mittels der CFD können bessere Turbomaschinen in kürzerer Zeit entwickelt werden, da bereits in frühen Entwicklungsstadien Aussagen über das spätere Betriebsverhalten getroffen werden können.

Gängige CFD-Werkzeuge setzen vom Anwender Spezialkenntnisse in numerischen Verfahren voraus. Die Durchführung von CFD-Experimenten erfordert einen gewissen Erfahrungsschatz und die Sicherheit im Umgang mit den Software-Werkzeugen. Ein Ingenieur, der in der Auslegung von Turbomaschinen ausgebildet wurde, verfügt i.d.R. nicht über detaillierte Kenntnisse in der Anwendung dieser Verfahren. Obwohl ein CFD-Experiment deutlich weniger Aufwand bedeutet als ein Experiment am Prüfstand, existieren Zugangshürden für den Entwicklungsingenieur. Es ist sinnvoll, diese Hürden soweit wie möglich abzubauen und dem Entwickler ein dem Problem angepaßtes Werkzeug an die Hand zu geben, mit dem er die Vorteile des CFD-Experiments voll ausschöpfen kann.

### 1.1 Aufgabenstellung

Es soll ein Programmsystem für die Nachrechnung und Optimierung hydraulischer Turbomaschinen radialer Bauweise erstellt werden. Der Zielanwender dieses Programmsys-

tems ist der im hydraulischen Entwurf geschulte Ingenieur, der *nicht* über tiefergehende Kenntnisse der numerischen Simulation von Strömungen in diesen Maschinen verfügt. Das Programmsystem soll diesem „Hydrauliker“ als Werkzeug dienen, wobei er keine Detailkenntnisse über dessen Wirkungsweise benötigen soll. Das Programmsystem soll ihn unterstützen beim Entwurf, der Nachrechnung und der Optimierung von hydraulischen Turbomaschinen. Somit ist besonderes Augenmerk auf die Bedienungsicherheit und die Automatisierung des CFD-Experiments zu legen.

Das Programmsystem soll integriert sein, d.h. es umfaßt neben der Geometriebeschreibung alle notwendigen Algorithmen für die Aufbereitung der Turbomaschine, damit sie simulierbar wird, das Präprozessing, die eigentliche Simulation, den CFD-Solver, und alle notwendigen Algorithmen zur problembezogenen Auswertung der Ergebnisse, das Postprozessing. Die Geometriemodellierung soll für unterschiedliche Simulationen mit unterschiedlicher Fragestellung einheitlich sein, um die Konsistenz der Daten zu gewährleisten. Zusätzlich soll das zu erstellende Programmsystem leicht erweiterbar sein, damit Fortschritte in der Simulationstechnik ohne grundlegende Änderungen einfließen können.

Abschließend soll die Leistungsfähigkeit des neu erstellten Programmsystems dargestellt werden, indem konkrete Problemstellungen aus der Entwicklung von hydraulischen Turbomaschinen gelöst werden. Dazu zählen die Kennlinien- bzw. -feldermittlung bei Radialpumpen und Francis-Turbinen, sowie die Optimierung von Laufschaufelgeometrien. Hierzu kommen unterschiedliche numerische Simulationsmethoden zur Anwendung. Schließlich soll die Leistungsfähigkeit des Programmsystems bei komplexeren Fragestellungen bewiesen werden, indem der Axial Schub einer Radialpumpe mittels CFD-Experiments ermittelt wird.

## 1.2 Stand der Technik

Für den Entwurf, die Simulation und die Optimierung von hydraulischen Turbomaschinen radialer Bauweise sind kommerzielle und akademische Softwarepakete verfügbar. Im Folgenden soll eine kurze Auswahl von Systemen vorgestellt werden, die Teilaspekte des Gesamtprozesses abdecken bzw. eine Lösung dieser Aufgabenstellung anbieten.

- CFTurbo [6] ist ein Programm für die Auslegung von Pumpen-, Ventilator-, Turbinen- und Kompressorbeschaufelungen sowie passender Spiralgehäuse. Es integriert sich in bestehende CAE-Infrastruktur und ermöglicht es dem Benutzer, die entworfene Beschaufelung mittels kommerzieller CFD-Tools zu testen.
- PumpLinx [4] ist ein integriertes System für Pumpen, dessen Schwerpunkt auf der Simulation des Betriebsverhaltens von Verdrängerpumpen liegt. Es enthält zusätzlich Module für die Untersuchung von Radialpumpen.
- Die Firma Ansys, Inc. bietet für ihr CFD-System CFX zusätzliche Programmmodule für den Entwurf von Beschaufelungen von Turbomaschinen an, den sog. „BladeModeler“. Für die Vernetzung von Turbomaschinen ist ein zusätzliches Modul „TurboGrid“ erhältlich. CFX ist ein umfassendes Softwaresystem für die Simulation von Strömungen und in erster Linie ein Werkzeug für Numeriker.



- Das Real Time Design System (RTD), s. LEPACH [36], wurde für den Entwurf, die Simulation und die Optimierung von Francis-Turbinen erstellt. Für schnelle Antwortzeiten im Optimierungsprozeß beinhaltet es ein Quasi-3D-Eulerverfahren. Für detailliertere Fragestellungen kommt ein RANS-Verfahren zur Anwendung. Das RTD kann als Vorgänger des hier vorgestellten Programmsystems gesehen werden.

## 1.3 Aufbau der Arbeit

Zunächst sollen im folgenden Abschnitt einige Grundlagen skizziert werden, die bei der numerischen Simulation radialer Strömungsmaschinen benötigt werden. Dabei liegt der Schwerpunkt auf der Beschreibung der Geometrie und der Auswertung der ermittelten Ergebnisse. In Abschnitt 3 wird das QNS-Verfahren untersucht und dessen Eignung für Optimierungsschritte begutachtet. In Abschnitt 4 werden die notwendigen Grundlagen erarbeitet, um das CFD-Experiment automatisiert ablaufen lassen zu können. In Abschnitt 5 werden die in den vorangehenden Abschnitten erarbeiteten Methoden genutzt, um Aufgabenstellungen aus der Entwicklung von radialen Turbomaschinen zu lösen.



# Kapitel 2

## Grundlagen

### 2.1 Strömungssimulation

Die Strömung eines inkompressiblen Fluids der Dichte  $\rho = \text{const.}$  und der Viskosität  $\nu = \text{const.}$  kann über Betrachtungen zu Erhaltung des Impulses und der Masse an einem differentiellen Volumenelement beschrieben werden. Wenn Temperatureinflüsse auf Viskosität oder Dichte vernachlässigt werden, so ist der Strömungszustand komplett über das Geschwindigkeitsfeld  $\vec{c} = (c_x, c_y, c_z)^T$  und den statischen Druck  $p$  im Feld  $\vec{x} = (x, y, z)^T$  für einen Zeitpunkt  $t$  beschrieben. Die Erhaltungsgleichungen für den inkompressiblen Fall, denen das Feld genügen muß, bestehen aus der Kontinuitätsgleichung (2.1) sowie den Navier-Stokes-Gleichungen, also dem Impulserhaltungssatz (2.2). Die Gleichungen lauten in Einsteinscher Summenkonvention wie folgt, vgl. z.B. ZIEREP [57]:

$$\frac{\partial c_i}{\partial x_i} = 0, \quad (2.1)$$

$$\frac{\partial c_i}{\partial t} + c_j \frac{\partial c_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[ \nu \left( \frac{\partial c_i}{\partial x_j} + \frac{\partial c_j}{\partial x_i} \right) \right]. \quad (2.2)$$

Dieses Gleichungssystem aus partiellen Differentialgleichungen beschreibt jedes inkompressible Strömungsproblem vollständig. Es besteht aus vier Gleichungen für vier unbekannte Größen. Diese Größen sind der statische Druck  $p$  und die drei Komponenten der Geschwindigkeit  $c_i$ . Der statische Druck  $p$  tritt nur in Form eines Gradienten auf, so dass das Druckniveau im gegebenen inkompressiblen Fall unerheblich ist, nur die Druckänderungen im Feld fließen in die Lösung des Gleichungssystems ein.

Das System gekoppelter, nichtlinearer partieller Differentialgleichungen ist nur für Einzelfälle unter der Annahme von starken Vereinfachungen analytisch lösbar, im Regelfall kann eine Lösung nur auf numerischen Wege ermittelt werden.

#### 2.1.1 Numerische Verfahren

Die direkte Lösung der Erhaltungsgleichungen (2.1) und (2.2) mittels numerischer Verfahren wird *Direkte Numerische Simulation* (DNS) genannt und kommt ohne jede

zusätzliche Modellannahme aus. Dabei werden alle Zustände im gesamten Strömungsfeld ermittelt, d.h. nicht nur die Makrostruktur, sondern jede Störung, die sich über die Zeit und den Raum ergibt, ist ein Ergebnis der Simulation. Jeder turbulente Wirbel, gleich welcher Größe und Energiegehalts, muß von der DNS erfaßt werden, damit das Verfahren zu einem sinnvollen Ergebnis führt. Numerische Verfahren basieren auf der diskreten Behandlung des stetigen Raumes. Diese räumliche Diskretisierung, *Gitter* oder *Netz*, muß somit kleiner sein als die kleinste mögliche Strömungsstruktur, die sich im Feld ergeben kann. TENNEKES UND LUMLEY [51] schätzen folglich die notwendige Anzahl von Zellen in einem Gitter über ein integrales turbulentes Längenmaß  $L_T$  und einem Mikro-Längenmaß  $\eta_T$ , der Kolmogoroff-Länge, ab:

$$\frac{L_T}{\eta_T} \sim \left( \frac{u_T \cdot L_T}{\nu} \right)^{3/4} \sim Re_T^{3/4} \Rightarrow N \sim \left( \frac{L_T}{\eta_T} \right)^3 \sim Re_T^{9/4}. \quad (2.3)$$

Die turbulente Reynoldszahl  $Re_T$  ergibt sich aus dem turbulenten Geschwindigkeitsmaß  $u_T$  und dem integralen Längenmaß  $L_T$ . Nach FERZIGER UND PERIC [21] beträgt die turbulente Reynoldszahl ca. 1% der Reynoldszahl, die die technische Anwendung charakterisiert, die also die makroskopische Strömung beschreibt. Das Wachstumsgesetz führt zu außerordentlich hohen Anforderungen, die die DNS an die Diskretisierung des Raumes stellt. Diese Anforderung sind derzeit für technische Strömungen gar nicht bzw. nicht sinnvoll erfüllbar.

Jedes andere Verfahren geht von einer Modellvorstellung aus, mit der die sehr kleinen und tendenziell chaotischen Strukturen der Strömung abgebildet werden sollen, damit das Rechengitter nur die relevante Makrostruktur der Strömung auflösen muß. Das qualitativ höchstwertige dieser Verfahren ist die Grobstruktursimulation, engl. *Large Eddy Simulation* (LES). Hierbei werden die wesentlichen Wirbel, die das Hauptströmungsfeld beeinflussen, komplett zeitlich und räumlich aufgelöst. Die Kleinstruktur der Strömung, die nicht mehr vom Rechengitter oder Zeitschritt aufgelöst werden können, werden mittels geeigneter Größen modelliert.

Das LES-Verfahren gewinnt für technische Aufgabenstellungen derzeit immer mehr an Bedeutung. Es stellt immer noch sehr hohe Anforderungen an den Speicherplatz und die Rechenkapazität. Das gängige Verfahren der Strömungssimulation ist die statistische Beschreibung der Turbulenz mittels der reynoldsgemittelten Navier-Stokes-Gleichungen, engl. *Reynolds-averaged Navier Stokes* (RANS).

## 2.1.2 Reynoldsgemittelte Navier-Stokes-Gleichungen

Der Ansatz von Reynolds besagt, dass die Strömung aufgeteilt werden kann in einen Mittelwert, der sich idealerweise geordnet und vorhersagbar verhält, sowie in eine ungeordnete Störgröße, die die Fluktuation der Strömungsgröße um den Mittelwert beschreibt. Allgemein kann also eine Strömungsgröße  $\Phi$  dargestellt werden als die zeitlich gemittelte Größe  $\overline{\Phi}$  und deren Fluktuation  $\Phi'$ , s. z.B. FERZIGER UND PERIC [21].

$$\Phi(x_i, t) = \overline{\Phi(x_i, t)} + \Phi'(x_i, t) \quad (2.4)$$

Wird diese Strömungsgröße über einen geeigneten Zeitraum  $T$  gemittelt, so heben sich die Fluktuationen gegenseitig auf und es bleibt der Mittelwert übrig:

$$\overline{\Phi}(x_i) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} \Phi(x_i, t) dt, \quad (2.5)$$

$$\overline{\Phi'(x_i, t)} = 0. \quad (2.6)$$

Mit dem Separationsansatz aus Gleichung (2.4) können nun die Erhaltungsgleichungen aus (2.1) und (2.2) gemittelt werden. Daraus erhält man die Reynoldsgemittelten Navier-Stokes-Gleichungen:

$$\frac{\partial \bar{c}_i}{\partial x_i} = 0, \quad (2.7)$$

$$\frac{\partial \bar{c}_i}{\partial t} + \bar{c}_j \frac{\partial \bar{c}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left[ \nu \left( \frac{\partial \bar{c}_i}{\partial x_j} + \frac{\partial \bar{c}_j}{\partial x_i} \right) - \overline{c'_i c'_j} \right]. \quad (2.8)$$

Durch die Mittelung ergeben sich nun die mittleren Strömungsgrößen  $\bar{p}(\vec{x}, t)$  und  $\bar{c}_i(\vec{x}, t)$ , sowie ein zusätzlicher Term  $\overline{c'_i c'_j}$ , der aus der Mittelung der konvektiven Terme resultiert und nicht wie die anderen Störgrößen durch die zeitliche Mittelung verschwindet. Dieser Reynolds-Spannungstensor enthält insgesamt sechs unterschiedliche Terme, für deren Bestimmung keine weiteren Gleichungen zur Verfügung stehen. Das Gleichungssystem ist somit unterbestimmt, d.h. es besteht ein Schließungsproblem.

### 2.1.3 Turbulenzmodelle

Für die Bestimmung der Störungsterme aus Gleichung (2.8) müssen Modellannahmen getroffen werden, damit das Gleichungssystem gelöst werden kann, die allerdings die Auswirkung der Turbulenz auf die Strömung physikalisch richtig abbilden müssen. Die Formulierung weiterer Gleichungen für die Errechnung dieser Störungsterme wird als Turbulenzmodellierung bezeichnet. Darunter wird die Modellierung der turbulenten Normal- und Schubspannungen verstanden.

Die derzeit am weitesten verwendete Näherung des Reynolds-Spannungstensors ist das Wirbelviskositätsmodell nach BOUSSINESQ [14]. Dabei wird die Belegung des Spannungstensors in Abhängigkeit vom mittleren Strömungsfeld gewählt. Lineare Wirbelviskositätsmodelle errechnen aus den Geschwindigkeitsgradienten unter Annahme einer turbulenten Wirbelviskosität  $\nu_t(\vec{x}, t)^T$  die Werte des Reynolds-Spannungstensors wie folgt:

$$-\overline{c'_i c'_j} = \nu_t \left( \frac{\partial \bar{c}_i}{\partial x_j} + \frac{\partial \bar{c}_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} k, \quad (2.9)$$

$$\text{mit } k = \frac{1}{2} \overline{c'_i c'_i}. \quad (2.10)$$

$k$  beschreibt die turbulente kinetische Energie, also die kinetische Energie der ungerichteten Störung der Geschwindigkeit. Das Kronecker-Symbol  $\delta_{ij}$  verknüpft den zweiten Term der rechten Seite mit der Diagonale des Tensors. Die Nebeneinträge des Reynolds-Spannungstensors werden nicht berücksichtigt, die Hauptdiagonale wird auf einen einzelnen Wert reduziert. Das Problem wird somit netto um vier Größen reduziert, statt der sechs Einträge im Reynolds-Spannungstensor müssen ersatzweise sinnvolle Werte für die Wirbelviskosität und die turbulente kinetische Energie ermittelt werden.

Ein sehr weit verbreitetes Modell nach der Wirbelviskositätsannahme ist das  $k$ - $\varepsilon$ -Modell nach LAUNDER UND SPALDING [35], das sich bei der Simulation von Turbomaschinen bereits vielfach bewährt hat und das in dieser Arbeit hauptsächlich verwendet wird.

### $k$ - $\varepsilon$ -Modell

Im  $k$ - $\varepsilon$ -Modell wird die Wirbelviskosität  $\nu_t$  im wesentlichen als Quotient der turbulenten kinetischen Energie  $k$  und der spezifischen Dissipationsrate  $\varepsilon$  berechnet,

$$\nu_t = C_\mu \frac{k^2}{\varepsilon}, \quad (2.11)$$

mit  $C_\mu$  als Modellkonstante. Für  $k$  und  $\varepsilon$  werden zusätzliche Transportgleichungen aufgestellt:

$$\frac{\partial k}{\partial t} + \bar{c}_i \frac{\partial k}{\partial x_i} = \frac{\partial}{\partial x_i} \left[ \left( \nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_i} \right] + P_k - \varepsilon, \quad (2.12)$$

$$\frac{\partial \varepsilon}{\partial t} + \bar{c}_i \frac{\partial \varepsilon}{\partial x_i} = \frac{\partial}{\partial x_i} \left[ \left( \nu + \frac{\nu_t}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_i} \right] + \frac{C_{\varepsilon 1} f_1 P_k - C_{\varepsilon 2} f_2 \varepsilon}{T}. \quad (2.13)$$

In Gleichung (2.12) und (2.13) ist  $P_k$  die Produktionsrate der turbulenten kinetischen Energie und wird definiert zu:

$$P_k = -\overline{c'_i c'_j} S_{ij} = 2\nu_t S_{ij} S_{ij}, \quad (2.14)$$

$$\text{wobei } S_{ij} = \frac{1}{2} \left( \frac{\partial \bar{c}_i}{\partial x_j} + \frac{\partial \bar{c}_j}{\partial x_i} \right) \quad (2.15)$$

Die Konstanten des  $k$ - $\varepsilon$ -Modells ergeben sich zu:

$$C_\mu = 0.09, \quad C_{\varepsilon 1} = 1.44, \quad C_{\varepsilon 2} = 1.92, \quad \sigma_k = 1.0, \quad \sigma_\varepsilon = 1.3. \quad (2.16)$$

Für die Turbulenzmodellierung ist es sinnvoll, die effektive Viskosität  $\nu_{\text{eff}}$  zu definieren, die sich aus der laminaren und der turbulenten zusammensetzt:

$$\nu_{\text{eff}} = \nu + \nu_t, \quad (2.17)$$

Somit kann der Reibungseinfluß der Turbulenz recht gut abgeschätzt werden.

## Wandbehandlung

Für die Betrachtung der Geschwindigkeiten in Wandnähe ist es sinnvoll, den dimensionslosen Wandabstand  $y^+$  und die dimensionslose Geschwindigkeit  $u^+$  zu definieren:

$$u^+ = \frac{\bar{u}_t}{u_\tau} , \quad (2.18)$$

$$y^+ = \frac{y_P u_\tau}{\nu} , \quad (2.19)$$

$$\text{wobei } u_\tau = \sqrt{\frac{\tau_w}{\rho}} . \quad (2.20)$$

Dabei ist  $\bar{u}_t$  die gemittelte tangentielle Geschwindigkeitskomponente der Strömung,  $y_P$  der Abstand des Mittelpunktes der wandnächsten Zelle und  $u_\tau$  die Schubspannungsgeschwindigkeit.

Die turbulente Strömung kann normal zu einer Wand in drei unterschiedliche Bereiche eingeteilt werden:

- Viskose Wandschicht
- Logarithmischer Bereich
- turbulente Kernströmung oder Außenströmung

Dabei gilt für die dimensionslose Geschwindigkeit  $u^+$  über den Wandabstand  $y^+$  der Zusammenhang aus Abbildung 2.1. In der viskosen Unterschicht entwickelt sich das Geschwindigkeitsprofil laminar, d.h.  $u^+ = y^+$ , dieser Zusammenhang gilt in einem Bereich bis  $y^+ \approx 5$ . Darüber geht das Geschwindigkeitsprofil über in den logarithmischen Bereich, d.h. es folgt dem *Universellen Logarithmischen Wandgesetz*:

$$u^+ = \frac{1}{\kappa} \ln y^+ + C . \quad (2.21)$$

Dabei beträgt der Wert der Kármán-Konstante  $\kappa \approx 0.41$ , die Konstante  $C$  gibt die Rauigkeit der Wand wieder, im hydraulisch glatten Fall ist  $C \approx 5.2$ . Der logarithmische Bereich, in dem Gleichung (2.21) gilt, wird mit  $y^+ \in [30; 500]$  angegeben.

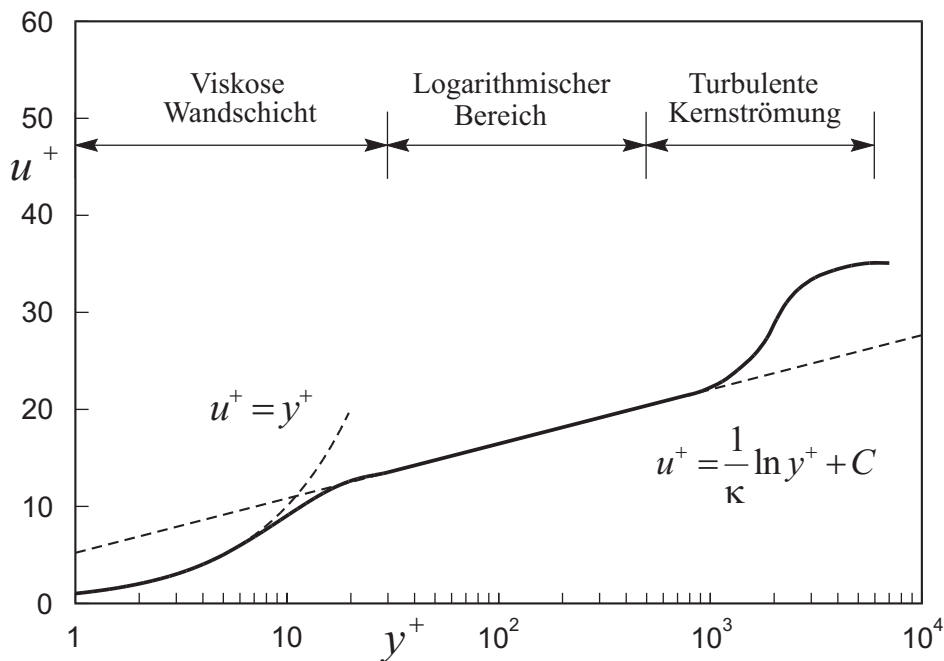
Strömungssimulationen, die das Geschwindigkeitsprofil in Wandnähe komplett auflösen und somit den Wandeinfluß ohne weiteres Modell abbilden, müssen für den Mittelpunkt der wandnächsten Zellen  $y^+$ -Werte von ca. 1 einhalten, d.h. sie müssen das gesamte Geschwindigkeitsprofil normal zur Wand abbilden. Bei großen Reynoldszahlen, die in Turbomaschinen meist erreicht werden, sind die Grenzschichten sehr dünn, so dass der Aufwand steigt, die Grenzschicht mit dem Rechengitter aufzulösen. Es ist sinnvoll, den Wandeinfluß über eine Wandfunktion abzubilden, die den Einfluß der Schubspannung bis in den logarithmischen Bereich widerspiegelt. Dadurch muß das Rechengitter nur bis in den logarithmischen Bereich aufgelöst werden. In diesem gilt:

$$u_\tau = C_\mu^{1/4} \sqrt{k} , \quad (2.22)$$

da die Strömung ein lokales Gleichgewicht aufweist, in dem sich die Produktion der turbulenten kinetischen Energie  $P_k$  und deren Dissipation  $\varepsilon$  gegenseitig aufheben, vgl. DURBIN UND PETTERSON REIF [18].

Die Grenzen des logarithmischen Bereichs sind problemabhängig, die Untergrenze wird von TRUCKENBRODT [54] mit  $y^+ = 60$  angegeben, die Obergrenze kann bei  $y^+ = 100$  liegen, für hohe Reynoldszahlen kann sie auch  $y^+ = 1000$  betragen, vgl. ADAMS [9].

Die  $y^+$ -Werte in einer Strömungssimulation sind Ergebnisse, d.h. sie können erst im Nachhinein ermittelt werden, wenn die Berechnung des gesamten Strömungsfelds vorliegt. Anhand der  $y^+$ -Werte muß das Simulationsergebnis geprüft werden, ob es die Forderungen des Wandreibungsmodells erfüllt. Ist dies nicht der Fall, so muß die Simulation mit angepaßten Netzen wiederholt werden. Im Rahmen der vorliegenden Arbeit wird die Wandreibung über die Wandfunktion abgebildet, für die wandnächste Zelle wird das Intervall  $y^+ \in [30; 300]$  angestrebt.



**Abbildung 2.1:** Geschwindigkeitsprofil einer turbulenten Grenzschicht an einer hydraulisch glatten Wand

#### 2.1.4 Quasi-Navier-Stokes-Verfahren (QNS)

In Abschnitt 2.1.2 wird das RANS-Verfahren inklusive des Schließungsproblems des Reynolds-Spannungstensors vorgestellt. Im Abschnitt 2.1.3 werden die Ansätze zur Lösung dieses Schließungsproblems skizziert, die Turbulenzmodelle. Sie führen zusätzliche skalare Größen ein, die zusammen mit dem hydrodynamischen Grundsystem gelöst werden müssen, und deren Konvergenz Rückwirkungen auf das Konvergenzverhalten des Gesamtsystems hat. In Gleichung (2.11) in Verbindung mit Gleichung (2.17) wird die Rückwirkung auf das Strömungs- und Druckfeld beschrieben. Mit der Annahme,



dass die effektive Viskosität konstant ist, kann auf die Lösung der Modellgleichungen für  $k$  und  $\varepsilon$  verzichtet werden:

$$\nu_t \approx \text{const} \implies \nu_{\text{QNS}} = C_t \cdot \nu. \quad (2.23)$$

Die Turbulenzkonstante  $C_t$  pauschaliert die Verluste durch die Turbulenz und ist für jede Problemstellung zu ermitteln. Generell ist anzunehmen, dass die turbulente Viskosität für den Anwendungsfall der hydraulischen Turbomaschine deutlich größer als die laminare Viskosität ist:

$$\begin{aligned} C_t &\in [10 \dots 1000] \\ \implies \nu_{\text{QNS}} &\gg \nu \\ \implies Re_{\text{QNS}} &\ll Re. \end{aligned} \quad (2.24)$$

In Verbindung mit Gleichung (2.3) wird klar, dass nicht nur zwei zu lösende Gleichungen wegfallen, sondern auch eine zusätzliche Ersparnis hinsichtlich der notwendigen Auflösung des Rechengebiets auftritt. Es ist zu erwarten, dass mit dem QNS-Verfahren deutlich schneller Lösungen ermittelt werden können als mit RANS-Verfahren, die Turbulenzmodelle benutzen.

Die reynoldsgemittelten Navier-Stokes-Gleichungen lassen sich zu den Quasi-reynoldsgemittelten Navier-Stokes-Gleichungen (QNS) vereinfachen:

$$\frac{\partial \bar{c}_i}{\partial t} + \bar{c}_j \frac{\partial \bar{c}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left[ C_t \cdot \nu \left( \frac{\partial \bar{c}_i}{\partial x_j} + \frac{\partial \bar{c}_j}{\partial x_i} \right) \right]. \quad (2.25)$$

Das QNS-Verfahren wird auch als QUASI-EULER-3D-Verfahren bezeichnet, s. SCHILLING ET AL. [48], FIEREDER [22], SCHMALHORST ET AL. [49].

### 2.1.5 CFD-Solver NS3D

Der im Rahmen dieser Arbeit verwendete CFD-Solver *NS3D* ist am Lehrstuhl für Fluidmechanik entwickelt und in mehreren Arbeiten erfolgreich verwendet worden. Er ist ein blockstrukturierter, Finite-Volumen-basierter RANS-Löser, der die Druck-Geschwindigkeitskopplung mithilfe des Druckkorrekturverfahrens vornimmt.

NS3D basiert auf einem frei verfügbaren FORTRAN-Code von FERZIGER UND PERIC [21], dessen grundlegende Eigenschaften NS3D somit auch aufweist. Die implementierten Verfahren sind bei BADER [11], SKODA [50] und EINZINGER [19] umfassend dargestellt.

NS3D kann parallel auf mehreren CPUs betrieben werden, er weist einige bewährte Turbulenzmodelle auf und ist dank der Vielzahl von verfügbaren Blockinterfaces sehr gut geeignet, um die Strömung in hydraulischen Turbomaschinen zu simulieren.

## 2.2 Programmiertechnik

### 2.2.1 Modularisierung komplexer Programmsysteme

Die geforderten Funktionen eines Programmsystems zur umfassenden Modellierung, Simulation und Optimierung von Turbomaschinen sollen unter einer einheitlichen grafischen Benutzeroberfläche zur Verfügung gestellt werden. Dabei variieren die Anforderungen an die einzelnen Programmteile stark: der CFD-Löser muß anderen Anforderungen genügen als ein Programmteil zur Manipulation der Geometrie der zu simulierenden bzw. optimierenden Turbomaschine.

Bei den laufezeitkritischen Anwendungen, also im wesentlichen beim CFD-Löser, ist ein nativ in Maschinencode vorliegendes Programm notwendig, das direkt ohne Umwege über Abstraktionsschichten ausgeführt werden kann. Die programmiertechnischen Fragestellungen bei der Implementierung eines CFD-Lösers sind in erster Linie der Speicherverbrauch und das Laufzeitverhalten. Diese Kern-Eigenschaften der Implementierung des Verfahrens sind weitaus wichtiger als Forderungen an Wartbarkeit und Erweiterbarkeit.

Die Programmteile, mit denen der Benutzer interagiert, sind i.d.R. ausreichend schnell, so dass hier das Augenmerk auf Bedienungsfreundlichkeit und -sicherheit liegt. Im wesentlichen warten diese interaktiven Programmteile auf Benutzereingaben, die eigentliche Datenverarbeitung ist deutlich schneller, als der Benutzer reagieren kann. Einzige Ausnahme sind aufwendige geometrische Operationen, denen ein größeres Augenmerk hinsichtlich der Laufzeit geschenkt werden muß.

Die Programmteile, die die grafische Benutzeroberfläche, engl. Graphical User Interface (GUI), implementieren, sollen außerdem leicht erweiterbar sein, d.h. es soll die Möglichkeit bestehen, dass das bestehende Programmsystem von Dritten um weitere Funktionen erweitert werden kann. Diese Funktionen können im konkreten Fall weitere Methoden der Netzgenerierung, der Auswertung von Strömungslösungen oder andere numerische Löser für weitergehende Problemstellungen sein, wie etwa Fluid-Struktur-Interaktion, Aero- oder Vibroakustik.

Mit der Modularisierung von Programmsystemen werden mehrere Ziele verfolgt, vgl. z.B. BOUDREAU ET AL. [13]:

**Wiederverwendbarkeit:** Software sollte in Form von Komponenten vorliegen, die in beliebiger, sinnvoller Form ohne große Anpassungen neu kombinierbar sein sollen.

**Erweiterbarkeit:** Existiert eine Schnittstellendefinition zwischen Modulen, so können neue Module erzeugt werden, die dieser Schnittstellendefinition genügen. Je nach Schnittstelle können neue Funktionen hinzugefügt werden.

**Wartbarkeit:** Einzelne Programmmodule können gegen neuere, verbesserte oder fehlerbereinigte Versionen ausgetauscht werden. Werden Module in mehreren Programmsystemen verwendet, so profitieren alle diese Programmsysteme von Verbesserungen in einem einzelnen Modul.

**Arbeitsteilung:** Die Aufteilung eines komplexen Gesamtprogramms kann in einzelne, weniger komplexe Module aufgeteilt werden. Durch fix definierte Modulschnittstellen können unterschiedliche Entwickler unterschiedliche Teilaspekte eines Programmsystems implementieren. Stehen die Schnittstellen zwischen den Modulen über einen längeren Zeitraum fest, so ist es nicht unbedingt notwendig, dass die verschiedenen Entwickler zusammenarbeiten.

Die Modularisierung eines Programmsystems benötigt anders als das monolithische Programm eine zusätzliche Verwaltungsfunktion für die Module. Sie muß z.B. Abhängigkeiten unter den Modulen kennen, die Schnittstellen für das Zusammenspiel der einzelnen Module bereitstellen und Module aktivieren bzw. deaktivieren können.

Module sind von Programmbibliotheken zu unterscheiden: Programmbibliotheken stehen i.d.R. bereits zur Übersetzungszeit komplett zur Verfügung und werden fest mit dem Programm verbunden. D.h. es wird zur Übersetzungszeit des Programms eine spezielle Implementierung zur Nutzung vorgesehen, die evtl. zur Ausführungszeit dynamisch hinzugeladen wird. Das nutzende Programmsystem erklärt seine Abhängigkeit von der konkreten Implementierung der Bibliothek im Vorfeld. Ein derart implementiertes Programm ist somit immer an eine bestimmte Bibliothek gebunden.

Für eine freiere Kombinierbarkeit von Programmteilen ist es notwendig, dass die Abhängigkeiten nicht bereits zur Übersetzungszeit festgelegt werden, sondern zur Ausführungszeit ermittelt werden können. Das bedeutet, dass ein Modul die Funktionen eines erst später erstellten anderen Moduls nutzen kann. Dieses Verhalten kann nach FOWLER [23] mit „Dependency Injection“ erreicht werden. Im Fall der Nutzung einer Bibliothek sucht das abhängige Modul aus den angebotenen Lösungen eine bestimmte heraus und nutzt genau diese. Das Modul A, das die Funktionen eines anderen Moduls nutzen möchte, erklärt bei Dependency Injection die abstrakte Beschreibung der Lösung. Es definiert, wie es ein anderes Modul ansprechen wird und was es als Ergebnis von ihm erwartet. Diese Beschreibung ist abstrakt, weil sie keinerlei Implementierungsdetails enthält. Ein Modul B, das diese Implementierung bereitstellt, füllt die abstrakt beschriebene Lösung nun konkret mit eigenen Algorithmen aus. Nun ist ein dritter Programmteil notwendig, der die Forderungen des Moduls A und die vom Modul B angebotene Lösung während der Ausführungszeit zusammenbringt. Dieses von FOWLER Assembler genannte Programmteil kann sich z.B. einfacher Textdateien bedienen, die es während der Laufzeit auswertet und die ohne spezielle Hilfsmittel angepaßt werden können.

Als Beispiel sei ein Modul genannt, das für eine Schaufel einer Turbomaschine das Gewicht berechnet. Die Schaufel sei durch eine abstrakte Klasse `Blade` beschrieben, deren Methoden die Oberflächengeometrie beschreiben. Die Implementierung der Funktion „Berechnung der Masse“ bezieht sich allein auf die Methoden der abstrakten Klasse. Nun ist es unerheblich, wie ein anderes Modul das Objekt `Blade` tatsächlich implementiert: es kann z.B. aus fest definierten Punkten bestehen, es kann aus der Skelettfläche und der Dickenverteilung erzeugt werden, oder es kann rein parametrisch aus Einzelprofilen erstellt werden. Nun kann für jede Schaufel, die die abstrakte Klasse `Blade` implementiert, die Masse berechnet werden. Es können auch weitere Module auf diese abstrakte Klasse zugreifen, und andere Funktionen dazu bereitstellen, wie z.B. Erstellen eines Netzes, Ermittlung des Trägheitstensors o.ä.

## 2.2.2 Frameworks

Viele komplexe Programmsysteme weisen neben der Kernfunktionalität eine Reihe von Komfortfunktionen auf, die die Verwendung vereinfachen und somit die Akzeptanz bei einem möglichen Anwender erhöhen können. Viele dieser implizit geforderten Funktionen sind nicht anwendungsspezifisch, sondern werden in mehreren unterschiedlichen Zusammenhängen verwendet. Beispiele für diese Komfortfunktionen sind z.B., ohne Anspruch auf Vollständigkeit:

- Änderungen rückgängig machen bzw. wiederherstellen
- Online-Hilfesysteme, evtl. kontextbezogen
- Template-System für Konfigurationsdateien
- Anpassbarkeit des Oberflächenlayouts
  - Schaltflächen
  - Menüpunkte
  - Tastaturkürzel
- Syntaxabhängige Darstellung von Textdateien
- Wiederherstellen des alten Programmzustands bei erneutem Starten

Dies sind reine Komfort- und Ergonomiefunktionen, die für die eigentliche Problemstellung nicht von Bedeutung sind. Sie erhöhen aber die Akzeptanz des Programmsystems beim Benutzer und verringern die Einarbeitungszeit. Ein Online-Hilfetext kann die Benutzungssicherheit erhöhen, da durch Bereitstellen von Zusatzinformationen zum richtigen Zeitpunkt Fehleingaben durch den Benutzer vermieden werden können.

In der Regel sind einzelne Funktionen, die in einem GUI-gestützten Programm aufgerufen werden können, auf verschiedenen Wege erreichbar, über:

1. globalen Menübefehl,
2. Schaltflächen,
3. Tastaturkürzel.

Alle drei Wege sollen zum Aufruf ein und desselben Codes führen. Es bietet sich an, einen einheitlichen Weg der Zuordnung einer Aktion zu diesen möglichen Aufrufmethoden zu finden. Lösungen, die eine Infrastruktur bereitstellen, so dass die Implementierung einer Aktion unter diesen Aufrufmethoden registriert werden können, werden „Framework“ genannt.

Als Implementierungsbasis wurde die Programmiersprache Java gewählt. Java ist abhängig davon, dass die standardisierte Java-Laufzeitumgebung „Java Virtual Machine“ (JVM) für ein Betriebssystem verfügbar ist. Die derzeit gängigen Betriebssysteme Microsoft Windows in den Versionen von XP, Vista und 7, Linux und Mac OS X sowie

einige Unix-Versionen verfügen über diese Java-Laufzeitumgebung, wodurch sich Java als Implementierungssprache für die nicht zeitkritischen Systembestandteile anbietet.

Für die Implementierungssprache Java stehen einige Frameworks zur Verfügung, u.a.:

**Swing Application Framework [17]:** Das Swing Application Framework ist als Java Standard Request (JSR) 296 in einer Referenzimplementierung vorhanden und soll somit standardisierter Bestandteil der Plattformdefinition von Java werden. O'CONNOR [39] beschreibt das Swing Application Framework als den Kern einer typischen Swing-Anwendung, mit dem ein Programmierer einen schnellen Einstieg bekommen kann und Lösungen für die gemeinsamen Elemente erhält: Architektur, Lebenszyklus- und Ressourcenmanagement, Eventmanagement, Nebenläufigkeit, Sitzungen und lokales Speichern. Das Swing Application Framework war als fester Sprachbestandteil von Java vorgesehen, ist aber für die nächste Version 7 herausgenommen worden, s. [7], das Framework selbst wird nicht weiterentwickelt.

Swing ist die Java-spezifische Implementierung von Bedienelementen einer grafischen Benutzeroberfläche wie z.B. Schaltern, Menüpunkten, Fenstern, Listen, Tabellen, etc.

**Eclipse Rich Client Platform [1]:** Eclipse ist vor allem als Entwicklungsumgebung (Integrated Development Environment, IDE) für Java-Programme bekannt. Die IDE ist selbst in Java geschrieben und kann als eigenständiges Framework verwendet werden. Eclipse ist quelloffen und kann im Rahmen der APACHE LIZENZ 2.0 [2] frei verwendet und modifiziert werden. Eclipse verwendet für die grafische Benutzeroberfläche das eigene SWT-Toolkit, das eine betriebssystemspezifische Implementierung erfordert.

**Netbeans Rich Client Platform [8]:** Netbeans ist ebenfalls eine Java-IDE, die wiederum selbst in Java implementiert worden ist. Es ist ebenfalls quelloffen verfügbar und unter einer kombinierten CDDL-GPL-LIZENZ [3] frei verwendbar und anpassbar. Anders als Eclipse verwendet es nur Swing und ist somit unabhängig vom Betriebssystem in jeder JVM einsetzbar. Netbeans RCP hat ein umfangreiches Modulsystem, das Versionierung, Aktualisierung und die Definition von Abhängigkeiten unterstützt. Es hat eine Vielzahl von Modulen für die Darstellung von GUI-Komponenten und für die Verwaltung von Dateien und deren Meta-Daten. Weiterführende Literatur zu Netbeans RCP ist deutschsprachig z.B. von BÖCK [12] oder englischsprachig von BOUDREAU ET AL. [13] verfügbar.

Die beiden letzten Frameworks haben den Vorteil, dass sie ausreichend viele Anwender haben und von großen Firmen unterstützt werden, so dass eine Perspektive besteht, dass diese Frameworks ausreichend lange weiterentwickelt werden. Zusätzlich ist im Internet eine umfangreiche Dokumentation für die Programmierung und Konfiguration erhältlich, und es gibt auch vielfältige weiterführende Literatur. Neben den hier aufgeführten existieren noch einige weitere Frameworks, wobei oftmals der Schwerpunkt auf die Unterstützung von serverbasierten Web-Applikationen liegt. Für die Implementierung des Programmsystems wurde das Netbeans RCP Framework gewählt, da es keine plattformspezifischen Elemente enthält.

### 2.2.3 Universelle Schnittstellendefinition

Ein Kernkonzept von Netbeans ist der sog. **Lookup**. Der **Lookup** ist ein Containerobjekt, das Instanzen von Objekten beinhaltet. Der Container kann abgefragt werden, ob er Instanzen zu bestimmten Klassendefinitionen enthält. Ein Programmteil, der den **Lookup** verwendet, kann von diesem ein Event erhalten, falls der Inhalt des **Lookup** sich ändert. Dieses Event ist mit einer Klassendefinition verbunden, so dass ein interessiertes Programmteil nur dann eine Nachricht erhält, wenn der Inhalt des **Lookup** sich hinsichtlich einer bestimmten Klassendefinition geändert hat.

Der **Lookup** wird bei Netbeans als universelle Schnittstelle zwischen Modulen verwendet. Ein Modul stellt einen **Lookup** zusammen, d.h. es erzeugt eine Sammlung von Instanzen, die über ihre jeweilige (abstrakte) Klasse markiert sind. Dieser Satz von Objekten wird anderen Modulen übergeben, die den Inhalt des **Lookup** prüfen können. Sind für bestimmte Funktionen notwendige Objekte enthalten, so kann dieses Modul mit diesen Objekten als Parameter aktiv werden.

In Netbeans verfügen fast alle GUI-Elemente über einen **Lookup**. Über diesen wird der Inhalt des jeweiligen Elements veröffentlicht. Je nachdem, welche GUI-Elemente vom Benutzer aktiviert werden, wird ein globaler **Lookup** zusammengestellt, der den Kontext der Benutzereingaben darstellt. Programmteile, die Aktionen auf Objekte definieren (**Actions**) können von diesem globalen **Lookup** Nachrichten beziehen, die als Parameter für diese Programmteile benötigt werden, wenn der Benutzer einen Inhalt aktiviert. Die **Actions** können sich abhängig von diesem Kontext selbst aktivieren und deaktivieren, wodurch die Benutzereingaben sinnvoll eingeschränkt werden. Der Benutzer stellt mit seiner Selektion also die Parameter zusammen, die die **Actions** als Parameter entgegennehmen. In Listing 2.1 ist skizziert, wie die tatsächliche Implementierung einer Action aussieht, die auf die Eingabe des Benutzers reagiert.

```
package de.diss;

public class DemoAction extends AbstractAction implements
    ContextAwareAction
{
    private Lookup lkp;

    public DemoAction() {
        this(Utilitites.actionsGlobalContext());
    }

    public DemoAction(Lookup lkp) {
        this.lkp = lkp;
    }

    public Action createContextAwareAction(Lookup lkp) {
        return new DemoAction(lkp);
    }

    public void actionPerformed(ActionEvent evt) {
```

```

    //Blade aus dem Lookup holen
    Blade bl = lkp.lookup(Blade.class);
    //nun kann bl beliebig verwendet werden.
}
}

```

**Listing 2.1:** Beispiel einer kontextsensitiven Action

Die Klasse `DemoAction` benutzt die Java-eigene Klasse `AbstractAction`, um die Basisfunktionalität bereitzustellen. Die Methode `actionPerformed` ist in dieser abstrakten Klasse nicht definiert und muß von `DemoAction` implementiert werden. Diese Methode beinhaltet den eigentlichen Algorithmus, den `DemoAction` dem Anwender anbietet.

Der Standard-Konstruktor ohne Parameter verweist auf den Konstruktor `DemoAction(Lookup lkp)`, und belegt diesen mit dem `Lookup` aus `Utilities.actionsGlobalContext()`. Dieser `Lookup` spiegelt die aktuelle Selektion in der grafischen Benutzeroberfläche wider. Die Methode `createContextAwareAction(Lookup lkp)` ist vom Netbeans-Modul definiert, das die einzelnen Aktionen zusammensucht und sie in die Menüstruktur einordnet, die Schaltflächen erstellt und die Tastenkürzel verwaltet. In Listing 2.2 ist skizziert, wie die Action per Konfigurationsdatei im Framework angemeldet wird. Die Zeile `<attr name="type" stringvalue="de.diss.Blade"/>` bestimmt dabei, welche Klasse im relevanten `Lookup` vorliegen muß, damit die `DemoAction` aktiv ist und aufgerufen werden kann. Die Zuordnung der `DemoAction` zu einem Tastenkürzel und die Einordnung in die Menüstruktur geschieht ebenfalls mittels Einträgen in einer Konfigurationsdatei.

```

<file name="de-diss-DemoAction.instance">
  <attr name="delegate" newvalue="de.diss.DemoAction"/>
  <attr name="instanceCreate" methodvalue="org.openide.awt.
    Actions.context"/>
  <attr name="type" stringvalue="de.diss.Blade"/>
</file>

```

**Listing 2.2:** Deklarative Registrierung der Action aus Listing 2.1 mittels Konfigurationsdatei

## 2.3 Geometriedarstellung

### 2.3.1 Kubische Splines

Kubische Splines stellen ein Verfahren zur Interpolation einer vorgegebenen Anzahl von Punkten  $P_i = (p_{xi}, p_{yi}, p_{zi})^T$  dar. Dabei wird eine parametrische Kurve  $\vec{x}(t) = (x(t), y(t), z(t))^T$  eingeführt, die einerseits durch die Punkte  $P_i$  gehen soll, andererseits auch mindestens zweimal stetig differenzierbar sein soll (krümmungsstetig).

Nach HOSCHEK UND LASSER [29] ist der kubische Spline beschrieben durch  $N + 1$  Punkte  $P_i$  sowie den zugeordneten Parameterwerten  $t_i \in [a; b]$ , die aufsteigend sortiert sein sollen:

$$a = t_0 < t_1 < \dots < t_{N-1} < t_N = b \quad (2.26)$$

$$\Delta t_i := t_{i+1} - t_i \quad (2.27)$$

Jedes benachbarte Punktepaar  $P_{i+1}, P_i$  wird über ein kubisches Kurvenstück verbunden:

$$x_i(t) = a_i(t - t_i)^3 + b_i(t - t_i)^2 + c_i(t - t_i) + d_i \quad (2.28)$$

$$t \in [t_i; t_{i+1}]$$

Es gilt die Forderung, dass für alle inneren Parameterwerte  $t = t_i, i \in [1; N - 1]$  folgende Bedingungen zu erfüllen sind:

$$x_i(t_{i+1}) = x_{i+1}(t_{i+1}) = p_{xi}, \quad (2.29)$$

$$x'_i(t_{i+1}) = x'_{i+1}(t_{i+1}), \quad (2.30)$$

$$x''_i(t_{i+1}) = x''_{i+1}(t_{i+1}), \quad (2.31)$$

d.h. die Funktionswerte sind identisch, die parametrische Kurve ist in den Schnittstellen der einzelnen Polynome tangential- und krümmungsstetig. Das Gleichungssystem, das sich aus den Gleichungen (2.28), (2.29)ff. ergibt, lautet wie folgt:

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ \Delta t_i^3 & \Delta t_i^2 & \Delta t_i & 1 \\ 0 & 2 & 0 & 0 \\ 6\Delta t_i & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 3\Delta t_i^2 & 2\Delta t_i & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} a_i \\ b_i \\ c_i \\ d_i \end{pmatrix} = \begin{pmatrix} x_i \\ x_{i+1} \\ x'_i \\ x'_{i+1} \\ x''_i \\ x''_{i+1} \end{pmatrix} \quad (2.32)$$

Dabei ist  $\Delta t_i = t_{i+1} - t_i$ . Für  $N + 1$  Punkte werden nun  $N$  Kurvenstücke  $x_i$  gesucht, die durch insgesamt  $4N$  Unbekannte  $a_i, b_i, c_i$  und  $d_i$  beschrieben werden. Dazu stehen nur  $4N - 2$  Gleichungen zur Verfügung. Unter Vorgabe von  $x'_0(t_0)$  und  $x'_{N-1}(t_N)$  oder  $x''_0(t_0)$  und  $x''_{N-1}(t_N)$  kann das Gleichungssystem bestimmbar gemacht werden. In der Regel wird an den Rändern keine Krümmung vorgegeben:

$$x''_0(t_0) = 0 \quad (2.33)$$

$$x''_{N-1}(t_N) = 0 \quad (2.34)$$

Kubische Splines, die mit dieser Randbedingung definiert sind, werden natürliche Splines genannt. Für eine räumliche Kurve werden drei Gleichungssysteme aufgestellt, d.h. Gleichung (2.32) wird für alle  $x_i, y_i$  und  $z_i$  aufgestellt und gelöst.



## Parametrisierung

In der Anwendung wird für die Interpolation einer Punktmenge nur die Reihenfolge vorgegeben, in der sie durchlaufen werden soll. Die zugehörigen Parameterwerte sind i.d.R. frei wählbar. Die Erzeugung der Parameterwerte  $t_i$  wird Parametrisierung genannt. Die Parametrisierung hat einen großen Einfluß auf die Kurvengestalt, s. Abbildung 2.2.

Die naheliegende Parametrisierung ist die äquidistante:

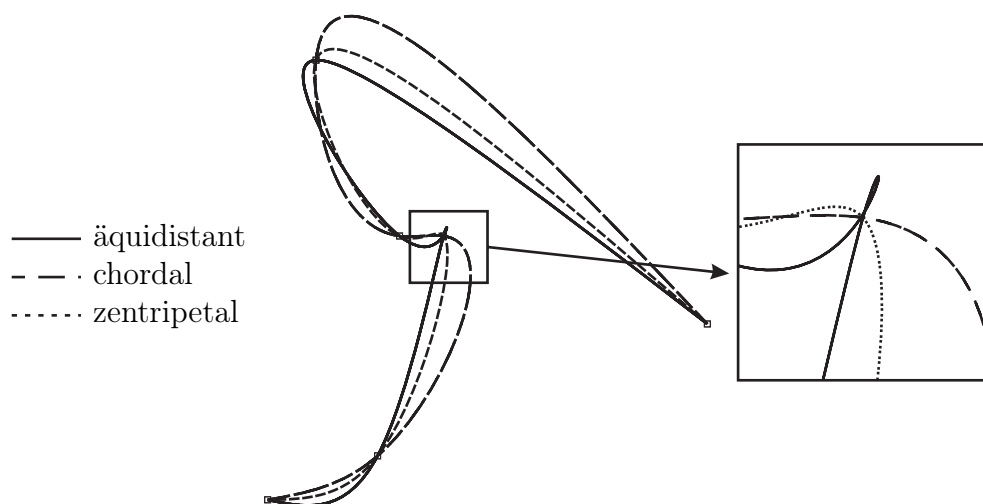
$$\begin{aligned} t_0 &= 0 \quad , \quad t_N = 1 \quad , \\ t_i &= \frac{i}{N} \quad , \quad i \in [1, N - 1] \quad . \end{aligned} \quad (2.35)$$

Die chordale Parametrisierung berücksichtigt die Abstände der Punkte untereinander:

$$\begin{aligned} t_0 &= 0, \quad t_N = 1 \quad , \\ d &= \sum_{i=0}^{N-1} |P_{i+1} - P_i| \quad , \\ t_i &= t_{i-1} + \frac{|P_{i+1} - P_i|}{d} \quad \text{für } i \in [1, N - 1] \quad . \end{aligned} \quad (2.36)$$

Die zentripetale Parametrisierung ist eine Abwandlung der chordalen:

$$\begin{aligned} t_0 &= 0, \quad t_N = 1, \quad d = \sum_{i=0}^{N-1} \sqrt{|P_{i+1} - P_i|} \quad , \\ t_i &= t_{i-1} + \sqrt{\frac{|P_{i+1} - P_i|}{d}} \quad \text{für } i \in [1, N - 1] \quad . \end{aligned} \quad (2.37)$$



**Abbildung 2.2:** Kubische Splines: äquidistante, chordale und zentripetale Parametrisierung

Für Anwendungen im Strömungsmaschinenbereich haben sich die chordale und die zentripetale Parametrisierung bewährt. Abbildung 2.2 zeigt, dass diese beiden Parametrisierungen deutlich weniger zum Überschwingen neigen als die äquidistante. PIEGL UND TILLER [41] empfehlen die zentripetale Parametrisierung für Punktmengen, bei deren Interpolation mit starken Krümmungen zu rechnen ist.

### 2.3.2 B-Splines

Eine weitere parametrische Formulierung einer Kurve ist der B-Spline, eine Verallgemeinerung der Bézier-Kurve. Der B-Spline ist definiert durch ein Kontrollpolygon  $P_i$ , die Ordnung  $k$  und einen Knoten- oder Trägervektor  $T$ . ROGERS [44] definiert den B-Spline wie folgt:

$$\vec{x}(t) = \sum_{i=0}^n P_i N_{i,k-1}(t), \quad a \leq t \leq b, \quad 2 \leq k \leq n+1 \quad . \quad (2.38)$$

Dabei ist  $N_{i,k}(t)$  die rekursiv definierte Basisfunktion:

$$N_{i,0}(t) = \begin{cases} 1 & \text{für } t \in [t_i; t_{i+1}] \\ 0 & \text{sonst.} \end{cases}$$

$$N_{i,k}(t) = \frac{(t - t_i)}{t_{i+k-1} - t_i} N_{i,k-1}(t) + \frac{(t_{i+k} - t)}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t) . \quad (2.39)$$

Quotienten im Ausdruck für  $N_{i,k}(t)$ , die im Nenner den Wert 0 ergeben, werden auf 0 gesetzt. Die Knoten  $t_i$ , die den einzelnen Punkten  $P_i$  zugeordnet werden, sind in einem Knotenvektor  $T$  definiert:

$$t_i \leq t_{i+1}$$

$$T = \{ \underbrace{a, \dots, a}_{k\text{-fach}}, t_k, t_{k+1}, \dots, t_{m-k}, \underbrace{b, \dots, b}_{k\text{-fach}} \}, \quad m = n + k \quad (2.40)$$

Aus der rekursiven Definition der Basisfunktion (2.39) ergibt sich für ein B-Spline der Ordnung  $k$  die Basisfunktion als ein Polynom vom Grad  $p = k - 1$ . Eine B-Splinekurve der Ordnung 4 besitzt eine kubische Basisfunktion.

Ist der Knotenvektor uniform belegt, d.h. für die inneren Knoten  $t_i$ ,  $i \in [k+1; m-k]$  gilt  $\Delta t_i = t_{i+1} - t_i = \text{const}$ , so spricht man von Uniform B-Splines (UBS). Ist  $T$  nicht uniform belegt, sondern das  $\Delta t_i$  z.B. mittels chordaler oder zentripetaler Parametrisierung ermittelt, erhält man Non-Uniform B-Splines (NUBS). Werden die Kontrollpunkte mit einem Wichtefaktor  $w$  versehen, so erhält man Non-Uniform Rational B-Splines (NURBS). Im wesentlichen werden für die vorliegende Arbeit NUBS verwendet.

### Eigenschaften der NURBS

Eine umfassende Beschreibung der NURBS sowie weiterer geometrischer Operationen für NURBS findet sich bei ROGERS [44], HOSCHEK UND LASSER [29] und bei PIEGL UND TILLER [41]. Zusammenfassend können folgende Eigenschaften aufgezählt werden:

1. NURBS der Ordnung  $k$  sind  $k - 1$ -fach stetig differenzierbar, ohne dass die Ableitung verschwindet. Dabei ist die  $l$ -te Ableitung des NURBS definiert durch die  $l$ -te Ableitung der Basisfunktion:

$$\vec{x}^{(l)}(t) = \sum_{i=0}^n N_{i,k-1}^{(l)}(t) P_i$$

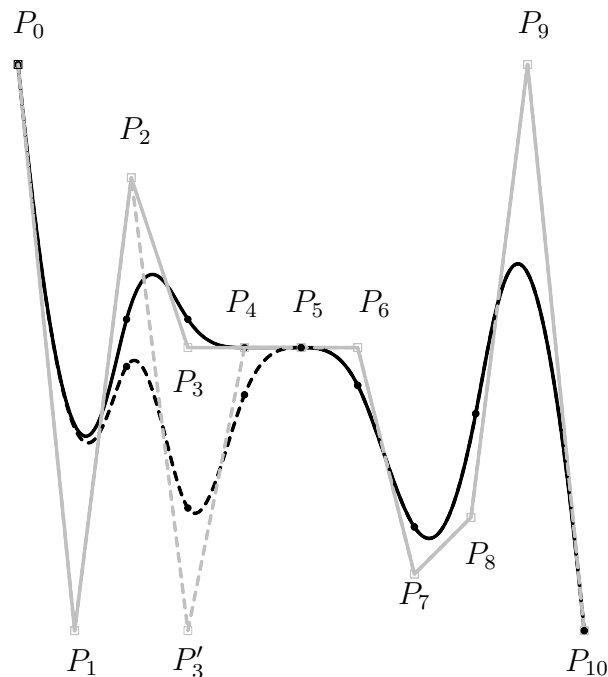
$$N_{i,k}^{(l)} = p \cdot \left( \frac{N_{i,k-1}^{(l)}}{t_{i+k} - t_i} - \frac{N_{i+1,k-1}^{(l)}}{t_{i+k+1} - t_{i+1}} \right). \quad (2.41)$$

Die Ableitung ist ebenfalls vektorwertig. Weitere Ableitungen  $l > k - 1$  haben den Wert  $\vec{0}$ .

2. Sind alle Knoten des Vektors  $T$  einfach belegt, so ist die Kurve überall  $C^{k-2}$  stetig. Bei  $m$ -facher Belegung eines Knotens  $t_i = t_{i+1} = \dots = t_{i+m}$  verringert sich in diesem Punkt die Stetigkeit auf  $C^{k-m-2}$ . Somit können durch entsprechende Belegung des Knotenvektors Unstetigkeiten wie Ecken mit NURBS abgebildet werden.
3. Die Kurve folgt der Allgemeinen Form des Kontrollpolygons, sie approximiert das Kontrollpolygon. Für  $k = 2$  fallen Kurve und Kontrollpolygon zusammen.
4. Eine affine Transformation der Kurve muß nur auf das Kontrollpolygon angewandt werden. Dadurch wird die gesamte Kurve transformiert.
5. Ein Kontrollpunkt  $P_i$  hat maximal Einfluß auf  $k$  Segmente der Kurve, jedes Kurvensegment zwischen zwei Werten des Knotenvektors wird von  $k$  Knoten beeinflusst. Dieses Kurvensegment liegt immer in der konvexen Hülle dieser Kontrollpunkte  $P_i$ .
6. Daraus folgt, wenn  $k$  Punkte kollinear sind, dann muß das Kurvensegment ein Teilstück der aus den kollinear liegenden Punkte erzeugten Gerade sein.
7. Der NURBS hat die Eigenschaft des *variation diminishing*, s. ROGERS [44]. Sie besagt, dass der NURBS nicht öfter um eine Gerade oszillieren kann als das zugrundeliegende Kontrollpolygon.

## 2.4 Geometriemodellierung

Basis des CFD-Experiments ist die Form der Turbomaschine, erweitert um die Kinematik. Es wird eine Beschreibung benötigt, mit der die Geometrie der Turbomaschine



**Abbildung 2.3:** NURBS der Ordnung  $k = 4$  mit 11 Kontrollpunkten. Die Reaktion auf Änderung von  $P_3$  zu  $P'_3$  ist gestrichelt gezeichnet. Die Segmentgrenzen sind durch Punkte gekennzeichnet. Im Ursprungs-NURBS sind  $P_3$  bis  $P_6$  kollinear, deshalb folgt das entsprechende Segment dem Kontrollpolygon.

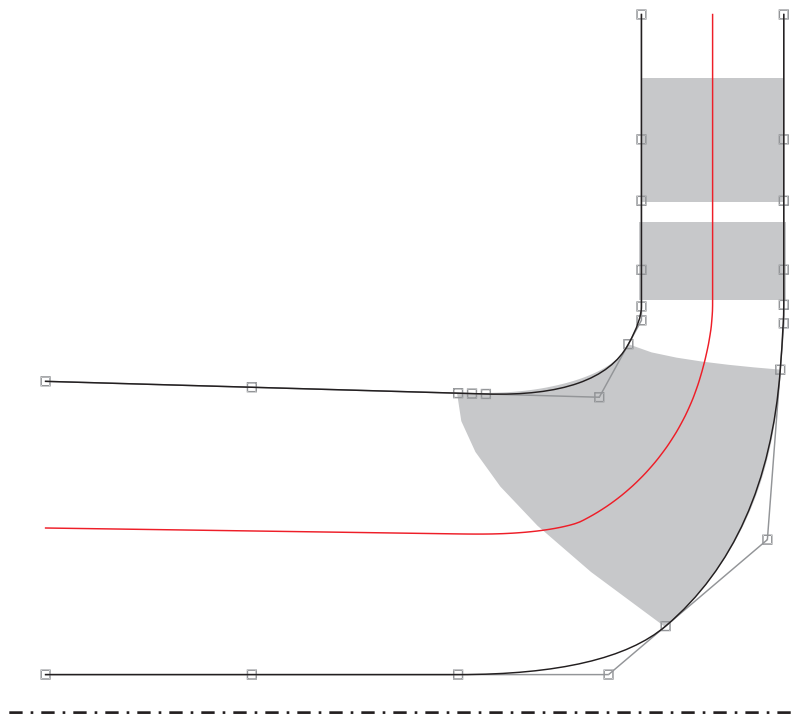
ausreichend genau modelliert werden kann, die aber auch geeignet ist, Modifikationen vorzunehmen, die das Betriebsverhalten der Maschine beeinflussen. Eng verbunden mit der Geometriemodellierung ist die Vernetzung und die Definition von Randbedingungen.

Das verwendete CFD-Verfahren basiert auf blockstrukturierten Netzen. Es ist sinnvoll, die Geometriemodellierung so zu gestalten, dass die Generierung dieser Netze ohne Konvertierungsverluste stattfinden kann. Die Anwendung des Finiten Volumenverfahrens erfordert eine Aufteilung des betrachteten Gesamtgebiets in endlich große Teilvolumina. Somit können auch nur endlich große Details berücksichtigt werden. Das verwendete Rechenetz ist immer eine Idealisierung des tatsächlich durchströmten Volumens, die Geometriemodellierung kann somit ebenfalls eine Idealisierung darstellen, die unwesentliche Details der tatsächlichen Geometrie nicht enthält.

### 2.4.1 Rotationssymmetrische Teile

Besonders bewährt bei der Beschreibung der rotationssymmetrischen Teile haben sich NURBS, s. 2.3.2. Sie werden bei LEPACH [36] und bei KRÄMER [32] erfolgreich für die Beschreibung der Meridiankontur eingesetzt. Diese Vorgehensweise wurde erwei-

tert, indem alle rotationssymmetrischen Teile mit NURBS dargestellt werden, d.h. der Radseitenraum, Saugmund bzw. Saugrohr können z.B. aus CAD-Modellen extrahiert und direkt vernetzt werden.

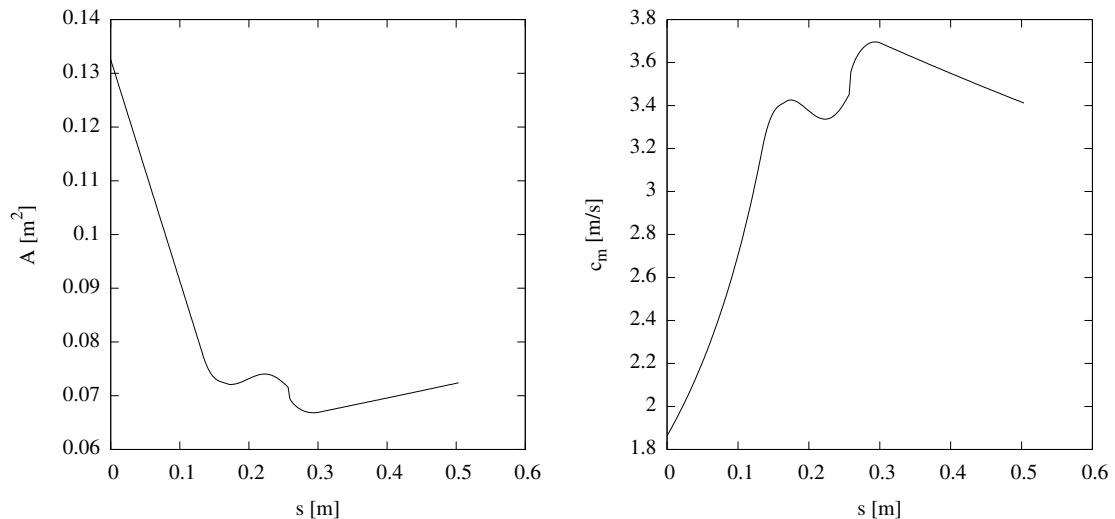


**Abbildung 2.4:** Meridiankontur bestehend aus Hub- und Shroudkontur als NURBS am Beispiel einer Modellturbine

Die Hub- und Shroudkontur sind im vorgestellten Programmsystem besonders ausgezeichnete Geometrieinformationen der Strömungsmaschine, nehmen sie doch den gesamten Nennvolumenstrom auf. So kann aus einfachen geometrischen Überlegungen eine gute Abschätzung für die mittlere Geschwindigkeit der Strömung in Meridianrichtung ermittelt werden:

$$\overline{c_m(s)} = \frac{Q}{A_m(s)}. \quad (2.42)$$

Dabei ist  $A_m(s)$  die wirksame Querschnittsfläche entlang der gedachten Laufkoordinate  $s$ ,  $c_m(s)$  die Meridiankomponente der Geschwindigkeit am Ort  $s$ . Die Ermittlung der Geschwindigkeitskomponente  $c_m$  ist notwendig bei der Festlegung der Geschwindigkeitsdreiecke. Die wirksame Querschnittsfläche kann aus zweidimensionalen Betrachtungen mit einer von KRONSNABL [33] modifizierten Delaunay-Triangulation ermittelt werden, beschrieben bei KRÄMER [32]. Ein Beispiel für einen ermittelten Flächenverlauf ist in Abbildung 2.5 abgedruckt. Der gezeigte Flächenverlauf bezieht sich auf die Meridiankontur aus Abbildung 2.4, die Versperrungswirkung der Schaufeln ist im Graphen für  $\overline{c_m}$  nicht berücksichtigt, weshalb im Bereich der Laufschaufel ein s-förmiger Verlauf zu beobachten ist.



**Abbildung 2.5:** Flächenverlauf einer Meridiankontur und daraus ermittelter  $\overline{c_m}$ -Verteilung am Beispiel einer Modellturbine

## 2.4.2 Schaufelflächen

Die Schaufelfläche wird je nach Anwendungsfall auf zweierlei Weise dargestellt. Für den allgemeinen Nachrechnungsfall genügt es, die benetzte Oberfläche der Schaufel zu beschreiben. Es hat sich bewährt, diese Fläche aufzuteilen in eine Druck- und eine Saugseite, so dass ein geometrischer Nasenpunkt entsteht. Diese Aufteilung geht auf die alternative Darstellung zurück, in der die Schaufel als Skelettfäche und Dickenüberlagerung betrachtet wird.

Alle im vorliegenden Programmsystem betrachteten Flächen sind strukturiert, d.h. sie sind topologisch viereckige Gebiete, haben also vier Berandungskurven. So können die Oberflächenkoordinaten  $(u, v)$  eingeführt werden. Alle Flächendarstellungen werden einheitlich in eine linearisierte Oberflächenform gebracht, d.h. sie werden durch ein Raster von einzelnen Punkten beschrieben, die in einheitlicher Anzahl in u- und v-Richtung vorliegen. Je nach Feinheit des gewünschten Oberflächennetzes müssen die Punkte interpoliert werden. Dies geschieht einheitlich mit kubischen Splines, so dass die Schaufelaufösung beliebig hochskaliert werden kann. Das Verfahren ist bei gleicher Ausgangsgeometrie eindeutig, die Krümmungsstetigkeit auf der Schaufeloberfläche ist gewährleistet.

### 2.4.3 Andere strömungsführende Teile

Alle nicht rotationssymmetrischen Bauteile, wie z.B. Spiralgehäuse, Krümmer, Saugrohre oder Diffusoren, müssen gesondert behandelt werden. Diese Bauteile sind für die Gesamtsimulation notwendig und werden i.d.R. nicht benötigt, um einzelne Lauf- und Leitschaufeln aufeinander abzustimmen, sie bestimmen aber wesentlich das Gesamt-Betriebsverhalten. Grundsätzlich gibt es hier zwei Strategien:

1. Beschreibung der Geometrie in einem externen CAD-Tool und nachträgliche Vernetzung.

Die Aufgabe der Geometriebeschreibung wird auf ein externes, kommerzielles CAD-Tool übertragen. Das Geometriemodell kann um fertigungsbezogene Informationen angereichert sein und steht anderen CAE-Tools zur Verfügung, z.B. für Untersuchung der Festigkeit oder Fertigbarkeit. Die Parametrisierung geschieht im CAD-Tool und hängt nur von der Leistungsfähigkeit desselben ab. Die Netzgenerierung erfolgt mit einem allgemeinen Netzgenerator, der allerdings speziell aufbereitete Geometrieinformationen benötigt.

2. Parametrische Beschreibung der Geometrie im Entwurfssystem inkl. Vernetzung, zusätzlich Exportfunktion der Oberflächen.

Dabei werden für das Entwurfssystem problemangepaßte Netzgeneratoren erzeugt, die im System parametrisch angesteuert werden können. Es muß neben dem Netzgenerator eine Schnittstelle vorgesehen sein, mit der die benetzten Oberflächen des Bauteils in einem allgemeinen CAD-Format exportiert werden können.

Die Strategie 1 deckt *alle* Bauteile ab, auch die o.g. rotationssymmetrischen und Schaufeln, sie ist die flexibelste, da sie nur noch durch das CAD-System und den Netzgenerator limitiert wird. Die Komplexität wird vom Ersteller des Simulationssystems auf den Benutzer übertragen: Er muß die geometrische Modellierung festlegen und die Netze in das geometrische Modell bringen. Im Fall unstrukturierter Netze kann dies auch weitgehend automatisiert geschehen, so dass auch umfangreiche Parametervariationen möglich werden. PARK UND LEE haben ein allgemeines automatisiertes Verfahren für die Erzeugung blockstrukturierter Netze auf Basis allgemeiner Geometrien vorgestellt, s. THOMPSON ET AL. [52], allerdings konnte es sich nicht allgemein durchsetzen. Somit bleiben je nach Problemstellung aufwendige Vorarbeiten beim Nutzer.

Die Strategie 2 verlagert den Aufwand hin zum Entwickler eines Simulationssystems. Ist ein solcher Netzgenerator verfügbar, so können vom Endanwender mit geringem Aufwand in der Aufbereitung Parameterstudien durchgeführt werden. Für Bauteile, die oft optimiert werden, ist es sinnvoll, diese Strategie zu wählen. Der Nachteil ist, dass eine Vielzahl unterschiedlicher Bauteile eine Vielzahl unterschiedlicher Netzgeneratoren benötigt werden.

## 2.5 Entwurfsverfahren

Entwurfsverfahren beschreiben die Prozesse, mit denen die geometrische Form einer Strömungsmaschine auf Grundlage der zu lösenden technischen Anforderungen festge-

legt wird. Die minimalen Anforderungen, die bekannt sein müssen, sind die notwendigen Förder- oder Fallhöhenintervalle, der erforderliche oder zur Verfügung stehende Bereich für den Volumenstrom, sowie eine zeitliche Gewichtung dieser einzelnen notwendigen Betriebspunkte. Hinzu kommen je nach Anwendungsfall zusätzliche Informationen wie Bauraumanforderungen, Aufstellungshöhe, Anforderungen an die Zuverlässigkeit der strömungstechnischen Anlage, Budgetrahmen, Netzfrequenz, Explosionssicherheit usw.

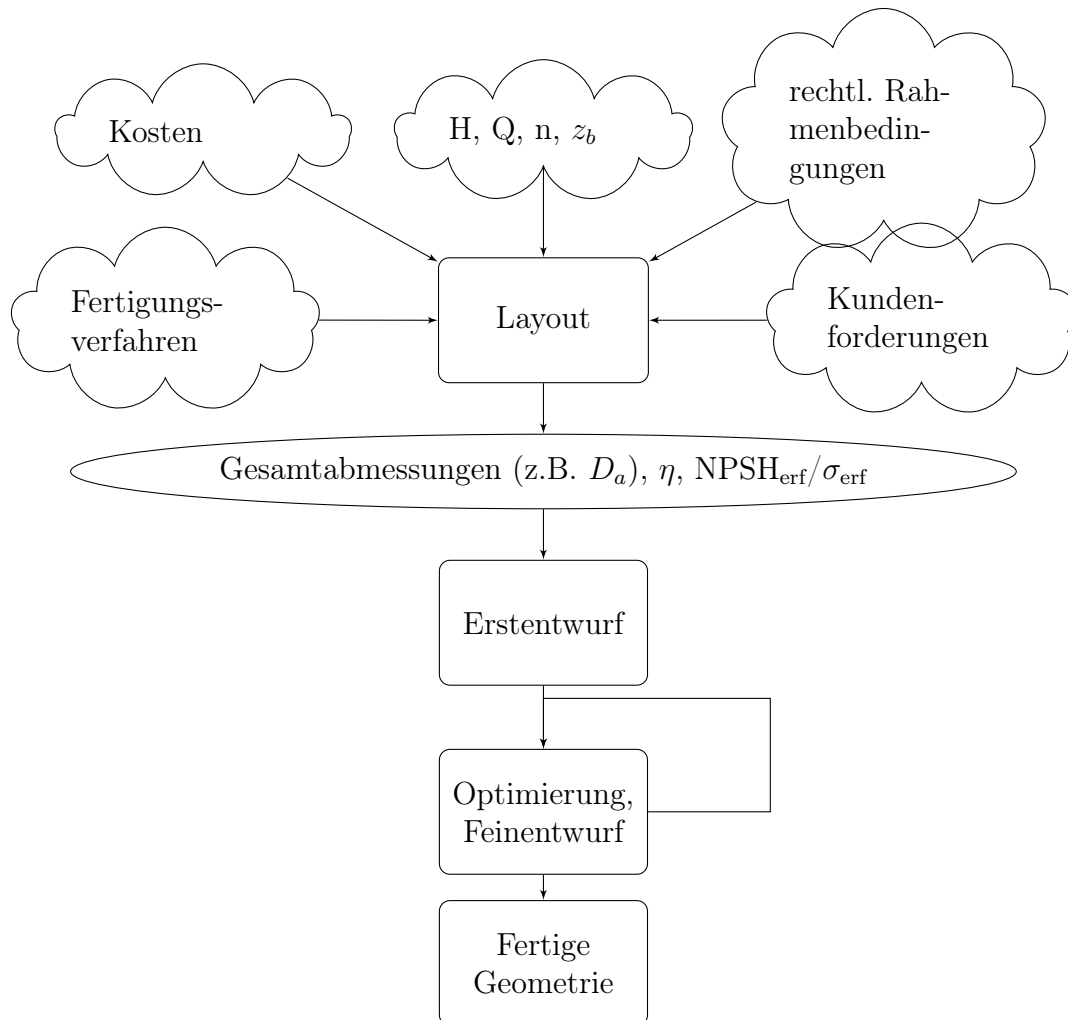


Abbildung 2.6: Entwurfsprozess

### 2.5.1 Dimensionierung, Layout

Auf Basis der Anforderungen werden die Gesamtabmessungen der Strömungsmaschine, das *Layout*, ermittelt. Hier werden neben Bauart der Strömungsmaschine - Radial, Halbaxial, Axial - der Durchmesser und die Drehzahl festgelegt. Der Layout-Prozess findet sowohl für Radialpumpen wie für Francis turbinen sehr früh im Gesamt-Entwicklungsprozess statt und umfasst neben den allgemeinen strömungstechnischen Überlegungen immer auch wirtschaftliche Aspekte. Im industriellen Umfeld ist er der Angebotserstellung zuzurechnen. Das Layout stützt sich immer auf empirische Daten, d.h. aus Erfahrungen mit bereits erstellten Anlagen, vgl. z.B. Abbildung 2.7.



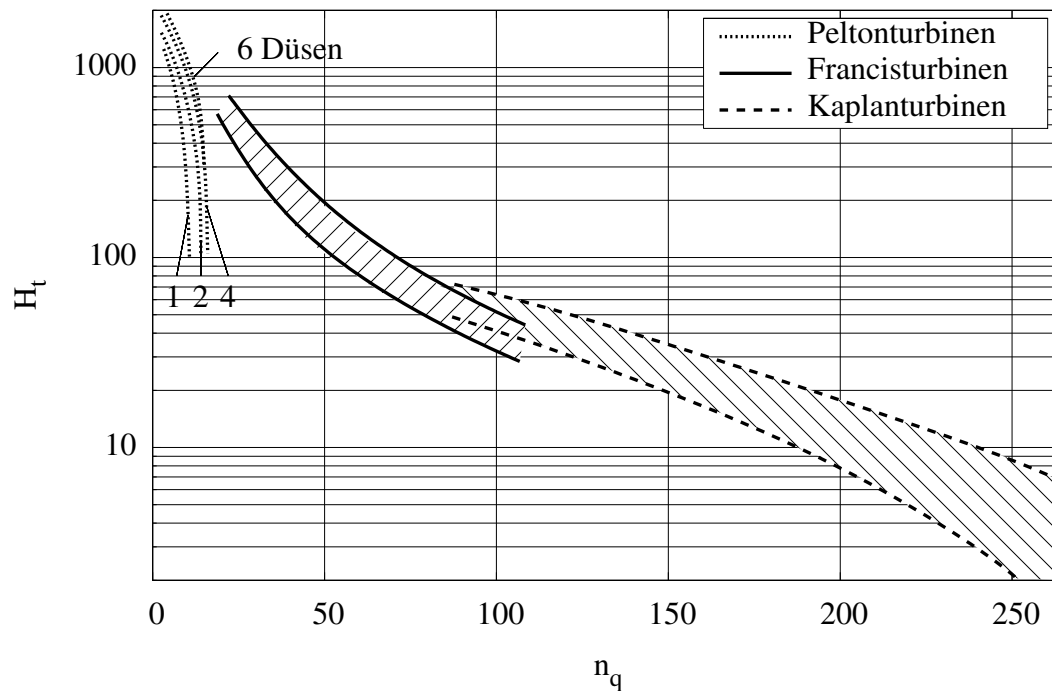


Abbildung 2.7: Bauartenauswahl von Turbinen nach Voith, vgl. RAABE [42]

## 2.5.2 Erstentwurf

Auf Basis des Layouts kann zunächst ein Erstentwurf festgelegt werden. Dieser Erstentwurf umfaßt Gestaltungsvorschläge für alle strömungsführenden Oberflächen, alle benetzten Oberflächen werden im Erstentwurf geometrisch definiert. In den Erstentwurf gehen die Anforderungen aus der Problembeschreibung und dem Layout ein. Der Erstentwurf der Strömungsmaschine ist eine Entwicklungsaufgabe und entweder der allgemeinen Forschung und Entwicklung oder der konkreten Projektarbeit zuzurechnen.

Die Erstentwurfverfahren können in zwei Gruppen aufgeteilt werden:

**Empirisch-analytische Verfahren:** Hierzu zählen alle „klassischen“ Entwurfsverfahren, wie z.B. von PFLEIDERER [40], RAABE [42] oder GÜLICH [26] vorgestellt. Diese Verfahren gehen zurück auf die Gestaltung der Strömungswinkel, um die spezifische Schaufelarbeit nach Gleichung (2.46) festzulegen. Zusammen mit empirisch ermittelten Faktoren, die die Reibungseffekte in der Strömungsmaschine berücksichtigen sollen, werden so die Meridiankontur und die Schaufelgeometrien festgelegt. Einige der Verfahren geben dabei lediglich Auskunft über die Gestaltung der Schaufelein- und -austrittswinkel, dem entwerfenden Ingenieur stellt sich die Aufgabe, den Winkelverlauf zwischen Vorder- und Hinterkante der Schaufel sinnvoll zu gestalten.

**Datenbankgestützte Verfahren** benötigen einen Erfahrungsschatz über bekannte Strömungsmaschinen und deren Betriebsverhalten. Sie nutzen die Stetigkeit des Lösungsraums für die Ermittlung unbekannter Lösungen. Eine Lösung für eine noch nicht verfügbare Geometrie wird aus existierenden Geometrien interpoliert, deren Layoutdaten zu den gewünschten benachbart sind. Ein Beispiel für ein interpolierendes, datenbankgestütztes Verfahren ist die Nutzung Künstlicher Neuronaler Netze, s. KRÄMER [32].

### 2.5.3 Optimierung

Auf Basis der mit den Erstenwurfsverfahren ermittelten Geometrien kann die Strömungsmaschine optimiert werden. Hierbei sind problembezogen unterschiedliche Zielfunktionen und Randbedingungen denkbar. Die Optimierung geschieht nach SCHILLING [46] mittels Multi-Level-CFD in unterschiedlichen Stufen: Zuerst wird mit schnell rechnenden, aber vereinfachten Modellen der Entwurf in die Nähe des absoluten Optimums gebracht. Diese schnell rechnenden Verfahren können z.B. Quasi-3D-Verfahren, Blade-Elemente-Methoden, reine Eulerverfahren oder Quasi-Navier-Stokes-Verfahren sein. In einem weiteren Schritt wird die so ermittelte fast optimale Geometrie mittels genauerer CFD-Simulationen, die im Regelfall rechen- und zeitaufwendiger sind, weiter optimiert. Somit kann recht schnell ein absolutes Optimum erreicht werden.

In der vorliegenden Arbeit wird der Integrated Design Process nach SCHMALHORST, SCHILLING, FLURL [49] angewendet, s. Abbildung 2.8, wobei als vereinfachtes Modell das Quasi-Navier-Stokes-Verfahren (QNS) angewendet wird. Ziel des QNS ist es, die Strömungsantwort auf Änderungen der Geometrie möglichst sofort ermitteln zu können.

## 2.6 Vernetzung

Die Vernetzung von Volumina erfordert grundsätzlich die folgenden Schritte:

1. Festlegen des zu vernetzenden Volumens über seine benetzte Oberfläche
2. Festlegen der Berandungskurven, die die einzelnen Oberflächenteilstücke verbinden
3. Verteilen von Punkten auf den Berandungskurven
4. Verteilen von Punkten auf den Berandungsflächen
5. Erzeugen der Netzzellen im Volumen

Bei der Anwendung von strukturierten Multiblockverfahren sind hinsichtlich der Vernetzung einige Einschränkungen zu berücksichtigen: Flächen müssen jeweils vier Berandungskurven aufweisen, damit eine orthogonale, parametrische Fläche in  $(i, j)^T$  angepaßt auf die Flächenkontur in den Raum projiziert werden kann. Für die Volumina herrscht die Einschränkung, dass ein (Teil-) Volumen jeweils sechs Berandungsflächen benötigt, damit der orthogonale Parameterraum in  $(i, j, k)^T$  in Koordinaten innerhalb

dieses Volumen transformiert werden kann. Um allgemeine Volumina vernetzen zu können, müssen i.d.R. Hilfsflächen innerhalb des Volumens definiert werden. An diesen Hilfsflächen werden die einzelnen Rechnernetze mittels Interfaces gekoppelt, so dass ein durchgängiges durchströmtes Volumen entsteht.

Methoden, um von den Punkten auf den Berandungskurven Flächen erzeugen zu können, sind z.B. die transfinite Interpolation, s. THOMPSON ET. AL. [52] oder die  $C^1$ -stetige Gordon-Coons-Flächen, s. HOSCHEK UND LASSER [29]. Diese Verfahren lassen sich ebenfalls anwenden, um von den Berandungsflächen Netzzellen in das Volumen zu interpolieren. Mit der transfiniten Interpolation sowie mit Gordon-Coons-Flächen können die inneren Punkte direkt errechnet werden, es sind keine zusätzlichen Iterationen notwendig.

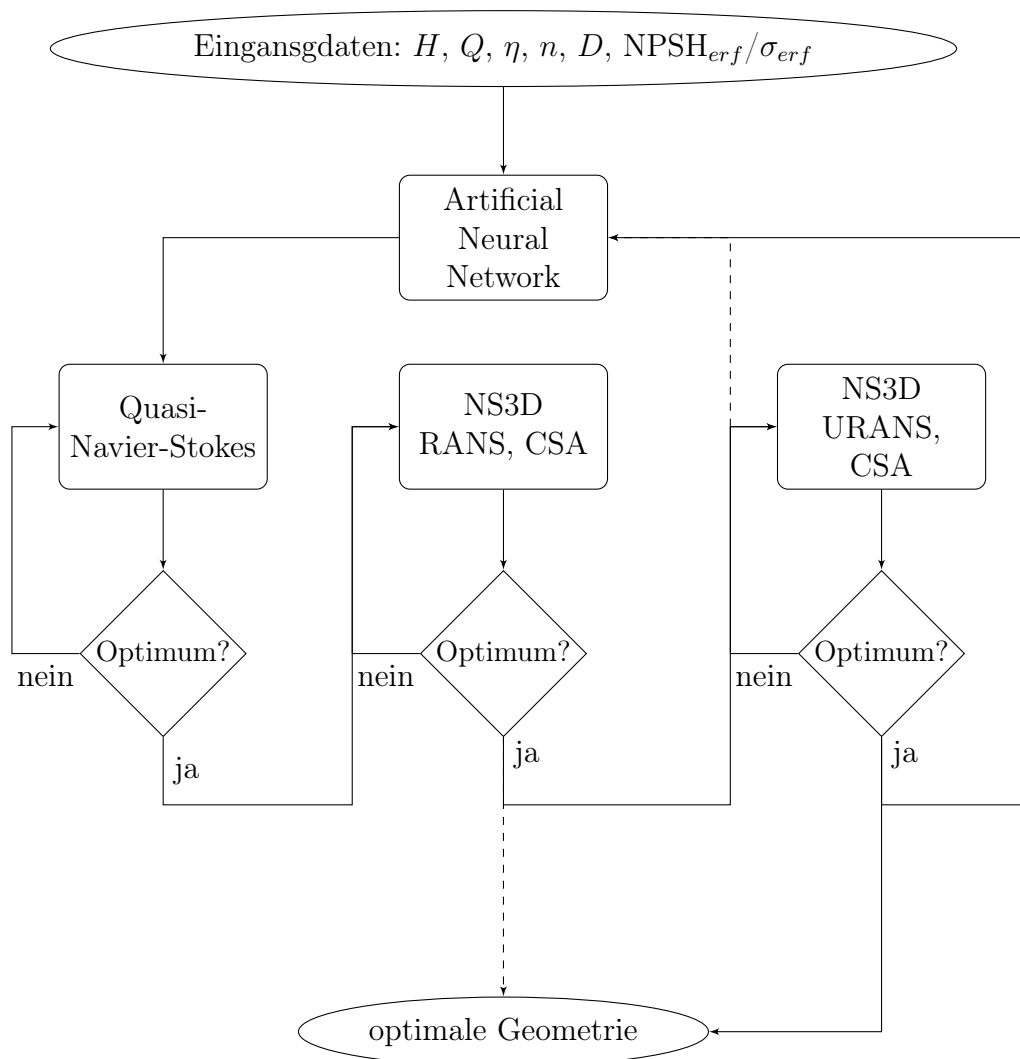


Abbildung 2.8: Multi-Level-CFD: Integrated Design Process

### 2.6.1 Verfügbare Interfaces

Der benutzte Solver NS3D ist blockstrukturiert, d.h. unterschiedliche Blöcke können durch verschiedene Interfaces miteinander verbunden werden. Die verfügbaren Interfaces sind im Folgenden aufgeführt.

**Matching Interface (M)** verknüpft zwei Blöcke bei Knotengleichheit. Die Knoten auf der Berandungskurve und im inneren der Kontaktfläche sind bei beiden Blöcken gleich. Im instationären Fall kann das Matching Interface nur Blöcke gleicher Kinematik verbinden, d.h. stehend – stehend und rotierend – rotierend.

**Non-Matching Interface (NM)** hat eine deckungsgleiche, aber nicht knotengleiche Berandungskurve. Die inneren Knoten sind frei wählbar. Das Non-Matching Interface verbindet im instationären Fall Blöcke gleicher Kinematik.

**Periodisches Interfaces (P)** wird benötigt um Einzelkanalrechnungen durchführen zu können. Es verbindet entweder rotationsperiodisch oder translatorisch periodisch, und ist nur sinnvoll bei gleicher Kinematik. Die Flächen müssen analog zum NM-Interface deckungsgleich sein, können aber um einen konstanten Winkel  $\varphi$  bzw. eine konstante Strecke  $\Delta s$  verschoben sein.

**Non-Matching Group Interface (NMG)** verbindet N Geberblöcke zu M Empfängerblöcken. Die Geberflächen müssen insgesamt deckungsgleich mit den Empfängern sein. Das NMG-Interface verbindet Blöcke gleicher Kinematik. Ist die Kontaktfläche ein Ring, so können im instationären Fall Geber und Empfänger unterschiedliche Geschwindigkeiten aufweisen.

**Ring-Interface (R)** verbindet zwei Blöcke unterschiedlicher Kinematik, die Kontaktfläche muß eine Rotationsfläche sein. Die inneren Knoten können frei gewählt werden.

**Non-Matching umfangsperiodisches Interface (NMU)** verbindet zwei Einzelkanäle in Strömungsrichtung. Die radialen Bergrenzungsänderer müssen auf jeweils einem konstanten Radius  $r$  liegen, die anderen Berandungskurven müssen durch einen konstanten Winkel  $\varphi$  ineinander überführbar sein. Es verbindet im instationären Fall Blöcke unterschiedlicher Kinematik. Die Teilung beider Einzelkanäle muß gleich sein. Das NMU-Interface funktioniert auch bei ungleicher Teilung der Kanäle, ist dann nur in speziellen Fällen konservativ.

**Smart Wall Interface (SW)** stellt eine Wand dar, in der mit nicht matchenden Knoten ein Loch geschnitten wird, s. Abbildung 2.9. Es verbindet Blöcke gleicher Kinematik.

**Stage Interface** verbindet zwei Einzelkanäle unterschiedlicher Teilung konservativ. Es werden in Umfangsrichtung gemittelte Werte ausgetauscht, in der radialen Richtung ergibt sich eine Schichtung der Werte. Das Stage Interface ist schwächer gekoppelt als das NMU-Interface, Zwischenergebnisse werden nur in der äußeren Iteration ausgetauscht und nicht während des Lösens des linearisierten Gleichungssystems.

Die Interfaces NM, NMG, NMU und SW wurden von HERRMANN [28] für den NS3D erweitert bzw. neu implementiert und sind dort im Einzelnen beschrieben. Die Möglichkeiten und Einschränkungen dieser Interfaces müssen beim Erzeugen der Blöcke berücksichtigt werden.

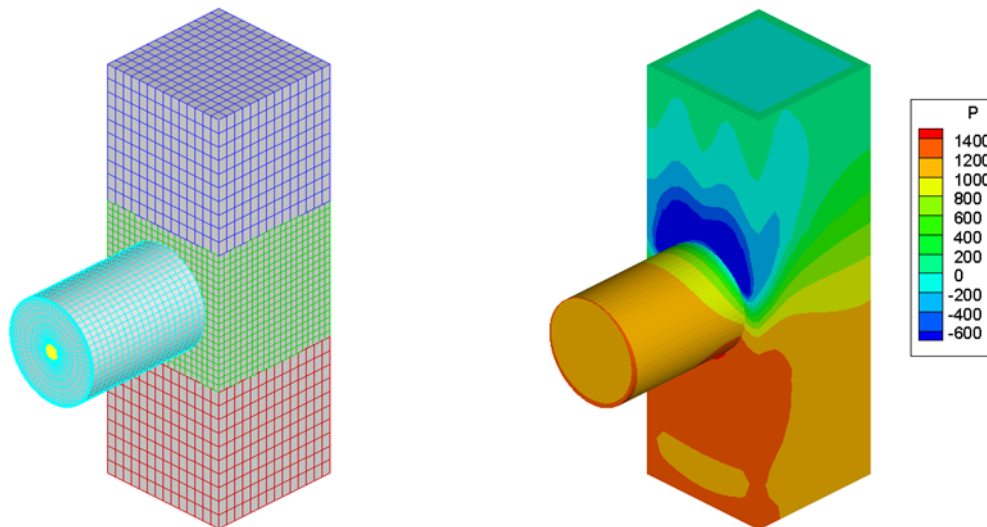


Abbildung 2.9: Beispiel für Smart-Wall Interface, s. HERRMANN [28]

## 2.6.2 Hierarchische Netzgenerierung

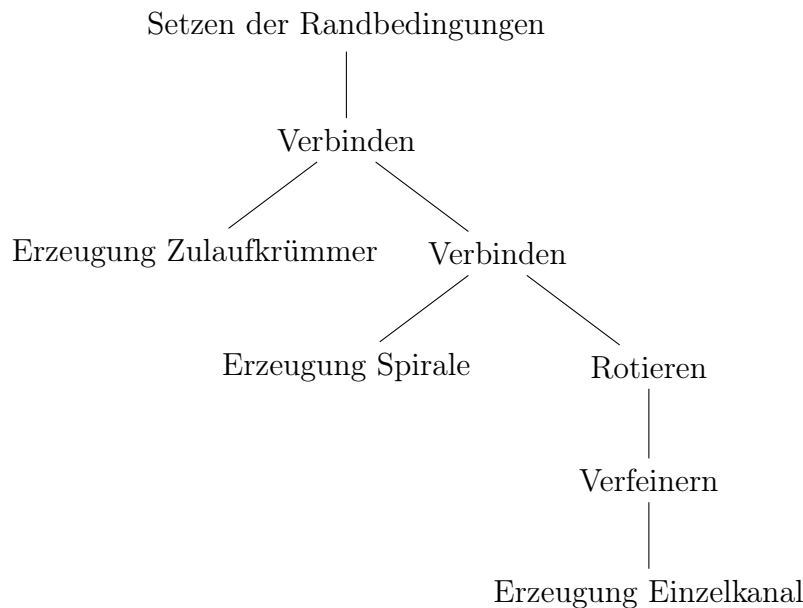
Je nachdem, welche Strategie der Geometriebeschreibung verfolgt wird, s. Abschnitt 2.4.3, sind die Rechnetze entweder bereits ein Teilprodukt des Geometriemodells oder müssen gesondert aufbereitet werden. Es ist sinnvoll, beide Strategien für ein Rechenetz nutzen zu können. Dazu bietet sich ein hierarchisches Verfahren der Vernetzung an, das die Vernetzung grundsätzlich in zwei unterschiedliche Schritte trennt:

1. Netzerzeugung
2. Netzbe- und -verarbeitung

Ein Netzerzeugungsschritt hat ein Rechenetz zum Ergebnis und benötigt lediglich Geometriegrößen als Eingabeparameter. Ein Schritt der Netzbearbeitung kann z.B. sein, den einzelnen Laufschaufelkanal entsprechend der Teilung zu kopieren, zu rotieren und die Konnektivitäten zwischen den Blöcken anzupassen. So entsteht aus einem Rechenetz eines einzelnen Schaufelkanals das eines kompletten Laufrades. Die Schritte zu Netzbe- und -verarbeitung sind einzelne, teils simple Operationen, die in Kombination mit Netzerzeugungsschritten das Erstellen komplexer Netze ermöglichen.

Der Ablaufplan einer Netzerstellung kann als Baumstruktur verstanden werden, wobei die Netzerzeugungsschritte jeweils einen äußeren Knoten (Blatt) darstellen. Die Netzbearbeitung ist ein innerer Knoten mit mindestens einem untergeordneten Blattelement. Das Erzeugen eines Rechnetzes für eine komplette Pumpe inkl. der lokalen Verfeinerung einzelner Blöcke kann als Baumstruktur wie in Abbildung 2.10 dargestellt

werden. Um das gewollte Ergebnis zu erreichen, ist der Baum rekursiv in Post-Order-Reihenfolge durchzugehen, d.h. für einen Knoten werden erst die Subknoten aufgerufen, danach erst wird die Vernetzungsvorschrift des Knotens selbst aufgerufen.



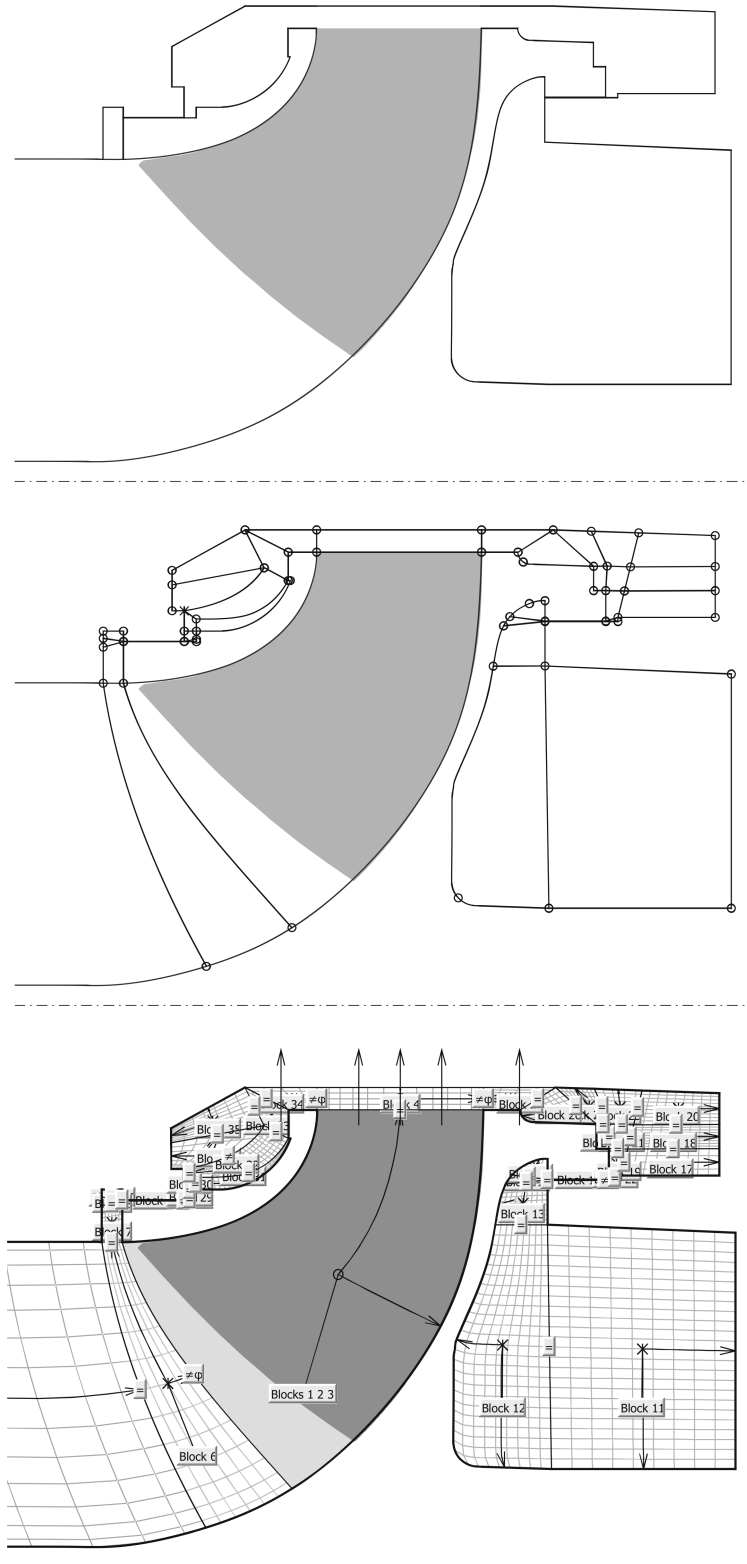
**Abbildung 2.10:** Hierarchische Vernetzung

### 2.6.3 Partitionierung

Die rotationssymmetrischen Bereiche werden in der Meridianansicht partitioniert. Die rotationssymmetrischen Wände werden als NURBS eingelesen und in der Meridianansicht dargestellt. Der Nutzer muß nun die inneren Grenzen einfügen, so dass Teilflächen entstehen, die jeweils 4 Berandungskurven aufweisen. Somit kann ein zweidimensionales Meridiannetz erstellt werden, das durch Rotation um die Drehachse in ein dreidimensionales strukturiertes Netz überführt werden kann. Randbedingungen und Interfaces sind bei der Partitionierung Eigenschaften der Berandungskurven. Um die Eingabe zu vereinfachen, wurde die Einschränkung gemacht, dass ein Interface nur ganze Blöcke verbinden kann, auch kann eine Berandungskurve nur eine einheitliche Randbedingung aufweisen. Dies muß beim Definieren der Grenzen berücksichtigt werden. Der verwendete Code NS3D weist diese Einschränkungen nicht auf.

### 2.6.4 Schaufelbereich

Der Schaufelbereich ist in der Meridianansicht eine Fläche mit 4 Berandungskurven, allerdings enthält er – anders als die rein rotationssymmetrischen Bereiche – ein nicht symmetrisches Hindernis, die Schaufel. Es ist möglich, diesen Bereich mit einem einzelnen Block zu vernetzen. Alternativ dazu kann der Strömungsraum in mehrere Blöcke aufgeteilt und die Zwischenräume mit weiteren Blöcken aufgefüllt werden, wobei i.d.R. ein Block um die Schaufeloberfläche gelegt wird. Diese Multi-Block-Netze werden je nach Gestalt des Netzes um die Schaufel in C- und O-Netze unterteilt.



**Abbildung 2.11:** Beispiel der Partitionierung in der Meridianansicht, CAD-Geometriemodell (oben), Partitionierung (mitte), Ansicht des Meridiannetzes (unten)

Obwohl nicht zwingend notwendig, ist es sinnvoll, die Schaufelbereiche in der Meridiankontur so zu vernetzen, dass in Hub-Shroud-Richtung eine feste Anzahl von Netzebenen konstruiert wird. Auf den so entstandenen Netzebenen, die auch als geometrische Strombahnen bezeichnet werden, läßt sich der Bereich zwischen den Schaufeln mit zweidimensionalen Ansätzen vernetzen. Die Schaufelnetzgenerierung wird im allgemeinen wie folgt vorgenommen:

1. Partitionierung des Strömungsbereichs,
2. Festlegen des Bereichs für das nicht rotationssymmetrische Schaufelnetz,
3. Festlegen der Netzebenen in Hub-Shroud-Richtung,
4. Verschneiden der Schaufeloberfläche mit den Netzebenen – Ermittlung der Schaufelprofile und schließlich
5. Festlegen der Vernetzung in Umfangs- und Strömungsrichtung auf den ermittelten Netzebenen.

### Schaufelnetze

Der Schaufelbereich kann strukturiert in einem einzelnen Block vernetzt werden. Dabei werden auf den Schaufelprofilen Punkte verteilt, die Vorder- und Hinterkante wird in geeigneter Weise mit der Einlaß- und Auslaßfläche verbunden. Mittels der o.g. Interpolationsverfahren kann ein zweidimensionales Netz auf den Hub-Shroud-Netzebenen erzeugt werden, wobei für alle Ebenen die gleiche Punkteverteilung beibehalten werden muß. Das algebraisch ermittelte Netz wird im Anschluß mittels einer elliptischen Differentialgleichung geglättet. Diese Vorgehensweise soll sicherstellen, dass die Netzebenen senkrecht aufeinander und auch senkrecht auf den Wänden stehen.

Die H-Netze haben einige Einschränkungen hinsichtlich der handhabbaren Geometrien. Um mehr Freiheitsgrade zu gewinnen, werden Multiblocknetze verwendet, die den Schaufelbereich typischerweise in 4 Blöcke aufteilen. Dazu wird um die Schaufel zuerst ein Netz normal auf die Schaufelgeometrie gelegt, im Regelfall also ein O-Netz. Für radiale Pumpen, deren Schaufelhinterkante durch Abdrehen auf einen einheitlichen Radius gebracht wird, kann auch ein C-Netz verwendet werden. Die Zwischenräume um das Schaufelnetz werden mit drei Blöcken aufgefüllt, dabei wird ein Block einlaßseitig, ein Block auslaßseitig und ein Block zwischen den Schaufelnetzen verwendet.

Für die Erstellung der O- und C-Netze stehen Netzgeneratoren verschiedener Ausprägung im IDS bereit. Neben variabel und umfangreich parametrisierten Netzgeneratoren sind Verfahren für die automatisierte Vernetzung des Schaufelbereichs implementiert worden. Diese Verfahren erfordern nur vier „Mußparameter“, um ein Netz basierend auf einer vorhandenen Schaufelgeometrie zu erzeugen: Die Dicke des O- bzw. C-Netzes und die Anzahl Knoten  $N_i$ ,  $N_j$  und  $N_k$  in diesem Netz. Aus den gegebenen Dimensionen und der Geometrie wird eine charakteristische Länge  $l_{char}$  ermittelt, die die mittlere Distanz zwischen zwei Knoten des Rechnernetzes darstellt. Neben den genannten „Mußparametern“ sind „Kannparameter“ vorgesehen, mit denen die automatisierte Schaufelvernetzung manuell verbessert werden kann.



Die automatisierte Vernetzung des Schaufelbereichs umfaßt die folgenden Teilschritte:

1. Festlegen der Netzebenen in Hub-Shroud-Richtung,
2. Erzeugen einer Offsetfläche um die Schaufel,
3. Verschneiden der Schaufel- und der Offsetfläche mit den Netzebenen,
4. Analyse der Vorder- und Hinterkante der entstandenen Profile hinsichtlich:
  - (a) Nasenform,
  - (b) Krümmung,
  - (c) Knotenzahl, um die Krümmung ausreichend beschreiben zu können.
5. Ermitteln der Extremwerte auf dem Offsetprofil in den Zylinderkoordinaten  $r$  und  $\varphi$ . Die Extremalpunkte werden auf das benachbarte Schaufelprofil bzw. die Einlaß- und Auslaßfläche projiziert und ergeben somit zusätzliche charakteristische Punkte für die inneren Grenzen der Blöcke.
6. Anhand der Extremalpunkte werden die Profile der Schaufel und der Offsetfläche in Teilprofile unterteilt und die inneren Interfaces der Füllblöcke definiert. Die Interfaces ergeben Teilflächen mit je vier Berandungskurven.
7. Auf den Teilprofilen werden unter Berücksichtigung von  $l_{char}$  Knoten verteilt.
8. Mit geeigneten Interpolationsverfahren werden die Randkurven in das Volumen hinein interpoliert.

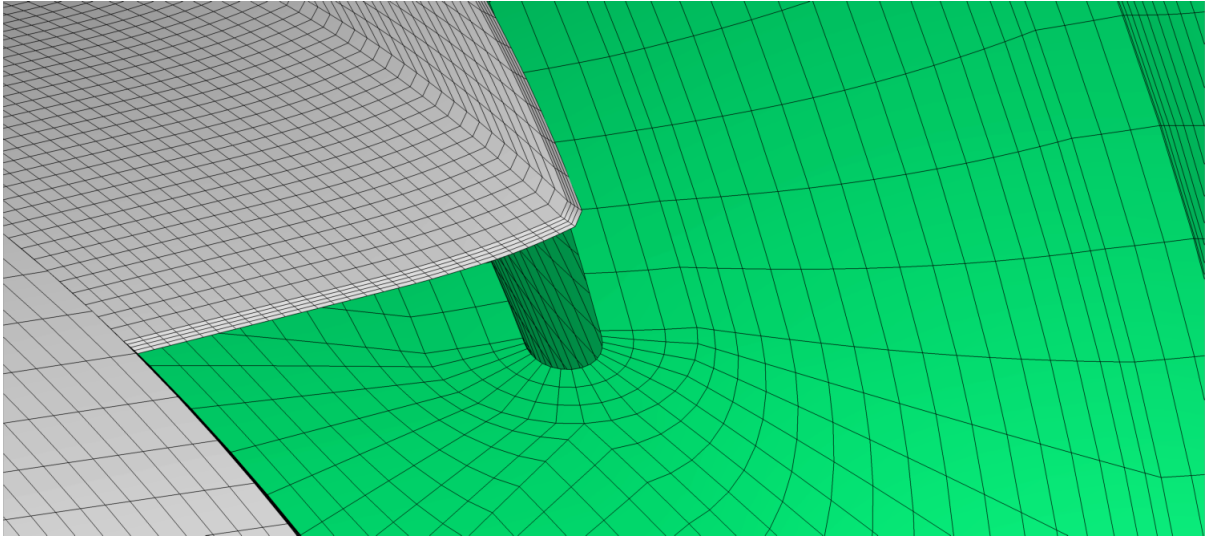
Radialpumpen können im Radseitenraum zusätzliche Schaufeln aufweisen, deren Aufgabe die Einstellung des Axialschubs ist. Außerdem werden sie eingesetzt, um Partikel aus dem Radseitenraum zu fördern. Sind diese Rückenschaufeln gekrümmt, so können sie mit den Vernetzungsmethoden der Hauptschaufeln behandelt werden. Laut GÜLICH [26] sind die Rückenschaufeln oftmals als rein radiale Rippen ausgeführt, die mit einem eigenen Netzgenerator modelliert werden. Der Bereich zwischen den Schaufeln wird als H-Netz dargestellt, das nicht die komplette Teilung ausfüllt, sondern um die Breite der Rippe schmaler erzeugt wird. In Meridianrichtung füllt die Rückenschaufel das ihr zugewiesene Rechengebiet mit vier Berandungskurven voll aus. Die Rückenschaufel wird komplett ohne zusätzliches Geometriemodell erzeugt und folgt somit der Strategie 1 aus Abschnitt 2.4.3.

## Kopfspalte

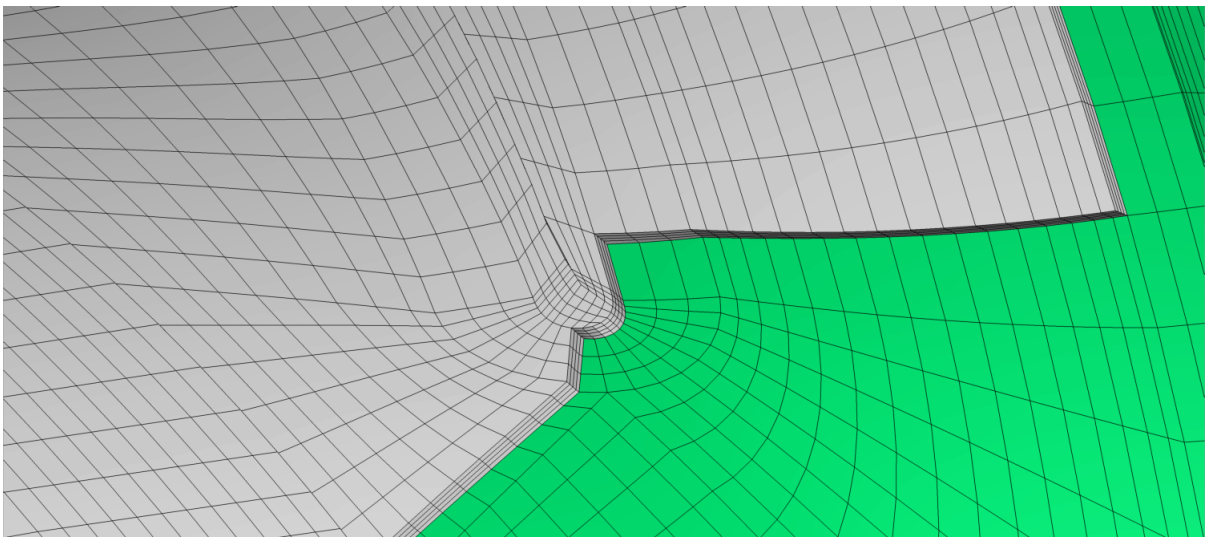
Für Kopfspalte bzw. offene Laufräder stehen zwei verschiedene Methoden zur Verfügung: Erstens das Vernetzen des Kopfspalts mit einem einzelnen Block, wobei das Interface nicht knotengleich per Smart-Wall-Interface bewerkstelligt wird, s. Abbildung 2.12. Zweitens kann die Topologie des Schaufelnetzes in das Kopfspaltvolumen kopiert werden. In einem weiteren Schritt wird der Bereich zwischen den Schaufeln aufgefüllt, dargestellt in Abbildung 2.13.

## 2.6.5 Rotationssymmetrische Bereiche

Die rotationssymmetrischen Bereiche sind abhängig von den Schaufelnetzen, die Schaufelnetze geben die Punktverteilung an der Kontaktfläche zu den rotationssymmetrischen Bereichen vor. Die rotationssymmetrischen Bereiche werden im Anschluß an die Schaufelbereiche vernetzt. Rotationssymmetrische Bereiche werden aus den ermittelten Meridianflächen direkt erzeugt, wobei zuerst die Meridianfläche mittels der Transfiniten Interpolation vernetzt wird, so dass ein Netz im Parameterraum  $(i, j)^T$  und dem physikalischen Raum  $(r, z)^T$  entsteht. Diese Netze können in einer zweidimensiona-



**Abbildung 2.12:** Vernetzung des Spaltes eines offenen Laufrads mittels eines einzelnen Blocks. Der Schaufelkanal und das Spaltnetz werden per Smart-Wall-Interface verbunden.



**Abbildung 2.13:** Vernetzung des Spaltes eines offenen Laufrads mit einem knotengleichen Multi-Block-Netz. Der Schaufelbereich im Spalt ist mit zwei Blöcken aufgefüllt.

len Vorschau dargestellt werden, so dass der Anwender eine gute Kontrolle über das spätere Rechnetz hat, s. Abbildung 2.11. Die ermittelten zweidimensionalen Punkte werden kopiert und entsprechend der zu erreichenden Teilung um einen Winkel  $\Delta\varphi$  um die Rotationsachse verschoben, so dass ein dreidimensionales Netz im Parameterraum  $(i, j, k)^T$  und im physikalischen Raum  $(r, \varphi, z)^T$  entsteht. Weisen die  $\varphi$ -Werte eines Nachbarnetzes in Umfangsrichtung keine Gleichverteilung auf, so werden die Netzlinien aufgefächert. Um zu starke Scherungen zu verhindern, wird nur ein maximaler Winkel zwischen der Achsebene und der erzeugten Netzlinie zugelassen.

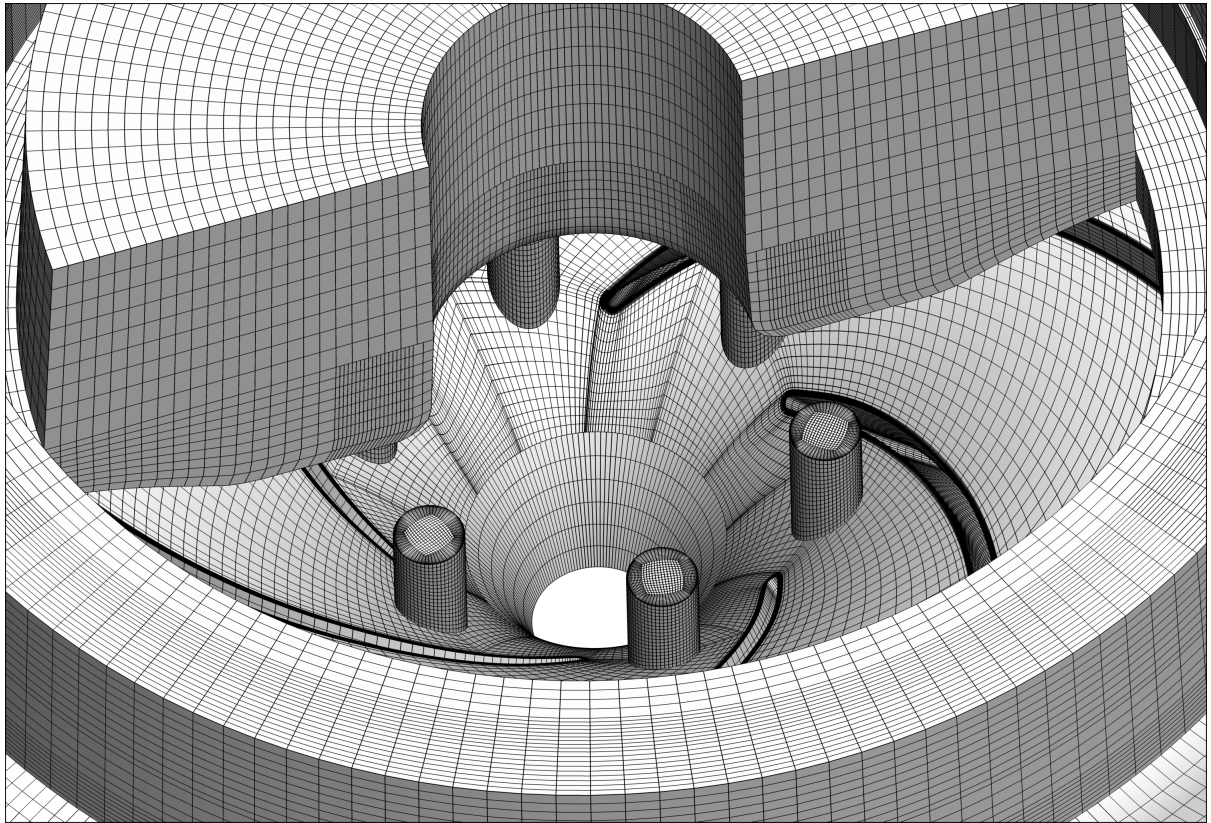
### 2.6.6 Entlastungsbohrung

Eine Bohrung muß immer als Nachbehandlung eines existierenden Netzes definiert werden. Die Bohrung wird aus zwei Netzen erzeugt, die einen Zylinder ergeben. Das äußere Netz hat eine O-Topologie, das innere Netz ist ein H-Netz. Der Zylinder wird per Smart-Wall-Interface mit den Partnerblöcken verbunden, wobei die Kontaktflächen beliebig geformt sein können, eine knotengleiche Anbindung ist nicht implementiert. Der Netzgenerator für die Entlastungsbohrung kann Bohrungen beliebiger Lage erzeugen, die Teilung der Bohrung kann unterschiedlich sein zur Teilung des Schaufelkanals. Ein konkretes Beispiel einer vernetzten Entlastungsbohrung ist in Abbildung 2.14 gegeben.

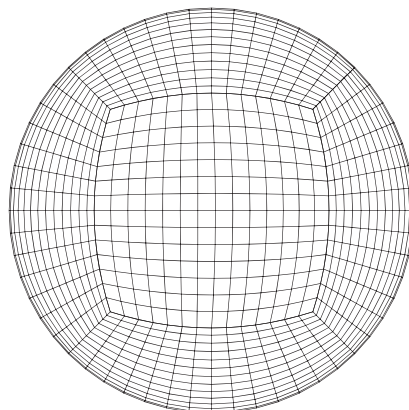
Das Bohrungsnetz wird über die Mittellinie und einen Radius definiert. Die Mittellinie muß so gewählt werden, dass der resultierende Zylinder komplett in die Blöcke eintaucht, die mit der Bohrung verbunden werden sollen. Im ersten Schritt wird ein zweidimensionales Basisnetz erzeugt, s. Abbildung 2.15, das dann entsprechend der Länge der Mittellinie der Bohrung verlängert wird. Die entstandene Linienschar wird in die Position des Bohrungszyinders transformiert und mit allen Oberflächen verschnitten, die die gleiche Drehgeschwindigkeit aufweisen wie die Bohrung selbst.

### 2.6.7 Spiralteilstücke

Bei der Einzelkanalsimulation von einstufigen Radialpumpen ist die Modellierung des Auslaßbereichs problematisch. Während in der Realität die Spirale das ausströmende Fluid sammelt und den radialen Geschwindigkeitsanteil teilweise in statischen Druck umwandelt, die Hauptströmung also im wesentlichen tangential verläuft, wird bei der Einzelkanalsimulation der Auslaß als rein radiales Bauteil modelliert, wobei aus Gründen der numerischen Stabilität die Strömung durch Querschnittsverengung beschleunigt wird. Der Netzgenerator für Spiralteilstücke soll eine zusätzliche Methode sein, den Auslaßbereich einer Einzelkanalsimulation zu modellieren. Dazu wird ein Teilstück einer logarithmischen Spirale erzeugt, das die gleiche Teilung aufweist wie das Laufrad, s. Abbildung 2.16. Die Flächendifferenz kann frei gewählt werden, es wird grundsätzlich ein Rechtecksquerschnitt verwendet. Die deckungsgleichen Flächenanteile werden mit periodischen Interfaces versehen, der an der einen Seite überstehende Teil ohne periodischen Partner wird mit der Randbedingung „Auslaß“ belegt. Durch Wahl der Parameter  $r_a$  und  $A_{out}$  kann das Spiralteilstück so variiert werden, dass der Flächenverlauf einer real ausgeführten Spirale annähernd wiedergegeben werden kann.



**Abbildung 2.14:** Netz einer Radialpumpe mit Entlastungsbohrungen, der druckseitige Radseitenraum ist zur Hälfte ausgeblendet



**Abbildung 2.15:** zweidimensionales Basisnetz der Entlastungsbohrung

Das Spiralteilstück ist stellungsabhängig, d.h. die relative Position des Impellers zum Spiralteilstück hat Auswirkungen auf das Ergebnis. Die mittlere Austrittsgeschwindigkeit am Auslass ist durch die Kontinuität gegeben mit

$$\overline{c_{m,out}} = \frac{Q}{A_{out}} . \quad (2.43)$$

Da die tangentielle Geschwindigkeit in der Spirale in erster Linie von  $u_{ref}$  abhängt, kann es am Auslaß zu Rückströmungen kommen, die eine saubere Konvergenz behindern. Somit ist zu erwarten, dass im Teillastbereich die Simulation eines solchen Strömungsgebiets instabil wird.

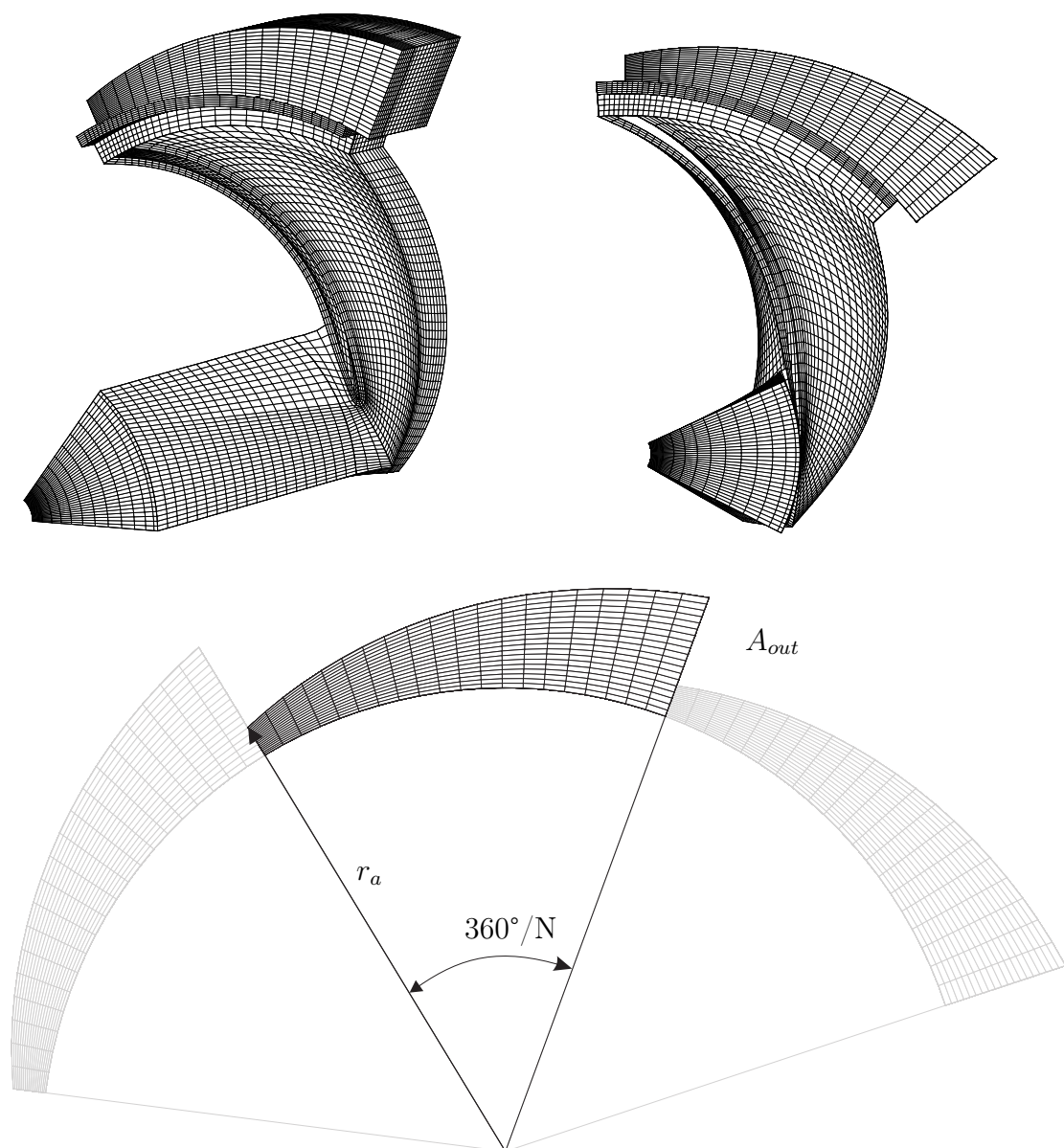


Abbildung 2.16: Spiralteilstück

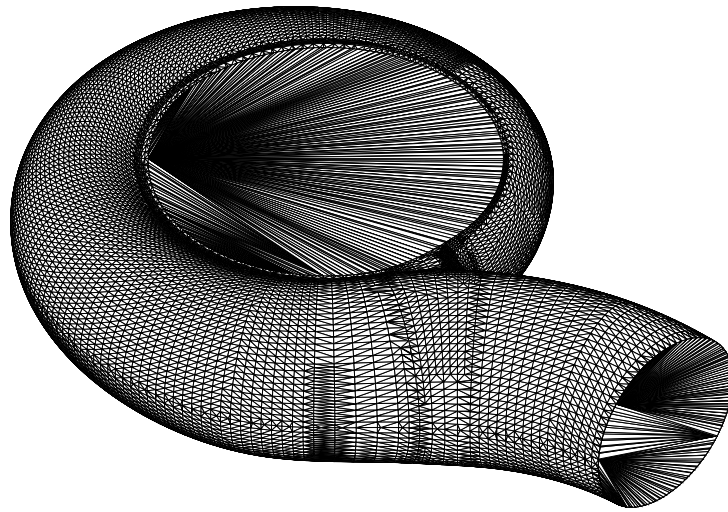


### 2.6.8 Spirale

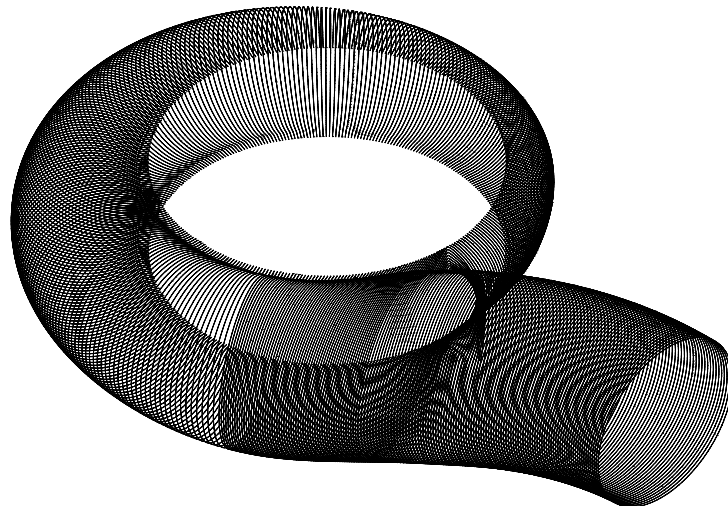
Die im CAD erstellte Geometrie des durchströmten Spiralvolumens wird als linearisierte Fläche im STL-Format eingelesen, s. Abbildung 2.17, Bild a). Das STL-Format, s. BURNS [15], beschreibt einen Körper durch seine Oberfläche, wobei ausschließlich Dreiecke zum Einsatz kommen. Das STL-Format ist unstrukturiert, es gibt keine Anforderungen an die Gestalt der verwendeten Dreiecke. Im ersten Vernetzungsschritt wird die Oberfläche mit tangential verlaufenden Ebenen geschnitten, wodurch die Fläche strukturiert wird, s. Abbildung 2.17, Bild b). Auf den Schnittebenen ist die Oberfläche nun eine stückweise lineare Kurve. Diese Berandungskurven können bei Bedarf in tangentialer Richtung mittels Spline-Interpolation geglättet werden. Die Berandungskurven werden genutzt, um unter Zuhilfenahme einer geeigneten Partitionierung in der Ebene der Kurve ein Flächennetz zu erzeugen. Durch die Verbindung der einzelnen zweidimensionalen Netze entlang der tangentialen Richtung entsteht das strukturierte Volumennetz, s. Abbildung 2.17, Bild c).

Der Netzgenerator für die Spiralgehäuse folgt der Strategie 2, die in Abschnitt 2.4.3 vorgestellt wird, d.h. der benetzte Fluidraum wird in einem externen CAD-Tool beschrieben und für die Simulation im IDS aufbereitet. Die Realisierung von Parameterstudien ist mit diesem Vernetzungsverfahren möglich, wenn die benötigten tangentialen Netzebenen direkt ohne den Umweg über ein CAD-System erzeugt werden. Da die Simulationen der Spiralgehäuse immer die Simulation eines Impellers bedingen, die selbst im stationären Fall für mehrere Stellungen des Impellers durchgeführt werden müssen, sind Parametervariationen bei den derzeitigen Laufzeiten der Simulation nicht praktikabel. Der verwendete Netzgenerator wurde von WUNDERER [56] implementiert.

a) STL-Ausgangsdaten



b) Zwischenschritt radiale Netzebenen



c) fertiges Netz

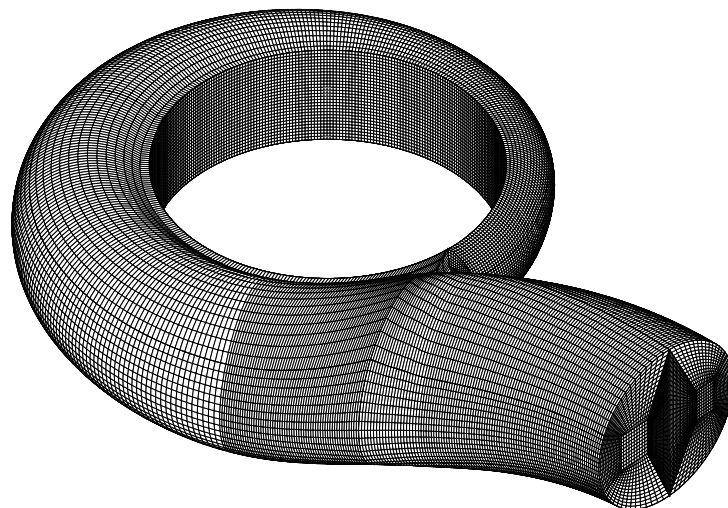


Abbildung 2.17: Schritte der Spiralvernetzung

## 2.7 Auswertung

### 2.7.1 Integrale Größen

In der hydraulischen Turbomaschine findet ein Energieaustausch statt: Entweder verrichtet das Laufrad der Maschine am Fluid Arbeit und führt diese Arbeit dem Fluid zu (Pumpe), oder das Fluid verrichtet die Arbeit am Laufrad, die Energie wird dem Fluid entzogen (Turbine). Dabei wird mechanische Leistung  $L_{mech}$  ausgetauscht. Zusätzlich entstehen durch die Energiewandlung Verluste, die es zu minimieren gilt.

Für die hydraulische Turbomaschine wird die spezifische Druckstutzenarbeit  $Y_{t,th}$  wie folgt definiert:

$$Y_{t,th} = gH_{t,th} = \frac{p_{out} - p_{in}}{\rho} + \frac{c_{out}^2 - c_{in}^2}{2} + g(z_{out} - z_{in}) \pm \frac{p_v}{\rho}, \quad (2.44)$$

$$Y_{t,th} = gH_{t,th} = \frac{1}{\rho} (p_{t,out} - p_{t,in}) \pm \frac{p_v}{\rho}. \quad (2.45)$$

Dabei gilt das positive Vorzeichen von  $\frac{p_v}{\rho}$  für die Pumpe und das negative für die Turbine.

Die spezifische Druckstutzenarbeit stellt für die Pumpe die pro Masseinheit an das Fördermedium übertragene Energie dar, die bis auf den Druckverlust  $p_v$  nutzbar ist. Im Fall der Turbine ist  $Y_{t,th}$  die vom Fluid in der Maschine abgegebene Gesamtleistung.

Im Schaufelbereich gilt gleichzeitig die durch die Schaufel umgesetzte Energie:

$$Y_{t,th} = gH_{t,th} = c_{u2}u_2 - c_{u1}u_1 \quad (2.46)$$

Es ist sinnvoll, alle Größen zu entdimensionieren, um Strömungsmaschinen unterschiedlicher Baugröße besser vergleichen zu können. Dazu wird der Durchmesser des Laufrades  $D$  und die Winkelgeschwindigkeit  $\omega$  herangezogen. Damit erhält man die Volumenzahl  $\phi$  und die Druckzahl  $\Psi$  sowie die dimensionslosen Größen für die Zirkulation  $\Gamma$  und den Druck  $P$ .

$$r_{ref} = \frac{D}{2} \quad (2.47)$$

$$u_{ref} = \omega \frac{D}{2} \quad (2.48)$$

$$\Psi = \frac{2gH}{u_{ref}^2} \quad (2.49)$$

$$\phi = \frac{Q}{\pi r_{ref}^3 \omega} \quad (2.50)$$

$$\Gamma = \frac{r \cdot c_u}{r_{ref}^2 \omega} \quad (2.51)$$

$$P = \frac{2p}{\rho r_{ref}^2 \omega^2} \quad (2.52)$$



Somit ergeben sich die referenzierten Größen wie folgt:

$$\Psi_t = P_{t,out} - P_{t,in} \quad (2.53)$$

$$\Psi_{t,th} = \Psi_t \pm P_v \quad (2.54)$$

$$\Psi_{t,th} = \frac{2}{u_{ref}^2} (c_{u2}u_2 - c_{u1}u_1) \quad (2.55)$$

Zusätzlich hat sich für die Beurteilung von Charakteristiken bei Pumpen der relative Volumenstrom  $q$  bewährt:

$$q = \frac{Q}{Q_{opt}} \quad (2.56)$$

Ist der optimale Volumenstrom  $Q_{opt}$  im Vorfeld nicht bekannt, wird alternativ auch der Volumenstrom im Auslegungspunkt  $Q_{ausl}$  als Bezugsgröße verwendet.

Für die Begutachtung von Strömungszuständen in Bauteilen, die strömungsumlenkend wirken, in denen aber keine Arbeit verrichtet wird und lediglich die Reibung den Energiegehalt ändert, ist es sinnvoll, zusätzlich die Größe  $\Psi_{rcu}$  einzuführen:

$$\Psi_{rcu} = 2\Gamma = 2 \frac{r_{loc} \cdot c_u}{r_{ref} u_{ref}} = 2 \frac{\omega r_{loc} \cdot c_u}{\omega r_{ref} u_{ref}} \quad (2.57)$$

$$\Psi_{t,th} = \Delta\Psi_{rcu} = \Psi_{rcu,out} - \Psi_{rcu,in} \quad (2.58)$$

Für das geführte System ist  $\Psi_{t,th}$  identisch mit einer Differenz aus  $\Psi_{rcu}$ , im ruhenden System ist  $\Psi_{t,th}$  gleich 0.  $\Psi_{rcu}$  ist auch für das ruhende System definiert und spiegelt das lokal wirkende Moment der Umfangskomponente der Strömung wider.  $\Psi_{t,th}$  und  $\Psi_t$  sind Differenzen, die sich aus den Feldwerten für  $P_t$  und  $\Psi_{rcu}$  ermitteln lassen.  $\Psi_{t,th}$  und  $\Psi_t$  spiegeln einen Energieaustausch wider. Bei der Auswertung eines CFD-Experiments ist es sinnvoll, die Größen  $\Psi_{rcu}$ , und  $P_t$  als Feldgrößen zu betrachten. Sie spiegeln die in einer Rechenzelle enthaltene Energie wider, nicht die abgegebene bzw. aufgenommene Leistung.

Werden die Bilanzierungsebenen korrekt gewählt, so entspricht die dimensionslose spezifische Schaufelarbeit  $\Psi_{t,th}$  der entsprechend entdimensionierten Wellenleistung der Turbomaschine:

$$L_{mech} = M \cdot \omega \quad (2.59)$$

$$\Psi_{mech} = \frac{2}{u_{ref}^2} \cdot \frac{M \cdot \omega}{\dot{m}} \quad (2.60)$$

$$\Psi_{t,th} \stackrel{!}{=} \Psi_{mech} \quad (2.61)$$

Wobei  $M$  das auf den Impeller wirkende Moment ist und  $\omega$  die Winkelgeschwindigkeit des Impellers.

Für die Pumpe gilt:

$$\begin{aligned}
 \text{sign}(M) = \text{sign}(\omega) &\Rightarrow \Psi_{t,th} > 0 \\
 \Psi_t > 0 \quad \wedge \quad P_v > 0 \\
 &\Rightarrow \Psi_t < \Psi_{t,th} \\
 \eta_{h,Pu} &= \frac{\Psi_t}{\Psi_{t,th}}
 \end{aligned} \tag{2.62}$$

Analog gilt für die Turbine:

$$\begin{aligned}
 \text{sign}(M) = -\text{sign}(\omega) &\Rightarrow \Psi_{t,th} < 0 \\
 \Psi_t < 0 \quad \wedge \quad P_v > 0 \\
 &\Rightarrow \Psi_t > \Psi_{t,th} \\
 \eta_{h,Tu} &= \frac{\Psi_{t,th}}{\Psi_t}
 \end{aligned} \tag{2.63}$$

Die Größen  $\Psi_t$  und  $\Psi_{t,th}$  werden im täglichen Gebrauch nicht vorzeichenbehaftet verwendet, ihre dimensionsbehafteten Entsprechungen werden als Förderhöhe oder Fallhöhe bezeichnet, wodurch das Vorzeichen ausgedrückt wird. Das Vorzeichen der Schaufelarbeit erschließt sich dann aus dem Kontext.

Der Betriebspunkt einer Strömungsmaschine kann durch die Volumen- und Druckzahl  $(\phi, \Psi)$  oder auch über die spezifische Drehzahl beschrieben werden:

$$n_q = n \cdot \frac{Q^{1/2}}{H^{3/4}} \cdot \frac{[H = 1m]^{3/4}}{[Q = 1m^3/s]^{1/2}} \tag{2.64}$$

Die spezifische Drehzahl spiegelt den aktuellen Betriebszustand der Turbomaschine wider. Es ist üblich, die spezifische Drehzahl im Optimalpunkt anzugeben, um die Radform zu charakterisieren. Die Radformkennzahl (RFK) ist somit die spezifische Drehzahl im Optimalpunkt  $n_{q,opt}$ . Die Radformkennzahl beschreibt die Radgeometrie, Turbomaschinen, die ein gleiches  $n_{q,opt}$  aufweisen, sind geometrisch ähnlich. Generell wird bei Angabe der spezifischen Drehzahl  $n_q$  als Radformkennzahl auf die Angabe des Zusatzindex *opt* verzichtet.

## 2.7.2 Lokale Größen

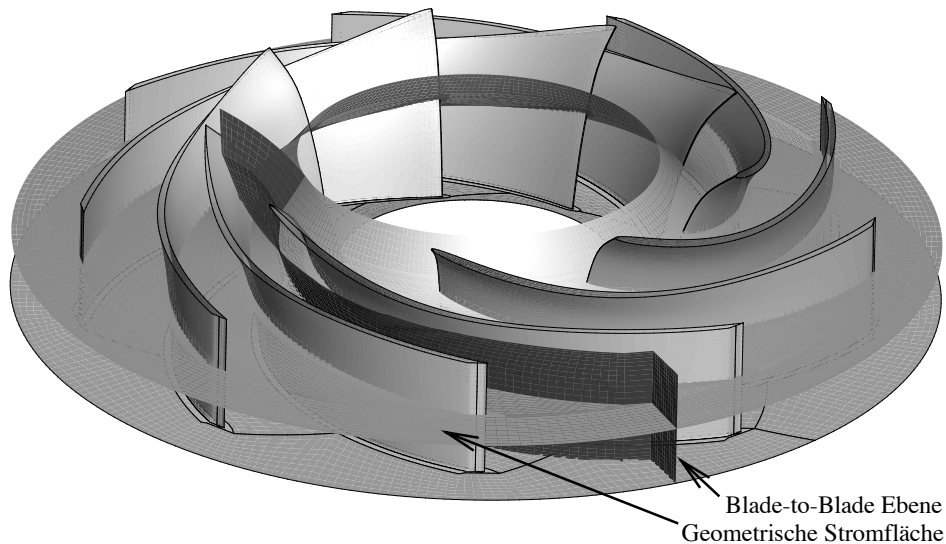
Das CFD-Experiment liefert als Ergebnis ein Feld, das im inkompressiblen Fall mit den Lösungsvariablen  $\vec{c} = (c_x, c_y, c_z)^T$ ,  $p$  und den Größen des Turbulenzmodells  $k$  und  $\varepsilon$  belegt ist. Für die Analyse lokaler Phänomene ist es sinnvoll, die Strömungslösung auf zweidimensionale Flächen zu reduzieren. Dabei bieten sich drei unterschiedliche Flächen an, s. Abbildung 2.18:

1. Die geometrische Stromfläche: Hierbei wird eine Rotationsfläche erzeugt, deren Spur im Achsschnitt eine Kurve ist, die die Hub- und Shroudkontur interpoliert. Sie ist eine gute Näherung einer Stromfläche. Diese Fläche ist i.d.R. zweifach

gekrümmt und nicht abwickelbar. Als Projektionsvorschrift bietet sich eine winkeltreue konforme Abbildung an. Dabei werden die Oberflächenkoordinaten eines Punktes auf der geometrischen Oberfläche  $(m, \varphi)$  in ebene Koordinaten  $(l, r_{ref} \cdot \varphi)$  transformiert:

$$l = r_{ref} \cdot \int_{m_2}^m \frac{dm}{r} - l_{max} \quad (2.65)$$

2. Die Blade-to-Blade-Ebene: Sie ist eine Fläche, die im rotationssymmetrischen Teil dem Achsschnitt entspricht. Im Schaufelkanal wird sie erzeugt, indem zwischen der Druckseite der einen Schaufel und der Saugseite der Nachbarschaufel interpoliert wird.
3. Die Normalenebene. Hier wird senkrecht zum Strömungskanal in Hub-Shroudrichtung eine Bilanzierungsfläche eingeführt. Sie ist i.d.R. eine Kegelmantelfläche. Sie dient in der Ermittlung von integralen Werten zur Verlustanalyse.



**Abbildung 2.18:** Flächendefinitionen für Postprocessing einer Strömungsmaschine am Beispiel eines offenen Pumpenlaufrads

### Ermittlung integraler Größen aus der CFD-Lösung

Die allgemeine Mittelung einer Strömungsgröße  $\Phi$  an einer Schnittfläche  $A = [\Delta A_0, \Delta A_1, \dots, \Delta A_i]$  wird abgebildet über eine impulsgemittelte Summe der Einzelwerte  $\Phi_i$  für die Flächenteilstücke  $\Delta A_i$ :

$$\bar{\Phi} = \frac{\sum_i \Phi_i \vec{c}_n \Delta A_i}{\sum_i \vec{c}_n \Delta A_i} = \frac{1}{Q} \sum_i \Phi_i \vec{c}_n \Delta A_i, \quad (2.66)$$

dabei ist  $\vec{c}_n$  die Geschwindigkeit normal auf die Fläche. Es bietet sich an, ausgezeichnete Netzebenen zu verwenden, damit keine Schnittoperationen benötigt werden, um die Werte für die Mittelung zu errechnen. Die verwendeten Werte entsprechen dann

direkt denen, die der Löser für die Berechnung verwendet und weisen keine Ungenauigkeiten hinsichtlich des geometrischen Schnitts und der Interpolation der knoten- oder zellzentriert vorliegenden Werte auf.

Da der Wert impulsgemittelt ist, kann er verschwinden, wenn z.B. durch die Bilanzierungsebene lediglich ein symmetrischer Wirbel fließt und das Profil des gemittelten Werts ebenfalls symmetrisch ist. Der impulsgemittelte Wert ist der Fluß der gemittelten Größe durch die Bilanzierungsebene. Sollen mittlere Geschwindigkeiten, mittlere skalare Größe o.ä. in solch einem Fall betrachtet werden, so bietet sich die Flächenmittelung an:

$$\bar{\Phi} = \frac{\sum_i \Phi_i \Delta A_i}{\sum_i \Delta A_i} = \frac{1}{A} \sum_i \Phi_i \Delta A_i . \quad (2.67)$$

Es ist zu beachten, dass so gewonnene integrale Größen nicht zur Verlustbetrachtung geeignet sind.

### Umfangsgemittelte Größen

Für die Begutachtung der Strömungslösung sind die mittleren Strömungsgrößen in einem Meridianschnitt wichtig, dazu wird für einen Kreis um die Maschinendrehachse  $r = \text{const}$  eine Mittelung benötigt:

$$\bar{\Phi} = \frac{\sum_{\varphi} \Phi \vec{c}_n d\varphi \cdot \delta r}{\sum_{\varphi} \vec{c}_n d\varphi \cdot \delta r} . \quad (2.68)$$

Dabei wird davon ausgegangen, dass die Werte zuerst mit einer Normalenebene aus dem Feld extrahiert worden sind, somit  $\vec{c}_n$  entsprechend definiert ist.

### Drehmoment

Das auf die Schaufel wirkende Moment kann direkt von der Oberfläche ermittelt werden. Die Schaufel ist durch die Randbedingung „Wand“ beschrieben, d.h. hier gilt, dass normal zu ihr kein Massenstrom auftreten kann. Die auf ein differentielles Wandelement wirkenden Kräfte sind die Druckkraft und die Reibkraft. Für das Moment um die Maschinendrehachse gilt:

$$M_z = (0, 0, 1)^T \cdot \left[ \sum_i (-p_i \cdot \vec{n}_i - \vec{\tau}_i |\vec{n}_i|) \times \vec{r}_i \right] \quad (2.69)$$

d.h. das Moment ergibt sich aus der Schubspannung und dem statischen Druck auf das finite Wandelement  $\Delta A_i$ , das am mittleren Ortsvektor  $r_i$  liegt und dessen Fläche durch den Normalenvektor  $\vec{n}_i$  beschrieben ist. Der Betrag des Normalenvektors  $\vec{n}_i$  ist der Betrag der Fläche,  $\vec{\tau}_i$  ist die Schubspannung, die an der Fläche auf das Fluid wirkt. Daraus ergibt sich die über die Welle ausgetauschte Leistung zu:

$$L_{\text{mech}} = M_z \cdot \omega = (0, 0, 1)^T \cdot \left[ \sum_i (-p_i \cdot \vec{n}_i - \vec{\tau} |\vec{n}_i|) \times \vec{r}_i \right] \cdot \omega_i. \quad (2.70)$$

$\omega_i$  ist die Winkelgeschwindigkeit des finiten Wandelements  $\Delta A_i$  um die Maschinendrehachse. Es wird vorausgesetzt, dass alle im Absolutsystem rotierenden Wände mit der An- oder Abtriebswelle verbunden sind. Analog zum Moment können über die Druck- und Reibkraft auf die rotierenden Wände auch die Radial- und Axialkraft auf den Impeller ermittelt werden. Diese Definition ist ebenfalls eindeutig, es treten keine Fragestellungen bzgl. der Systemgrenzen auf.

### Kennzahlen für Kavitationsneigung

Im Pumpenbau ist die wesentliche Kennzahl für die Kavitationsneigung einer Pumpe der Haltedruck, d.h. der Totaldruck am Laufradeintritt  $p_{t, \text{in}}$  vermindert um den Dampfdruck  $p_{va}$ , bzw. die Haltedruckhöhe, engl. Net Positive Suction Height (NPSH), s. GÜLICH [26]. Dabei wird der Druck im Saugstutzen betrachtet, bei dem im Pumpenlauf die Kavitation beginnt:

$$\text{NPSH} = \frac{p_{t, \text{in}} - p_{va}}{\rho \cdot g}. \quad (2.71)$$

Der NPSH-Wert wird in Metern angegeben. NPSH-Angaben sind gebräuchlich für gerade beginnende, *inzipiente* Kavitation, gekennzeichnet als  $\text{NPSH}_i$ , und für den Betriebspunkt, in dem durch Kavitation bereits 3% der Förderhöhe der Pumpe eingebüßt werden, bezeichnet als  $\text{NPSH}_{3\%}$ .  $\text{NPSH}_i$  muß im realen Experiment optisch ermittelt werden,  $\text{NPSH}_{3\%}$  kann dagegen am Prüfstand gemessen werden. In der CFD-Simulation kann aus jeder inkompressiblen Strömungslösung  $\text{NPSH}_i$  ermittelt werden: Dazu wird aus dem Gesamtfeld der minimale statische Druck  $\min(p_s)$  ermittelt. Über einen Korrekturdruck  $\Delta p_{\text{korrr}} = p_{va} - \min(p_s)$  kann der Druck am Laufradeintritt  $p_{t, \text{in}}$  auf das Niveau gesetzt werden, zu dem Kavitation zuerst eintreten würde. Da im Regelfall der Mindestdruck auf der Laufschaufeloberfläche auftritt, genügt es, aus dem Druckbeiwert  $c_p$  auf der ersten Laufschaufel den statischen Druck zu ermitteln. Für die Ermittlung des  $\text{NPSH}_{3\%}$ -Werts sind in der CFD-Simulation weitere Modellbildungsmaßnahmen notwendig, die das Kavitationsverhalten abbilden können, s. FROBENIUS [24]

Im Turbinenbau ist dagegen die Thoma-Zahl  $\sigma$  gebräuchlich:

$$\sigma = \frac{p - p_{va}}{\rho g H} \quad (2.72)$$

Die Thoma-Zahl charakterisiert für ein gegebenes Druckfeld die Kavitationsneigung und kann analog zu  $\text{NPSH}_i$  aus dem Ergebnis einer CFD-Simulation mit inkompressiblem Stoffgesetz ermittelt werden. Die Thomazahl  $\sigma$  bezieht sich ebenfalls auf den niedrigsten statischen Druck im Feld, je nach Maschine wird ein  $\sigma_{\text{krit}}$  angegeben, das die Maschine einhalten muß, damit keine Kavitation entsteht. Bei LEPACH [36] ist folgender Zusammenhang für  $\sigma_{\text{krit}}$  angegeben:

$$\sigma_{\text{krit}} \leq \sigma_{\text{anl}} = \frac{1}{H_t} \left( \frac{p_b - p_{va}}{\rho g} - h_s \right) \quad (2.73)$$

Dabei ist  $p_b$  der barometrische Außendruck und  $h_s$  die Saughöhe.

### Automatisch ermittelte Größen

Für die Ermittlung integraler Größen an ausgezeichneten Netzebenen sind geeignet:

1. Einströmflächen für die Ermittlung des entdimensionierten Totaldrucks am Einlaß  $P_{t, \text{in}}$
2. Ausströmflächen für die Ermittlung des entdimensionierten Totaldrucks am Auslaß  $P_{t, \text{out}}$
3. rotierende Wände für die Ermittlung der entdimensionierten Wellenleistung  $\Psi_{\text{mech}}$

Aus diesen Werten und mit den Betrachtungen aus (2.62) und (2.63) läßt sich ohne Kenntnis der tatsächlich zugrundeliegenden Geometrie für eine beliebige CFD-Simulation einer radialen Turbomaschine die entdimensionierte Totaldruckdifferenz des betrachteten Systems  $\Psi_{t, \text{sys}}$ , dessen Wellenleistung  $\Psi_{\text{mech}}$  und somit auch zuverlässig der Wirkungsgrad des betrachteten Systems  $\eta_{\text{sys}}$  ermitteln:

$$\Psi_{t, \text{sys}} = P_{t, \text{out}} - P_{t, \text{in}} \quad (2.74)$$

$$\eta_{\text{sys}} = \begin{cases} \Psi_{t, \text{sys}} \cdot \Psi_{\text{mech}}^{-1} & \Psi_{t, \text{sys}} < \Psi_{\text{mech}} \\ \Psi_{t, \text{sys}}^{-1} \cdot \Psi_{\text{mech}} & \Psi_{t, \text{sys}} > \Psi_{\text{mech}} \end{cases} \quad (2.75)$$

Die so ermittelten Systemwerte sind unempfindlich gegenüber möglichen Fehlerquellen bei Schnitten, wie z.B. einer schlechten Wahl der Position der Schnittebenen. Es ist zu beachten, dass das ermittelte  $\eta_{\text{sys}}$  immer kleiner sein muß als ein korrekt aus Bilanzierungsebenen im Strömungsfeld ermitteltes  $\eta_h$ .

# Kapitel 3

## Echtzeitverhalten

Für den Integrated Design Process, s. Abschnitt 2.5.3, wird das QNS-Verfahren für die Vorooptimierung vorgeschlagen, um den möglichen Lösungsraum schnell abarbeiten zu können und zügig den Bereich des absoluten Optimums ermitteln zu können. Dies setzt voraus, dass die mit dem QNS-Verfahren ermittelten Systemantworten bzw. deren Änderungen denen des RANS-Verfahrens entsprechen. In [22] untersucht FIEREDER das QNS-Verfahren für die Simulation von Radialpumpen und untersucht das allgemeine Lösungsverhalten. Im Folgenden soll nun die Eignung des QNS-Verfahrens für die Vorooptimierung nachgewiesen werden.

### 3.1 Anwendung des QNS-Verfahrens

Für die pauschale Turbulenzkonstante  $C_t$  konnte FIEREDER [22] nachweisen, dass sie in einem breiten Bereich ohne Auswirkungen auf das Endergebnis gewählt werden kann. Die effektive Viskosität  $\nu_{\text{QNS}}$  ist für Radialpumpen ca. 150 mal größer als die laminare Viskosität, also  $C_t \approx 150$ . Die Abhängigkeit der integralen und lokalen Ergebnisse von dem pauschalierten Turbulenzfaktor  $C_t$  ist allerdings gering, so dass  $C_t$  fast beliebig im Intervall von 10 . . . 1000 gewählt werden kann. Die endgültige Wahl des Parameters kann also rein von Stabilitäts- und Geschwindigkeitsüberlegungen abhängig gemacht werden.

Da die Viskosität stark erhöht wird, muß die Wandbehandlung beim QNS-Verfahren angepaßt werden. Mit der Haftbedingung würde infolge der hohen Viskosität die Wandreibung deutlich überschätzt werden, so dass alle Wände reibungsfrei betrachtet werden müssen. Anstelle der Forderung, dass die Relativgeschwindigkeit des Fluids an der Wand 0 ist, wird im QNS-Verfahren nur gefordert, dass die Normalkomponente der Relativgeschwindigkeit an der Wand 0 ist. Durch das komplette Vernachlässigen der Reibung an der Wand bestehen beim QNS-Verfahren keine Einschränkungen hinsichtlich der  $y^+$ -Werte.

Ein Vorteil des QNS-Verfahrens ist, dass es mit demselben Löser auf denselben Netzen arbeiten kann wie eine RANS-Simulation. Es muß keine zusätzlich Infrastruktur für Vernetzung, Simulation oder Postprocessing geschaffen werden, d.h. einmal für das QNS-Verfahren aufbereitete Geometrien stehen auch für feinere Analysen zur Verfügung. Die Netze werden analog zu den RANS-Netzen erzeugt, wobei infolge der verrin-

gerten Reynolds-Zahl die Netze nur ca. 10% der Zellen einer RANS-Simulation benötigen. Ein QNS-Netz kann also aus einem RANS-Netz erzeugt werden, indem in jede Indexrichtung die Anzahl der Knoten halbiert wird, wodurch die Anzahl der Zellen auf 12.5% sinkt, umgekehrt kann ein QNS-Netz durch Verfeinern in jede Indexrichtung in ein RANS-Netz überführt werden.

Im Bereich kleiner  $n_q$  ist das QNS-Verfahren nur bedingt geeignet: Die auftretenden Sekundärströmungen im Schaufelkanal müssen durch das diskrete Rechenetz wiedergegeben werden, so dass die gewünschte Vergrößerung nicht möglich ist. Zusätzlich fallen bei niedrigen  $n_q$  Reibungseffekte deutlicher ins Gewicht, so dass die pauschalisierte Berücksichtigung der Turbulenz kein geeignetes Verlustmodell mehr darstellt.

## 3.2 Integrale Größen

In Abbildung 3.1 ist die Abweichung integraler Größen in Abhängigkeit vom pauschalierten Turbulenzkoeffizienten dargestellt. Die relative Abweichung  $\delta$  ist für eine beliebige integrale Größe  $\Phi$  wie folgt definiert:

$$\delta(\Phi) = \frac{\Phi_{\text{QNS}}}{\Phi_{\text{RANS}}} \quad (3.1)$$

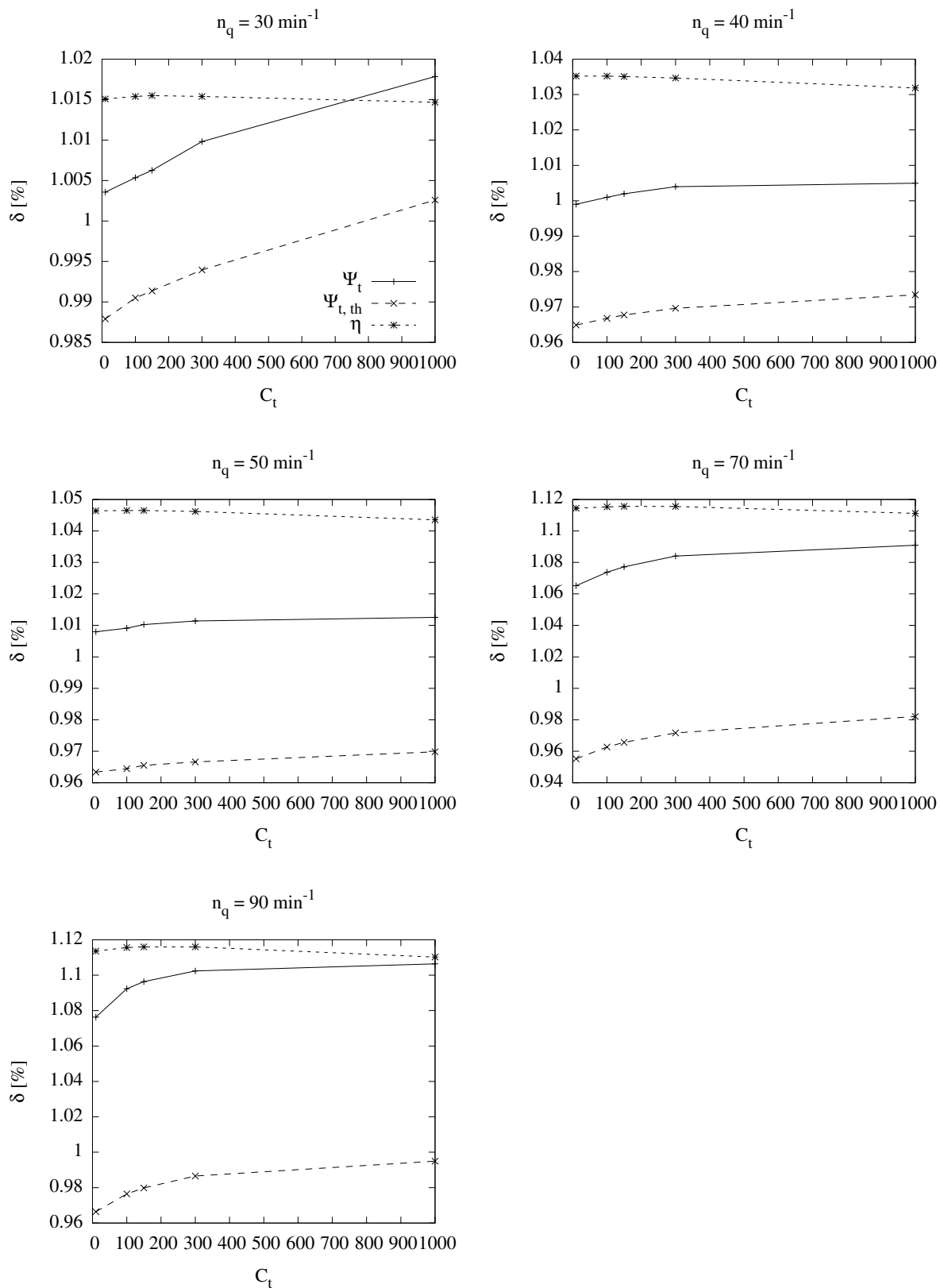
Es ist gut zu erkennen, dass der QNS-Wirkungsgrad unabhängig vom gewählten  $C_t$  und durchweg höher als bei der RANS-Simulation ist. Die dimensionslose Schaufelarbeit bzw. theoretische Totaldruckzahl  $\Psi_{t,th}$  und somit die aufgewendete Leistung ist für QNS im gesamten Bereich geringer, lediglich bei  $n_q = 30 \text{ min}^{-1}$  und  $C_t = 1000$  erreicht sie den Wert der RANS-Simulation. Die Förderhöhe bzw. Totaldruckzahl ist durchweg höher, wobei die Abweichungen mit  $n_q$  steigen. Für  $n_q = 30 \text{ min}^{-1}$  sind die QNS-Werte noch in einem schmalen Band um die RANS-Werte,  $\pm 2\%$ , und steigen auf eine Überbewertung der Förderhöhe von ca. 12 % an der betrachteten  $n_q$ -Obergrenze.

Die Analyse der verwendeten Geometrien der Radialpumpen für  $n_q = 70 \text{ min}^{-1}$  und  $n_q = 90 \text{ min}^{-1}$  hat gezeigt, dass für beide  $n_q$  der Optimal- und der Auslegungspunkt weit auseinander liegen. Für die Radialpumpe  $n_q = 90 \text{ min}^{-1}$  liegt der Optimalpunkt bei  $\phi_{opt} \approx 0.8 \cdot \phi_{ausl}$ , für  $n_q = 70 \text{ min}^{-1}$  noch niedriger bei  $\phi_{opt} \approx 0.7 \cdot \phi_{ausl}$ . In den ermittelten jeweiligen Optimalpunkten der beiden Radialpumpen reduziert sich die Abweichung der vom QNS-Verfahren ermittelten integralen Werte von den Werten des RANS-Verfahrens, sie beträgt maximal 7%, s. Abbildung 3.2. Der ermittelte Wirkungsgrad  $\eta$  hängt nur in geringem Maße vom pauschalierten Turbulenzparameter  $C_t$  ab und variiert vor allem mit  $n_q$ .

Eine Möglichkeit, vom Ergebnis der QNS-Rechnung auf ein mögliches RANS-Ergebnis zu schließen, ist es,  $\eta_{\text{QNS}}$  zu ermitteln, einen Korrekturfaktor aus dem Graphen für  $\delta(\eta)$  aus Abbildung 3.2 zu ermitteln und  $\Psi_{t,QNS}$  und  $\Psi_{t,QNS}$  zu gleichen Teilen zu verändern, so dass das ermittelte korrigierte  $\eta$  erreicht wird. Die so korrigierten Werte stellen dann eine gute Abschätzung der mit RANS zu ermittelnden Totaldruck- und Dralldifferenzen dar.

Für die Festlegung der zu nutzenden  $C_t$ -Werte ergibt sich aus Abbildung 3.1 kein zwingender Zusammenhang. Je niedriger  $C_t$  gewählt wird, umso besser stimmt die





**Abbildung 3.1:** Abweichung der integralen Größen  $\Psi_t$ ,  $\Psi_{t,th}$  und  $\eta$  von der pauschalen Turbulenzkonstante  $C_t$

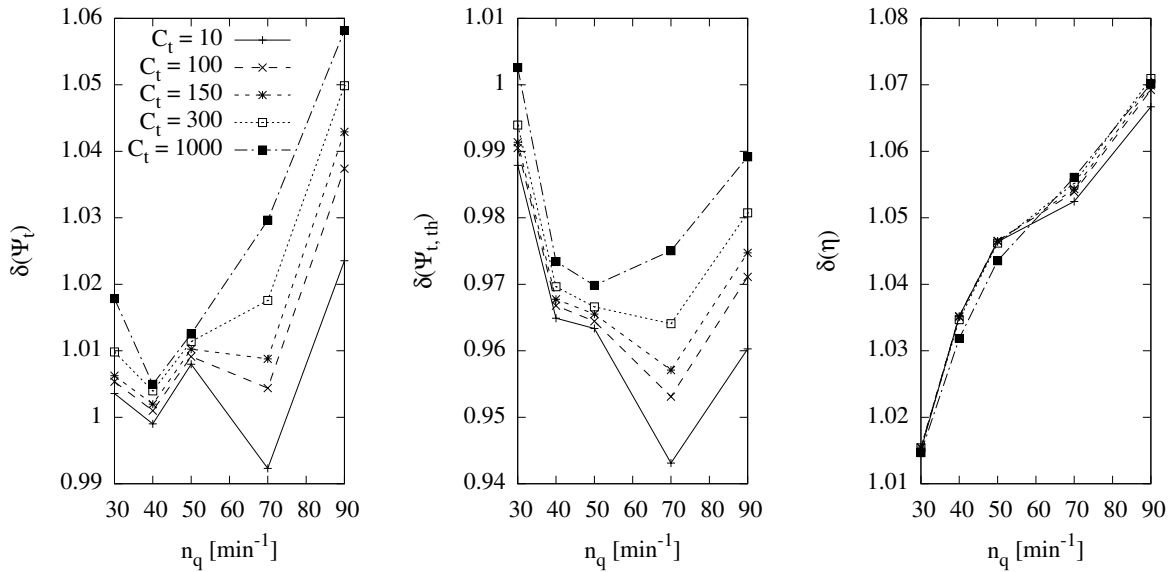


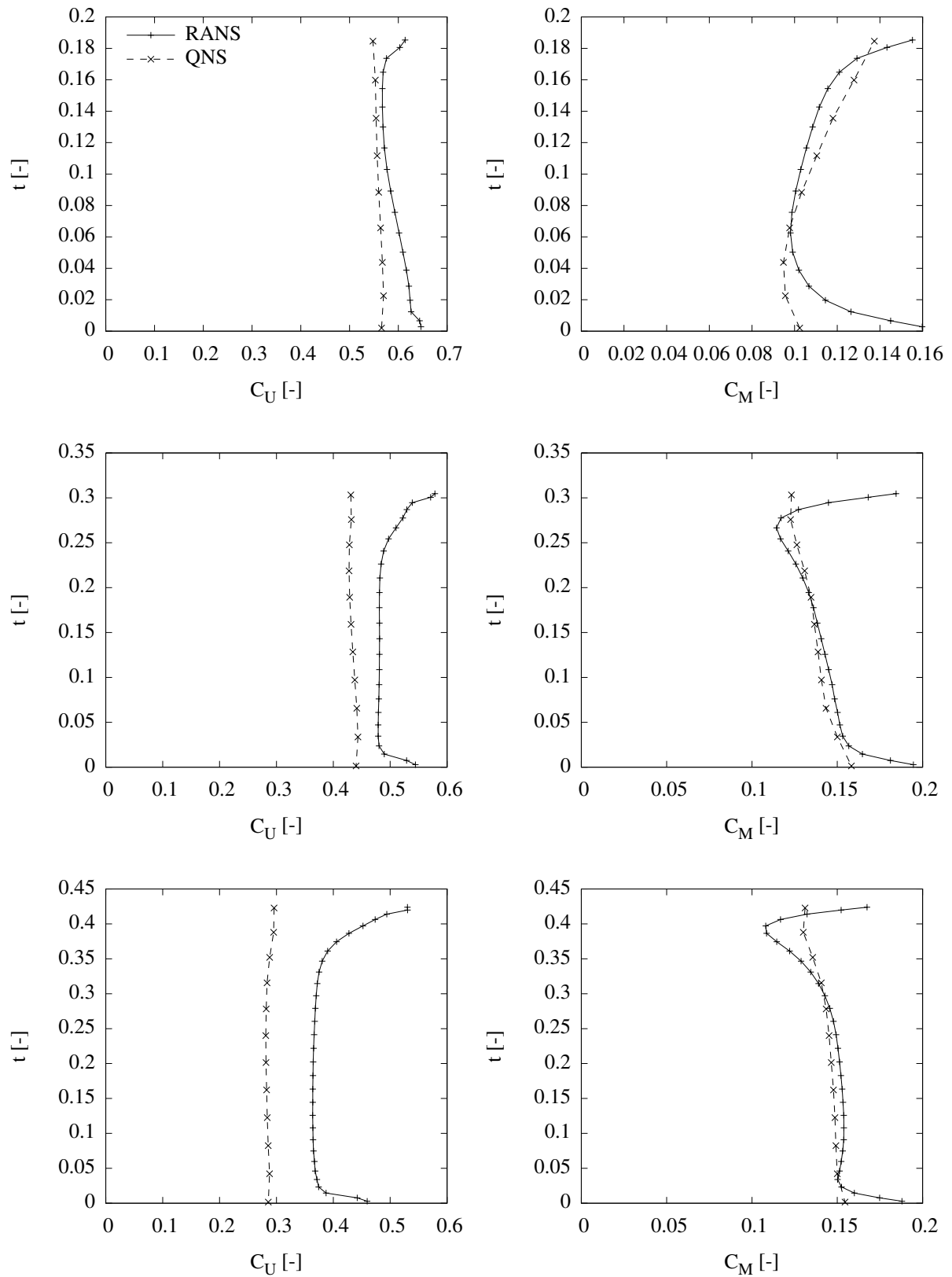
Abbildung 3.2: Abhängigkeit integraler Werte von  $C_t$ , korrigiert für hohe  $n_q$

Totaldruckerhöhung, je höher  $C_t$  gewählt wird, umso besser stimmt die notwendige Umlenkung. Im Folgenden soll für Radialpumpen mit den  $n_q = 30 \text{ min}^{-1}$ ,  $50 \text{ min}^{-1}$  und  $70 \text{ min}^{-1}$  untersucht werden, inwiefern das QNS-Verfahren für die Vorooptimierung geeignet ist. Dabei wird  $C_t = 10$  gesetzt, um eine möglichst gute Näherung der Totaldruckerhöhung zu erhalten. Für  $n_q = 30 \text{ min}^{-1}$  wird aus Stabilitätsgründen  $C_t = 100$  gesetzt.

### 3.3 Lokale Größen

In Abbildung 3.3 sind die umfangsgemittelten Abströmprofile beider Simulationen kurz hinter dem Laufschaufelaustritt über der auf den Radius des Laufrads referenzierten Kanalbreite gegenübergestellt. Die  $C_m$ -Verteilungen entsprechen sich recht gut, berücksichtigt man die stark verringerte Auflösung der Lösung der QNS-Simulation und den fehlenden Einfluß der Haftbedingung an der Wand. Die  $C_u$ -Verteilung zeigt deutliche Unterschiede: die QNS-Lösung weist eine geringere Umfangsgeschwindigkeit auf, d.h. das von der QNS-Simulation ermittelte  $\Psi_{t,th}$  liegt wie erwartet unter dem einer RANS-Lösung.

In den Abbildungen 3.4, 3.5 und 3.6 sind die Druckverteilungen um die Schaufel der Radialpumpenlaufräder zu sehen, die mit dem QNS- und dem RANS-Verfahren ermittelt worden sind. Die Druckverteilungen zeigen eine sehr gute Übereinstimmung, gerade das Druckprofil an der Eintrittskante der Pumpe ist gut abgebildet, was bei der Beurteilung der Kavitationsneigung hilfreich ist. Es sind deutliche Unterschiede in der Netzfeinheit zu erkennen, feinere Strukturen können im QNS-Verfahren nicht aufgelöst werden. Jedoch ist eine gute Übereinstimmung der Druckprofile zu beobachten. Das erreichte Druckniveau am Ende der Schaufel ist bei der QNS-Simulation etwas höher als bei der RANS-Simulation, was sich mit den Beobachtungen aus dem vorangegangenen Abschnitt deckt.



**Abbildung 3.3:** Geschwindigkeitsverteilung nach der Austrittskante für QNS, RANS,  $n_q = 30 \text{ min}^{-1}$  (oben),  $n_q = 50 \text{ min}^{-1}$  (mitte),  $n_q = 70 \text{ min}^{-1}$  (unten)

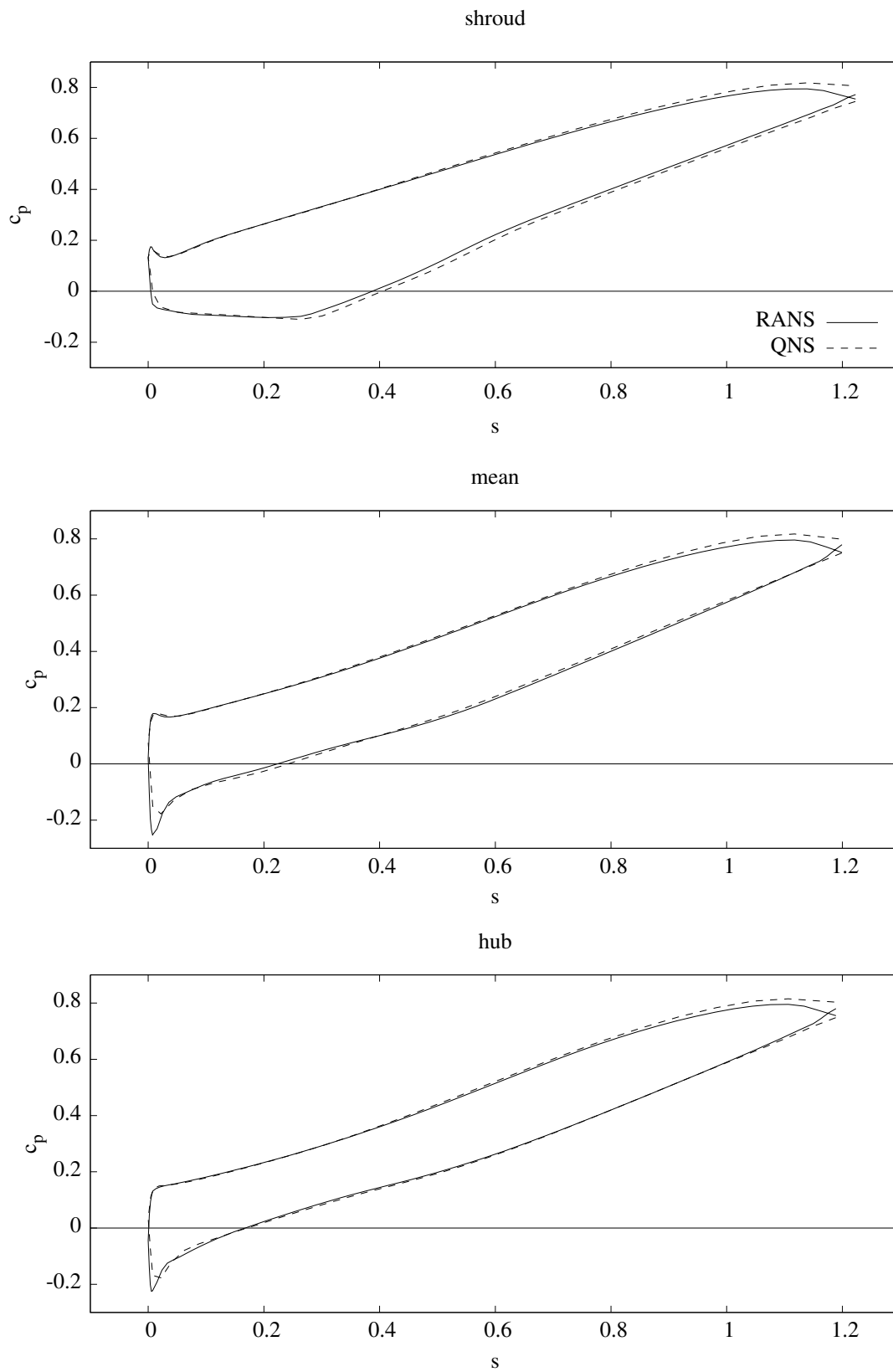


Abbildung 3.4: Druckverteilung QNS, RANS,  $n_q = 30 \text{ min}^{-1}$

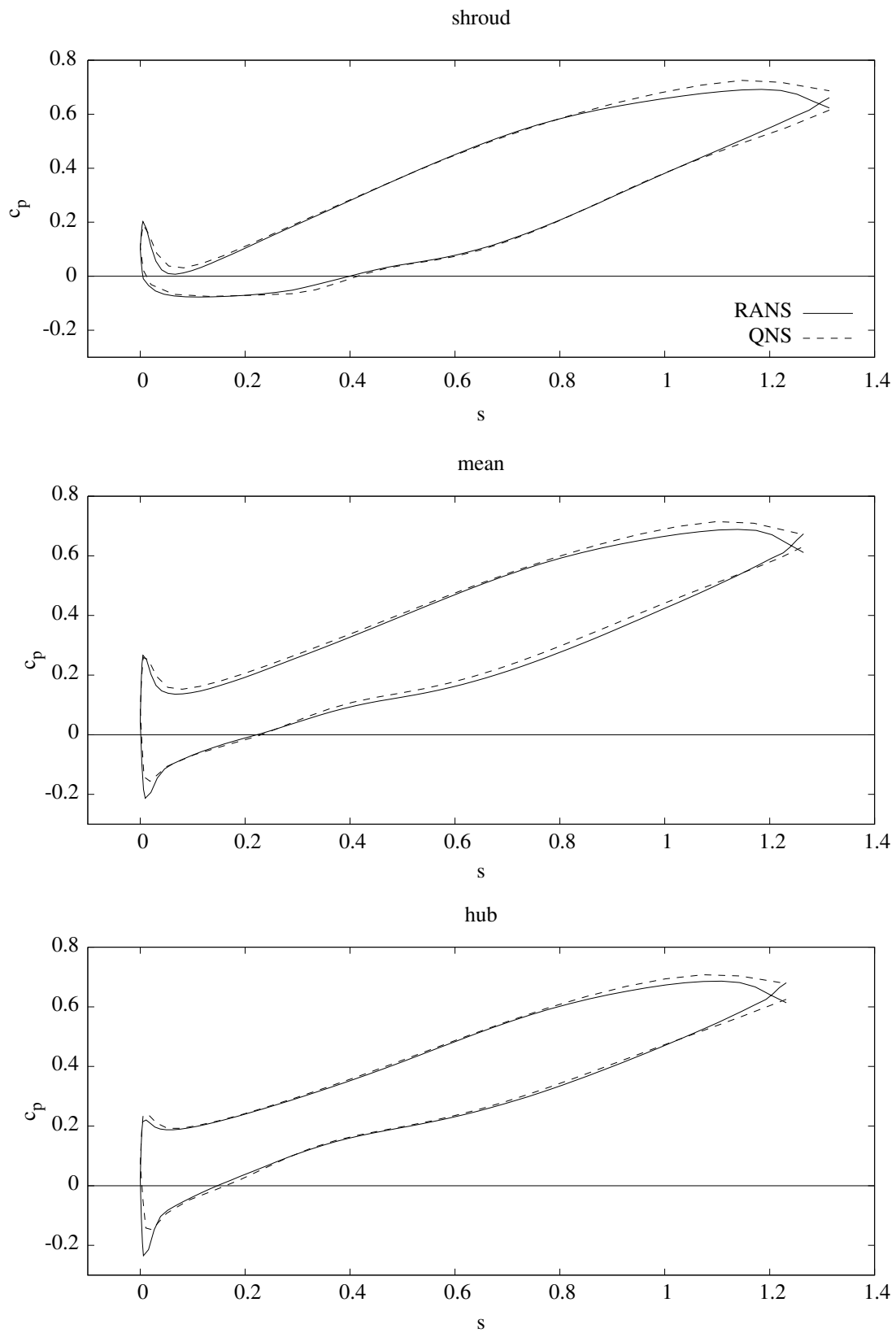


Abbildung 3.5: Druckverteilung QNS, RANS,  $n_q = 50 \text{ min}^{-1}$

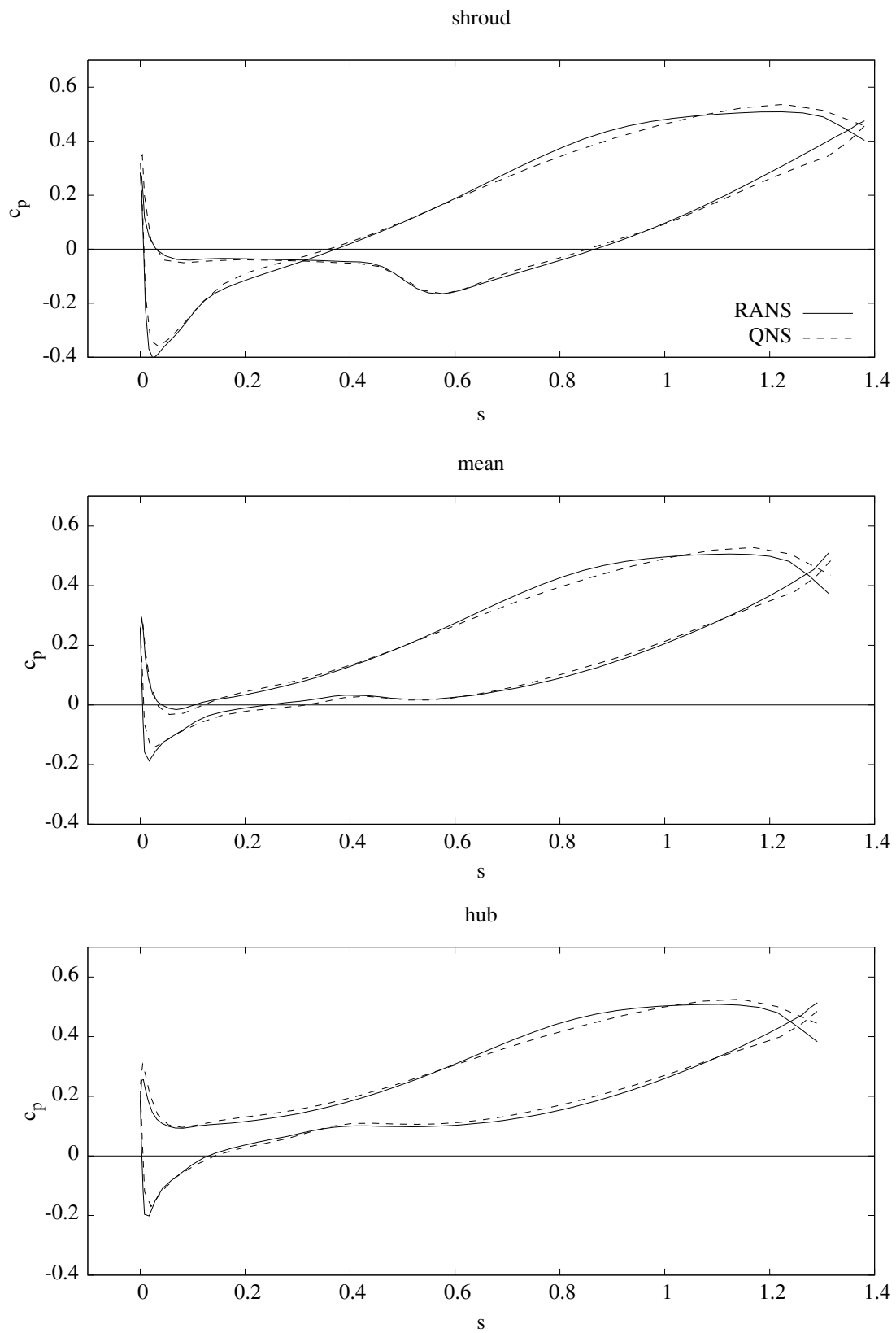


Abbildung 3.6: Druckverteilung QNS, RANS,  $n_q = 70 \text{ min}^{-1}$

## 3.4 Verhalten bei Änderungen

Um die Tauglichkeit des QNS-Verfahrens für die Begutachtung von Geometrieänderungen zu prüfen, wurden für 3 ausgewählte Pumpen Variationsrechnungen durchgeführt, wobei die variierten Geometrien jeweils mit QNS-Verfahren sowie dem RANS-Verfahren simuliert wurden. Die Pumpen haben die Radformkennzahlen  $n_q = 30 \text{ min}^{-1}$ ,  $n_q = 50 \text{ min}^{-1}$  und  $n_q = 70 \text{ min}^{-1}$ . Als integrale Größen werden die Druckzahl der Förderhöhe  $\Psi_t$ , die theoretische Schaufelarbeit  $\Psi_{t,th}$ , das Druckzahläquivalent der von der Schaufel geleisteten mechanischen Arbeit  $\Psi_{mech}$ , der minimale Druckbeiwert an der Schaufel  $c_{p,min}$  und die Wirkungsgrade des Laufrads  $\eta_{h,La}$  und der nach Gleichung (2.75) definierte Wirkungsgrad des betrachteten Strömungsgebiets  $\eta_{sys}$  ermittelt. Als wesentliche Eingangsgrößen wurden die Schaufelwinkel  $\beta_{s1}$  und  $\beta_{s2}$ , der Umschlingungswinkel  $\Delta\varphi$  und die Schaufellänge in der Meridiankontur  $\Delta l$  in einem Bereich von  $\pm 10\%$  um den Wert der Ausgangsgeometrie variiert.

Es ist zu erwarten, dass die Systemantworten für die beiden Modellierungen sich grundlegend unterscheiden. Da das QNS-Verfahren für die schnelle Optimierung eingesetzt werden soll, sollten sich allerdings die Änderungen der integralen Antworten auf Änderungen der Eingangsgrößen gleich oder zumindest ähnlich verhalten. Lokale Minima und Maxima der integralen Größen sollten für beide Verfahren an gleichen Orten liegen. Abbildung 3.7, Abbildung 3.8, Abbildung 3.9 sowie Abbildung 3.10 zeigen die Systemantworten, jeweils referenziert auf den integralen Wert, der sich für die Ausgangsgeometrie ergibt.

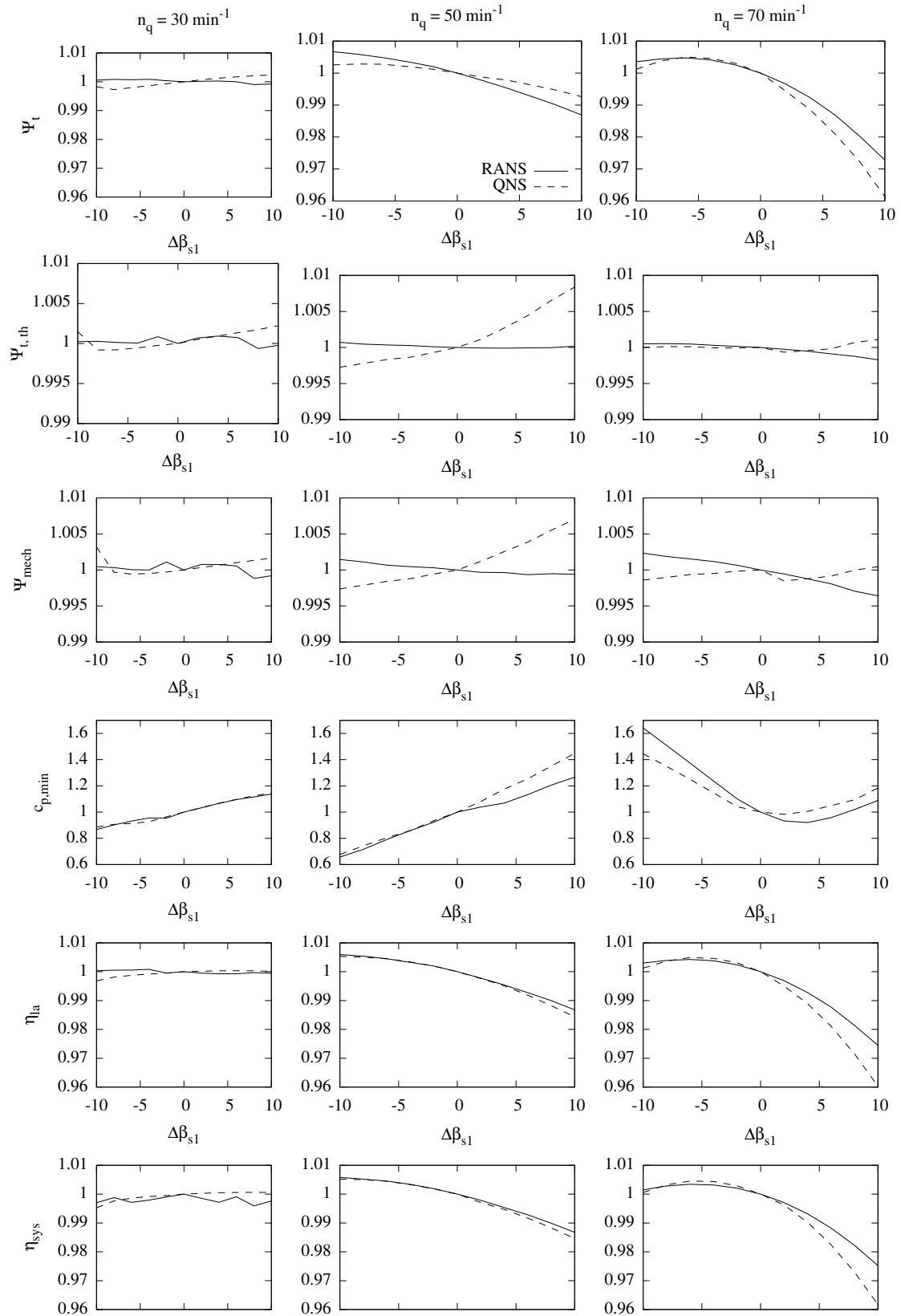


Abbildung 3.7: Normierte integrale Systemantworten für  $\Delta\beta_{s1}$  für  $n_q = 30, 50$  und  $70 \text{ min}^{-1}$



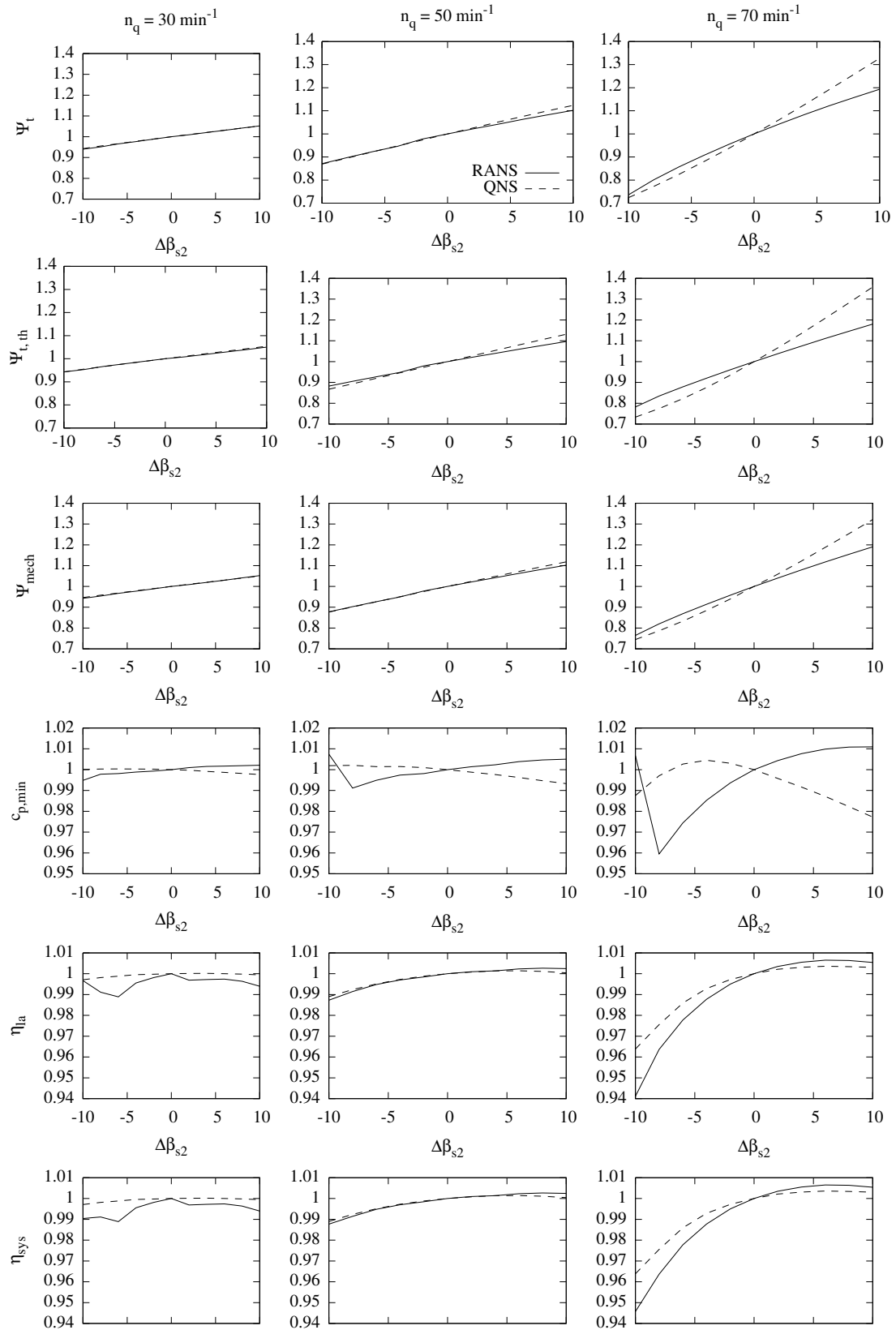


Abbildung 3.8: Normierte integrale Systemantworten für  $\Delta\beta_{s2}$  für  $n_q = 30, 50$  und  $70 \text{ min}^{-1}$

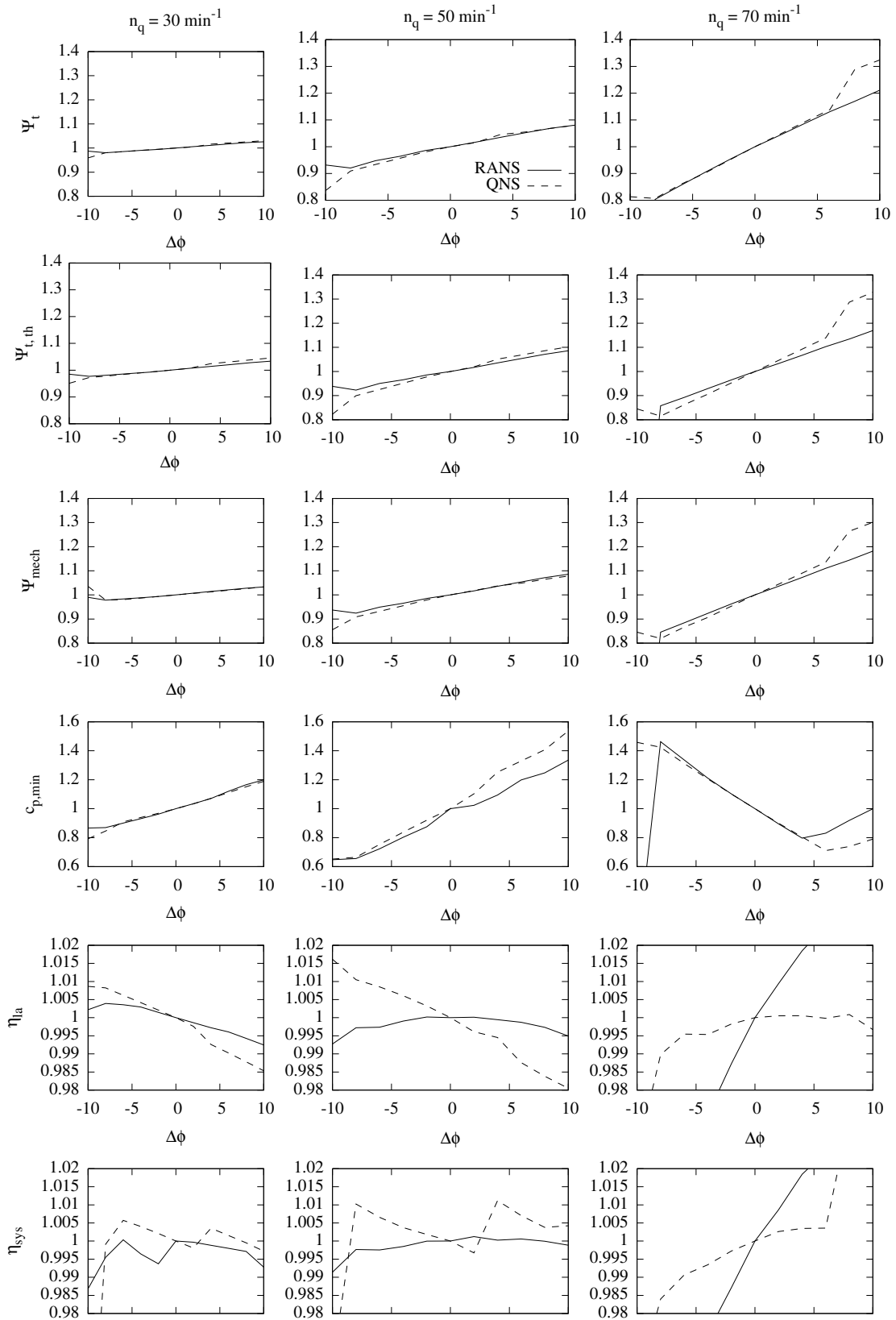


Abbildung 3.9: Normierte integrale Systemantworten für  $\Delta\varphi$  für  $n_q = 30, 50$  und  $70 \text{ min}^{-1}$

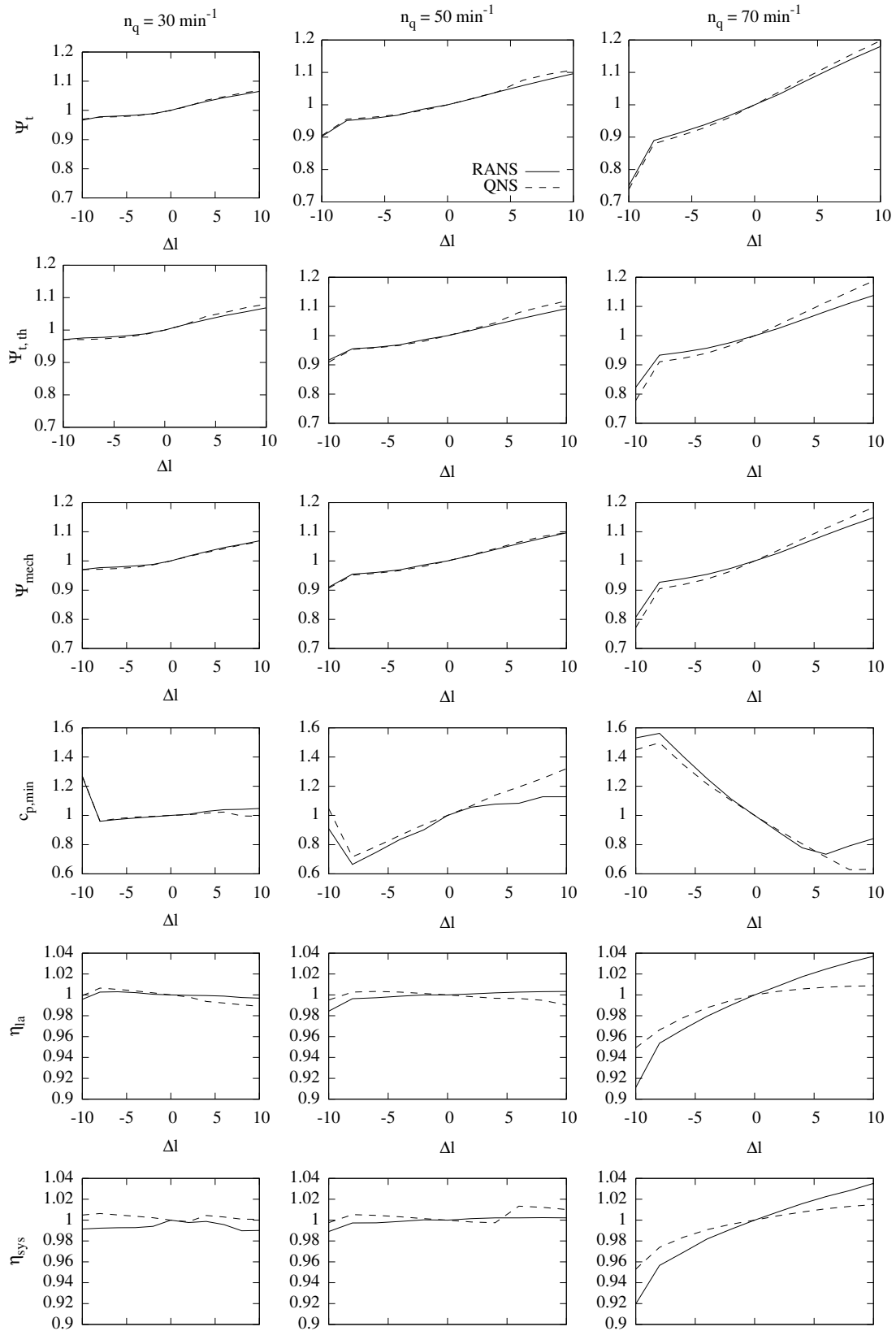


Abbildung 3.10: Normierte integrale Systemantworten für  $\Delta l$  für  $n_q = 30, 50$  und  $70 \text{ min}^{-1}$

### 3.5 Laufzeitverhalten

Das Laufzeitverhalten wurde nicht unter reproduzierbaren Bedingungen ermittelt, die durchgeführten Rechnungen liefen auf einem Arbeitsplatzrechner, der parallel zusätzliche Programme auszuführen hatte. Die einzelnen Simulationen wurden nicht wiederholt, um diese Störeinflüsse abschätzbar zu machen. Dennoch kann aus den insgesamt 264 Simulationen ein Bild gewonnen werden, wie sich die Laufzeit des QNS-Verfahrens zum RANS-Verfahren verhält. In Tabelle 3.1 sind die ermittelten Laufzeiten nach  $n_q$  geordnet angegeben. Die relative Laufzeit bezieht sich dabei auf die schnellste Simulation im RANS-Verfahren. Es zeigt sich, dass das QNS-Verfahren lediglich zwischen 5 - 10 % der Laufzeit einer RANS-Simulation benötigt.

$n_q$ [ $min^{-1}$ ]	Verfahren	Laufzeit [s]			rel. Laufzeit		
		min	Ø	max	min	Ø	max
30	QNS	10	16	29	0.06	0.10	0.18
	RANS	164	597	4619	1.00	3.64	28.16
50	QNS	5	8	14	0.03	0.05	0.09
	RANS	160	237	392	1.00	1.48	3.58
70	QNS	10	18	36	0.04	0.07	0.14
	RANS	258	552	1161	1.00	2.14	4.50

**Tabelle 3.1:** Laufzeitverhalten QNS und RANS

### 3.6 Beurteilung des QNS-Verfahrens

Die Graphen in Abbildung 3.7, Abbildung 3.8, Abbildung 3.9 sowie Abbildung 3.10 zeigen immer dann gute Übereinstimmung der relativen Systemantworten für das RANS und das QNS-Verfahren, wenn diese hoch ausfallen. Die Fälle, in denen die relativen RANS-Systemantworten gering ausfallen, führen teilweise zu schlechten Übereinstimmungen mit der QNS-Systemantwort. Auch in den Randbereichen, bei extremen Änderungen der Geometrie, kommt es zu größeren Abweichungen, s. u.a. Abbildung 3.9,  $n_q = 70 \text{ min}^{-1}$ . Vor dem Hintergrund der geringen Laufzeitkosten des QNS-Verfahrens sind diese Abweichungen jedoch vertretbar, wenn die QNS-Rechnungen im Sinne des Multi-Level-CFD-Ansatzes verwendet und für eine Vorooptimierung herangezogen werden.

Eine weitere Überprüfung der mit Hilfe des QNS-Verfahrens ermittelten Optima, zumindest hinsichtlich der integralen Werte, muß jedoch noch durch einen Vergleich mit RANS-Simulationen durchgeführt werden.

# Kapitel 4

## Automatisierung des CFD-Experiments

Im vorangegangenen Abschnitt wurde ein schnelles Verfahren für die Optimierung von Turbomaschinen untersucht, mit dem Ziel, die Rechenzeiten für den Optimierungsprozeß möglichst gering zu halten. Im nun folgenden Abschnitt sollen die Zeiten reduziert werden, die der Anwender von CFD-Verfahren für das Erstellen eines numerischen Experiments benötigt. Das CFD-Experiment soll weitestgehend automatisiert durchgeführt werden können, wenn die notwendigen Ausgangsdaten zur Verfügung stehen.

Ein automatisiert durchgeführtes CFD-Experiment muß folgenden Anforderungen genügen:

- Zuverlässigkeit, d.h. das CFD-Experiment sollte im Rahmen der Modellierung zu einem korrekten Ergebnis kommen.
- Robustheit, d.h. Fehleingaben müssen vermieden, identifiziert und wenn möglich berichtigt werden, schlimmstenfalls muß das CFD-Experiment mit entsprechender aussagekräftiger Fehlermeldung abgebrochen werden.
- Effizienz, d.h. die angebotenen Rechnerressourcen sind bestmöglich zu nutzen.

Diese Vorgaben hängen in erster Linie von dem verwendeten CFD-Löser ab. Jedoch weisen CFD-Löser i.A. eine große Anzahl von Parametern auf, die das Rechenverhalten beeinflussen. Ein kundiger CFD-Experimentator kennt diese und weiß sie der Situation entsprechend einzusetzen. Es gilt also, das Verhalten des CFD-Lösers im Einzelnen kennenzulernen, die wichtigsten Parameter zu identifizieren und die entsprechenden Werte vorzubelegen.

Der Punkt Robustheit im Sinne der Anwendungssicherheit ist dagegen eine Anforderung an die Programmiertechnik. Mögliche Fehleingaben müssen beim Entwurf der Benutzerschnittstelle bestenfalls im Vorfeld identifiziert werden und abgefangen werden. Dabei helfen Programmiergrundsätze wie z.B. „Don't Repeat Yourself“ (DRY), s. HUNTS ET. AL [30]. Hinsichtlich der notwendigen Parameter bedeutet das, dass Daten, die an mehreren Stellen des Programms benötigt werden, nur ein einziges Mal gespeichert werden sollen, um Widersprüche zu vermeiden. Das bedeutet, dass für das

CFD-Experiment ein Datenmodell aufgebaut werden muß, das Parameter und Ergebnisse der Simulation eindeutig adressierbar macht und ohne Redundanzen abbildet.

## 4.1 Effiziente Shared Memory Parallelisierung

Der vorliegende Code NS3D ist nach dem MPI-Standard parallelisiert, d.h. er kann mehrere der in aktuellen Rechnern vorhandenen Rechenkerne gleichzeitig nutzen, alternativ können mehrere Rechner über ein Netzwerk gekoppelt werden. Dabei wird in beiden Fällen das CFD-Experiment auf Basis der vorhandenen Blöcke aufgeteilt, d.h. jeder Prozess umfaßt mindestens einen Block. Die Werte an den Blockinterfaces werden dann zwischen den Prozessen ausgetauscht. Diese Form der Parallelisierung hat zwei Ziele:

1. Die Rechengeschwindigkeit kann durch die Parallelisierung deutlich erhöht und die Wartezeiten auf Simulationsergebnisse somit verkürzt werden.
2. Die Parallelisierung über mehrere Rechner ermöglicht es, sehr viel umfangreichere Gitter zu verarbeiten und etwaige Einschränkungen von Hardware und Betriebssystemen zu überwinden.

Damit die Parallelisierung funktioniert, müssen die vorhandenen Blöcke auf die verschiedenen Rechenkerne bzw. Rechner aufgeteilt werden. Die derzeit angebotenen Rechner verfügen über mindestens zwei Rechenkerne. Das FLM verfügt über mehrere Rechner, die mit insgesamt 4 Kernen arbeiten, und zusätzlich 4 sogenannte Hyperthreading-Kerne aufweisen. Aus Sicht des Betriebssystems sind somit 8 verschiedene Rechenkerne verfügbar. Um diese angebotene Rechenleistung auch im automatisierten CFD-Experiment nutzbar zu machen, ist eine automatisierte und optimale Verteilung der Blöcke auf verschiedene Prozesse notwendig.

### 4.1.1 Abschätzung der Anzahl möglicher Lösungen

Der Lösungsraum der Problemstellung „Verteilung von  $n$  Blöcken auf  $k$  Prozesse“ ist endlich, da das Problem diskret beschrieben ist. Grundsätzlich ist es also möglich, Brute-Force-Methoden bei der optimalen Verteilung der Blöcke anzuwenden. Für die Verteilung von  $n$  Blöcken auf  $k$  Prozesse bestehen insgesamt  $k^n$  Möglichkeiten:

$$M = k^n . \tag{4.1}$$

Gilt die Forderung, dass jeder Prozess mindestens einmal in der Kombination vorkommen muss, so verringert sich diese Zahl um  $(k - 1)^n$  Möglichkeiten für all diejenigen Verteilungen, bei denen ein Block fehlt, und dies in allen Permutationen. Dabei sind allerdings die Möglichkeiten mehrfach abgezogen, bei denen nur 2 Blöcke fehlen, usw. Dies muß rekursiv berücksichtigt werden, da im nächsten Schritt wieder die Lösungen mit 3 Blöcken zuviel hinzugerechnet worden sind. Somit ergibt sich als Zahl der Möglichkeiten,  $n$  Blöcke auf genau  $p$  Prozesse zu verteilen, wie folgt:

$$M = k^n + \sum_{i=1}^{k-1} \left[ -1^i \binom{k}{k-i} (k-i)^n \right] \quad (4.2)$$

Gilt zusätzlich, dass alle Gruppen untereinander austauschbar sind, so sind ebenfalls alle Permutationen herauszurechnen:

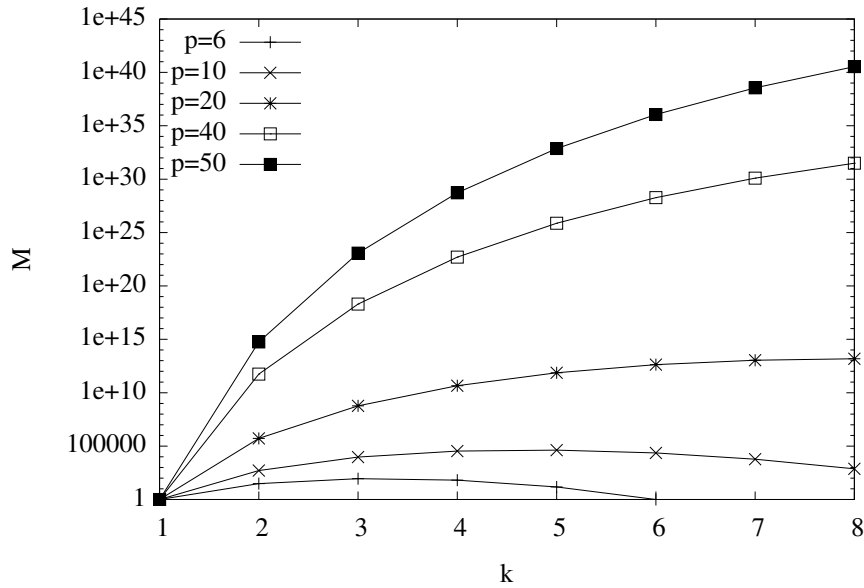
$$M = \frac{1}{k!} \left( k^n + \sum_{i=1}^{k-1} \left[ -1^i \binom{k}{k-i} (k-i)^n \right] \right) \quad (4.3)$$

Gängige Desktop-PC verfügen im Jahr 2010 über 2 bis 8 Kerne, d.h. es besteht die Möglichkeit, bis zu 8 Prozesse sinnvoll parallel zu betreiben, ohne dass die Kommunikation der Prozesse über das Netzwerk geleitet werden muß. Die Anzahl der Blöcke in einer Strömungssimulation variiert zwischen 1 für die Simulation der Umströmung einer einzelnen Schaufel bis hin zu über 100 für komplexere Aufgabenstellungen, bei denen Radseitenräume, Spiralgehäuse, Bohrungen, Zulaufgeometrien oder Saugrohre aufgelöst werden. In Abbildung 4.1 ist das Wachstum der möglichen Kombinationen  $M$  für derzeit typische Anzahlen von Prozessen dargestellt.

Das schnelle Wachstum zeigt deutlich, dass die Auswertung aller möglichen Aufteilungen schnell einen Umfang annimmt, der durch den Zugewinn an Geschwindigkeit durch die optimale Parallelisierung nicht gerechtfertigt ist. Es ist sinnvoll, nur einen Teil der Lösungen auszuwerten und schnell eine möglichst gute Lösung zu erzeugen. Die erreichbare Rechenzeit einer Strömungssimulation ist nicht nur durch das Gleichgewicht von Knotenverteilung und Aufwand für den Datenaustausch bestimmt. Auf einem durchschnittlichen Rechenknoten laufen neben der eigentlichen Strömungssimulation auch andere betriebssystemeigene Prozesse, die die Rechenkerne unterschiedlich stark in Anspruch nehmen. Erfahrungsgemäß sind die Unterschiede zwischen einer guten und der perfekten Verteilung sehr gering, so dass der gesteigerte Aufwand, die perfekte Aufteilung zu finden, nicht gerechtfertigt ist.

## 4.1.2 Zielfunktion

Bei der Parallelisierung des Simulationsexperiments werden die einzelnen Teilgebiete, die sich durch die Blockstruktur ergeben, auf verschiedene Prozesse aufgeteilt. Diese Prozesse laufen auf verschiedenen CPUs. Die Teilgebiete werden repräsentiert durch eine Problematrix und deren Randbedingungen. Die Randbedingungen eines Teilgebietes hängen von den Ergebnissen anderer Teilgebiete ab, d.h. die Teilgebiete müssen untereinander fortwährend entsprechend ihrer Konnektivitäten Zwischenergebnisse austauschen. Dieser Datenaustausch ist die Synchronisierungsschranke, an der alle Prozesse warten müssen, bis alle Zwischenergebnisse vorliegen und ausgetauscht worden sind. Die Gesamtgeschwindigkeit der Berechnung ohne den Kommunikationsaufwand wird somit immer durch die Berechnungsgeschwindigkeit des langsamsten Teilgebiets bestimmt. Der Austausch der Zwischenergebnisse innerhalb eines Teilgebiets wird direkt über die problembeschreibende Matrix abgebildet und ist gegenüber dem Aufwand für den Austausch zwischen zwei Teilgebieten auf verschiedenen CPUs vernachlässigbar.



**Abbildung 4.1:** Anzahl der möglichen Kombinationen  $M$  für verschiedene Blockanzahl  $p$  und  $k \in [1..8]$  Prozesse

Die Gesamtdauer einer Iteration  $t_i$  kann aufgeteilt werden auf die Rechenzeit zur Ermittlung der Teillösung  $t_{r,i}$  und die Zeit für den Datenaustausch  $t_{k,i}$ :

$$t_i = t_{r,i} + t_{k,i} \quad (4.4)$$

Die Rechenzeit  $t_{r,i}$  für die Ermittlung eines Zwischenergebnisses ist abhängig von Größe und Struktur der problembeschreibenden Matrix und korreliert mit der Anzahl der Rechenzellen, die das Teilgebiet umfasst: Die geometrischen Rechenzellen werden direkt in Einträge der Lösungsmatrix umgesetzt. Um die Wartezeit von einzelnen Prozessen kurz zu halten, muß sichergestellt werden, dass möglichst alle Zwischenergebnisse zum gleichen Zeitpunkt vorliegen, d.h. die Prozesse müssen die Teilgebiete in gleicher Zeit lösen können.

Unter der Annahme, dass alle Prozesse auf CPUs mit gleicher Bearbeitungsgeschwindigkeit laufen, gilt die Forderung:

$$N_n = N_m, \quad (4.5)$$

d.h. alle Teilgebiete weisen möglichst die gleiche Anzahl von Rechenzellen auf.

Die Kommunikationszeit ist abhängig von der Kommunikationsgeschwindigkeit zwischen den Prozessen und dem Umfang der Kommunikation, d.h. der Anzahl der ausgetauschten Daten. Die Kommunikation wird von einem Masterprozess zentral verwaltet. Im ungünstigsten Fall können nur je zwei Prozesse zu einem Zeitpunkt miteinander kommunizieren. Somit ist die Obergrenze der Gesamtdauer des Austausch der Zwischenergebnisse die Summe der Dauer des Austausch der Teil-Zwischenergebnisse der einzelnen Blöcke.



Ziel der Verteilung der Blöcke auf die Prozesse ist es, die gleiche Anzahl Knoten je Prozess zu erhalten und dabei eine minimale Anzahl von Austauschbeziehungen der Blöcke über Prozessgrenzen hinweg zu erzeugen.

NS3D benutzt für den Datenaustausch zwischen den Prozessen die Funktion `MPI_sendrecv`. Dadurch wird das Senden und Empfangen der Teildaten in einem Funktionsaufruf gekapselt. MPI kann mehrere unabhängige Nachrichten zwischen den Prozessen parallel abwickeln. Die Maximaldauer der Kommunikation kann durch diese Parallelisierung teilweise deutlich unterschritten werden.

Um die Güte einer beliebigen Zuordnung der Blöcke zu den Prozessen zu beurteilen, sind zwei Kostenfunktionen notwendig: Eine, die das Knotengleichgewicht ausdrückt, und eine, die den mutmaßlichen Umfang der notwendigen Interprozesskommunikation bewertet. Das vorgestellte Problem ist allgemein bekannt als die Partitionierung eines vorgegebenen Graphen, bei dem sowohl Knoten-, wie auch Kantengewichte berücksichtigt werden. Bei der gegebenen Aufgabenstellung sind die Blöcke die Knotenpunkte, die Interfaces zwischen den Blöcken sind die Kanten des Graphen.

Die Gleichverteilung der Zellen auf die Blöcke kann nun abgebildet werden über die maximale Abweichung der Anzahl der Zellen eines Prozesses  $i$  von der theoretisch optimalen Anzahl je Prozess:

$$K = \max(N_i - \frac{N_{tot}}{p}) \quad (4.6)$$

Für den Kommunikationsaufwand wird der sogenannte  $k_{cut}$ -Wert herangezogen. Er summiert die Anzahl der Verbindungen zwischen unterschiedlichen Prozessen. In der vorliegenden Implementation wird er noch auf den maximalen  $k_{cut}$  Wert-normiert, der sich ergibt, wenn alle Knoten unterschiedlichen Prozessen zugeordnet werden.

Die Globale Zielfunktion wird über ein Blending der beiden Kostenfunktionen erzeugt. Die globale Kosten für eine Aufteilung ergeben sich zu:

$$K_{total} = \alpha \cdot K + (1 - \alpha) \cdot k_{cut} \rightarrow 0 \quad (4.7)$$

### 4.1.3 Heuristische Aufteilung

Eine Referenzimplementierung eines heuristischen Algorithmus zur Partitionierung von Graphen ist METIS. Er ist in zwei unterschiedlichen Varianten als ausführbares Programm erhältlich. Der zugrundeliegende Algorithmus ist erstmals von KARYPIS UND KUMAR [31] beschrieben. Der Algorithmus existiert in einer frei verfügbaren Referenzimplementierung, er kann als Kommandozeilenprogramm verwendet werden und kann darüber hinaus als C/C++-Bibliothek in eigene Programme eingebunden werden. Schwerpunkt bei METIS ist das schnelle, parallelisierte quasi-optimale Partitionieren sehr großer Graphen. METIS wird erfolgreich angewandt, um unstrukturierte Gitter zu partitionieren. Dabei sind die Einzelvolumina die Knoten, die Nachbarschaftsverhältnisse zwischen den Volumina die Kanten des Graphen. Dadurch ergeben sich Graphen, die um Größenordnungen mehr Elemente aufweisen als die betrachteten blockstrukturierten Probleme.

Ein weiterer Algorithmus zur Partitionierung von Graphen ist von COMELLAS UND SAPPENA [16] vorgestellt worden, der sog. ANTS-Algorithmus. Er basiert auf der Annahme, dass eine global quasioptimale Aufteilung aus lokal optimalen Lösungen aufgebaut werden kann. Dabei werden den Knotenpunkten des Graphen Optimierungsmarken zugeordnet, sogenannte „ants“ (Ameisen). Das Verfahren ist iterativ: während jeder Iteration werden die Optimierungsmarken entlang der Kanten des Graphen verschoben und die Farben der Knoten des Graphen verändert. Das Verschieben der Optimierungsmarke geschieht dabei mit der Wahrscheinlichkeit  $p_m$  in Richtung des benachbarten Knotens, der die lokale Kostenfunktion am schlechtesten erfüllt. Mit der Wahrscheinlichkeit  $1 - p_m$  wird die Optimierungsmarke auf einen beliebigen Nachbarpunkt bewegt.

Nach dem Bewegen der Optimierungsmarken wird die Farbe des Knotens festgelegt, auf dem die Marke zeigt: mit der Wahrscheinlichkeit  $p_c$  wird der Knoten dem Prozeß zugeordnet, mit dem die lokale Kostenfunktion minimiert wird, mit der Wahrscheinlichkeit  $1 - p_c$  einem zufällig gewählten Prozeß.

Der ursprüngliche Algorithmus fordert, dass nach jeder Iteration die Balance wiederhergestellt wird, wobei nicht beschrieben wird, wie die Punkte umzufärben sind. In der Implementierung für das vorliegende Problem wird nur sichergestellt, dass alle  $k$  Farben in der Partitionierung enthalten sind.

Ist die globale Kostenfunktion niedriger als in der vorangegangenen Iteration, so wird als Zwischenergebnis die Verteilung und das neue relative Optimum als Ergebnis der globalen Kostenfunktion gespeichert, ansonsten wird das vorangegangene lokale Optimum fortgeschrieben. Das Verfahren wird abgebrochen, wenn das globale Optimum oder die maximale Anzahl von Iterationen erreicht wurden. Die notwendige lokale Kostenfunktion ergibt sich aus der Anzahl der Nachbarblöcke, die nicht dem gleichen Prozeß zugeordnet sind.

Die vorgeschlagenen Werte für die verwendeten Konstanten sind:

$$p_c = p_m = 0.9 . \quad (4.8)$$

Durch das Verwenden von zufälligen Ereignissen soll verhindert werden, dass der Algorithmus in einem lokalen Minimum festläuft, ohne ein globales Minimum zu finden. Der Nachteil des Verfahrens ist, dass es nicht deterministisch ist, d.h. ein zweimaliges Ausführen des Algorithmus bei gleichen Startbedingungen kann zu unterschiedlichen Ergebnissen führen.

#### 4.1.4 Laufzeitverhalten praktischer Beispiele

Zur Beurteilung des Laufzeitverhaltens wird der Speedup der Parallelisierung betrachtet. Dieser ist definiert als Quotient der auf  $k$  Prozessen parallelisierten Rechnung zur nicht parallelisierten Laufzeit  $t_1$ .

$$S_k = \frac{t_k}{t_1} \quad (4.9)$$

Nr.	Typ	Anzahl Blöcke	Zellen	Größter Block
1	Einzelkanalrechnung	6	68 816	25 652
2	Einzelkanalrechnung, saugseitiger RSR	20	141 600	33 264
3	Einzelkanalrechnung, beide RSR, Entlastungsbohrung	39	308 598	86 400
4	Einzelkanalrechnung, beide RSR, Entlastungsbohrung, händisch aufgespaltene Blöcke	41	308 598	43 200

**Tabelle 4.1:** Parallelisierte Rechengitter

Die Effizienz der Parallelisierung  $E$  bemißt sich aus dem Quotienten des Speedups und der Anzahl Rechenkerne und ist somit ein relativer Speedup.

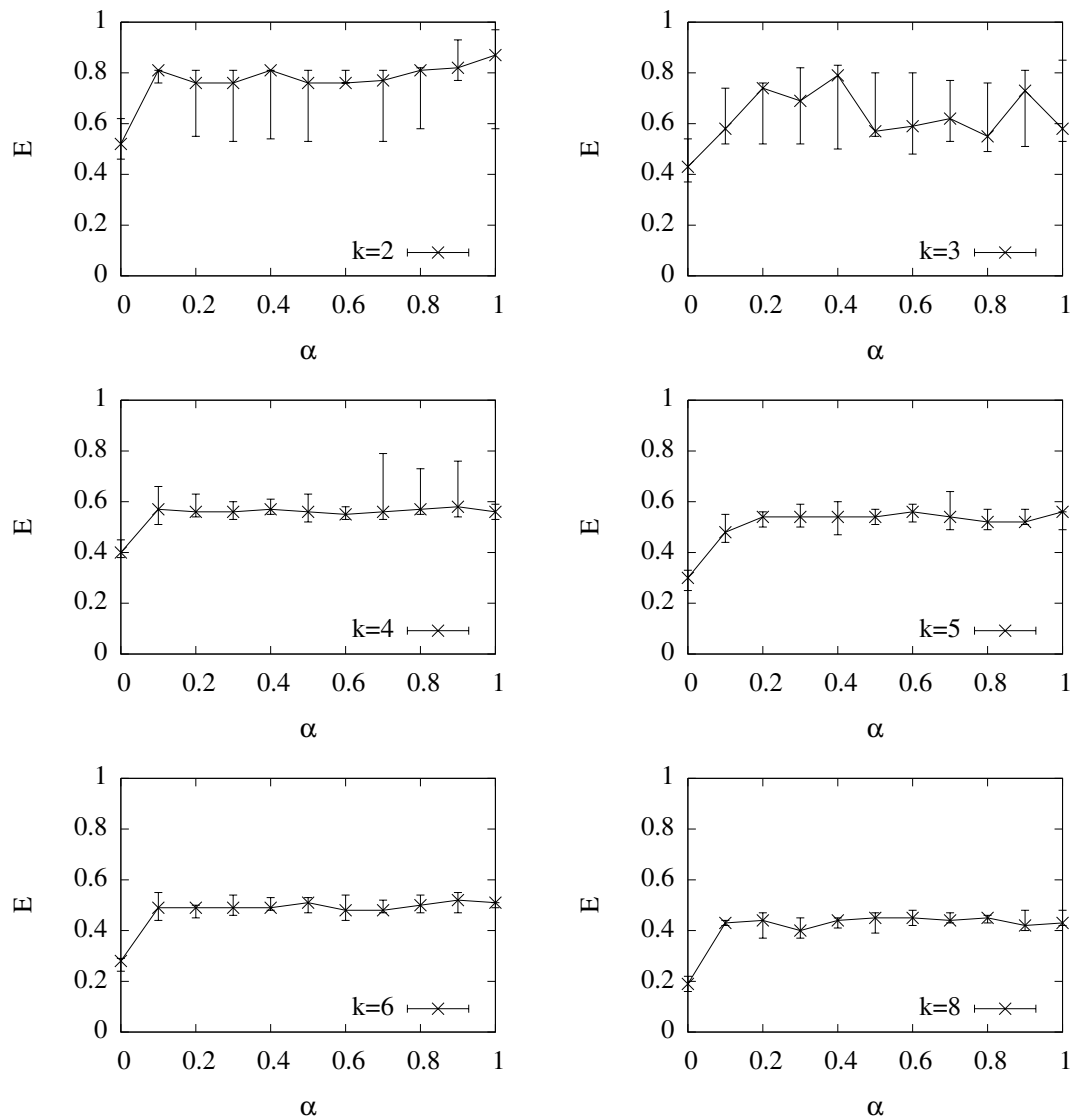
$$E = \frac{S_k}{k} \quad (4.10)$$

Im besten Fall ist mit einer Effizienz von 1 zu rechnen, dann wird die gesamte hinzugenommene Rechenleistung für die Simulation genutzt. Der Kommunikations- und Synchronisationsoverhead sorgt allerdings für eine Verringerung der Effizienz. Nach LE-PACH [36] steigt dieser Aufwand mit steigender Parallelisierung, so dass die Effizienz abnimmt.

Zur Beurteilung des Laufzeitverhaltens wurden 4 typische Gitter simuliert, s. Tabelle 4.1. Im Fall 1 wurde die Laufzeit bis zum Erreichen des Konvergenzkriteriums betrachtet, in den anderen Fällen wurde nach den ersten 200 Iterationen abgebrochen. Die Zeiten für die Festplattenausgabe wurden nicht abgezogen. Die Rechnungen wurden auf einem handelsüblichen PC durchgeführt, s. Tabelle 4.2. Die Blockverteilung wurde mit dem ANTS-Algorithmus vorgenommen, wobei der Blendingfaktor von 0 bis 1 in Schritten von 0.1 variiert wurde. Um Störeinflüsse des allgemeinen Rechnerbetriebs auszuschließen, wurde jede Rechnung fünfmal wiederholt, wobei der Median der Rechnungen in die Auswertung eingeflossen ist. Für die Beurteilung des Speedups wurde die jeweils schnellste Rechnung aus der Variation der Blendingfaktoren herangezogen.

Es hat sich gezeigt, dass die Laufzeiten für die genannten Testfälle auch bei gleicher Partitionierung in einem recht breiten Bereich schwanken. Selbst auf einem Rechner, auf dem lediglich das Betriebssystem mit seinen Hintergrundprozessen und der NS3D ausgeführt werden, sind Laufzeitschwankungen um den Median bei  $\alpha = const.$  größer oder gleich der Schwankung des Medians der Effizienz über  $\alpha$ , s. Abbildung 4.2. Lediglich für  $\alpha = 0$  ist ein deutlicher Einbruch zu verzeichnen. Teilweise konnte über ein Anhalten und Neustarten des MPICH2-Hintergrunddienstes die Rechenzeit verkürzt werden. Die Auswirkung der nicht beeinflussbaren Faktoren auf die Laufzeit ist größer als der mittlere Unterschied zwischen einer guten und sehr guten Aufteilung.

Bei einigen Beispielen konnte eine Effizienz der Parallelisierung größer als 1 erzielt werden, d.h. die Ausführungsgeschwindigkeit ist stärker gestiegen als die Zahl der ver-

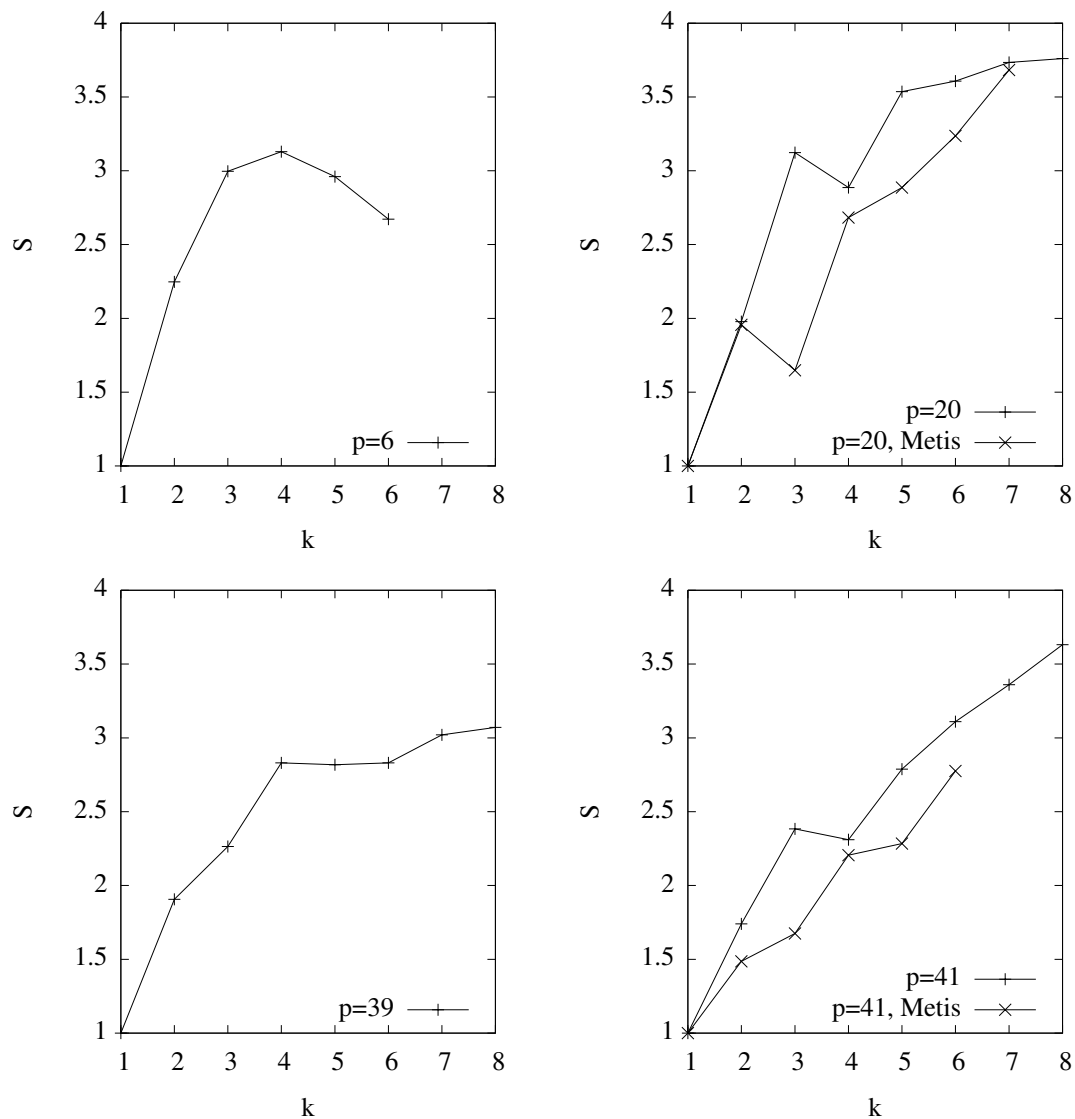


**Abbildung 4.2:** Effizienz für die Parallelisierung von 41 Blöcken auf  $k$  Prozesse in Abhängigkeit von  $\alpha$

Prozessor	Core i7-920
Anzahl Kerne	4+4 Hyperthreading
Speicher	12 GB
Betriebssystem	Microsoft Windows Server 2003
MPI-Implementation	MPICH2-1.0.8

**Tabelle 4.2:** Verwendete Hardware

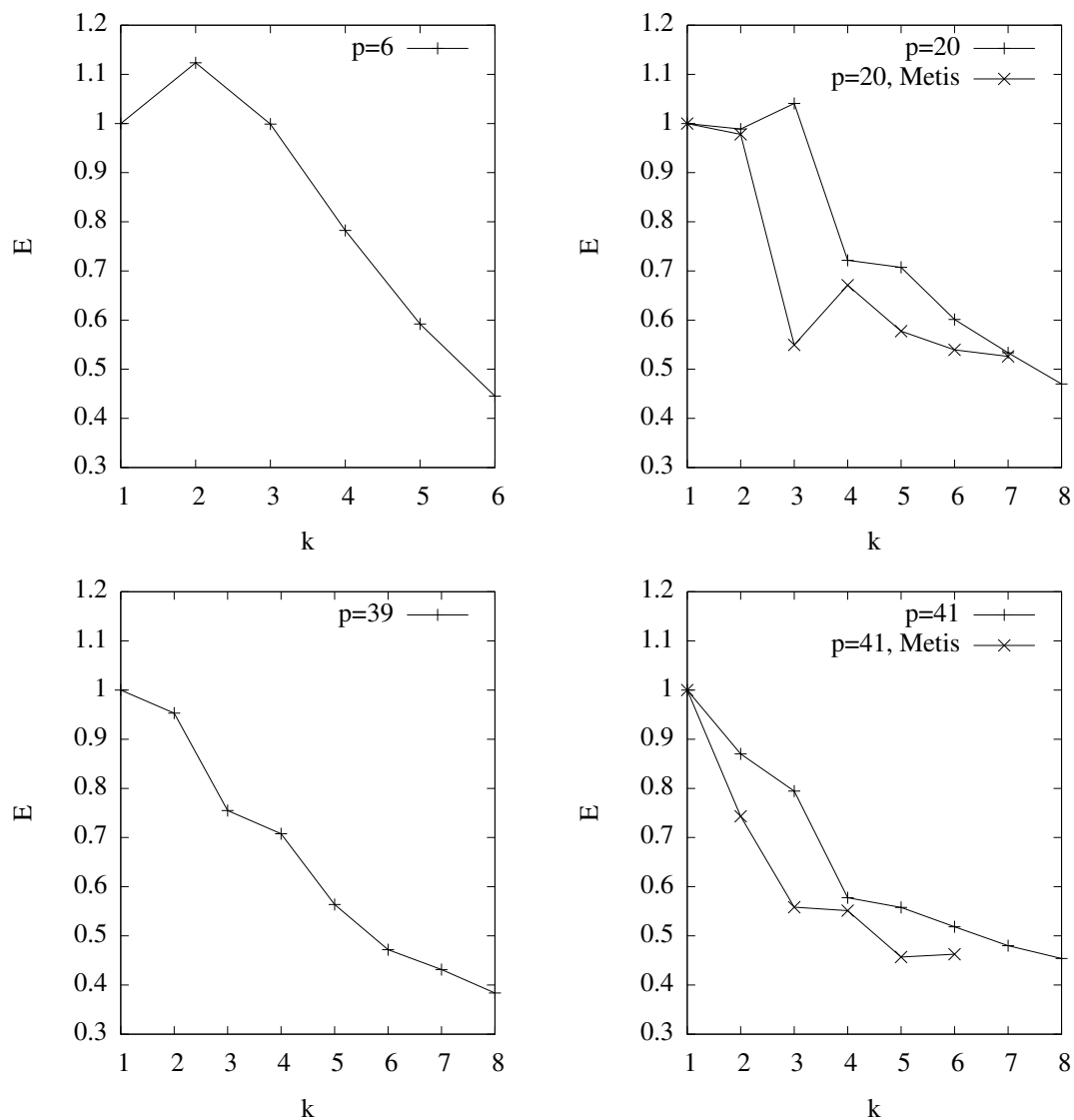
wendeten Rechenkerne. Dieser Effekt kann darauf zurückgeführt werden, dass die Teilprozesse kleiner sind als die Ursprungsprozesse und u.U. besser den schnellen Zwischenspeicher (Cache) der Rechenkerne ausnutzen können als der größere Einzelprozess. Ein Hinweis darauf ist, dass in beiden Fällen, in denen eine Effizienz größer 1 beobachtet werden kann, die Anzahl der Zellen pro Prozeß ca. 35.000 beträgt. Aus Abbildung 4.4 geht außerdem hervor, dass die Prozeßverteilung mit METIS i.d.R. eine schlechtere Effizienz hervorbringt als die ANTS-Prozeßverteilung. Dies ist ein Hinweis darauf, dass für das Laufzeitverhalten die Gleichverteilung der Prozeßgewichte höheres Gewicht hat als die Reduktion des  $k_{cut}$ -Wertes. Insgesamt ist METIS nicht bzw. nur bedingt als Lösung für das Problem „quasioptimale Aufteilung von  $N$  Blöcken auf  $k$  Prozesse“ geeignet. Sowohl bei 20 wie auch bei 41 Blöcken wurden für  $k = 8$  keine Aufteilungen gefunden.



**Abbildung 4.3:** Speedup der Shared-Memory-Parallelisierung mit ANTS und METIS für  $k \in [1, 8]$

Die beiden Fällen „39 Blöcke“ und „41 Blöcke“ sind identisch, nur dass beim zweiten der jeweils größte Block aufgeteilt wurde, bis der größte Block weniger als  $1/8$  der Zellen aufweist. Dadurch kann insgesamt ein etwas höherer Speedup erzielt werden, die Laufzeit für  $k = 1$  ist zudem etwas geringer für das Modell mit den 41 Blöcken. Während beim Fall „39 Blöcke“ ab  $k = 4$  kein nennenswerter Speedup erzielt werden kann verläuft der Speedup für den Fall „41 Blöcke“ fast linear bis zu  $k = 8$ . Ein automatisiertes Aufteilen des jeweils größten Blocks ist eine denkbare Option, um die Gesamtlaufzeit eines CFD-Experiments zu verkürzen. Allerdings muß dann im Postprocessing berücksichtigt werden, dass diese Blöcke nach der Simulation wieder zusammengesetzt werden.

Beim Fall „41 Blöcke“ ist ein Einbruch des Laufzeitverhaltens beim Schritt von 3 auf 4 Prozesse zu beobachten. In der Gesamtübersicht Abbildung 4.2 kann man für  $k = 4$  allerdings ablesen, dass durchaus eine Effizienz von  $E = 0.8$  erreicht worden



**Abbildung 4.4:** Effizienz der Shared-Memory-Parallelisierung mit ANTS und METIS für  $k \in [1, 8]$

---

ist, was einem Speedup von  $S_4 = 3.2$  entspricht. Diese Effizienz würde der mittleren Effizienz für  $k = 3$  entsprechen. Es bleibt offen, ob mit einer angepaßten Kostenfunktion Aufteilungen gefunden werden können, die diese Effizienz sicher erreichen, oder ob die nicht beeinflussbare Streuung der Werte durch das Betriebssystem hier einen größeren Einfluß hat.

## 4.2 Radialpumpen

Für die Simulation einer Radialpumpe werden die Gestalt ihrer Meridiankontur benötigt, die Geometrie der Schaufel sowie die Betriebspunktdaten  $Q$  und die erwartete Förderhöhe  $H$ . Um Größeneffekte auf die Simulation ausschließen zu können, werden Drehzahl, Durchmesser und Stoffparameter standardisiert, s. Tabelle 4.3. Aus  $Q$ ,  $H$  und der Drehzahl  $n$  wird die spezifische Drehzahl  $n_q$  ermittelt, anhand derer die Parameter kategorisiert werden. Ziel ist die Ermittlung eines gültigen Parametersatz in Abhängigkeit von  $n_q$ .

$D$	$n$	$\rho$	$\mu$	Re
340 mm	1000 $\text{min}^{-1}$	1000 $\text{kg}/\text{m}^3$	0.001 $\text{Pa} \cdot \text{s}$	$3.026 \cdot 10^6$

**Tabelle 4.3:** Geforderte Standardwerte des Pumpenexperiments

Das automatische CFD-Experiment für Radialpumpen wird beschränkt auf die Simulation eines Einzelkanals, wobei mindestens eine Laufschaufel vorhanden sein muß. Die Berücksichtigung von Nachleitbeschaufelung und Rückführstufen ist prinzipiell möglich. Somit können auch mehrstufige Pumpen simuliert werden. Radseitenräume oder Spiralgehäuse werden nicht berücksichtigt. Sie können von Hand nachträglich hinzugefügt werden.

Um die Einflußfaktoren für ein CFD-Experiment zu ermitteln, wird eine Pumpenbaureihe aus dem von von KRÄMER [32] erstellten Künstlichen Neuronalen Netz erzeugt. Die Betriebspunkt- und Solldaten, d.h. die spezifische Drehzahl, die Volumen- und Druckzahl im Optimalpunkt, die Schaufelzahl  $Z$  sowie die sich daraus ergebenden Daten für optimalen Volumenstrom und die zu erreichende Förderhöhe der einzelnen Modelle sind in Tabelle 4.4 aufgeführt. Der Schwerpunkt der Pumpenreihe wird in den Bereich für niedrige  $n_q$  gelegt, um das Verhalten des Programmsystems für diesen Bereich genauer beurteilen zu können. Alle vorliegenden Simulationen wurden mit dem Standard- $k$ - $\varepsilon$ -Modell und der logarithmischen Wandfunktion durchgeführt.

### 4.2.1 Partitionierung des Strömungsbereichs

Basis des Strömungsbereichs sind die Meridiankonturen, die dessen Gesamtabmessung bestimmen. Sie müssen entweder als geordnete Liste diskreter Punkte vorliegen oder als NURBS-Kurvenpaar im Format IGES. Die diskreten Punkte werden automatisch mit einer NURBS-Kurve mit dem Grad 3 approximiert, wobei die Anzahl der Kontrollpunkte auf  $2/3$  der Anzahl der Ausgangspunkte gesetzt wird. Erfahrungsgemäß werden die Punktlisten mit diesen Einstellungen ausreichend gut approximiert, die Tendenz, um die Ausgangspunkte zu schwingen, ist nur gering ausgeprägt. Diese Approximation deckt Knicke in der Meridiankontur nicht ab.

Der Einlaß der automatisierten Rechnung ist die lineare Verbindung der Anfangspunkte von Hub- und Shroudkontur. Entsprechend werden die Endpunkte der beiden Konturen linear verbunden, um die Auslaßfläche zu definieren. Für die Simulation einer einzelnen Schaufel sind nun ausreichend viele Begrenzungen vorhanden. Allerdings werden



$n_{q,\text{opt}} [min^{-1}]$	$\phi_{\text{opt}} [-]$	$\Psi_{t,\text{opt}} [-]$	$Q_{\text{opt}} [m^3/s]$	$H_{t,\text{soll}} [m]$	$Z [-]$
10	0.0051	1.180	0.0083	19.0277	6
15	0.0114	1.1653	0.0184	18.8404	7
20	0.0197	1.1473	0.0319	18.5277	7
25	0.0302	1.131	0.0488	18.2681	7
30	0.0411	1.0889	0.0664	17.5907	7
40	0.0637	0.9946	0.103	16.065	9
50	0.0857	0.8994	0.1384	14.5255	9
70	0.1239	0.7347	0.2003	11.8665	10
90	0.1594	0.6214	0.2576	10.0378	10

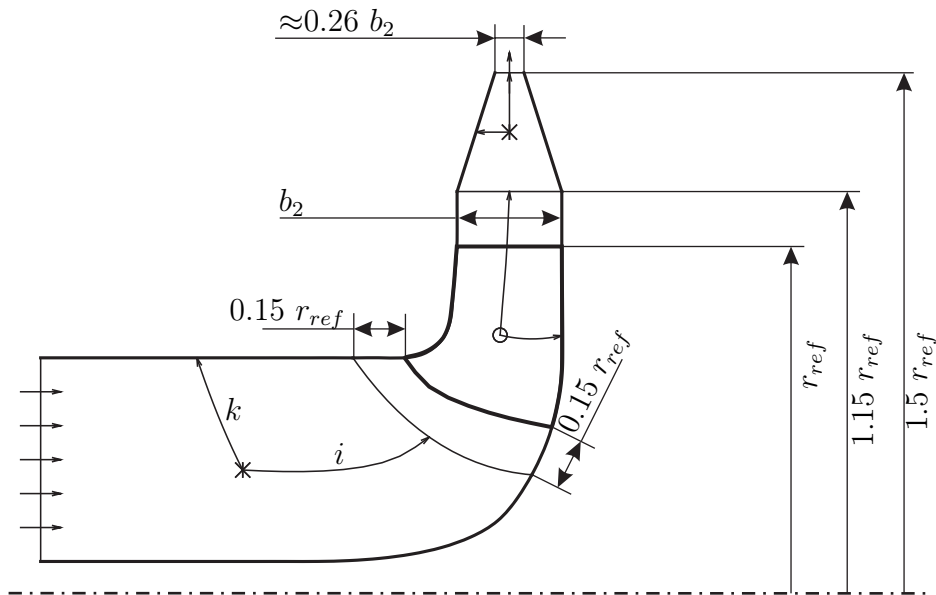
**Tabelle 4.4:** Auslegungsdaten der Pumpenreihe nach KRÄMER [32]

für die bessere Parallelisierbarkeit und die Vernetzung innere Grenzen zwischen den rein rotationssymmetrischen Teilen und den Schaufeln erzeugt. Diese inneren Grenzen werden wie folgt erzeugt:

1. Die Meridiankontur wird linear interpoliert, so dass eine mittlere Linie (Mean) erzeugt wird.
2. Die Vorder- und Hinterkante der Schaufeln wird mit der Hub-, Mean und Shroud-kontur verschnitten.
3. Auf den Konturen werden Kontrollpunkte erzeugt, und zwar je  $0.15 \cdot r_{\text{ref}}$  stromauf vor der Vorderkante und  $0.15 \cdot r_{\text{ref}}$  stromab hinter der Hinterkante.
4. Aus diesen Kontrollpunkten werden neue Berandungskurven als NURBS des Grades 2 erzeugt.
5. Falls sich die Bereiche zwischen zwei Schaufeln überlappen, werden die per Offset ermittelten Grenzen gelöscht und nur die mittlere verwendet.

Im Fall einer Radialpumpe mit einer einzelnen Schaufel ergeben sich nun drei Rechengebiete, ein rotationssymmetrisches einlaßseitig, das Schaufelgebiet und ein auslaßseitiges. Meridiankonturen mit konstanter Breite  $b_0$  im Auslaßbereich, die also in der Flächenentwicklung infolge der Radienerweiterung wachsende Querschnitte aufweisen, erzeugen je nach  $n_q$  eine starke Diffusorwirkung, die das Konvergenzverhalten negativ beeinflusst. Diese Geometrie führt zu Rückströmungen über die Auslaßfläche, was die Rechnung destabilisiert, da Informationen über das angesaugte  $k$  und  $\varepsilon$  nicht zur Verfügung stehen und aus dem vorliegenden Feld extrapoliert werden müssen. Eine künstliche Unterbindung der Rückströmung am Auslaß per Randbedingung ist zwar möglich, stabilisiert die Simulation aber nicht in dem Maße wie eine Geometriekorrektur.

Das auslaßseitige Rechengebiet kann nachträglich verändert werden, ohne die Meridiankontur zu beeinflussen. Ein Verjüngung der Meridiankontur auf ca.  $1/4$  der Auslaßfläche



**Abbildung 4.5:** Aufteilung des Rechengebiets einer Radialpumpe

an der Laufradhinterkante hat sich als gut stabilisierend erwiesen. Insgesamt ergibt sich für die Radialpumpe eine Geometrie im Meridianschnitt wie in Abbildung 4.5.

Als Randbedingungen werden am Einlaß die Geschwindigkeit normal zur Einlaßfläche sowie initiale Werte für  $k$  und  $\varepsilon$  vorgegeben. Am Auslaß wird ein statischer Druck  $p_s = 0$  vorgegeben. Dies ist die gängige Kombination für die Simulation inkompressibler Strömungen. Im Fall der Pumpe ist die Förderhöhe bei vorgegebenem Volumenstrom das direkte Ergebnis der Simulation.

## 4.2.2 Vernetzungsparameter

### Hub-Shroud-Richtung

Der dimensionslose Abstand  $y^+$  des wandnächsten Knotens bestimmt über die Wandfunktion die Schubspannung auf das Fluid und somit die Verluste infolge der Wandreibung, s. Abschnitt 2.1.3. Der  $y^+$ -Wert ist ein Ergebnis der Strömungslösung und muß herangezogen werden, um die Gültigkeit des Wandreibungsmodells zu prüfen. Die  $y^+$ -Werte müssen in einem Bereich zwischen 30 und 300 liegen, damit das Modell gültig ist, das der Wandfunktion zugrunde liegt. Die Anforderungen an diese  $y^+$ -Werte müssen nun übersetzt werden in Anforderungen an die Netzgenerierung.

Die Netzgenerierung in Hub-Shroud-Richtung geschieht wie folgt:

1. Festlegen der Berandungslinien in der Meridianebene, je vier pro Rechengebiet
2. Festlegen der Punkteverteilung auf den Berandungslinien
3. Interpolation der Randpunkte in die Meridianfläche
4. Rotieren des ebenen zweidimensionalen Netzes um die Maschinenachse zum Erzeugen des dreidimensionalen Netzes

Die Berandungslinien liegen als parametrische Kurven vor, so dass beliebige Punkteverteilungen auf der normierten Bogenlänge erzeugt werden können. Die benötigten Parameterwerte  $s$  werden über eine Abbildungsfunktion  $s(t)$  aus einer Gleichverteilung ermittelt. Die Abbildungsfunktion lautet:

$$s(t) = b_1 t + (3 - 4b_1 + b_2)t^2 + (-2 + 5b_1 - 3b_2)t^3 + 2(b_2 - b_1)t^4, \quad (4.11)$$

$$b_1 \in [0; 1], b_2 \in [0; 1].$$

Die Verteilungsfunktion ist ein Polynom, das durch die Punkte  $(0, 0)$  und  $(1, 1)$  geht, die Parameter  $b_1$  und  $b_2$  sind die Steigungen des Polynoms in diesen Punkten. Werden sie zwischen 0 und 1 gewählt, so bleibt das Polynom streng monoton. Die Möglichkeit, Punkte zu einer Seite hin zu verdichten, ist bei dem vorgegebenen Polynom begrenzt. Das von KRONSCHNABL [34] vorgestellte Béziervverfahren wird nicht benutzt, da es für den Schaufelbereich nicht implementiert ist. Um die Anzahl der zu ermittelnden Parameter zu begrenzen, wird Symmetrie der Verteilung gefordert:

$$b_1 \stackrel{!}{=} b_2 = b. \quad (4.12)$$

Neben dem Verdichtungsfaktor ist die Anzahl der Knoten in Hub-Shroudrichtung insgesamt ausschlaggebend für den Wandabstand. Die wandnächste Zelle für ein Netz mit  $n_k$  Knoten in Hub-Shroudrichtung wird auf der Berandungskurve mit dem Parameterwert

$$t_1 = \frac{1}{n_k - 1}$$

ermittelt, somit ist der Wandabstand abhängig von den Parametern  $n_k$  und  $b$ , s. Abbildung 4.6.

Die Wandfunktion ist gültig für den Bereich  $y^+ \in [30, 300]$ , so dass es für den tatsächlichen Wandabstand ebenfalls einen Gültigkeitsbereich gibt. Die  $y^+$ -Werte werden klassiert betrachtet, es wird eine Untergrenze für den Wandabstand erzeugt, in der der Schwerpunkt der  $y^+$ -Werte in der Klasse 30-100 liegt sowie eine Obergrenze, in der der Schwerpunkt in der Klasse 100-300 liegt. Die endgültige  $y^+$ -Verteilung für den groben und feinen Bereich sind in Abbildung 4.7 aufgeführt. Es ist zu beachten, dass Pumpen mit niedrigem  $n_q$  bei feiner Auflösung fast immer Netzzellen im Bereich von  $y^+ < 30$  aufweisen. Pumpen mit hohem  $n_q$  haben bei grober Auflösung fast immer einige Zellen im Bereich für  $y^+ > 300$ .

Die ermittelten Ergebnisse für  $s_{grob}(t_1)$  und  $s_{fein}(t_1)$  wurden gemittelt und dann mit einer Exponentialfunktion angenähert, s. Abbildung 4.8. Die Regressionsfunktion für den Parameterwert für die wandnächste Zelle lautet:

$$s_{soll}(t_1) = 0.5474 \cdot n_q^{-0.9233}. \quad (4.13)$$

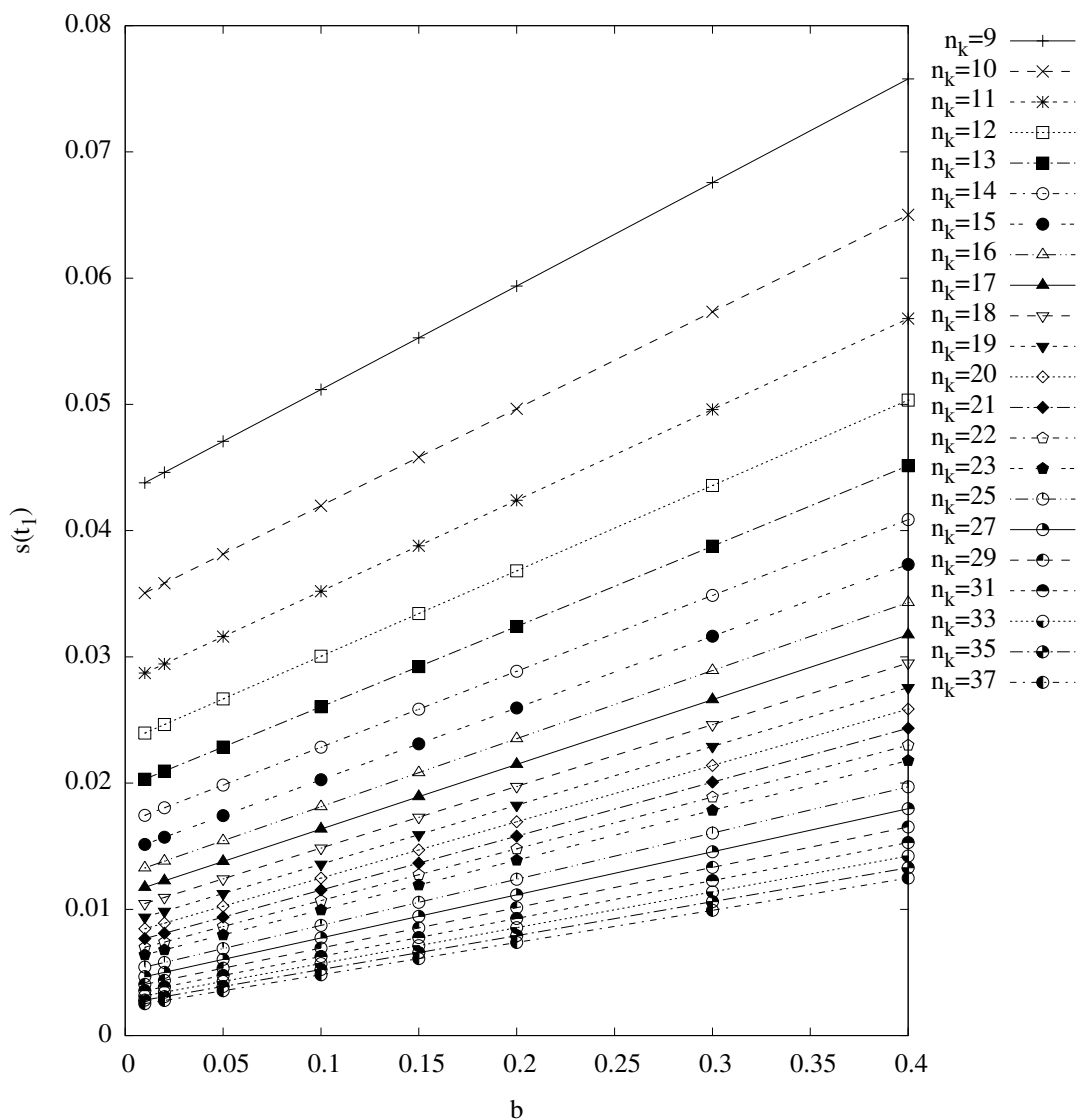
Aus (4.13) kann für ein gegebenes  $n_q$  mit (4.11) ein gültiges  $b$  ermittelt werden. Aus Abbildung 4.6 kann leicht abgelesen werden, dass für einige  $s(t_1)$  die Anzahl der Netzebenen  $n_k$  nicht frei wählbar ist, sondern für die gegebene Verdichtungsfunktion und dem geforderten Wandabstand eine minimale Anzahl von Strombahnen benötigt wird, damit das  $y^+$ -Kriterium nicht verletzt wird. Um die Wahl von  $n_k$  zu erleichtern, kann der Verdichtungsfaktor  $b$  begrenzt werden:

$$b \stackrel{!}{\geq} 0.01. \quad (4.14)$$

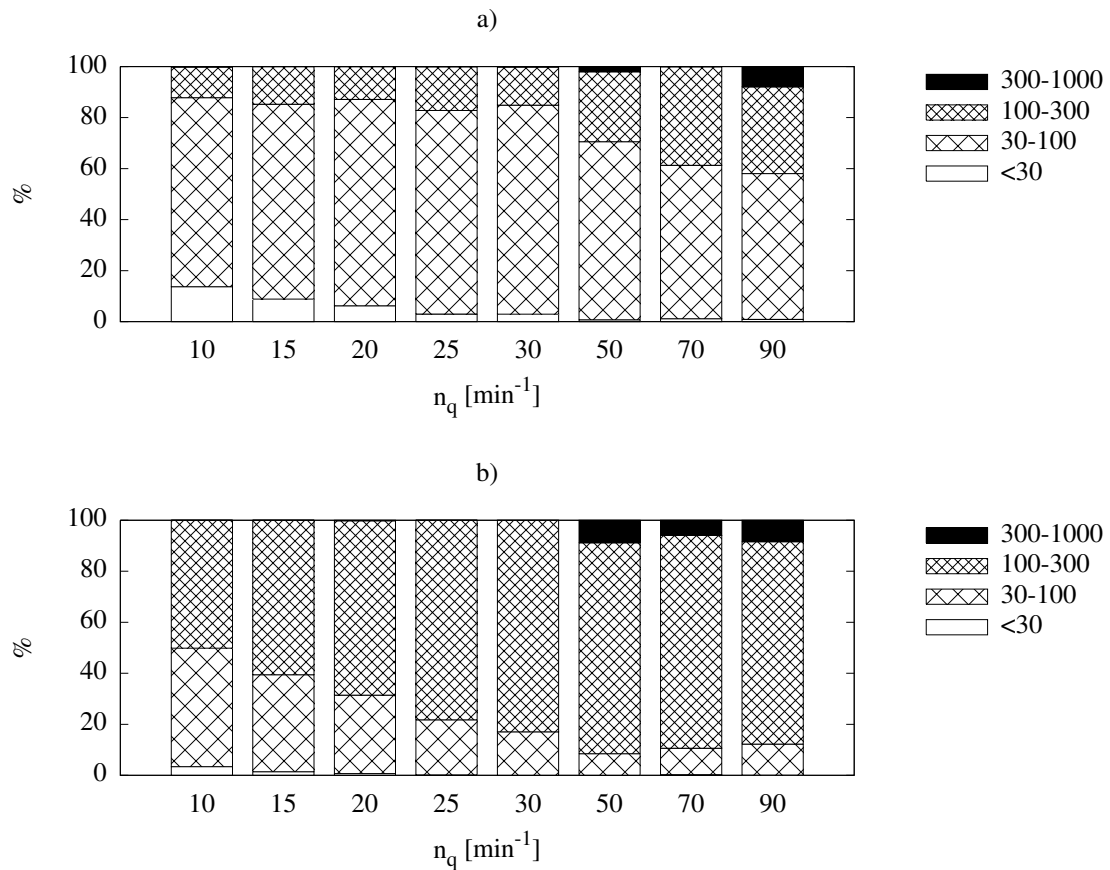
Ein weiteres Kriterium für die Wahl von  $n_k$  kann das Expansionsverhältnis der wandnächsten Zelle zu ihrer Nachbarzelle in Richtung der Kernströmung sein. Dieses Verhältnis kann direkt aus der Verteilungsfunktion abgeleitet werden:

$$\frac{s(t_2) - s(t_1)}{s(t_1)} \stackrel{!}{\simeq} x, \quad (4.15)$$

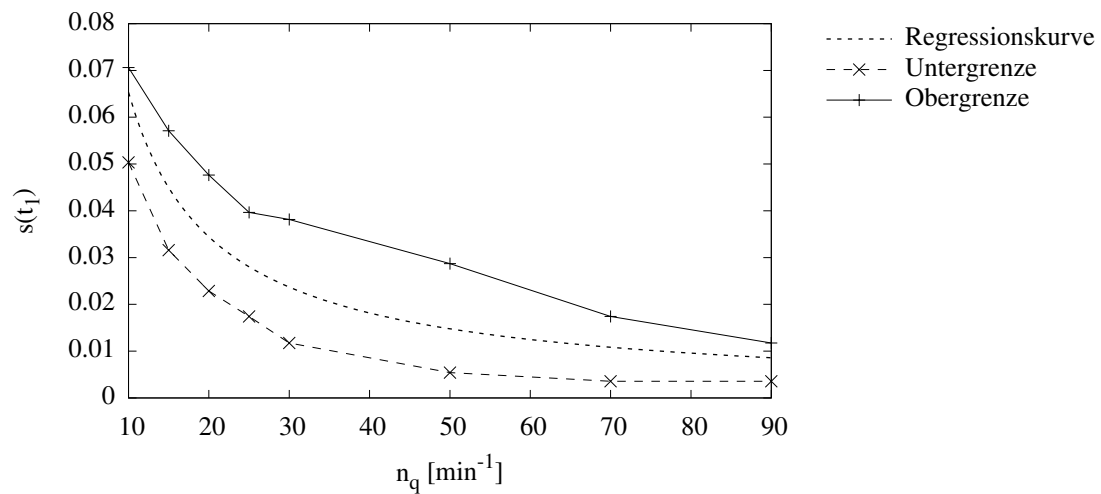
wobei die zweite Zellschicht die  $x$ -fache Dicke der ersten Zellschicht aufweist. Sinnvolle Werte für  $x$  bewegen sich im Bereich  $x \in [1, 2]$ . Die Parameter, die sich aus diesen Überlegungen ergeben sind in Abbildung 4.9 aufgeführt.



**Abbildung 4.6:** Parameterwerte für die wandnächste Zelle in Abhängigkeit von Verdichtungsfaktor  $b$  und der Anzahl Netzebenen in Hub-Shroudrichtung  $n_k$



**Abbildung 4.7:** Klassenverteilung der  $y^+$ -Werte mit Schwerpunkt  
 a)  $y^+ \in [30; 100]$  und  
 b)  $y^+ \in [100; 300]$



**Abbildung 4.8:** Ermittelter Gültigkeitsbereich des Abstandes der wandnächsten Zelle  $s(t_1)$  über  $n_q$

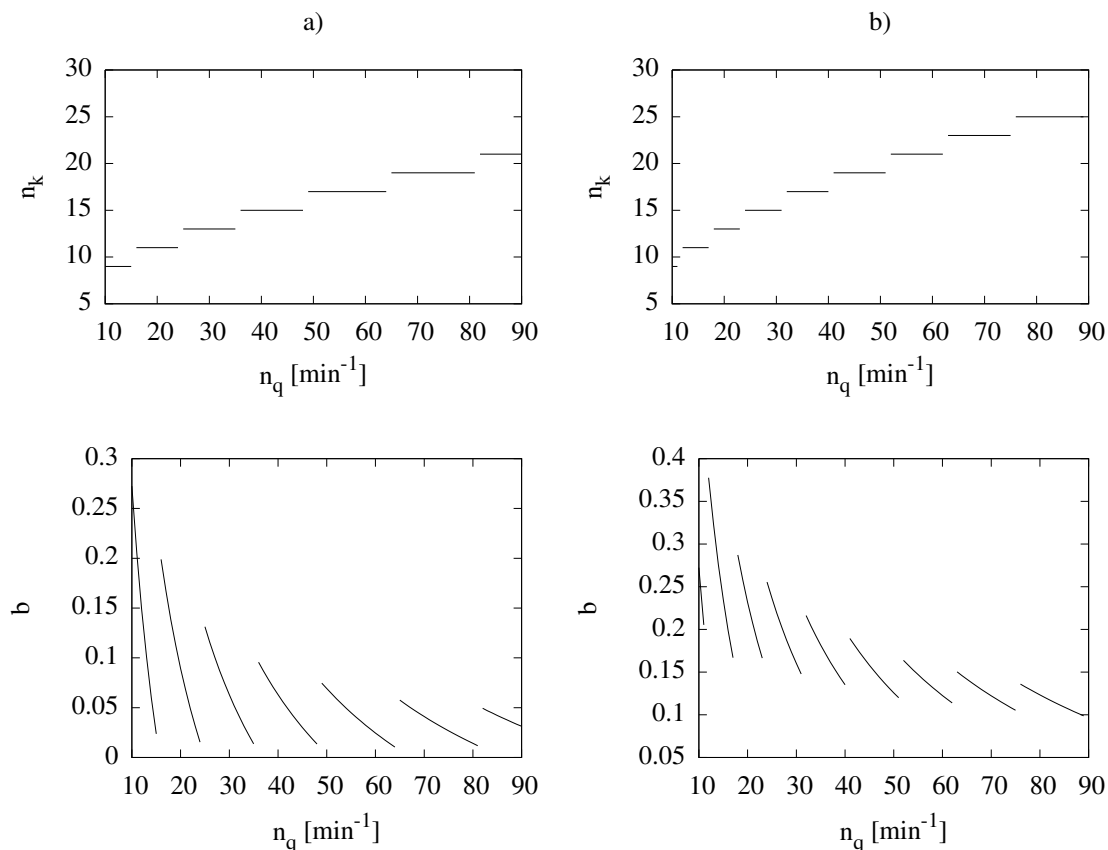


Abbildung 4.9: Sollwerte für  $n_k$  und  $b$ , a)  $b_{min} \geq 0.01$ , b)  $x \simeq 2$

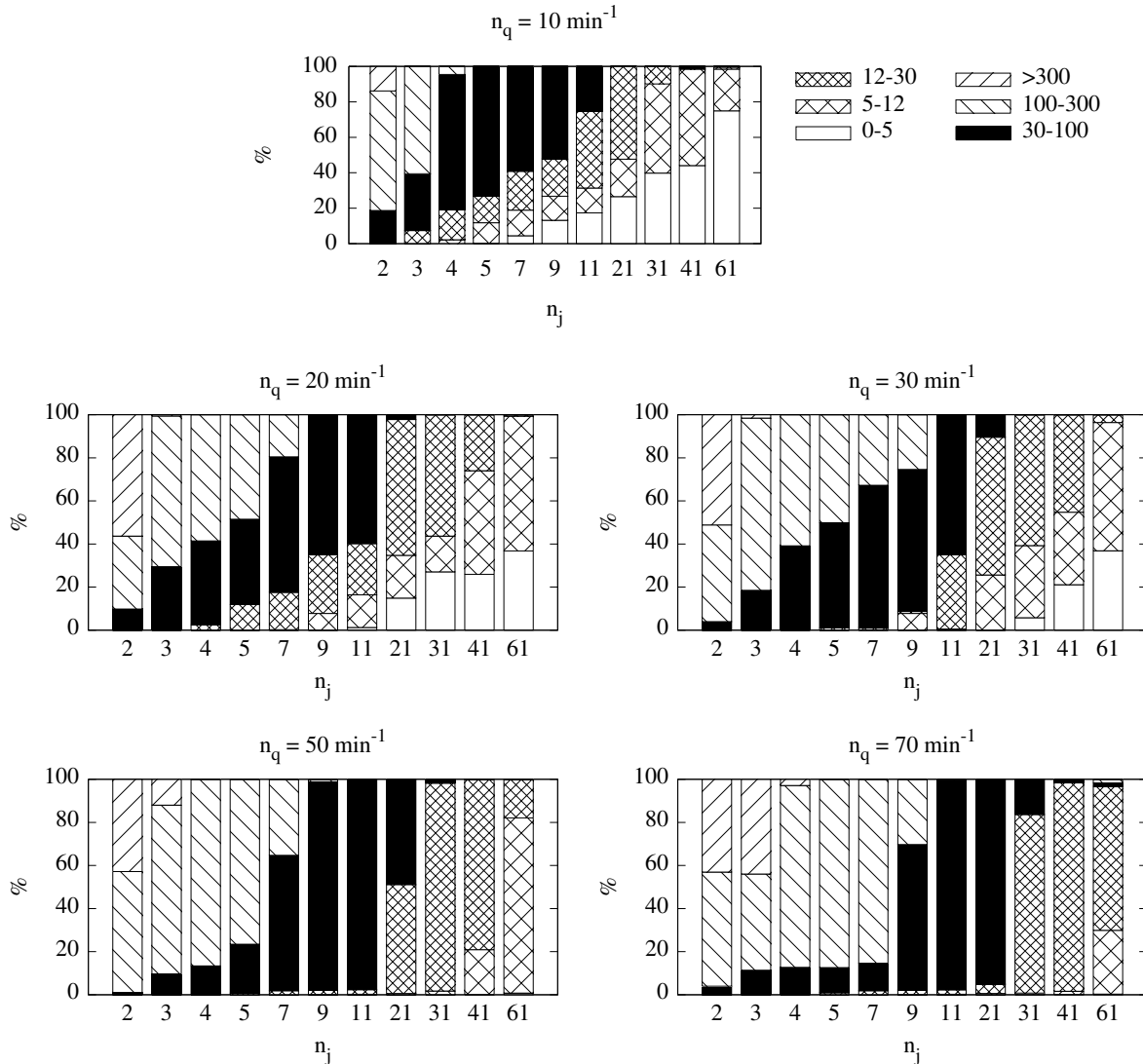
### Schaufelbereich

Für die Vernetzung des Schaufelbereichs wird der automatisierte C-Netzgenerator für Radialpumpen verwendet, s. Kap. 2.6.4. Die Standardeinstellung für die Dicke des C-Netzes beträgt 2% des Referenzradius, d.h. in allen vorliegenden Fällen  $3.4\text{mm}$ . Die Knoten im C-Netz werden in der Richtung normal zur Schaufel gleich verteilt, d.h. es wird keine Verzerrungsfunktion zur Schaufel hin benutzt. Um die  $y^+$ -Werte auf der Schaufel zu beurteilen, wird nur die Anzahl der Zellschichten  $n_j$  variiert. Betrachtet werden die Radialpumpen  $n_q = 10 \text{ min}^{-1}$ ,  $n_q = 20 \text{ min}^{-1}$ ,  $n_q = 30 \text{ min}^{-1}$ ,  $n_q = 50 \text{ min}^{-1}$  und  $n_q = 70 \text{ min}^{-1}$ , der Schwerpunkt liegt also im Bereich für niedrige  $n_q$ .

Analog zur Betrachtung der Vernetzung eines einzelnen Profils bei KRONSCHNABL [34] wird für die Schaufeloberfläche ein mittlerer dimensionsloser Netzabstand  $\overline{y^+}$  ermittelt, der die Netzauflösung charakterisiert. KRONSCHNABL verwendet für  $\overline{y^+}$  eine ungewichtete Mittelung, in der vorliegenden Arbeit wird  $\overline{y^+}$  flächengemittelt ermittelt. Es wird ein Bereich für  $\overline{y^+}$  erwartet, in dem die integralen Werte stabil sind und nicht oder nur in geringem Maße auf Änderungen der Netzauflösung reagieren.

Die Übersicht in Abbildung 4.10 zeigt, dass es für niedrige  $n_q$  keinen Bereich gibt, in dem das  $y^+$ -Kriterium nach Abschnitt 2.1.3 voll erfüllt werden kann. Die Werte der Relativgeschwindigkeit an der Schaufeloberfläche variieren im Bereich von  $w = 0$  im

Staupunkt bis maximal zur Größenordnung der Umfangsgeschwindigkeit  $u_{ref}$  an der Austrittskante. Anders als bei der Meridiankontur durchlaufen die Geschwindigkeiten immer einen Bereich mit sehr geringen  $y^+$ -Werten. Für höhere  $n_q$  existieren breitere Bereiche, in denen die  $y^+$ -Kriterien erfüllt werden können.



**Abbildung 4.10:** Verteilung der  $y^+$ -Klassen für Pumpen verschiedener  $n_q$  und unterschiedlicher Knotenanzahl  $n_j$  normal zur Schaufeloberfläche

In Abbildung 4.11 kann ein stabiler Bereich für  $\overline{y^+} \in [20; 100]$  beobachtet werden, unabhängig vom  $n_q$  der Pumpe. Lokal wird das  $y^+$ -Kriterium verletzt. Das bedeutet, dass die Werte für  $\Psi_t$  und  $\Psi_{t,th}$  nur in geringem Umfang variieren. Die beiden Werte bewegen sich parallel, so dass der Quotient  $\eta$  eine nennenswerte Abhängigkeit hinsichtlich  $\overline{y^+}$  aufweist. KRONSCHNABL [34] ermittelt für die Profilmströmung und das LCL-Turbulenzmodell einen stabilen Bereich von  $10 \leq \overline{y^+} \leq 55$ . dass dieser Wert geringer ist, erscheint plausibel, werden doch in beiden Fällen der jeweilige Nasenbereich deutlich feiner aufgelöst. Dort auftretende geringe  $y^+$ -Werte tragen bei gleicher Gewichtung der Einzelwerte stärker zum Mittelwert bei als bei der Flächenmittelung.

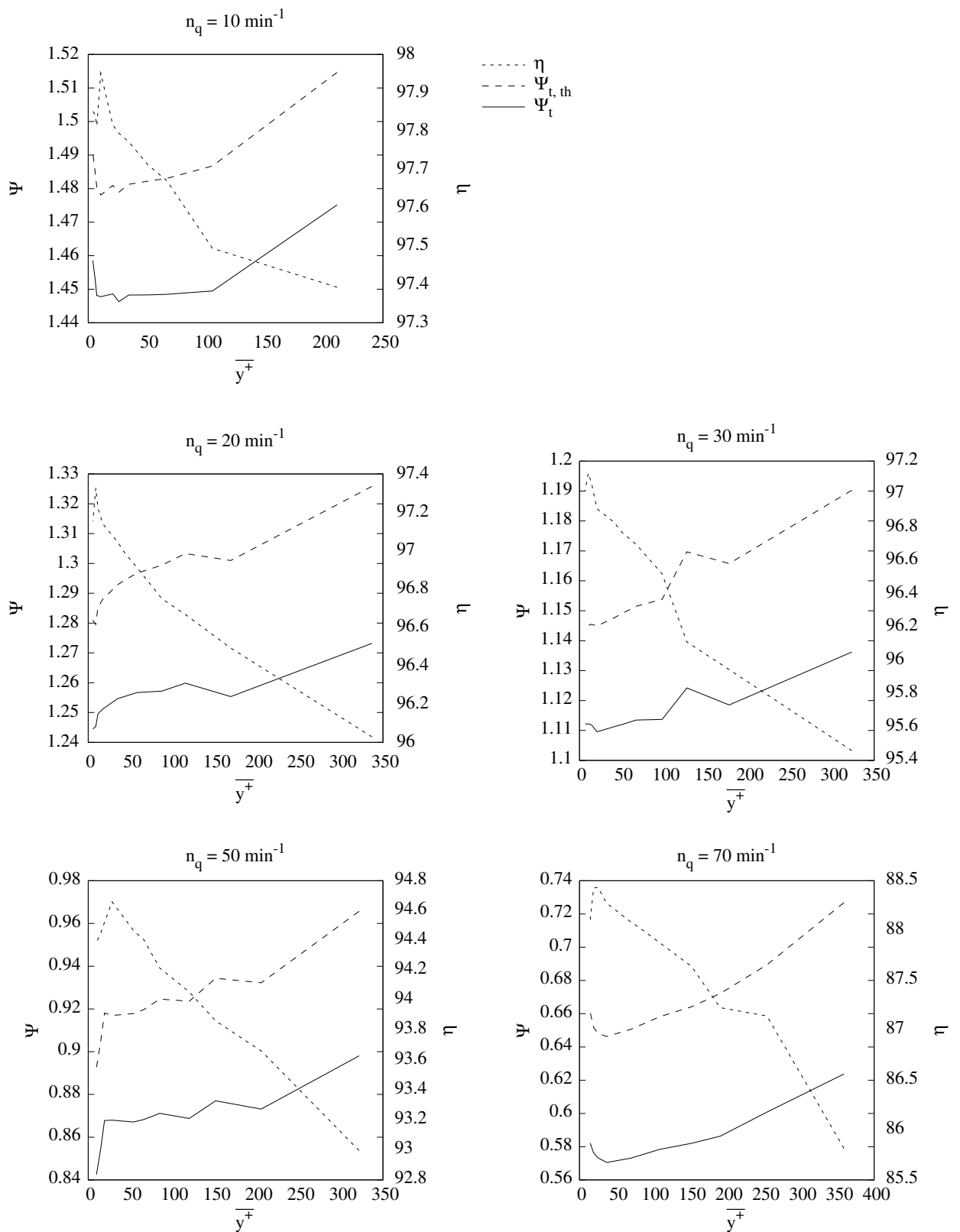


Abbildung 4.11: Integrale Größen bei unterschiedlicher Schaufelvernetzung als Funktion von  $\bar{y}^+$



Hinsichtlich des automatisierten CFD-Experiments und der zuverlässigen Ermittlung der Charakteristik von  $\Psi_t$ ,  $\Psi_{t,th}$  und  $\eta$  über verschiedene Volumenströme ist es sinnvoll, für  $\phi = \phi_{opt}$  einen mittleren Wert von  $\overline{y^+} \approx 60$  zu wählen. Somit können für Betriebspunkte  $\phi \gg \phi_{opt}$  noch ausreichend niedrige Werte für  $\overline{y^+}$  erreicht werden und für  $\phi \ll \phi_{opt}$  kann sichergestellt werden, dass  $\overline{y^+} > 20$  eingehalten wird.

Die betrachteten Netze variieren einzig in der Anzahl der Knoten im C-Netz normal auf die Schaufel. Die Knoten sind gleichverteilt. Mit der vorgegebenen Verzerrungsfunktion  $s(t)$  für die Knoten normal zu Schaufel kann der Wert für  $y_p$  genau eingestellt werden:

$$s(t) = \frac{e^{\lambda t} - 1}{e^\lambda - 1}, \lambda \neq 0. \quad (4.16)$$

Mit dem Zusammenhang zwischen  $y_p$  und  $y^+$  aus Abbildung 4.12 kann nun ein  $y_p$ , soll aus dem geforderten  $\overline{y^+} = 60$  ermittelt werden. Mit der bekannten Dicke des C-Netzes lassen sich analog zur Hub-Shroudrichtung optimale Vernetzungsparameter  $\lambda_{opt}(n_q, n_j)$  ermitteln, s. Abbildung 4.13.

Es bleibt noch zu überprüfen, ob der extrapolierte Bereich für  $n_q = 90 \text{ min}^{-1}$  zutrifft, und wie der Verlauf von  $\lambda_{opt}$  für den Bereich  $10 \text{ min}^{-1} \leq n_q \leq 20 \text{ min}^{-1}$  ist. Ein linearisiertes Abknicken ist unwahrscheinlich. Allerdings ist die Abweichung des erzielten  $\overline{y^+}$  im Bereich  $10 \text{ min}^{-1} \leq n_q \leq 20 \text{ min}^{-1}$  nur gering, so dass auf eine eingehendere Betrachtung verzichtet werden kann.

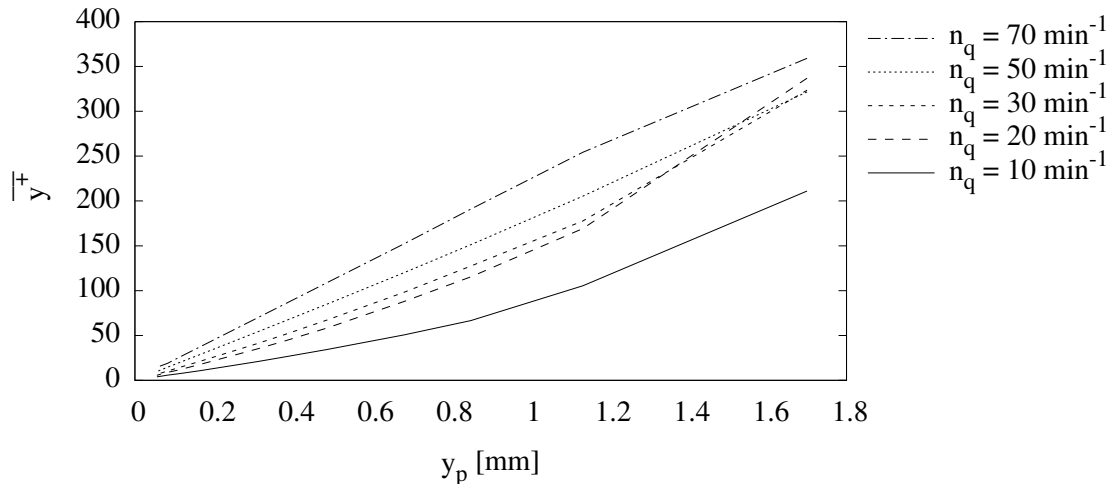


Abbildung 4.12:  $\overline{y^+}$  als Funktion von  $y_p$  für verschiedene  $n_q$

### 4.2.3 Konfiguration des Solvers

Für den Solver stehen einige Parameter bereit, die sein Verhalten hinsichtlich der Stabilität und der Geschwindigkeit beeinflussen. Im wesentlichen handelt es sich dabei um Unterrelaxationsfaktoren, die die Berücksichtigung des neu ermittelten Lösungsvektors von Iteration zu Iteration dämpfen. Die Solverparameter, die den höchsten Einfluß auf Stabilität und Geschwindigkeit haben, sind die Unterrelaxation der Impulsgleichung

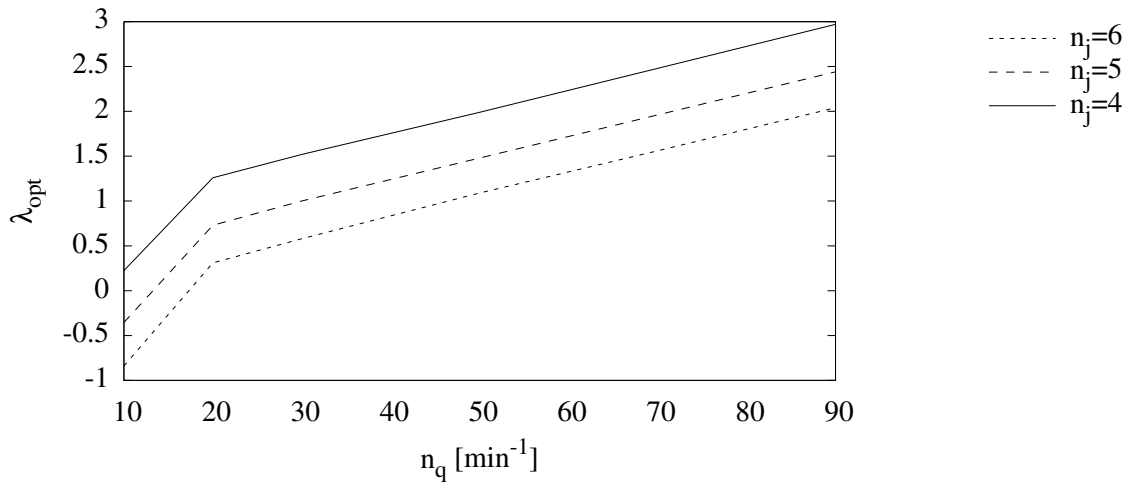


Abbildung 4.13:  $\lambda_{opt}(n_q)$  als Funktion von  $n_j$

$\alpha_{uvw}$ , die Unterrelaxation der Druckkorrekturgleichung  $\alpha_p$  sowie die Unterrelaxation des Updates des Lösungsvektors zwischen den einzelnen Impulsgleichungen  $\alpha_{upd}$ .

FERZIGER UND PERIC [21], auf deren Arbeit der verwendete CFD-Code NS3D beruht, geben folgendes Stabilitätskriterium an:

$$\alpha_{uvw} + \alpha_p \leq 1. \quad (4.17)$$

Die Erfahrungen mit verschiedenen Simulationen haben gezeigt, dass dieses Stabilitätskriterium nicht ausreicht, um in jedem Fall zu einer Lösung zu gelangen. Um eine bessere Übersicht über das Konvergenzverhalten zu erlangen, wurden Variationsrechnungen für die gesamte Pumpenreihe durchgeführt. Dabei wurden identische Netze, die auf den Forderungen aus Abschnitt 4.2.2 basieren, mit veränderten Unterrelaxationsfaktoren simuliert. Die Variation der Parameter wurde in folgenden Grenzen durchgeführt:

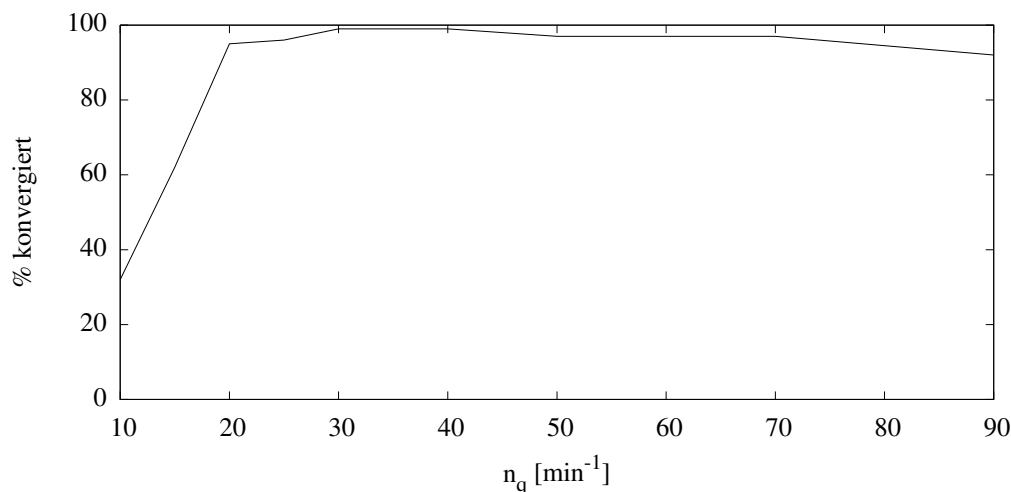
$$\begin{aligned} 0.1 &\leq \alpha_{uvw} \leq 0.6, \\ 0.1 &\leq \alpha_p \leq 0.6, \\ 0.3 &\leq \alpha_{upd} \leq 0.8. \end{aligned} \quad (4.18)$$

Es wurden für jedes  $n_q$  300 Wiederholungen der Rechnungen durchgeführt. Die Parameter wurden zufällig belegt, und bei allen Simulationen wurde mit dem Upwind-Discretization-Scheme gerechnet. Als Kriterium für das Konvergenzverhalten wird die Anzahl der Iterationen  $n_I$  betrachtet, die benötigt wurde, um einen Residuenabfall auf  $r_{max} \leq 10^{-6}$  zu erreichen. Wurde dieses Residuenniveau nicht innerhalb von 10.000 Iterationen erreicht, wurde die Simulation abgebrochen und als nicht konvergiert gezählt. Das Laufzeitverhalten konnte nicht berücksichtigt werden, da im Multi-Userbetrieb für die Vielzahl der Rechnungen keine konstanten Rahmenbedingungen gewährleistet werden konnten. Hilfsweise muß davon ausgegangen werden, dass eine kleine Anzahl von Iterationen  $n_I$  eine kurze Laufzeit bedeutet.

Neben  $n_I$  wurden für das Laufrad die integralen Werte  $\Psi_t$ ,  $\Psi_{t,th}$  und  $\eta_{La}$  ermittelt, um die Auswirkung der Änderungen der Unterrelaxationsfaktoren auf die Simulationsergebnisse beurteilen zu können.

### Konvergenzverhalten

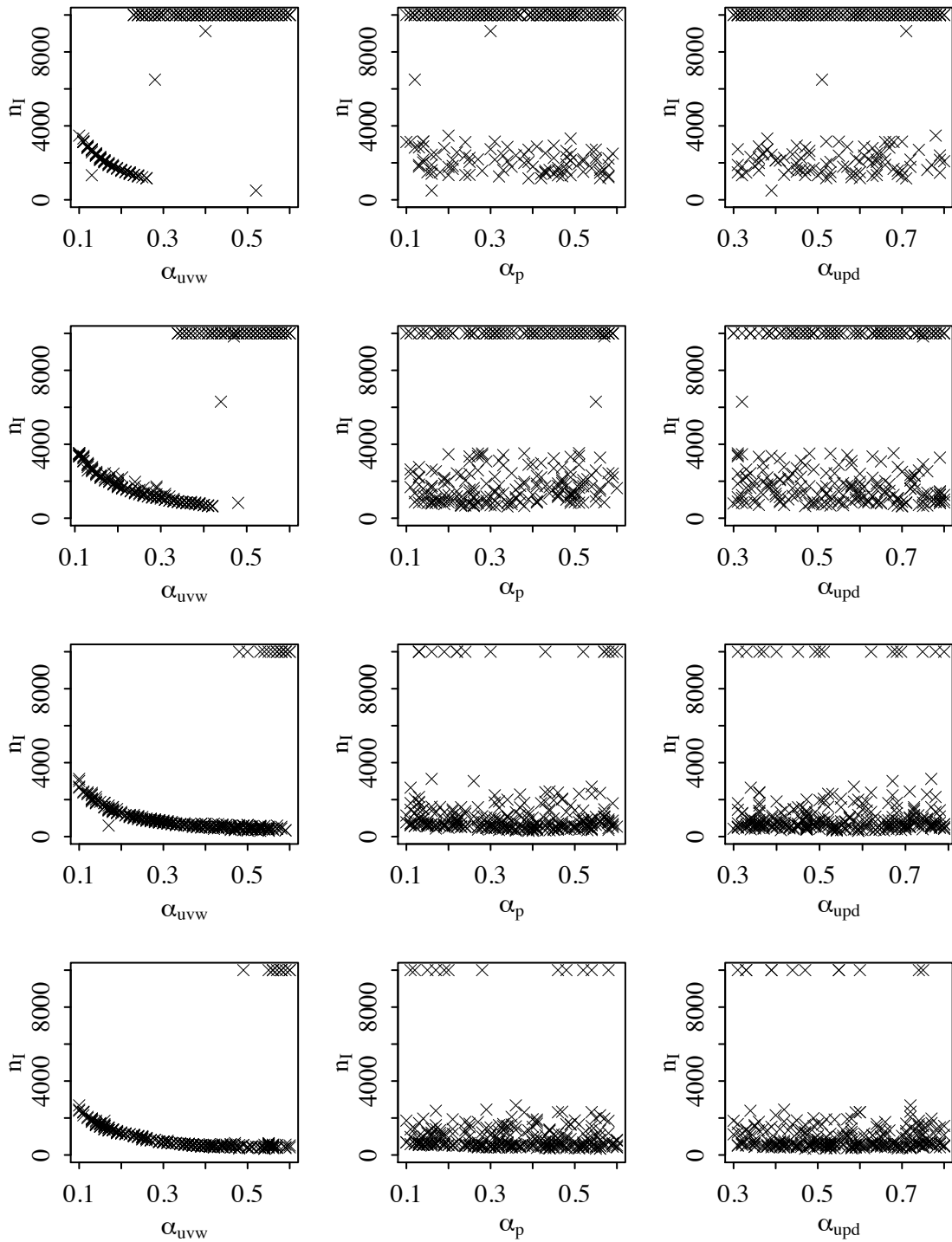
Alle Simulationen blieben stabil, d.h. es trat nie Divergenz auf. Im Regelfall konvergierten je  $n_q$  mehr als 90% aller Simulationen, unabhängig von den gesetzten Werten. Für den Bereich geringer Schnellläufigkeit,  $n_q \leq 20 \text{ min}^{-1}$ , ist zu beobachten, dass das Konvergenzverhalten sich stark verschlechtert. Für  $n_q = 10 \text{ min}^{-1}$  sind lediglich 32% der Simulationen konvergiert, s. Abbildung 4.14.



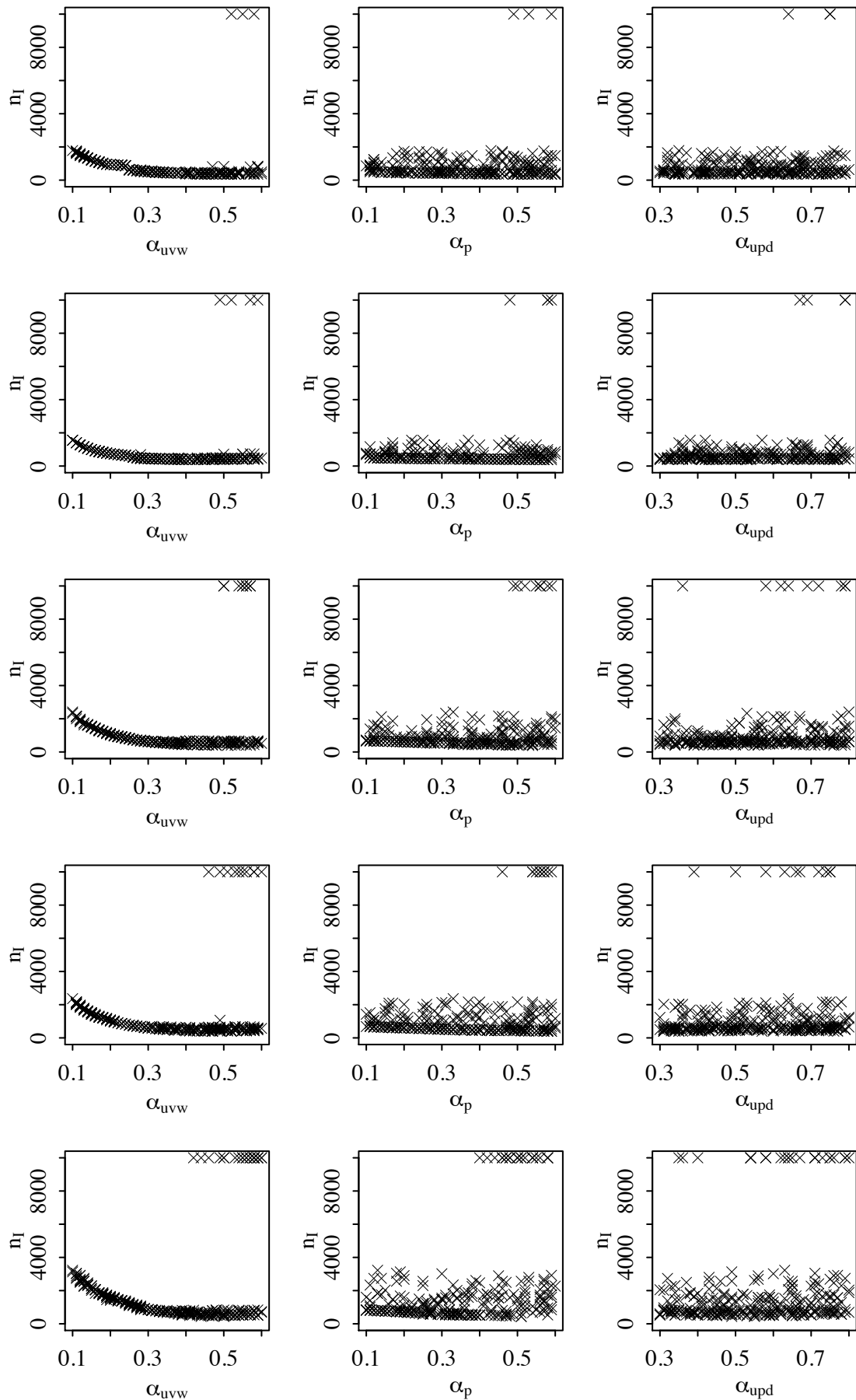
**Abbildung 4.14:** Konvergenzverhalten der Parametervariationen

In Abbildung 4.15 und 4.16 kann das Iterationsverhalten  $n_I$  in Abhängigkeit der Variation der  $\alpha$ -Werte abgelesen werden. Es ist deutlich erkennbar, dass der Haupteinflussfaktor für die Anzahl der Iterationen die Unterrelaxation der Impulsgleichungen  $\alpha_{uvw}$  ist. Der Einfluß von  $\alpha_p$  ist dagegen deutlich geringer. Wider Erwarten ist kein Zusammenhang zwischen  $\alpha_{upd}$  und der Geschwindigkeit oder der Stabilität zu sehen.

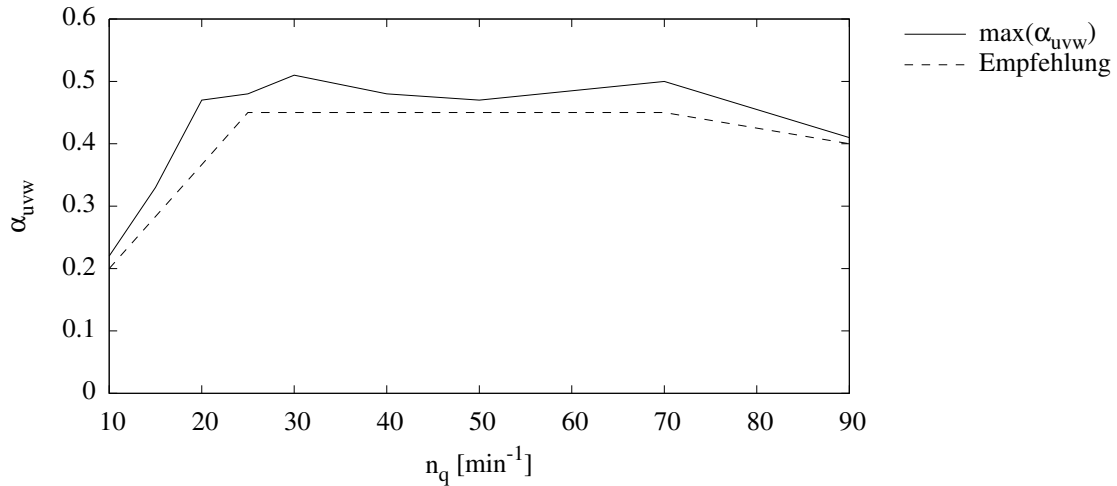
Eine weitere Analyse der Daten bestätigt das Stabilitätskriterium aus (4.17). Es existieren nur vereinzelte Simulationen, bei denen trotz Verletzung dieses Kriteriums eine konvergierte Lösung erreicht werden konnte, diese weisen erwartungsgemäß eine sehr geringe Zahl von Iterationen auf. Aus den Graphen für  $n_I(\alpha_{uvw})$  läßt sich in Abhängigkeit von  $n_q$  ein problembezogenes Stabilitätskriterium ableiten, s. Abbildung 4.17.



**Abbildung 4.15:** Anzahl der Iterationen  $n_I$  in Abhängigkeit von  $\alpha_{uvw}$ ,  $\alpha_p$  und  $\alpha_{upd}$ , für  $n_q = 10 \text{ min}^{-1}$ ,  $15 \text{ min}^{-1}$ ,  $20 \text{ min}^{-1}$  und  $25 \text{ min}^{-1}$  (von oben nach unten)



**Abbildung 4.16:** Anzahl der Iterationen  $n_I$  in Abhängigkeit von  $\alpha_{uvw}$ ,  $\alpha_p$  und  $\alpha_{upd}$ , für  $n_q = 30 \text{ min}^{-1}$ ,  $40 \text{ min}^{-1}$ ,  $50 \text{ min}^{-1}$ ,  $70 \text{ min}^{-1}$  und  $90 \text{ min}^{-1}$  (von oben nach unten)



**Abbildung 4.17:** Stabilitätsgrenze des Konvergenzverhaltens und Empfehlung für  $\alpha_{uvw}$

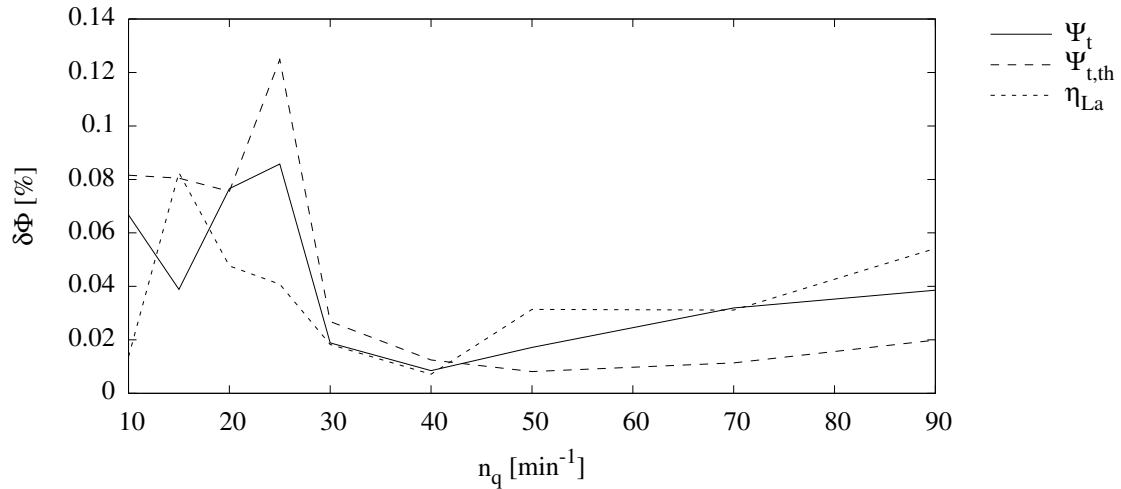
### Integrale Werte

Bei den integralen Werten ist eine sinnvolle Maßzahl für die Zuverlässigkeit der Simulation die mittlere relative Abweichung eines Einzelergebnis vom Median der integralen Werte aller konvergierten Lösungen:

$$\delta\Phi = \frac{1}{n \cdot \phi} \sum_{i=1}^n |\Phi_i - \bar{\Phi}| \cdot 100\% . \quad (4.19)$$

Dabei ist  $\bar{\Phi}$  der Median des Werts  $\Phi$ , nicht der Mittelwert.

Die mittlere relative Abweichung sagt aus, wie weit im Mittel aller Simulationen der konkrete Wert vom Median aller Simulationen entfernt liegt. Je kleiner dieser Wert ist, umso geringer ist die Abhängigkeit des Simulationsergebnisses von den gewählten Parametern für  $\alpha_{uvw}$ ,  $\alpha_p$  und  $\alpha_{upd}$ . Abbildung 4.18 zeigt die ermittelten relativen Abweichungen in Abhängigkeit von  $n_q$ . Sie bewegen sich deutlich unterhalb 0.1%. Die ermittelten Ergebnisse sind als unabhängig von den verwendeten Parametern zu betrachten, eine Simulation, die ein Residuum von  $10^{-6}$  aufweist, zeigt ein stabiles Ergebnis.



**Abbildung 4.18:** Stabilität des Konvergenzverhaltens am Beispiel der integralen Werte über  $n_q$

### 4.3 Francis-Turbine

Im vorangehenden Abschnitt wurde die Konfiguration des CFD-Experiments für Radialpumpen sehr kleinschrittig untersucht. Im Folgenden sollen vor allem durch Vergleiche mit den für Pumpen ermittelten Zusammenhänge die Ergebnisse auf Francis-Turbinen übertragen werden. Die Betrachtung wird deshalb auf fünf Francis-Turbinen beschränkt, s. Tabelle 4.5. Durchmesser, Drehzahl und die Stoffwerte werden ebenfalls wie bei der Radialpumpe gewählt, s. Tabelle 4.3. Die vom KNN ermittelten Turbinengeometrien wurden nicht angepaßt, d.h. das Laufrad wurde ohne weitere Nachbehandlung verwendet, auch wenn es nicht die geforderte Umlenkung erbrachte. Die Leitschaufelöffnung wurde angepaßt, so dass der Drall nach dem Leitrad dem Auslegungspunkt entspricht. Alle Rechnungen wurden mit Diskretisierungsverfahren höherer Ordnung (MINMOD) durchgeführt.

$n_{q,opt} [min^{-1}]$	$\phi_{opt} [-]$	$\Psi_{t,opt} [-]$	$Q_{opt} [m^3/s]$	$H_{t, soll} [m]$	$Z [-]$
30	0.079	1.676	0.127	27.08	17
40	0.126	1.568	0.204	25.33	15
60	0.229	1.354	0.370	21.87	13
90	0.340	1.029	0.549	16.62	11
120	0.342	0.705	0.553	11.39	9

**Tabelle 4.5:** Auslegungsdaten der Turbinenreihe

### 4.3.1 Partitionierung des Strömungsbereichs

Die Konfiguration des Strömungsbereichs der Francis-Turbine geschieht analog zu Radialpumpe, nur dass mehrere Schaufeln – Traverse, Leitapparat, Laufrad – berücksichtigt werden müssen. Für die Aufteilung des Strömungsgebiets in einzelne Rechendomains werden ebenfalls an Hub-, Mean- und Shroud-Kontur stromauf und stromab Hilfspunkte erzeugt, durch die die Begrenzungs-Splines der Rechendomains gelegt werden. Zusätzlich werden auf diesen Kurven zwischen zwei Schaufeln Splinekurven als Interfacegrenzen erzeugt. Ist die per Offset ermittelte Grenze einer Schaufel weiter entfernt als dieser Interfacespline, wird sie gelöscht, nur der Interfacespline bleibt erhalten. Somit können zwischen zwei Schaufeln, die ausreichend weit voneinander entfernt sind, bis zu zwei zusätzliche rotationssymmetrische Netzblöcke eingefügt werden. Ein typisches Ergebnis der automatischen Aufteilung ist in Abbildung 4.19 dargestellt.

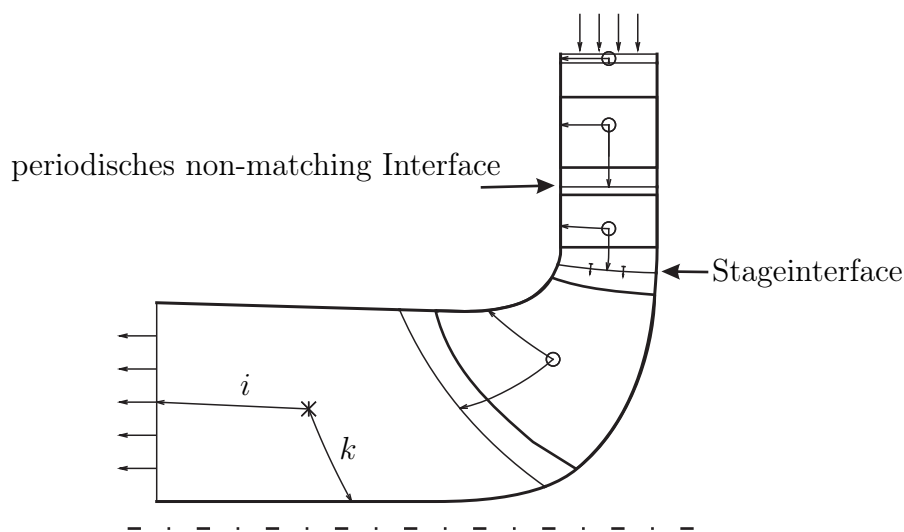


Abbildung 4.19: Aufteilung des Rechengebiets einer Francisturbine

### Randbedingungen

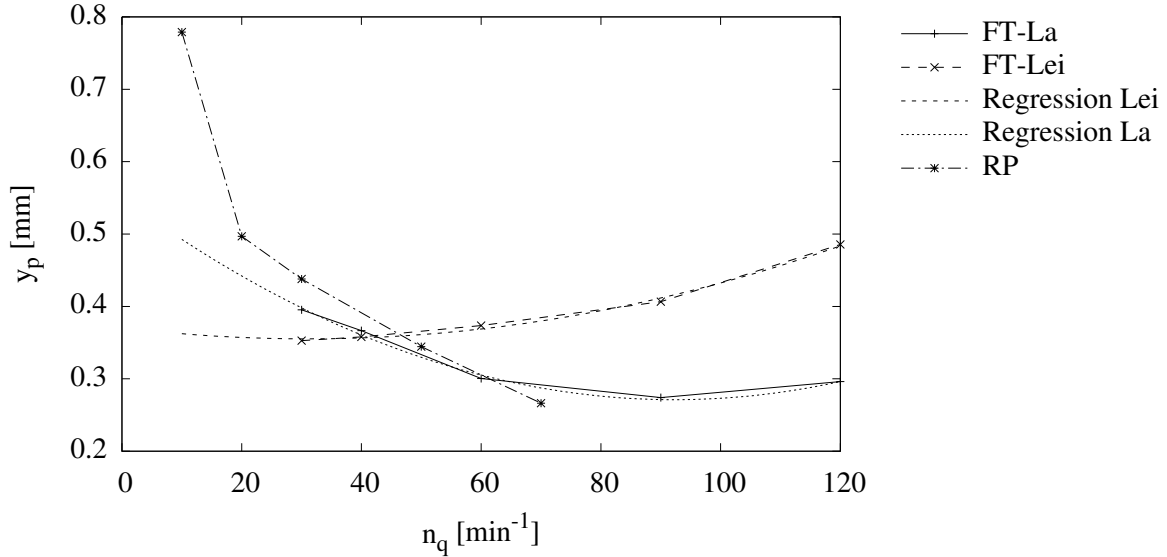
Als Randbedingungen werden am Einlaß die Geschwindigkeit in Zylinderkoordinaten und initiale Werte für  $k$  und  $\varepsilon$  vorgegeben. Am Auslaß wird ein statischer Druck  $p_s = 0$  vorgegeben. Anders als bei der Pumpe ist mit erfolgter Simulation das ermittelte Ergebnis nicht notwendigerweise das richtige, je nach Stellung des Leitapparats variieren Fallhöhe und Umlenkung. Typischerweise muß die Öffnung der Leitschaufeln angepaßt werden, so dass der Drall  $\Psi_{rcu}$  vor der Laufschaufel den Auslegungsdaten entspricht. Unter dieser Vorgabe können Fallhöhe, Umlenkung und Wirkungsgrad analysiert werden.

### 4.3.2 Vernetzungsparameter

Es konnte sowohl von KRONSCHNABL [34] wie auch im vorangegangenen Abschnitt 4.2.2 nachgewiesen werden, dass für bestimmte  $\bar{y}^+$ -Intervalle die ermittelten integralen



Werte für die Laufschaufel stabil bleiben, so dass diese Forderung hinsichtlich  $\overline{y^+}$  an die Vernetzung der Beschaufelung einer Francis-Turbine übernommen werden kann. Es ist allerdings zu beachten, dass der Zusammenhang zwischen  $y_p$  und  $\overline{y^+}$  im Laufrad wie in Abbildung 4.12 für die Francisturbine nicht gelten kann.



**Abbildung 4.20:** Verlauf des wandnächsten  $y_p$  für  $\overline{y^+} \approx 60$  über  $n_q$

Da die Anzahl der Schaufeln in der Francis-Turbine deutlich höher ist als in einer Radialpumpe mit gleichem  $n_q$ , soll zumindest im Laufrad die Dicke des O-Netzes geringer gewählt werden als bei der Pumpe. Aus dem  $y_{p,opt}$  aus Abbildung 4.20 kann nun ein  $\lambda = f(n_q, n_j, y_{p,max})$  ermittelt werden, so dass hier kein einheitlicher Graph für  $\lambda_{opt}$  abgedruckt werden kann.

Da die Strömungsgeschwindigkeiten im Leitschaufelbereich aufgrund der höheren Radien deutlich anders sind als im Laufschaufelbereich ergeben sich dementsprechend zwei unterschiedliche Verläufe für  $y_{p,opt}$ . Die ermittelten Werte für  $y_{p,opt}$  können für das Laufrad wie folgt angenähert werden:

$$y_{p,opt,La} = 3.27 \cdot 10^{-8} n_q^2 - 6.0361 \cdot 10^{-6} n_q + 5.4972 \cdot 10^{-4}. \quad (4.20)$$

Im Leitschaufelbereich ist hingegen folgende Regression gültig:

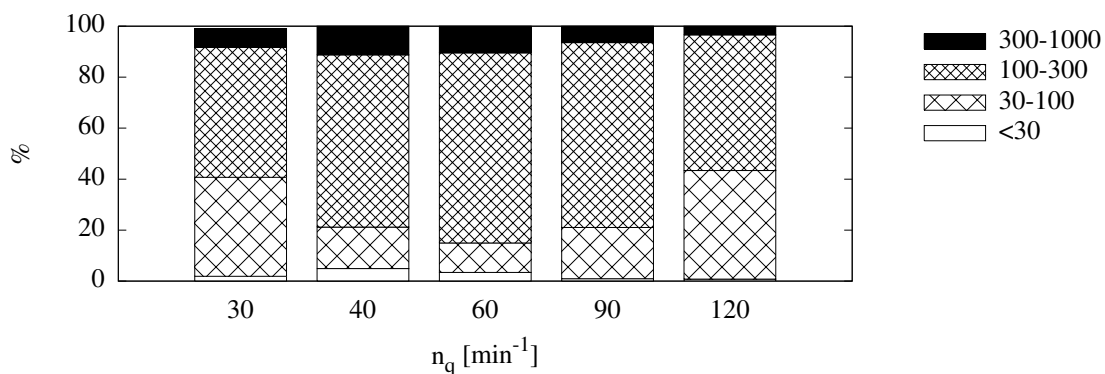
$$y_{p,opt,Le} = 1.62 \cdot 10^{-8} n_q^2 - 1.0037 \cdot 10^{-6} n_q + 3.7078 \cdot 10^{-4}. \quad (4.21)$$

Auf Basis dieser Regressionsfunktionen kann  $y_{p,opt}$  ermittelt werden. Aus einem vorgegebenen Wert für  $n_j$  und der Dicke des O-Netzes um die Schaufel kann ein  $\lambda_{opt}$  ermittelt werden. Der Traversenbereich wurde nicht untersucht, es ist aber anzunehmen, dass die Werte für die Leitschaufel ohne große Einbußen an Genauigkeit auf die Traversenvernetzung übertragen werden können.

Für die Hub-Shroud-Vernetzung liegen andere Strömungsverhältnisse als bei der Radialpumpe vor:

- Der Einlassbereich umfaßt neben der Traverse auch die Leitschaufel. Die auftretenden Geschwindigkeiten sind im äußeren Bereich geringer als bei der Simulation der Pumpe. Bei der Einzelkanalrechnung der Pumpe spielt dieser Bereich entweder keine Rolle, da er mit reibungsfreien Wänden modelliert wird, oder die Strömung wird durch eine Verengung der Meridiankontur künstlich beschleunigt. In der Tendenz ist damit zu rechnen, dass die ermittelten Vernetzungsempfehlungen aus Abschnitt 4.2.2 zu fein sind.
- Entgegengesetzt dazu verhält sich der Saugrohrbereich der Francis-Turbine. Während im Saugmund der Pumpe von einer drallfreien Zuströmung ausgegangen wird, benötigt die Francis-Turbine in diesem Bereich eine Drallkomponente, so dass die absolute Geschwindigkeit in diesem Bereich steigt und die  $y^+$ -Werte höher ausfallen als bei der Pumpe. Die Vernetzungsempfehlungen für die Pumpe sind hier tendenziell zu grob.

Das zur Verfügung stehende Vernetzungsverfahren benutzt eine einheitliche Anzahl von Netzebenen in Hub-Shroud-Richtung und eine einheitliche Verzerrungsfunktion, so dass diesen beiden Effekten nicht entgegengewirkt werden kann. Allerdings sind die mit dem Vernetzungsvorschlag aus der Pumpenreihe erzielbaren  $y^+$ -Werte ausreichend gut im zulässigen  $y^+$ -Intervall, s. Abbildung 4.21.



**Abbildung 4.21:** Klassen der  $y^+$ -Werte im Hub-Shroud-Netz über  $n_q$

### 4.3.3 Konfiguration des Solvers

Anders als bei der Pumpe wird die Strömung in der Francis-Turbine beschleunigt, d.h. die Strömungsrichtung folgt den fallenden Druckgradienten, wodurch das Konvergenzverhalten des Solvers deutlich stabiler wird. Die Auswirkung der Leitschaufelstellung auf das Geschwindigkeitsfeld ist deutlich höher als das  $n_q$  der Maschine. Für die Unterrelaxationsfaktoren haben sich folgende Wertebereiche als vorteilhaft erwiesen:

$$\alpha_{uvw} \in [0.45...0.3] , \quad (4.22)$$

$$\alpha_p \in [0.15...0.1] . \quad (4.23)$$

Analog zur Pumpe gilt auch hier: Ist das Residuum der Druckkorrektur höher als das der Impulsgleichungen, sollte das Verhältnis  $\alpha_{uvw}/\alpha_p$  erhöht werden, sind die Residuen der Impulsgrößen deutlich höher als das der Druckkorrektur, so gilt folgende Empfehlung:

$$\frac{\alpha_{uvw}}{\alpha_p} \geq 5 , \quad \alpha_{uvw} \leq 0.2 . \quad (4.24)$$

Im Regelfall ist für die Einzelkanalrechnung der Francisturbine im Optimalpunkt eine Simulation möglich, die voll mit einem Diskretisierungsschema 2. Ordnung, z.b. MIN-MOD, gerechnet ist. Dies setzt voraus, dass das Netz ausreichend orthogonale Zellen aufweist und die Leitschaufel korrekt eingestellt ist.



# Kapitel 5

## Anwendung

### 5.1 Einzelkanalsimulationen

#### 5.1.1 Pumpenreihe

Um die Zulässigkeit der in Abschnitt 4.2 ermittelten Vernetzungs- und Solverparameter zu beurteilen, wurde die gesamte Pumpenreihe aus Tabelle 4.4 nach den ermittelten Vorgaben vernetzt und im Auslegungspunkt simuliert. Zusätzlich wurde für jede Pumpe eine Charakteristik für den Bereich  $0.6 \cdot \phi_{opt}$  bis  $1.3 \cdot \phi_{opt}$  ermittelt, wobei eine Schrittweite von  $0.1 \cdot \phi_{opt}$  gewählt wurde. Pro Pumpe wurden 8 Simulationen durchgeführt, wobei ermittelte Ergebnisse als initiale Lösung für den Nachbarpunkt dienten. Die Simulation wurde aus Stabilitätsgründen zuerst mit UDS durchgeführt und anschließend auf einen MINMOD-Anteil von 0.4 erhöht.

Nach Abschnitt 4.2.2 wird für die Schaufel gefordert, dass der flächengemittelte dimensionslose Wandabstand  $\overline{y^+}$  sich im Intervall  $[20, 100]$  bewegt. Die Schaufelnetze wurden nach der Vorschrift aus Abbildung 4.13 erzeugt. Die damit erreichten  $\overline{y^+}$ -Werte für alle Betriebspunkte können in Abbildung 5.1 abgelesen werden. Einen Sonderfall stellt die Radialpumpe  $n_q = 90 \text{ min}^{-1}$  dar: Hier mußte die C-Netz-Dicke auf  $0.015 \cdot r_{ref}$  reduziert werden, so dass  $\overline{y^+}$  sich deutlich verringert.

Die Vernetzung in Hub-Shroud-Richtung wurde mit der Forderung nach einem maximalen Wandschicht-Expansionsfaktor von 2 ermittelt. Aus der Anzahl Netzebenen  $n_k$  in Hub-Shroud-Richtung wird eine charakteristische Zell-Länge ermittelt, die für die Wahl der übrigen Netzauflösung verwendet wird. Somit sind neben  $n_q$  vor allem die Länge der Schaufel, bestimmt durch Meridiankontur und Umschlingungswinkel, die Teilung und die Zulaufänge bestimmend für die Gesamt-Zellanzahl  $N$ . Die Dimensionen des Rechennetzes sind in Tabelle 5.1 aufgeführt.

Das automatisch aufgeteilte Rechengebiet wurde nur hinsichtlich der Randbedingungen überprüft: Es mußten am Auslass sowie einlaßseitig auf der Hubkontur des Rechengebiets die Randbedingungen manuell angepaßt werden. Ansonsten wurde das Netz wie automatisch ermittelt verwendet. Die Pumpen mit  $n_q \geq 20 \text{ min}^{-1}$  wurden ohne initiale Lösung mit den Solvereinstellungen aus Abbildung 4.17 gerechnet und konvergierten ohne Benutzereingriff. Lediglich die Pumpen  $n_q = 10 \text{ min}^{-1}$  sowie  $n_q = 15 \text{ min}^{-1}$  er-

$n_q [min^{-1}]$	10	15	20	25	30	40	50	70	90
$n_k$	11	13	15	17	19	19	25	31	35
$N$	82560	57876	60900	90256	74034	39780	61776	91200	64056

**Tabelle 5.1:** Netzdimensionen der Pumpenreihe

forderten Benutzereingriffe, bei beiden wurden die Unterrelaxationsfaktoren  $\alpha_{uvw}$  und  $\alpha_p$  herabgesetzt.

Die Kennlinie wurde mit dem Modul *autorun*, das für das IDS implementiert wurde, aufgesetzt und automatisch gerechnet. Dieses Modul ermittelt aus den Vorgabedaten die Randbedingungen für die verschiedenen Betriebspunkte, startet und überwacht Einzelsimulationen und ermittelt aus den Bilanzierungsebenen die integralen Werte der Pumpe, so dass Charakteristiken automatisiert verfügbar sind. Die Netze werden von *autorun* nicht angepaßt, die Solvereinstellungen gelten global für alle Einzelsimulationen.

Während alle Simulationen für die bezogenen Volumenströme  $\phi \geq \phi_{opt}$  stets mit den Einstellungen konvergierten, mit denen auch die Simulation im Auslegungspunkt konvergierte, waren bei einigen  $n_q$  für die Betriebspunkte  $0.6 \dots 0.8 \cdot \phi_{opt}$  teilweise erhebliche manuelle Anpassungen der Solverparameter notwendig. Konvergenzprobleme äußerten sich beim NS3D im wesentlichen dadurch, dass die Residuen der Lösungsgrößen um einen Wert pendeln, ohne dass es zu einem Abfall des Gesamtniveaus kommt. Bei Konvergenzproblemen muß generell die Unterrelaxation erhöht werden, d.h. die Werte für  $\alpha_{uvw}$  und  $\alpha_p$  müssen nach unten korrigiert werden. Für den Fall, dass das Residuum der Druckkorrektur deutlich höher ist als das Residuenniveau der Impulsgrößen, reagiert der Solver sehr gut auf die einseitige Verringerung von  $\alpha_{uvw}$  und konvergiert in wenigen Iterationen. Wenn die Druckkorrektur hingegen nominell konvergiert ist, die Residuen der Impulsgrößen allerdings noch eine bis eineinhalb Größenordnungen darüber liegen, müssen beide Werte angepaßt werden, wobei im Regelfall lediglich das Niveau des Mittelwerts geändert wird. Gelegentlich hilft es, hier  $\alpha_p$  eine Größenordnung unter  $\alpha_{uvw}$  zu wählen.

Die Pumpengeometrien, bei denen der Auslegungspunkt den Optimalpunkt gut trifft, weisen für die Betriebspunkte  $0.6 \dots 0.8 \cdot \phi_{opt}$  eine Tendenz zur Instabilität der Simulation auf, da in diesem Betriebsbereich tendenziell Teillastwirbel im Saugmund einsetzen. Die endliche räumliche Auflösung durch die Netzzellen ist bei der gewählten Netzdichte zu grob, um die kleinen Strukturen einsetzender Wirbelbildung erfassen zu können, zusätzlich können instationäre Effekte durch das gewählte Einzelkanalmodell nicht abgebildet werden. Treten diese in den Vordergrund, kann keine Konvergenz für den stationären Fall erzielt werden. Für die Radialpumpen  $n_q = 20 min^{-1}$  bis  $n_q = 30 min^{-1}$  konnte keine auskonvergierte Lösung für 40% MINMOD für die Betriebspunkte  $0.6 \dots 0.7 \cdot \phi_{opt}$  ermittelt werden. In den Charakteristiken aus Abbildung 5.3 sind die entsprechenden Kurven um die Werte für reine UDS-Diskretisierung ergänzt.

Die Werte für  $\phi$  wurden nicht um geschätzte Spaltvolumenströme ergänzt, so dass der Verlust durch den Förderstrom in den Gehäusespalten komplett vernachlässigt wurde.

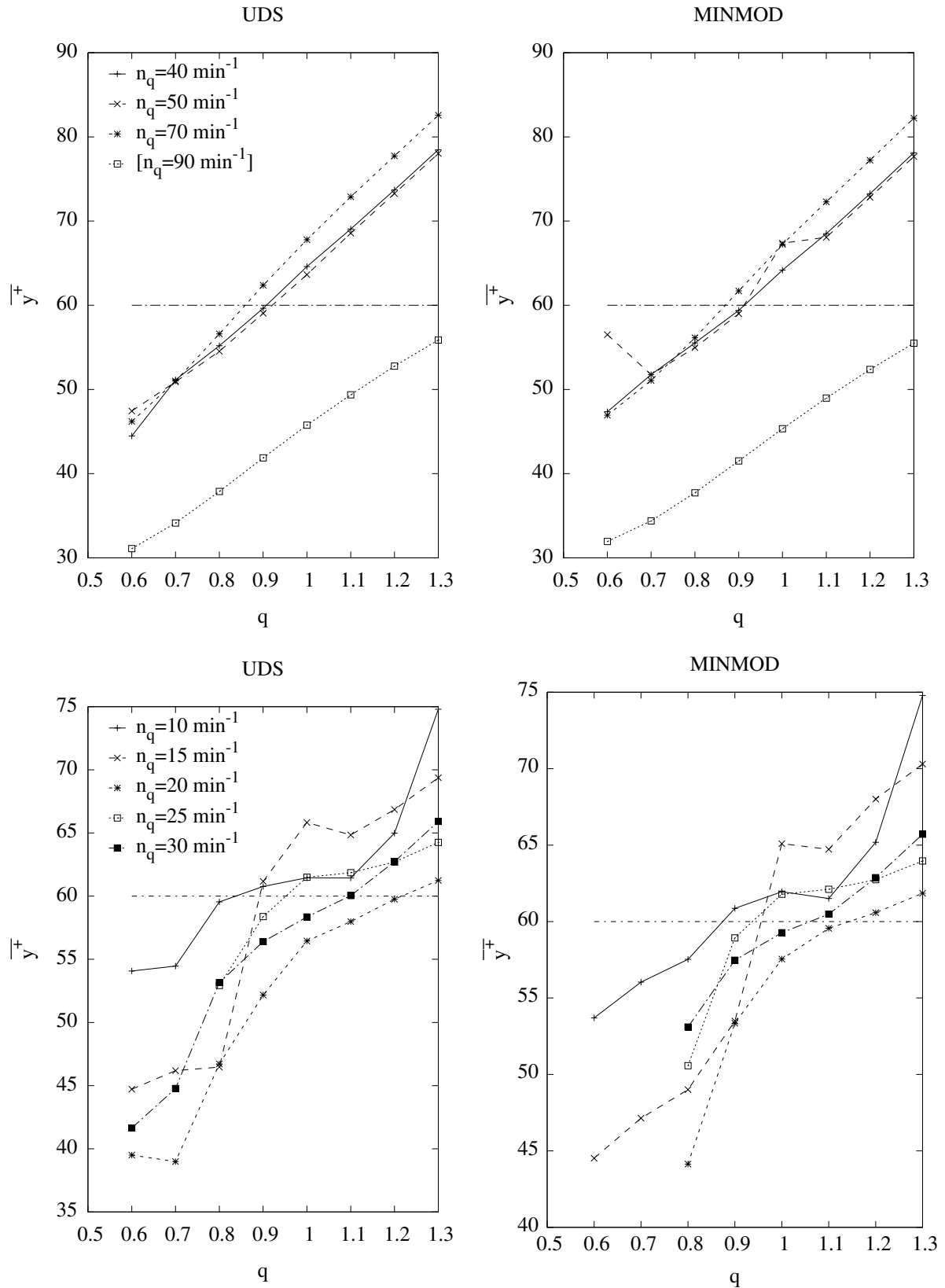


Abbildung 5.1: Pumpenreihe: Ermittelte  $\overline{y^+}$ -Werte in Abhängigkeit vom relativen Volumenstrom und  $n_q$  der Pumpenreihe

Der Volumenstrom durch den Gehäusespalt hat neben der Erhöhung des Bruttovolumenstroms vor allem eine Änderung der Anströmung der Schaufel im Shroudbereich zur Folge, die durch pauschale Erhöhung des Gesamtvolumenstroms nicht abgebildet werden kann.

Im Bereich höherer  $n_q$  sind der Verlauf des Wirkungsgrads wie auch der Totaldruckerhöhung  $\Psi_t$  plausibel und decken sich mit den Erwartungen an eine Pumpenkennlinie unter Vernachlässigung der Gehäusespaltverluste, vgl. GÜLICH [26]. Es ist zu beobachten, dass die beiden Radialpumpen mit dem hohen  $n_q = 70 \text{ min}^{-1}$  und  $n_q = 90 \text{ min}^{-1}$  nicht die Förderhöhe erbringen, die gefordert ist, s. Abbildung 5.2. Der Auslegungspunkt entspricht hier nicht dem Optimalpunkt und liegt bereits im Überlastbereich, d.h. hier besteht für die Erstausslegung noch Optimierungsbedarf.

Der Bereich niedriger  $n_q$  ist in Abbildung 5.3 dargestellt. Die Wirkungsgradverläufe werden mit fallendem  $n_q$  immer weniger verständlich, der vorhergesagte Wirkungsgrad ist für  $n_q \approx 15 \text{ min}^{-1}$  fast unabhängig vom bezogenen Volumenstrom  $\phi$  und bei  $n_q = 10 \text{ min}^{-1}$  reziprok zum Volumenstrom. Obwohl keine Vergleichsmessungen vorliegen, sind diese Kennlinienformen als unrealistisch zu bewerten.

Um die grundsätzliche Eignung der numerischen Simulation für diese Fälle zu untersuchen, wurde die Radialpumpe  $n_q = 20 \text{ min}^{-1}$  parallel mit der kommerziell verfügbaren Software CFX untersucht, wobei die gleichen Netze wie bei der Simulation mit dem NS3D verwendet wurden. In Abbildung 5.4 sind die mit CFX und NS3D ermittelten Kennlinien gegenübergestellt, wobei in CFX das  $k\text{-}\varepsilon$ -Modell und ergänzend das empfohlene SST-Modell verwendet wurden. Während die Netze identisch sind und die Turbulenzmodelle ähnlich, weist CFX eine grundsätzlich andere Wandbehandlung auf. Es ist hinsichtlich der Wandreibung freier bezüglich der  $y^+$ -Werte, da es über eine skalierende Wandfunktion verfügt. Abbildung 5.5 zeigt die Verteilung der  $y^+$ -Werte auf der Schaufel für die NS3D-Simulationen, wobei die vier Quartile und der Median angegeben sind. Die Spanne der  $y^+$ -Werte für die niedrigen  $n_q$  ist größer, wobei insbesondere der höhere Anteil niedriger  $y^+$ -Werte auffällt, sowie die Verbreiterung des zweiten Quartils. Skalierbare Wandfunktionen greifen in diesem Intervall ein, was eine Erklärung für die fehlerhafte Verlustvorhersage in diesem Bereich ist. Da mit niedrigeren  $n_q$  die Verluste durch Reibung stärker ins Gewicht fallen als die Mischungsverluste infolge innerer Reibung, vgl. z.B. GÜLICH[26], fallen Abweichungen in der Modellierung dieses Mechanismus stärker ins Gewicht.



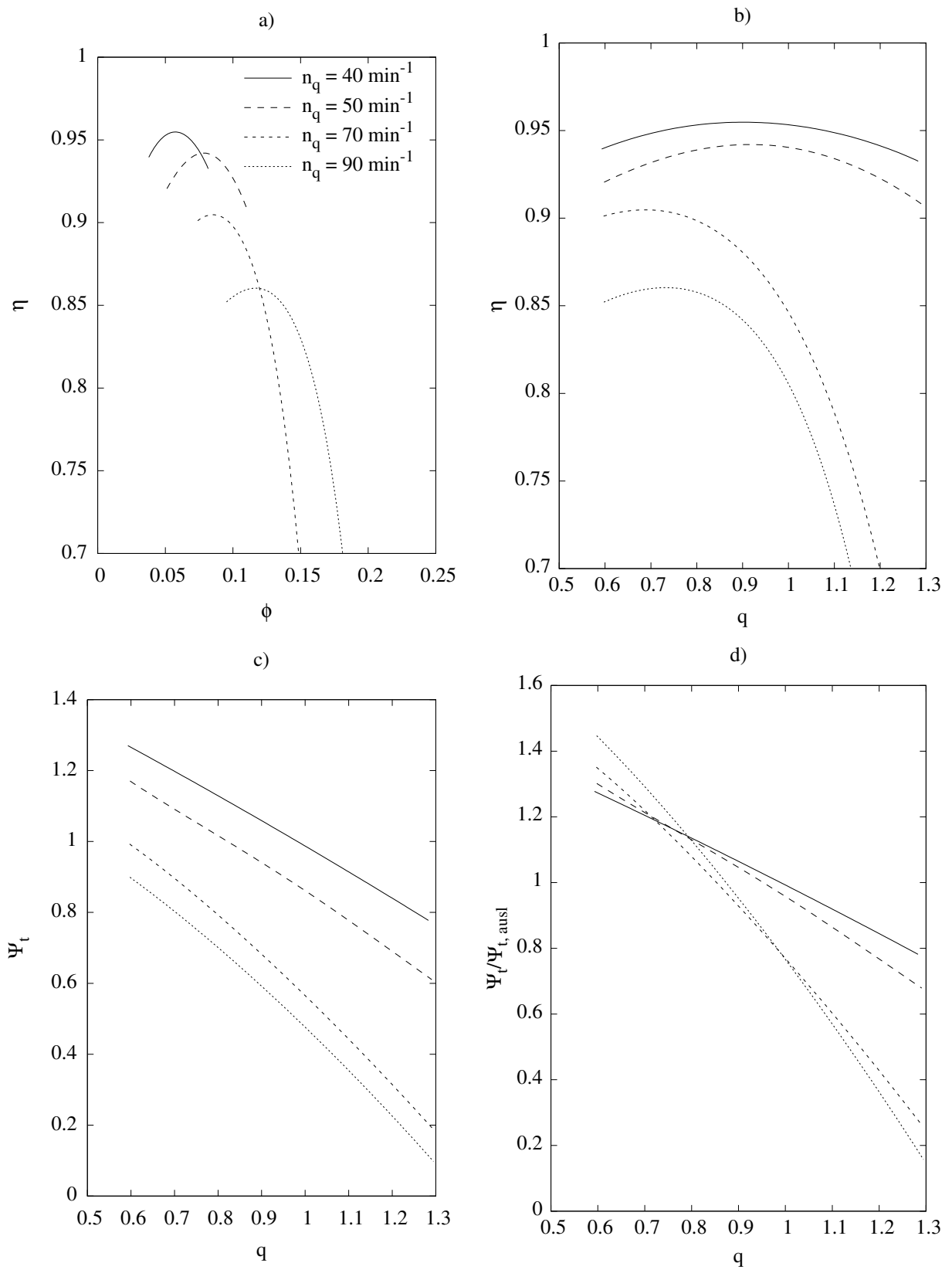


Abbildung 5.2: Charakteristiken der Pumpenreihe für  $n_q \geq 40 \text{ min}^{-1}$

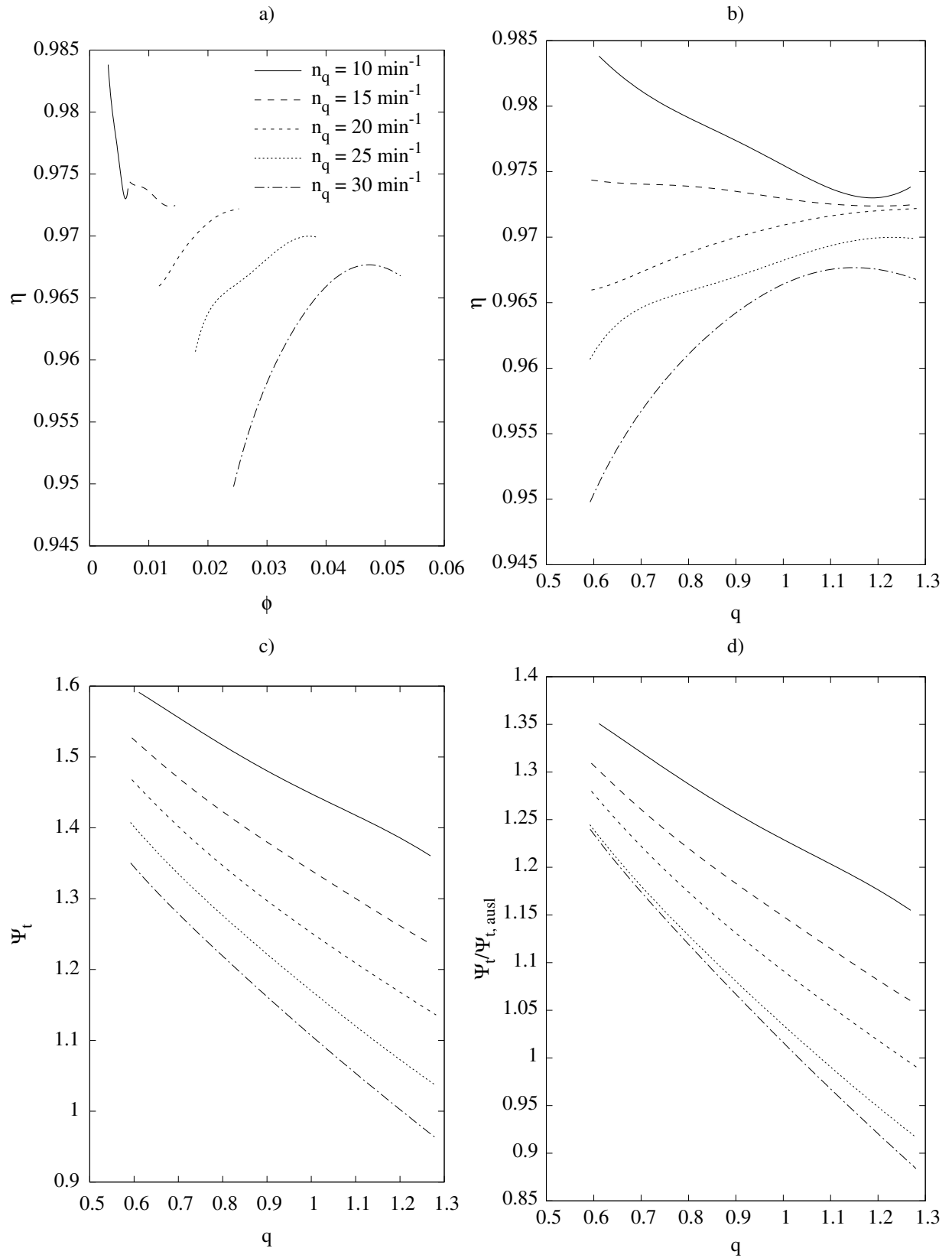


Abbildung 5.3: Charakteristiken der Pumpenreihe für  $n_q \leq 30 \text{ min}^{-1}$

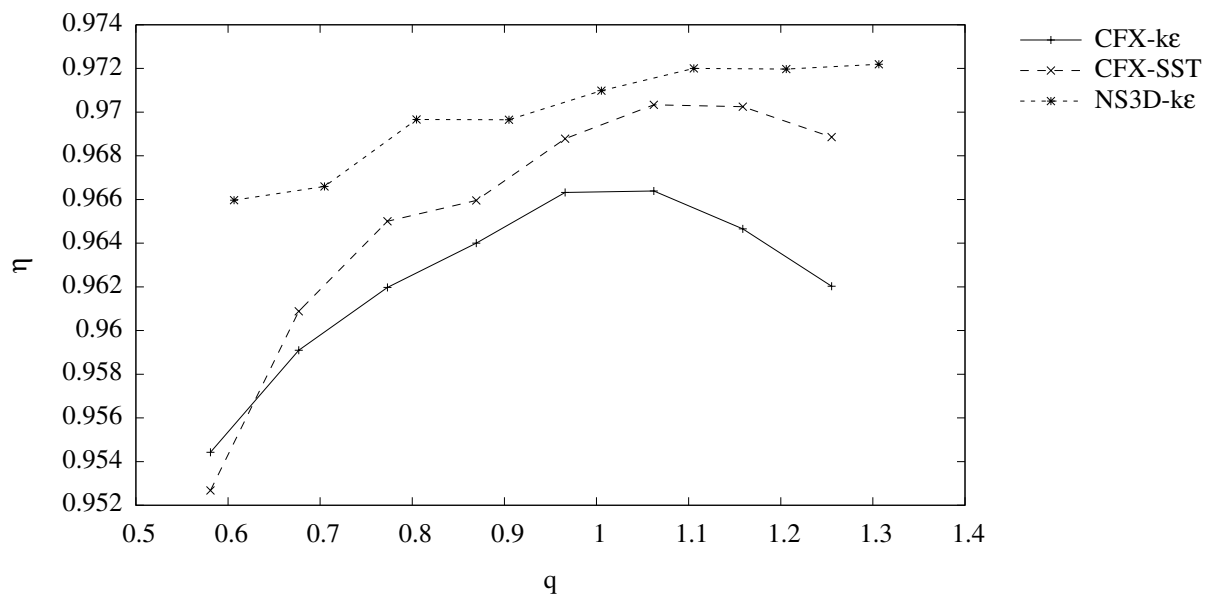


Abbildung 5.4: Wirkungsgradvorhersage der Pumpenreihe mit CFX und NS3D für  $n_q = 20 \text{ min}^{-1}$

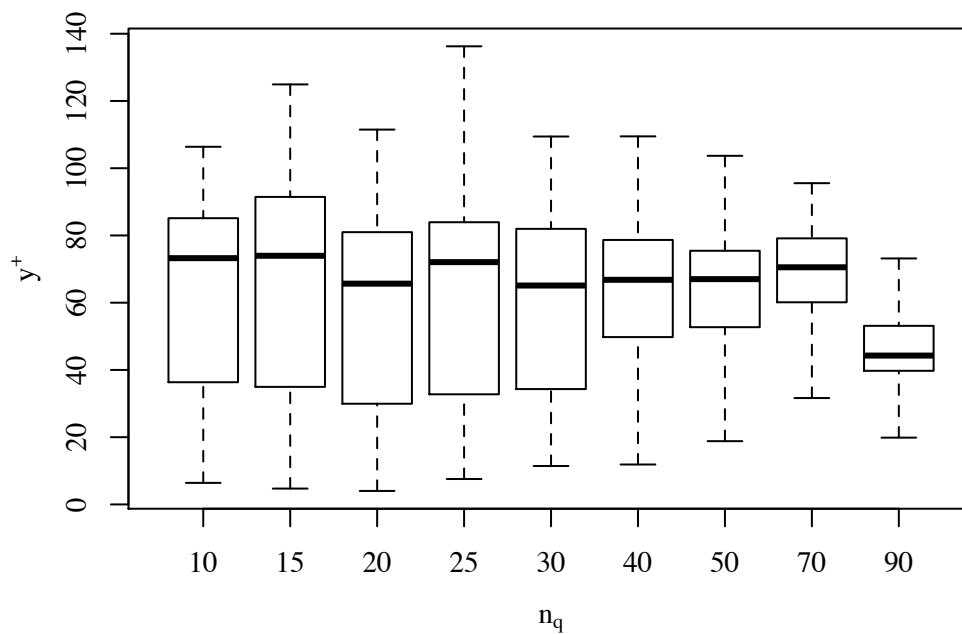


Abbildung 5.5:  $y^+$ -Verteilung für die Pumpenreihe auf der Schaufel

### 5.1.2 Turbinenkennfeld

Um die Funktion des Programmsystems hinsichtlich der Simulation von Turbinen zu prüfen, wurde ein Kennfeld einer Francisturbine ermittelt. Dazu wurde mit dem KNN von KRÄMER [32] die Geometrie einer Turbine mit der spezifischen Drehzahl  $n_q = 44.146 \text{ min}^{-1}$  erzeugt und auf den geforderten Optimalpunkt angepaßt, s. Tabelle 5.2. Der Simulation wird ein gemessenes Kennfeld für diesen Turbinentyp gegenübergestellt, das zum Vergleich der durch Simulation gewonnenen Daten herangezogen werden kann, s. SCHILLING [47].

$n_{q,\text{opt}}[\text{min}^{-1}]$	$\phi_{\text{opt}}[-]$	$\Psi_{t,\text{opt}}[-]$	$Q_{\text{ausl}}[\text{m}^3/\text{s}]$	$H_{t,\text{soll}}[\text{m}]$	$Z[-]$
44.146	0.153	1.563	0.2473	25.25	15

**Tabelle 5.2:** Auslegungsdaten der simulierten Turbine

Da das CFD-Experiment als Einzelkanalsimulation mit Laufrad, Leitschaufel und Traverse durchgeführt wurde, konnten Spirale und Saugrohr nicht berücksichtigt werden. Die Netze wurden nach den Maßgaben des Abschnitt 4.3.2 erstellt. Als Basis diente die Simulation des Optimalpunkts. Von hier aus wurde ohne Anpassung des Leitapparats der Volumenstrom unter Beibehaltung des Anströmwinkels der Traverse variiert. Durch sukzessives Öffnen bzw. Schließen des Leitapparats können dann parallel dazu Betriebspunkte angefahren werden, bei denen ebenfalls der Volumenstrom variiert wurde. Insgesamt wurden 6 verschiedene Leitradstellungen mit je 7 unterschiedlichen Volumenströmen simuliert.

Als Ergebnis erhält man aus der Simulation eine Schar von Kurven für  $H_t(Q)$ ,  $\eta(Q)$ . Um die Größen dimensionslos aufzutragen, bedient man sich bei der Turbine nicht eines Diagramms in  $\phi$  und  $\Psi$ , sondern es werden üblicherweise die folgenden reduzierten Größen definiert:

$$n'_1 = \frac{nD}{\sqrt{H}} \cdot \frac{[\sqrt{H = 1\text{m}}]}{[D = 1\text{m}]} \quad (5.1)$$

$$Q'_1 = \frac{Q}{D^2\sqrt{H}} \cdot [(D = 1\text{m})^2 \cdot \sqrt{H = 1\text{m}}] \quad (5.2)$$

Das angegebene  $H$  ist das  $H_t$  des jeweiligen Betriebspunkts. In der Simulation wird allerdings nur ein Teil der Fallhöhe ermittelt, nämlich der Anteil, der in Traverse, Leitapparat und Laufrad verloren geht bzw. umgesetzt wird. Die Gesamtfallhöhe ergibt sich zu:

$$H_t = h_{v,Sp} + h_{v,SR} + H_{St} \quad (5.3)$$

$$H_{St} = h_{v,Tra} + h_{v,Lei} + H_{t,La} \quad (5.4)$$

Die Stufenfallhöhe  $H_{St}$  ist das Ergebnis der CFD-Simulation. Die Verlusthöhen der Spirale  $h_{v,Sp}$  und des Saugrohrs  $h_{v,SR}$  sind unbekannt und nur durch die deutlich aufwendigere Simulation der kompletten Turbine zu ermitteln. Um das gemessene Kennfeld und

die CFD-Simulation vergleichen zu können, müssen die Verlusthöhen dieser Elemente abgeschätzt werden. Dazu wird im Optimalpunkt folgende Annahme getroffen:

$$\begin{aligned}
 H_{t,th} &= \eta \cdot H_t \\
 H_{t,th,St} &= \eta_{St} \cdot H_{St} \\
 H_{t,th} &\stackrel{!}{=} H_{t,th,St} \\
 \implies H_t &= \frac{\eta_{St}}{\eta} \cdot H_{St}
 \end{aligned} \tag{5.5}$$

Somit ergibt sich für einen Punkt, in dem  $\eta_{St}$ ,  $\eta$  und  $H_{St}$  bekannt ist, für die Verluste der nicht simulierten Elemente folgender Zusammenhang:

$$h_{v,Sp} + h_{v,SR} = h_v = H_t - H_{St} = \frac{\eta_{St}}{\eta} \cdot H_{St} - H_{St} = H_{St} \left( \frac{\eta_{St}}{\eta} - 1 \right) \tag{5.6}$$

Mit der Annahme, dass die Verluste in Spirale und Saugrohr proportional mit dem Quadrat des Volumenstrom verlaufen, kann gefordert werden:

$$\begin{aligned}
 h_v &= \xi \cdot Q^2 \\
 \implies \xi &= \frac{H_{St,0}}{Q_0^2} \left( \frac{\eta_{St}}{\eta} - 1 \right)
 \end{aligned} \tag{5.7}$$

Somit kann aus einem Referenzpunkt  $Q_0$  und  $H_{St,0}$  ein Verlustbeiwert für die Verluste in Saugrohr und Spirale definiert werden, mit dem sich diese Verluste in Abhängigkeit von  $Q$  abschätzen lassen.

Da die tatsächliche Fallhöhe lediglich abgeschätzt wird, ist die Darstellung in  $n'_1$  und  $Q'_1$  nicht sinnvoll, da in beiden Fällen der Term  $\sqrt{H}$  im Nenner steht und das Kennfeld der Simulation stark verzerrt wird. Für Turbinen kann alternativ die Darstellung in  $E_{\omega D}$  und  $Q_{\omega D}$  gewählt werden:

$$E_{\omega D} = \frac{gH}{\omega^2 D^2} \tag{5.8}$$

$$Q_{\omega D} = \frac{Q}{\omega D^3} \tag{5.9}$$

## Simulationsverhalten

Die Simulation wurde im Optimalpunkt bei Nennvolumenstrom und -fallhöhe gestartet. Auf Basis dieses Simulationsergebnisses wurde die Linie konstanter Leitschaufelöffnung gerechnet. Von dieser Linie aus wurden in Richtung größerer und kleiner Öffnungswinkel Linien konstanter Leitschaufelöffnungen ermittelt, wobei zunächst wieder der Punkt  $H_{t,St} = H_{opt,St}$  eingestellt wurde. Durch das Anfahren dieser Punkte konnte jedem dieser Basispunkte ein Wirkungsgrad  $\eta_{exp}$  zugeordnet werden, mit dem der Verlustbeiwert

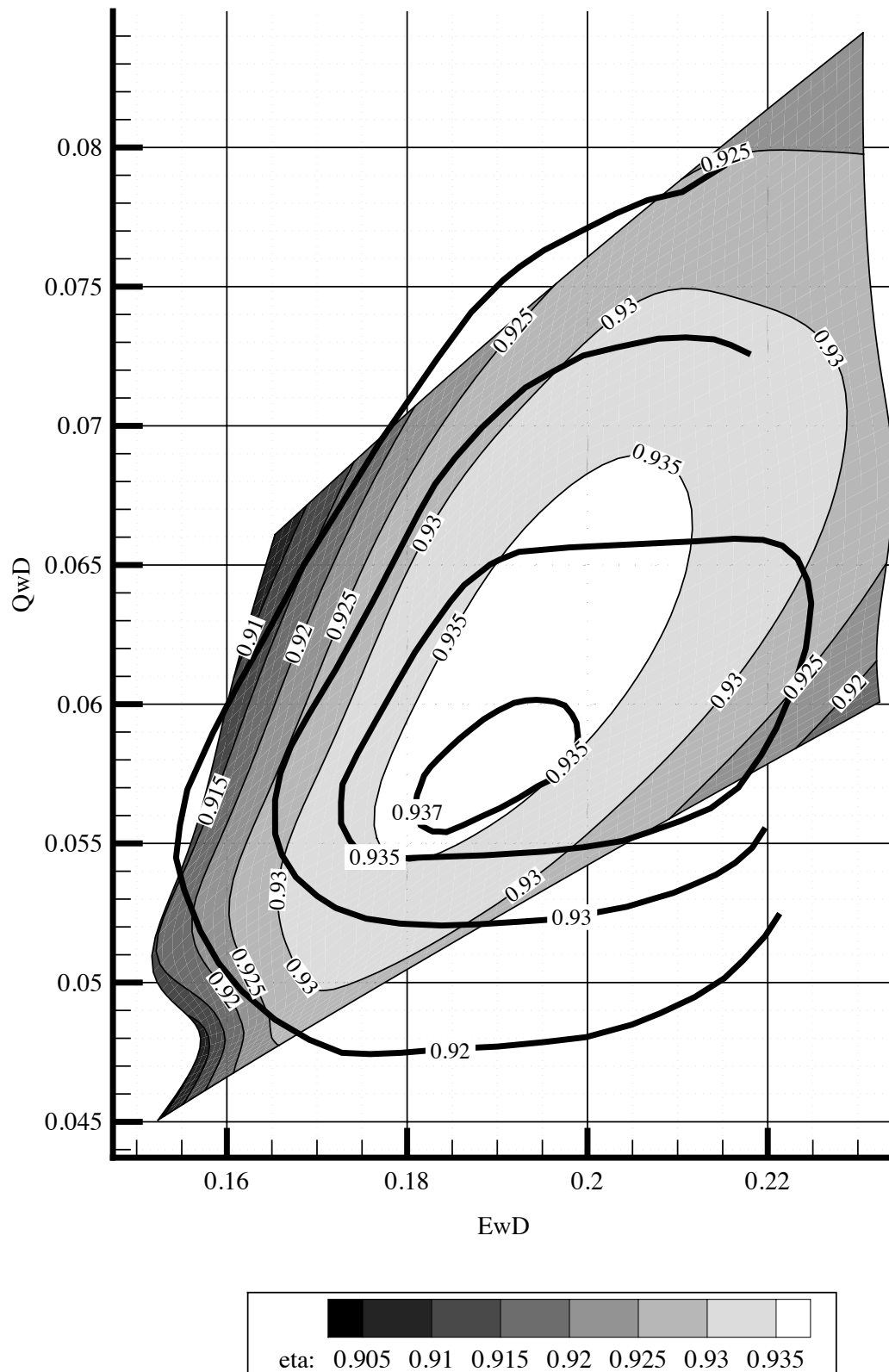


Abbildung 5.6: Kennfeld der Simulation und Iso- $\eta$ -Linien der Messung für die FT44.

$\xi$  ermittelt werden konnte. Diese Vorgehensweise erlaubt eine gute Parallelisierung des Simulationsexperiments, da voneinander unabhängige Betriebszustände simuliert werden können.

Im Laufe der Simulationen hat sich gezeigt, dass für Volumenströme im Bereich  $Q/Q_{opt} \leq 0.75$  keine zufriedenstellenden Ergebnisse zu erwarten sind. In diesem Bereich tendiert die Einzelkanalsimulation dazu, im Totwasser der Nabe zu rezirkulieren. Hier stößt die Modellvorstellung eines stationären, repräsentativen Einzelkanals an seine Grenzen. In Richtung Überlast ist die Simulation stabil, Instabilitäten traten im behandelten Bereich nicht auf.

### Bewertung

In Abbildung 5.6 sind die Isolinien der Messung und das mittels CFD ermittelte Kennfeld übereinandergeplottet. Die Deckung entlang der Linie  $E_{\omega D} = 0.193$  zeigt eine befriedigende Übereinstimmung. Insgesamt ist der Wirkungsgradverlauf der Messung in Richtung größerer Fallhöhen deutlich flacher als in der Simulation. Die Simulation kann die Gestalt der Isolinien nur verzerrt wiedergeben. Die Ermittlung der restlichen Verluste  $h_v$  nach Gleichung (5.7) ist wahrscheinlich in dem breiten Bereich nicht zulässig, den das Kennfeld abdeckt. Zusätzlich ist noch zu beachten, dass die Geometrie der zugrundegelegten Turbine nur näherungsweise der vermessenen Geometrie entspricht, und die Abweichungen hierdurch begründet sein können.

## 5.2 Automatisierte Optimierung

Mit dem automatisierten CFD-Experiment und der stark verkürzten Laufzeit des Quasi-Navier-Stokes-Modells erschließt sich die Möglichkeit, Simulationen zu nutzen, um einen automatisierten Optimierungsvorgang durchführen zu lassen. Analog zu THUM [53], die in ihrer Arbeit mit der Response Surface Methode (RSM) einen Diffusor sowie die strömungsführenden Bauteile einer Francis-Spiralturbine optimieren ließ, soll die Pumpenreihe von KRÄMER [32] im Bereich großer  $n_q$  optimiert werden. Als Optimierungsverfahren wurde ein genetischer Algorithmus gewählt, prinzipiell könnte jedoch jedes andere Optimierungsverfahren angewendet werden, das ein System mit mehreren Entwurfsvariablen hinsichtlich mehrerer Systemantworten optimieren kann. Der genetische Algorithmus weist den Vorteil auf, dass er keine Annahmen zur Stetigkeit des Lösungsraumes trifft, sondern auf einfachen besser/schlechter Entscheidungen basiert. Zusätzlich können mit diesem evolutionsbasierten Verfahren auch ganzzahlige Eingangsgrößen abgebildet werden, die bei der RSM nicht ohne weiteres anwendbar sind.

### 5.2.1 Genetische Algorithmen

THUM [53] erläutert Genetische Algorithmen bzw. Evolutionsstrategien als Such- und Optimierungsverfahren, die sich am Vorbild der Evolution und Genetik orientieren. Die einzelnen Entwurfsvariablen werden als Gene betrachtet, die Summe der Gene erzeugen

dann ein Chromosom. Aus einer konkreten Belegung der einzelnen Gene im Chromosom wird ein Individuum, dessen „Phänotyp“ die Systemantwort im Ergebnisraum ist.

Entsprechend dem Vorbild der Natur wird eine Population von Individuen erzeugt, die durch unterschiedliche, meist zufällige Belegung der Gene definiert sind. Diese Individuen werden im Ergebnisraum betrachtet, aus der Population werden neue Individuen rekombiniert, mutiert und selektiert. Diese neue Generation wird wieder hinsichtlich ihres Verhaltens im Ergebnisraum bewertet. Aus der Ausgangsgeneration werden die nicht geeigneten Lösungen aussortiert; dieses Vorgehen wird i.A. „*survival of the fittest*“ genannt. Die Kindgeneration wird erzeugt aus Vermischung der Eigenschaften der Elterngeneration auf Basis jedes einzelnen Gens. In die Kindgeneration werden durch Mutation wieder zufällige Änderungen eingebracht. Je nach Codierung können auch diskret belegte Entwurfsvariablen in die Betrachtung einfließen. Durch das Zufallsmoment der Mutation kann der Suchraum komplett ausgewertet werden, wenn eine ausreichende Anzahl von Generationen betrachtet wird.

Die genetischen Algorithmen sind einfach zu implementieren, universell anwendbar und robust hinsichtlich des Ergebnisses. Da sie auf einer Vielzahl von unabhängigen Einzelbetrachtungen basieren, sind sie gut parallelisierbar. Die Anwendung von Evolutionsstrategien für Optimierungsprobleme sind beschrieben bei BÄCK [10] und RECHENBERG [43], genetische Algorithmen bei GOLDBERG [25].

Für die vorliegende Problemstellung wird die Bibliothek JGAP von MEFFERT [38] verwendet, die neben einer Implementierung eines genetischen Algorithmus auch die Möglichkeit der genetischen Programmierung umfasst.

### Verwendete Codierung

Als Eingangsgrößen werden die Geometrieparameter der Schaufel auf der äußeren (Shroud), einer mittleren (Mean) und der inneren (Hub) Strombahn benutzt, wobei jeweils der vordere Skelettflächenwinkel  $\beta_{s1}$ , der Austrittswinkel  $\beta_{s2}$ , der Umschlingungswinkel  $\Delta\varphi$ , die Länge der Schaufel in der Meridianansicht  $\Delta l$ , sowie die Anzahl der Schaufeln  $Z$  im Chromosom codiert wird. Die Schaufel selbst ist auf 9 Strombahnen definiert, wobei die Strombahnen 2, 3, 4, 6, 7 und 8 durch Querabgleich erzeugt werden.

Auf den Strombahnen sind die Skelettflächen mit B-Splines beschrieben, deren Kontrollpolygon 5 Punkte aufweist. Die Austrittskante der Schaufel bleibt fix, d.h. der erste Punkt des Kontrollpolygons ergibt sich aus der Rückrechnung der Koordinaten des letzten Punktes über  $\Delta\varphi$  und  $\Delta l$ . Die inneren Kontrollpunkte werden gleichverteilt, wobei der 1. innere und der letzte innere Kontrollpunkt so korrigiert werden, dass sich die im Chromosom codierten Schaufelwinkel  $\beta_{s1}$  und  $\beta_{s2}$  ergeben. Somit ist die Erscheinungsform („Phänotyp“) für einen Chromosomensatz eindeutig definiert.

Als Basischromosom wird die Belegung aus dem KNN verwendet. Um diese Ausgangsparameter wird ein Intervall definiert, innerhalb dessen die Gene vom GA variiert werden dürfen. Die Startbelegung ist abhängig von der zu optimierenden Pumpengeometrie, die Intervalle werden so weit gefaßt, dass im Extremfall nicht realisierbare Geometrien möglich sind. Es sind 13 Gene im verwendeten Chromosom vorhanden. Die Ausgangspopulation wird mit 65 Individuen ausgestattet, deren Chromosome zu-



fällig innerhalb der Variationsgrenzen belegt werden. Insgesamt werden 7 Generationen ausgewertet.

### Zielfunktion

Die Ziele des Optimierungslaufs werden wie folgt definiert:

1. Erreichen der geforderten Förderhöhe. Die Förderhöhe wird aus der Totaldruckdifferenz zwischen Bilanzierungsebenen vor und hinter dem Laufrad ermittelt. Diese Bedingung ist eine „Muß-Bedingung“.
2. Maximierung des Wirkungsgrads. Die aufgewendete Leistung wird aus dem Moment um die z-Achse infolge des Drucks auf die Schaufeloberfläche ermittelt, somit wird ein Wellenwirkungsgrad zugrunde gelegt.
3. Minimierung der Kavitationsneigung. Dazu wird die Verteilung des statischen Drucks an der Schaufeloberfläche herangezogen und der minimale Wert vermindert um den Dampfdruck herangezogen. Diese dimensionslose Differenz  $c_{p,min}$  soll maximiert werden.

Nicht beachtet werden Kriterien, die die Stabilität der Strömung beeinflussen können, z.B. das Abströmungs- oder das Anströmungsprofil. Es wird nur der Auslegungspunkt betrachtet, Teillast- oder Überlastverhalten gehen nicht in die Optimierung ein.

Die Förderhöhe muß erreicht werden, deshalb geht sie als Randbedingung in die Auswertung ein, es wird eine Unterschreitung von 5% zugelassen:

$$\Psi_t \geq 0.95 \cdot \Psi_{t,soll} \quad (5.10)$$

Die Zielfunktion bzw. die Systemantwort ergibt sich aus:

$$y = 100 \cdot \eta + 100 \cdot (h_{min} + 10) \quad (5.11)$$

$$y \longrightarrow max$$

$$h_{min} = \frac{u_{ref}^2}{2g} c_{p,min} \quad (5.12)$$

$$c_p = \frac{p_s - p_{va}}{\frac{\rho}{2} u_{ref}^2} \quad (5.13)$$

Die Rechnung erfolgt inkompressibel, bei der Auswertung wird der statische Druck an der Einlaßfläche auf  $p_{s,in}$  jeweils so festgesetzt, dass sich im Einlaß der jeweils geforderte NPSH-Wert ergibt. Der Druckbeiwert  $c_p$  wird auf der Schaufeloberfläche ermittelt. Aus den Vorbetrachtungen hat sich ergeben, dass  $c_{p,min}$  im Start negativ sein wird. JGAP erwartet für die Optimierung eine Zielfunktion, die positive Werte zurückgibt. Dies wird gewährleistet, indem der  $h_{min}$ -Wert pauschal um 10 m erhöht wird. Da  $h_{min}$  im Optimum wahrscheinlich im Bereich  $0 m \leq h_{min} \leq 2 m$  liegen wird, bewegt sich der Ausdruck im Bereich  $0 \leq 100 \cdot (h_{min} + 10) \leq 1200$ . Da der Wirkungsgrad  $\eta$  maximal

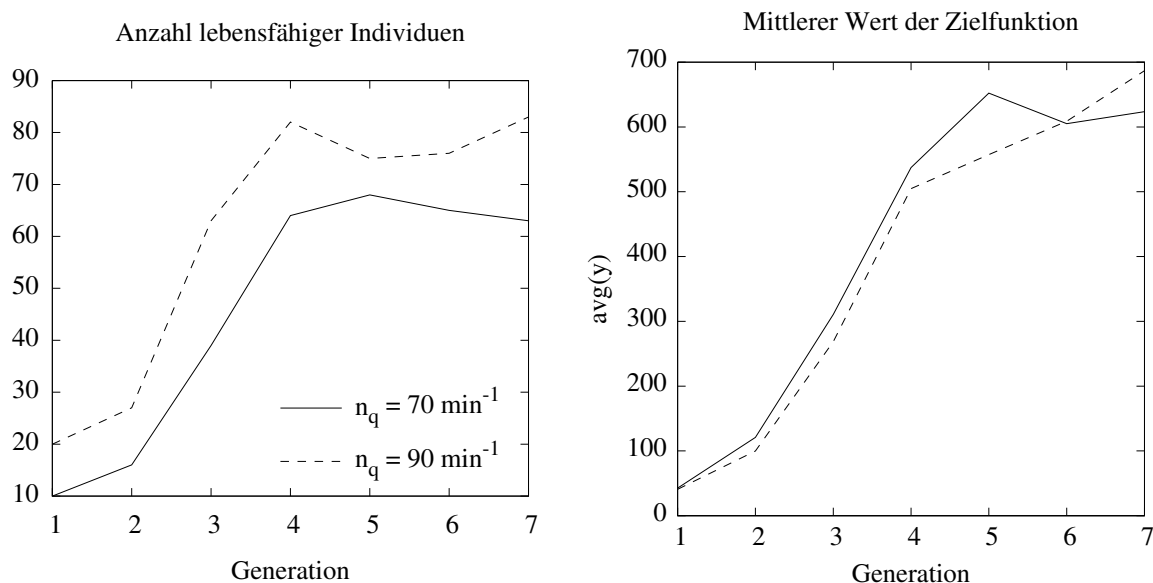
100 % erreichen kann, ist das Ziel Minimierung der Kavitationsneigung ca. zwölfmal stärker gewichtet.

Der vom GA benutzte Wirkungsgrad ist der automatisch ermittelte Systemwirkungsgrad  $\eta_{\text{sys}}$ , der aus der Totaldruckdifferenz zw. Einlaß- und Auslaßfläche des simulierten Strömungsgebiets sowie der Wellenleistung infolge Druckmoments ermittelt wird, s. Abschnitt 2.7.2.

Ist die Rechnung nicht durchführbar, da z.B. die Schaufelgeometrie sich selbst durchdringt, so wird als Funktionswert für die Zielfunktion 0 zurückgegeben. Wird die Randbedingung nicht erreicht, so wird ebenfalls eine 0 zurückgegeben. Das zugrundeliegende Individuum wird somit als ungültig markiert und aus der Population entfernt.

### Simulationsverfahren

Als Verfahren wird das QNS-Verfahren benutzt, s. Abschnitt 2.1.4, d.h. die Turbulenz wird über die pauschale Erhöhung der laminaren Viskosität abgebildet. Gleichzeitig wird das Rechenetz sehr grob aufgelöst, um die Rechenzeit pro Individuum so gering wie möglich zu halten. Als Konvergenzkriterium wird ein maximales Residuum von  $10^{-5}$  eingestellt. Simulationen, die innerhalb von 500 Iterationen dieses Kriterium nicht unterschreiten, werden in der Auswertung nicht berücksichtigt.



**Abbildung 5.7:** Konvergenzverhalten des genetischen Algorithmus für Radialpumpe  $n_q = 70 \text{ min}^{-1}$  und  $n_q = 90 \text{ min}^{-1}$

### Konvergenzverhalten

Beide Optimierungsläufe können nicht mit der vollen Anzahl von 65 Individuen in der ersten Generation durchgeführt werden, s. Abbildung 5.7. Die Anzahl lebensfähiger Individuen wird für beide Beispiele von Generation zu Generation gesteigert, bis in der

vierten Generation die Sollpopulationsgröße erreicht bzw. überschritten wird. Parallel dazu steigt der mittlere Wert der Zielfunktion von Generation zu Generation. Für die Radialpumpe  $n_q = 70 \text{ min}^{-1}$  ist nach der fünften Generation keine Verbesserung des durchschnittlichen Wertes mehr zu beobachten, die Optimierung ist hinsichtlich dieses Wertes abgeschlossen. Für die Radialpumpe  $n_q = 90 \text{ min}^{-1}$  ist bis zur siebten Generation ein steter Anstieg des durchschnittlichen Wertes der Zielfunktion zu verzeichnen, es ist offen, ob in einer eventuellen achten Generation noch eine weitere Verbesserung zu erreichen wäre.

## 5.2.2 Ergebnisse

### Radialpumpe $n_q = 70 \text{ min}^{-1}$

Die Ausgangsgeometrie entspricht der Pumpe mit  $n_q = 70 \text{ min}^{-1}$  (RP70) aus der Pumpenreihe, s. Abschnitt 4.2. Die ermittelten Betriebsdaten im Auslegungspunkt sind in Tabelle 5.3 aufgelistet. Als minimale Haltedruckhöhe wird ein Wert von  $\text{NPSH}_{\text{erf}} = 2.25 \text{ m}$  gefordert, dies entspricht einem statischen Druck im Saugmund von  $p_{s,\text{in}} = 19620 \text{ Pa} \hat{=} 2 \text{ m}$  bei einem Dampfdruck von  $p_{va} = 3800 \text{ Pa}$ .

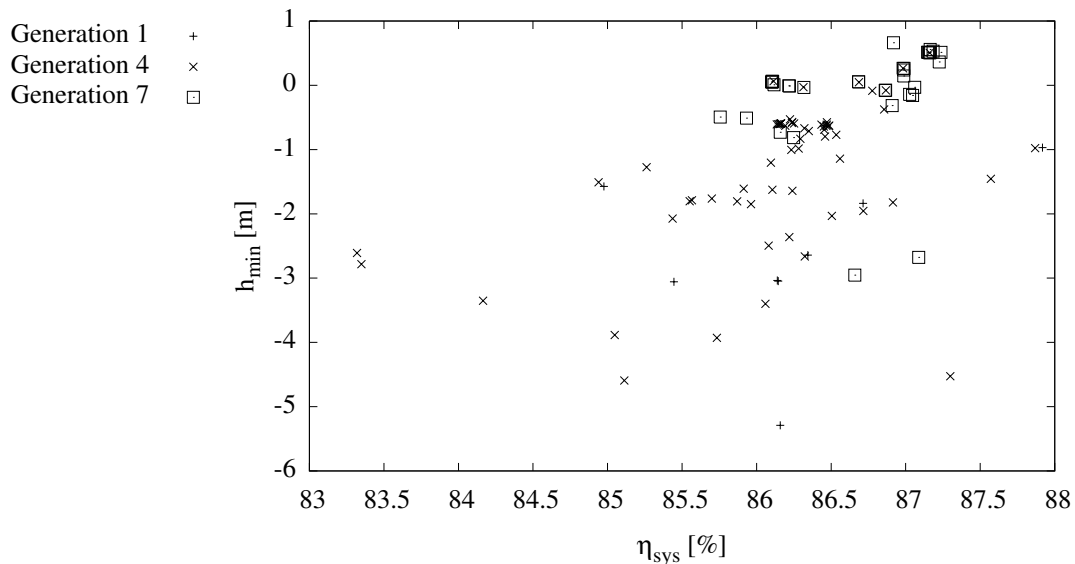
	$Z$ [-]	$H_t$ [m]	$\eta_{La}$ [-]	$c_{p,\text{min}}$ [-]	$\text{NPSH}_i$ [m]
Ausgangsdaten	10	9.12	0.878	-0.23	5.99
Bestes Individuum	10	13.11	0.928	0.03	1.78

**Tabelle 5.3:** Daten für die Radialpumpe  $n_q = 70 \text{ min}^{-1}$

Die Berechnung eines einzelnen Individuums benötigt ca. 15s, somit ist die erwartete Gesamtlaufzeit des Optimierungslaufs ca. 105min. Diese Laufzeit kann noch durch Parallelisierung verkürzt werden.

Da der Suchraum sehr weit gefaßt ist, werden in der ersten Generation nach 155 Versuchen lediglich 10 Individuen gefunden, die vernetzbar, simulierbar sind und die die Randbedingung für die Förderhöhe erfüllen. Dieser Wert steigt bis zur 3. Generation auf die Sollzahl von ca. 65 Individuen, d.h. erst in der 3. Generation sind ausreichend viele Individuen in der Population vorhanden, die alle Bedingungen erfüllen. In Abbildung 5.8 kann man gut erkennen, dass von der 4. zur 7. Generation die Systemantworten der Population sich zu einem Optimum hin orientieren. Auch in der 7. Generation sind Ausreißer hinsichtlich Wirkungsgrad und Kavitationsneigung zu beobachten, allerdings häufen sich die Systemantworten im Bereich  $\eta_{sys} \approx 87\%$  und  $h_{min} \approx 0.5 \text{ m}$ . Je nach Anforderung kann nun ein Individuum mit besserem Verhalten hinsichtlich der Kavitation oder im Wirkungsgrad aus der Population gewählt werden.

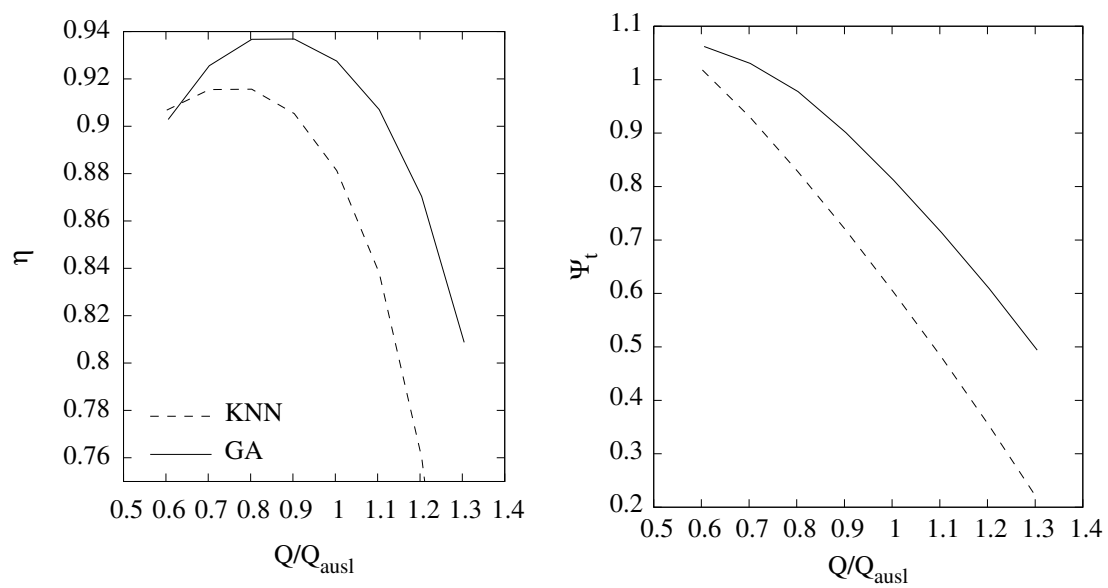
Nach 7 Generationen wird das Individuum mit der höchsten Summen-Systemantwort in einer RANS-Simulation noch einmal genauer berechnet. Der automatische Lauf unterschreitet wie gefordert die Mindest-Haltedruckhöhe  $\text{NPSH}_{\text{erf}}$ . Durch die Optimierung im Auslegungspunkt wird die Kennlinie des Entwurfs deutlich verbessert, s. Abbildung 5.9. Der Optimalpunkt rückt näher an den Auslegungspunkt heran, das Gesamtniveau steigt an. Der Schnittpunkt der Charakteristiken liegt im Teillastbereich im



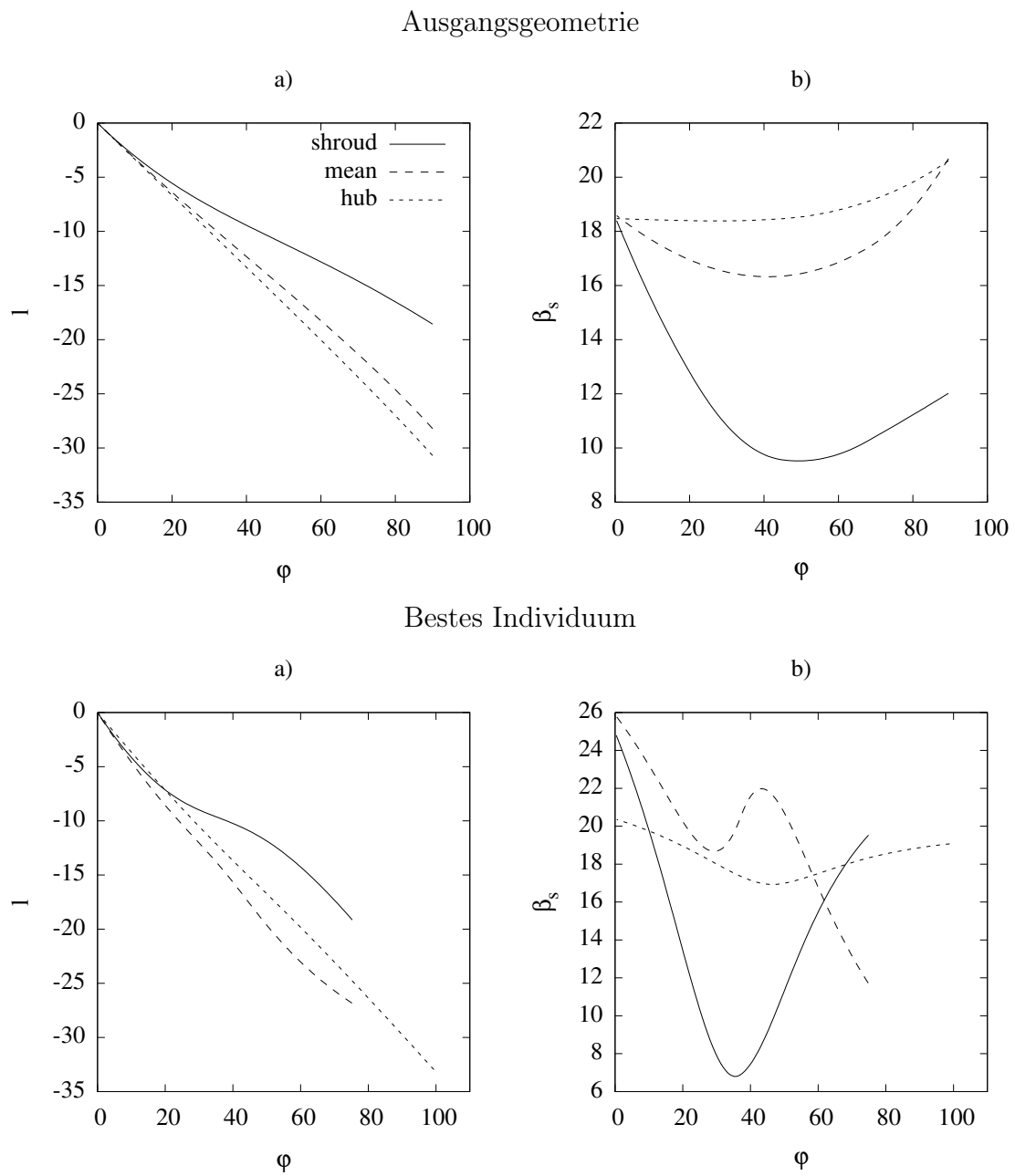
**Abbildung 5.8:** Systemantworten der Generationen 1, 4 und 7 für die Radialpumpe  $n_q = 70 \text{ min}^{-1}$

Bereich um  $q \approx 0.62$ , d.h. erst unterhalb dieses relativen Volumenstroms ist der Wirkungsgrad der alten Geometrie besser, allerdings ist die Förderhöhe der optimierten Geometrie im gesamten betrachteten Bereich oberhalb der Ausgangsgeometrie.

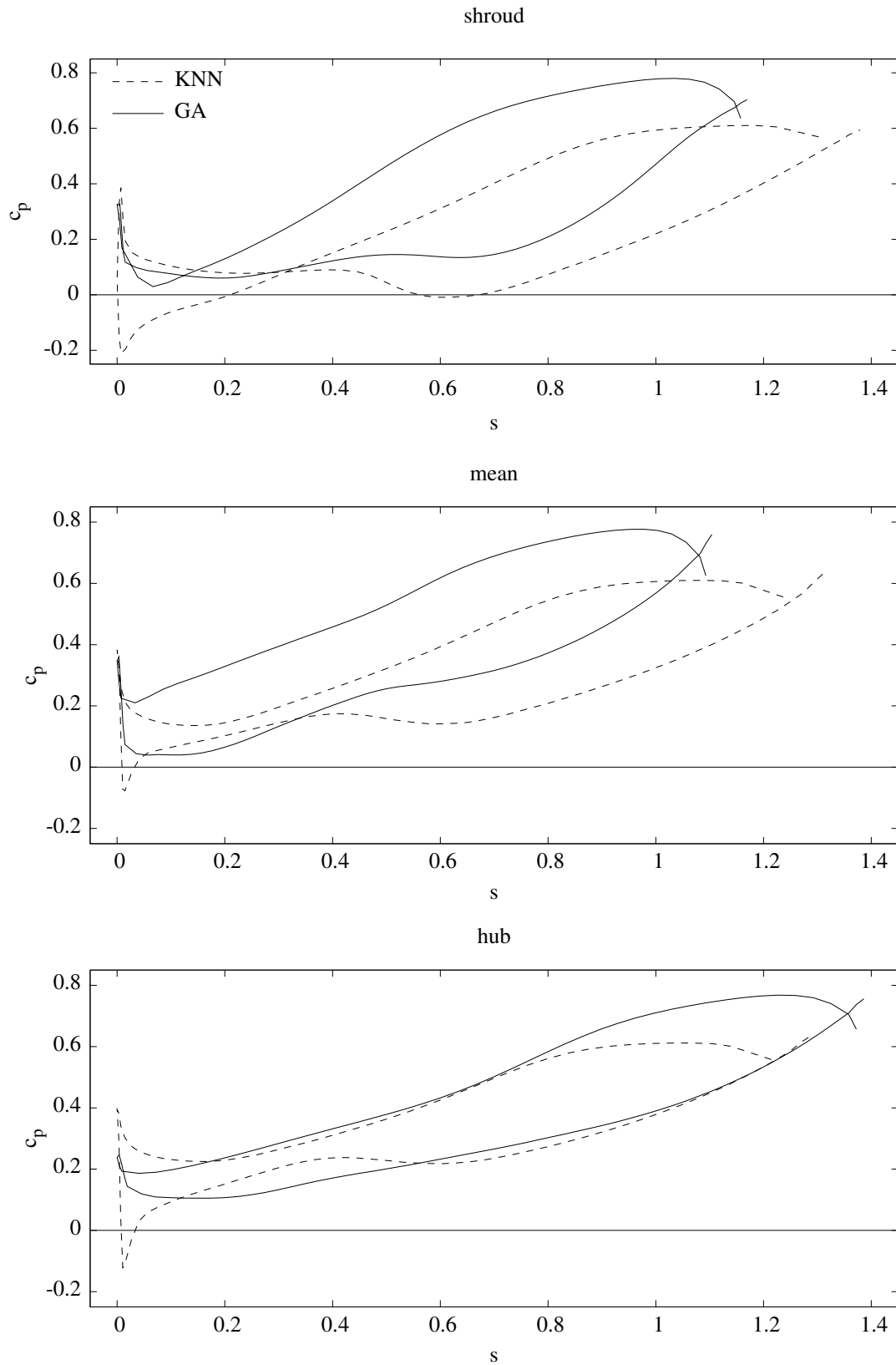
Die Optimierung hat bislang Fragestellungen der Festigkeit ignoriert: Die ermittelte Schaufelgeometrie ist im mittleren Stromfaden und im Shroudbereich deutlich kürzer, s. Abbildung 5.10. Die Druckverläufe sind fülliger, d.h. die lokalen Belastung auf die Schaufel sind ebenfalls größer, s. Abbildung 5.11. Somit bleibt zu prüfen, ob der ermittelte optimierte Entwurf ausreichend dimensioniert ist. Alternativ kann über analytische Kriterien oder über eine mitgeführte Simulation der Struktur die Festigkeit als zusätzliche Randbedingung in die Optimierung einfließen.



**Abbildung 5.9:** Wirkungsgradverlauf und Totaldruckerhöhung der Radialpumpe  $n_q = 70 \text{ min}^{-1}$  für die Ausgangsgeometrie (KNN) und nach vollem Optimierungslauf mit manuellem Eingriff



**Abbildung 5.10:** Optimierung der Radialpumpe  $n_q = 70 \text{ min}^{-1}$ ,  
a) Konforme Abbilder der Stromfäden und  
b) Schaufelwinkerverläufe



**Abbildung 5.11:** Optimierung der Radialpumpe  $n_q = 70 \text{ min}^{-1}$ , Druckverteilungen um die Schaufel  
Ausgangsgeometrie (KNN),  
Optimum des genetischen Algorithmus (GA)

**Radialpumpe**  $n_q = 90 \text{ min}^{-1}$ 

Die Optimierung der Radialpumpe  $n_q = 90 \text{ min}^{-1}$  (RP90) geschieht analog zur RP70, nur dass bei der Ausgangsgeometrie bereits einige Optimierungsmaßnahmen vorgenommen worden sind. Die Ausgangsgeometrie entspricht somit nicht dem Originalzustand aus dem Künstlichen Neuronalen Netzwerk. Als minimale Haltedruckhöhe wird ein Wert von  $\text{NPSH}_{\text{erf}} = 2.83 \text{ m}$  gefordert, dies entspricht einem statischen Druck im Saugmund von  $p_{s,in} = 24525 \text{ Pa} \hat{=} 2.5 \text{ m}$  bei einem Dampfdruck von  $p_{va} = 3800 \text{ Pa}$ .

Die Rechenzeiten bei der RP90 sind höher, da das QNS-Netz in Hub-Shroud-Richtung feiner aufgelöst wird, um eine konstante Zellgröße zu erhalten. Die nicht parallelisierte Laufzeit für die 7 Generationen bei einer Population von 65 Individuen beträgt ca. 6h. Im Verlauf der Optimierung wird die Soll-Populationsgröße etwas früher erreicht als für  $n_q = 70 \text{ min}^{-1}$ , s. Abbildung 5.7. Die Häufung der Werte um den Bereich  $h_{\text{min}} \approx 0 \text{ m}$  und  $\eta \approx 87\%$  deutet darauf hin, dass in den 7 Generationen eine sehr gute Lösung für das gegebene Problem gefunden wird, s. Abbildung 5.12.

	$Z$	$H_t$ [m]	$\eta_{La}$	$c_{p,min}$ [-]	$\text{NPSH}_i$ [m]
Ausgangsdaten	10	8.68	0.875	-0.16	5.40
Bestes Individuum	10	12.1	0.920	0.01	2.60

**Tabelle 5.4:** Daten für die Radialpumpe  $n_q = 90 \text{ min}^{-1}$

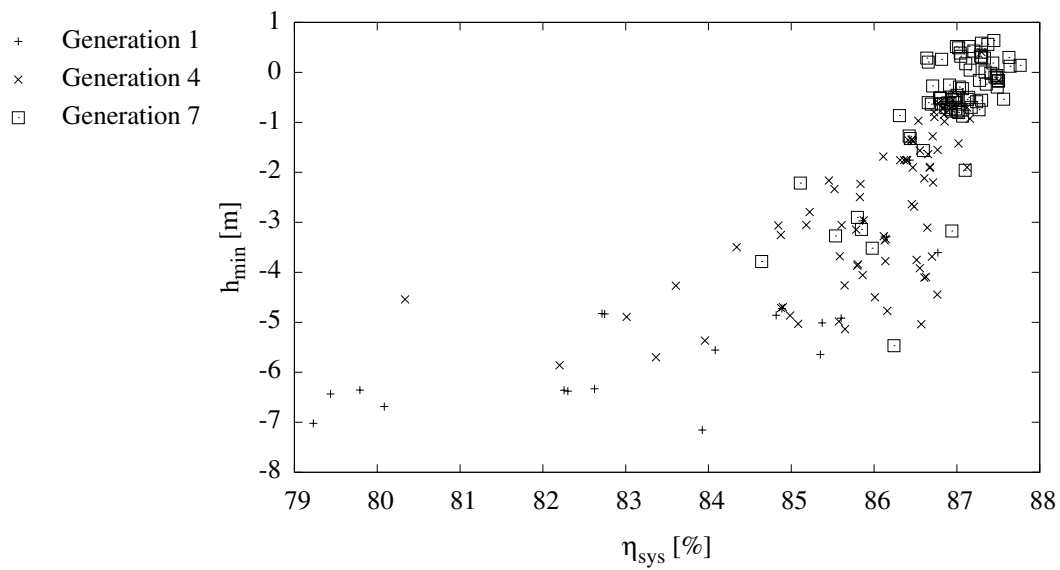
Die erreichte Steigerung des Wirkungsgrads ist vergleichbar mit der Steigerung bei der RP70, die Kavitationsneigung wird ebenfalls in entsprechendem Maße verbessert. Ebenfalls offen bleibt die Frage nach der Festigkeit der neuen Geometrie, die Schaufel der RP90 ist an der Shroudkontur ähnlich verkürzt wie die der RP70, s. Abbildung 5.14, die Druckverläufe sind ähnlich stark geändert, s. Abbildung 5.15.

Die Optimierung im Optimalpunkt ist auch für den Verlauf des Wirkungsgrads über verschiedene Lastbereiche vorteilhaft, s. Abbildung 5.13. Bei diesem Beispiel zeigt sich, dass die Stabilität der Strömung nicht Gegenstand der Optimierung ist. Für  $Q/Q_{\text{ausl}} \leq 0.8$  tritt sowohl bei der manuell angepassten Ausgangsgeometrie wie bei der optimierten Geometrie bereits ein Teillastwirbel auf, der für  $Q/Q_{\text{ausl}} = 0.6$  zu einem scheinbaren Anstieg des Laufrad-Wirkungsgrads führt. Dieses Phänomen ist auf die Lage der Bilanzierungsebenen zurückzuführen und wird in Abschnitt 5.3.1 betrachtet.

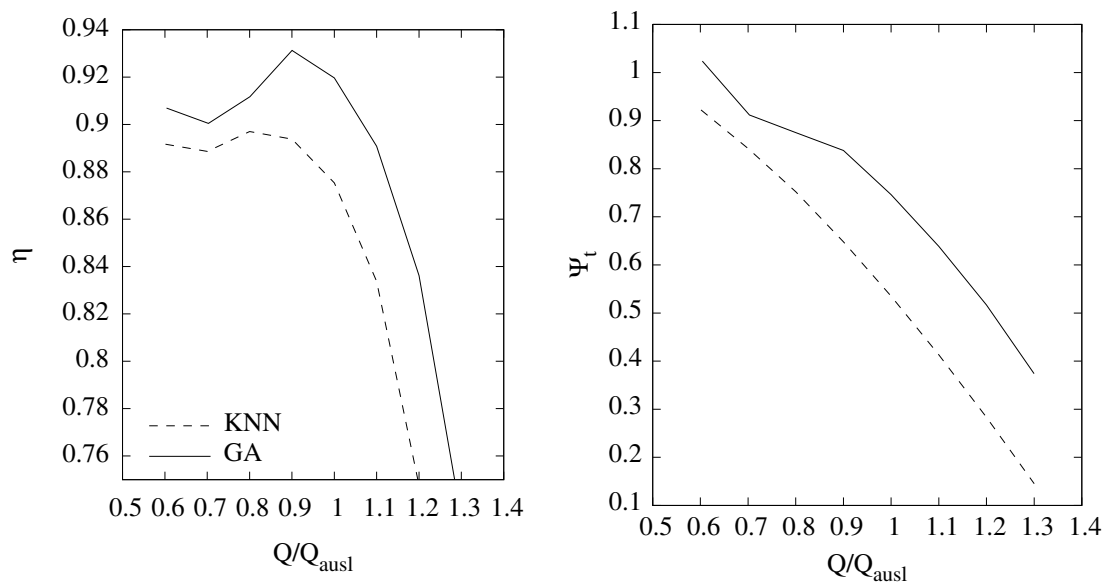
**Bewertung**

Das vorgestellte Verfahren kann ohne Benutzereingriff strömungsoptimale Pumpengeometrien erzeugen. Beide betrachteten Radialpumpen konnten entscheidend hinsichtlich der Zielgrößen verbessert werden. Der rechentechnische Aufwand dieser Optimierung ist begrenzt, die Rechnungen konnten an einem Arbeitsplatz-PC innerhalb eines Arbeitstages durchgeführt werden, ohne dass Potentiale bei der Parallelisierung ausgeschöpft werden mußten. Strömungstechnisch bleibt für beide Radialpumpen noch die Fragestellung, inwiefern eine Einbeziehung der Meridiankontur in die Optimierung eine weitere



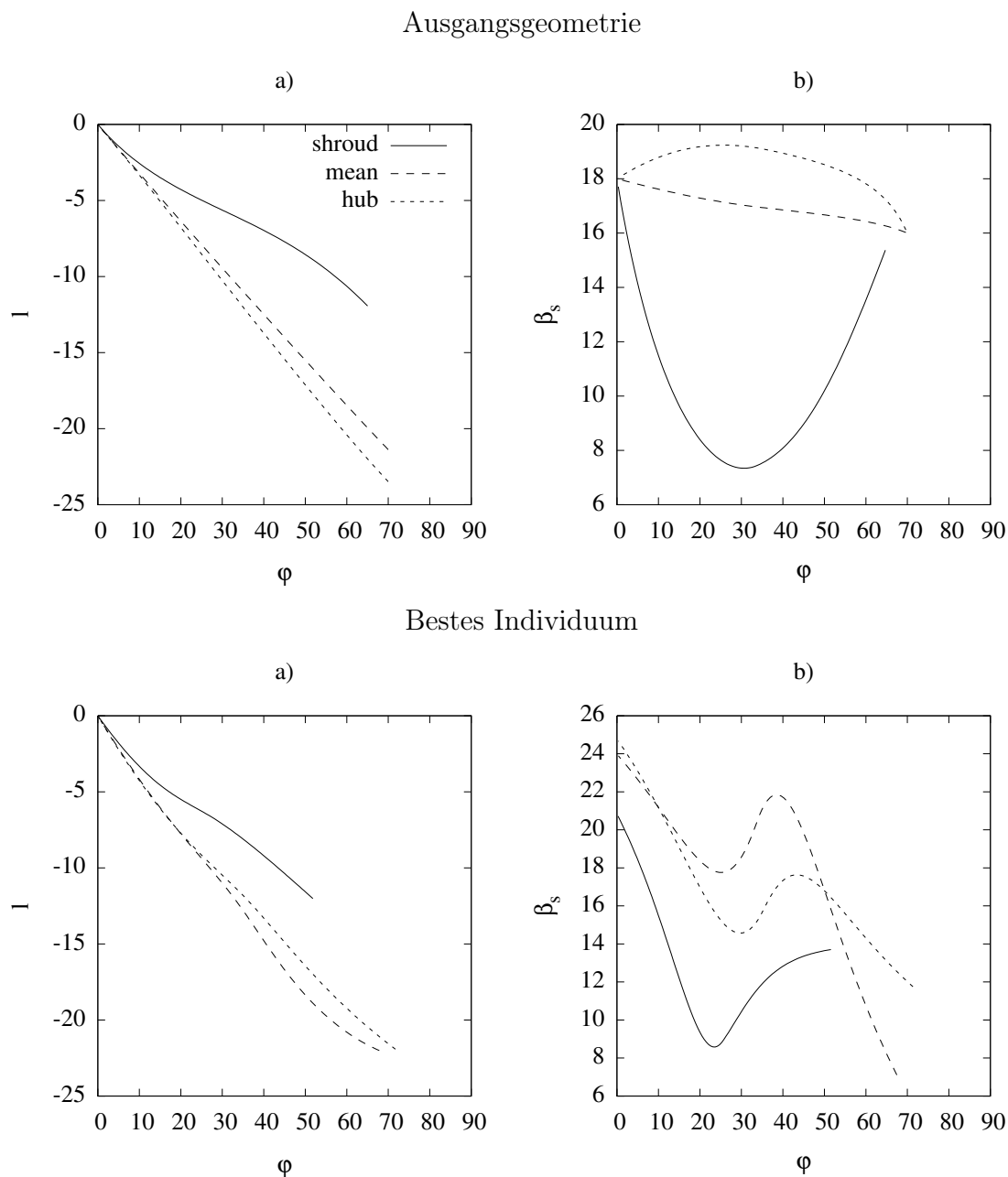


**Abbildung 5.12:** Systemantworten der Generationen 1, 4 und 7 für die Radialpumpe  $n_q = 90 \text{ min}^{-1}$

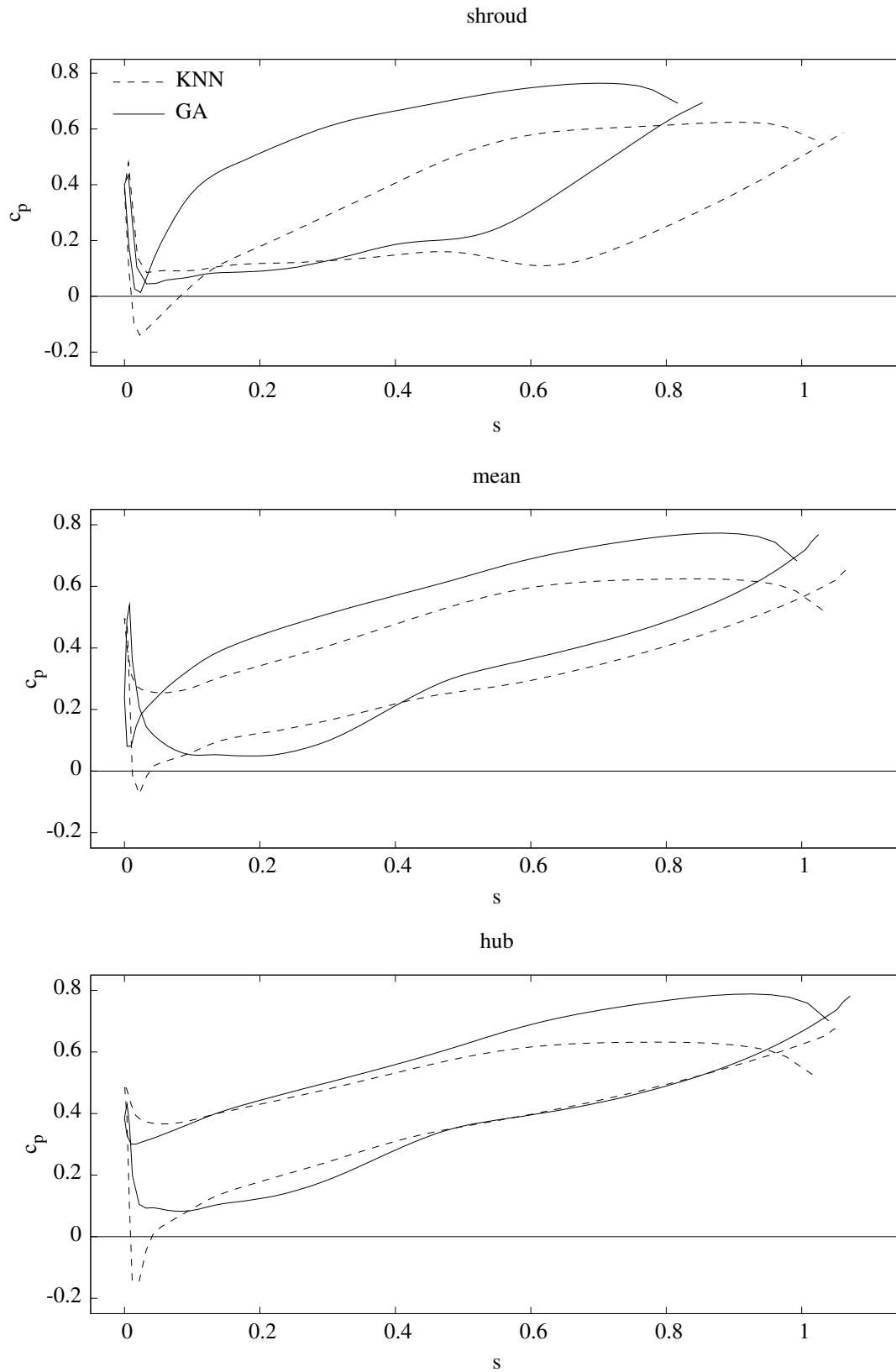


**Abbildung 5.13:** Wirkungsgradverlauf und Totaldruckerhöhung der Radialpumpe  $n_q = 90 \text{ min}^{-1}$  für die Ausgangsgeometrie (KNN) und nach vollem Optimierungslauf

Verbesserungen des Verhaltens der Geometrie bewirken kann. Analytische Festigkeitsbetrachtungen oder einfache strukturelle Simulationen könnten ebenfalls mit einbezogen werden, um weitere Randbedingungen und Zielgrößen abbilden zu können. Mit der Vergrößerung der Anzahl der Modellvariablen und der Randbedingungen geht die Vergrößerung der benötigten Population einher. Die zusätzlich benötigte Rechenkapazität erscheint relativ gering. Das vorgestellte Simulationsverfahren ist vielversprechend, was die tatsächliche Ermittlung eines Optimums angeht, es kann aber bereits in



**Abbildung 5.14:** Optimierung der Radialpumpe  $n_q = 90 \text{ min}^{-1}$ ,  
 a) Konforme Abbilder der Stromfäden und  
 b) Schaufelwinkerverläufe



**Abbildung 5.15:** Optimierung der Radialpumpe  $n_q = 90 \text{ min}^{-1}$ , Druckverteilungen um die Schaufel Ausgangsgeometrie (KNN), Optimum des genetischen Algorithmus (GA)

der vorliegenden, simplen Implementierung dem Hydraulikingenieur wertvolle Hinweise liefern, inwiefern eine vorgegebene initiale Geometrie noch zu verbessern ist.

## 5.3 Axialschub einer Radialpumpe

Nachdem im vorangegangenen Abschnitt das QNS-Verfahren genutzt wurde, um einen schnell konvergierenden, automatischen Optimierungslauf zu implementieren, wird im Folgenden das vorgestellte Programmsystem genutzt, um eine existierende Pumpe nachzurechnen und den zu erwartenden Axialschub zu ermitteln. Die in Abschnitt 2.6 vorgestellten Vernetzungsverfahren wurden genutzt, um eine mit Entlastungsbohrungen versehene Radialpumpe mit einem sechsschaufeligen Laufrad und der Schnellläufigkeit  $n_q = 53 \text{ min}^{-1}$  zu simulieren. Die Pumpengeometrie und die Messdaten wurden von der Firma KSB, Frankenthal, zur Verfügung gestellt. Die Hauptdaten der simulierten Pumpe sind in Tabelle 5.5 aufgelistet. Die Pumpe verfügt über sechs Entlastungsbohrungen, mit einem Durchmesser von je  $d_B = 10.5 \text{ mm}$ , jeweils in Kanalmitte auf einem Durchmesser von  $d = 70 \text{ mm}$  angebracht. Die Spalte im saug- wie im druckseitigen Radseitenraum haben jeweils eine radiale Erstreckung von  $\Delta r = 0.1 \text{ mm}$ .

Anders als in Abschnitt 4.2 gefordert, wurde die vorliegende Pumpe komplett in den Ausführungsdaten simuliert, da die Reynolds-Zahl kleiner ist als die des Normexperiments mit  $D = 340 \text{ mm}$  und  $n = 1000 \text{ min}^{-1}$  und somit die Anforderungen an die Netzfeinheit etwas geringer ausfallen.

D [mm]	n [ $\text{min}^{-1}$ ]	$H_{t,\text{opt}}$ [m]	$Q_{\text{opt}}$ [ $\text{m}^3/\text{s}$ ]	$\Psi_{t,\text{opt}}$ [-]	$\phi_{\text{opt}}$ [-]	Re [-]
224	1299	10.73	0.059	0.906	$9.74 \cdot 10^{-2}$	$1.71 \cdot 10^6$

**Tabelle 5.5:** Entwurfsdaten zur Nachrechnung des Axialschubs der Radialpumpe mit  $n_{q,\text{opt}} = 53 \text{ min}^{-1}$

Die Axialkraft  $F_{ax}$  kann dimensionslos beschrieben werden über den Axialschubbeiwert  $c_{f,ax}$ :

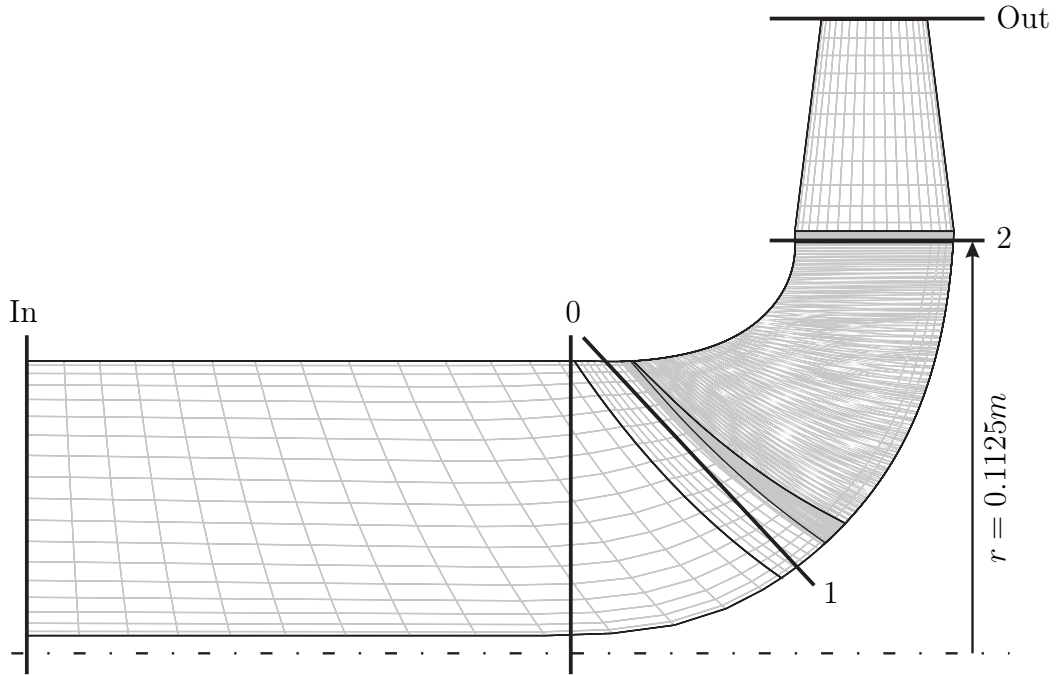
$$c_{f,ax} = \frac{F_{ax}}{p_{\text{ref}} \cdot A_{\text{ref}}} = \frac{8 \cdot F_{ax}}{\pi \rho u_{\text{ref}}^2 D_{\text{ref}}^2} = \frac{32}{\pi} \cdot \frac{F_{ax}}{\rho \omega^2 D_{\text{ref}}^4} \quad (5.14)$$

Aus dieser Definition läßt sich leicht herauslesen, dass die Axialkraft quadratisch mit der Drehzahl wächst und der Laufraddurchmesser das Ansteigen des Axialschubs mit dem Exponenten 4 bestimmt. Der Axialschub ist im Wesentlichen eine Differenz zwischen dem Axialschub infolge der Druckverteilung im saugseitigen Radseitenraum, dem Axialschub infolge der Druckverteilung im druckseitigen Radseitenraum und der Impulskraft am Laufradeintritt. Alternativ kann die Impulskraft am Eintritt ersetzt werden durch die Druckkraft auf die Wände innerhalb des Laufrads. Dies geschieht analog zur Ermittlung des Wellenmoments automatisch an allen rotierenden Wänden.

Anders als z.B. bei WILL ET AL. [55] stehen nur die integralen Werte für den Axialschub zur Verfügung, ein Vergleich der Druckverteilungen im Radseitenraum ist bei der vorliegenden Simulation nicht möglich. Der gemessene Axialschub ist systemdruckunabhängig, d.h. der Druckanteil im Saugmund wurde herausgerechnet. Entsprechend wird das statische Druckniveau im Saugmund im Postprocessing einheitlich auf  $p_s = 0 \text{ Pa}$  gesetzt.

### 5.3.1 Einzelkanalrechnung

Um das Konvergenzverhalten und die Vernetzung der Pumpe sowie die Lage des Optimalpunkts besser beurteilen zu können, wurde zuerst eine Simulationsreihe einer Einzelkanalrechnung ohne Radseitenraum durchgeführt, analog zu Abschnitt 5.1.1. Dabei wurde die etwas gröbere Auflösung gewählt, d.h. der Verzerrungsfaktor  $b$  wurde auf 0.01 limitiert, vgl. Abbildung 4.9. Mit dieser Maßgabe werden 19 Netzebenen in Hub-Shroud-Richtung benötigt, um die Mindestanforderungen an die Wandabstände zu erfüllen.



**Abbildung 5.16:** Meridianansicht des Rechnetzes der Einzelkanalrechnung ohne RSR, Bilanzierungsebenen

Die Einzelkanalsimulation ohne RSR wurde mit einem reinen UDS gerechnet, da lediglich die Lage des Optimalpunkts und die zu erwartenden  $y^+$ -Werte von Interesse waren. Das genutzte Rechnetz weist insgesamt 37 548 Zellen auf und ist wie die Bilanzierungsebenen für die Ermittlung von  $\Psi_t$ ,  $\Psi_{t,th}$ ,  $\eta_h$  und der Leistung  $L$  in Abbildung 5.16 dargestellt.

Es wurden folgende Bilanzierungen verwendet:

$$\Psi_{t,th} = \Psi_{rcu,2} - \Psi_{rcu,1} \quad (5.15)$$

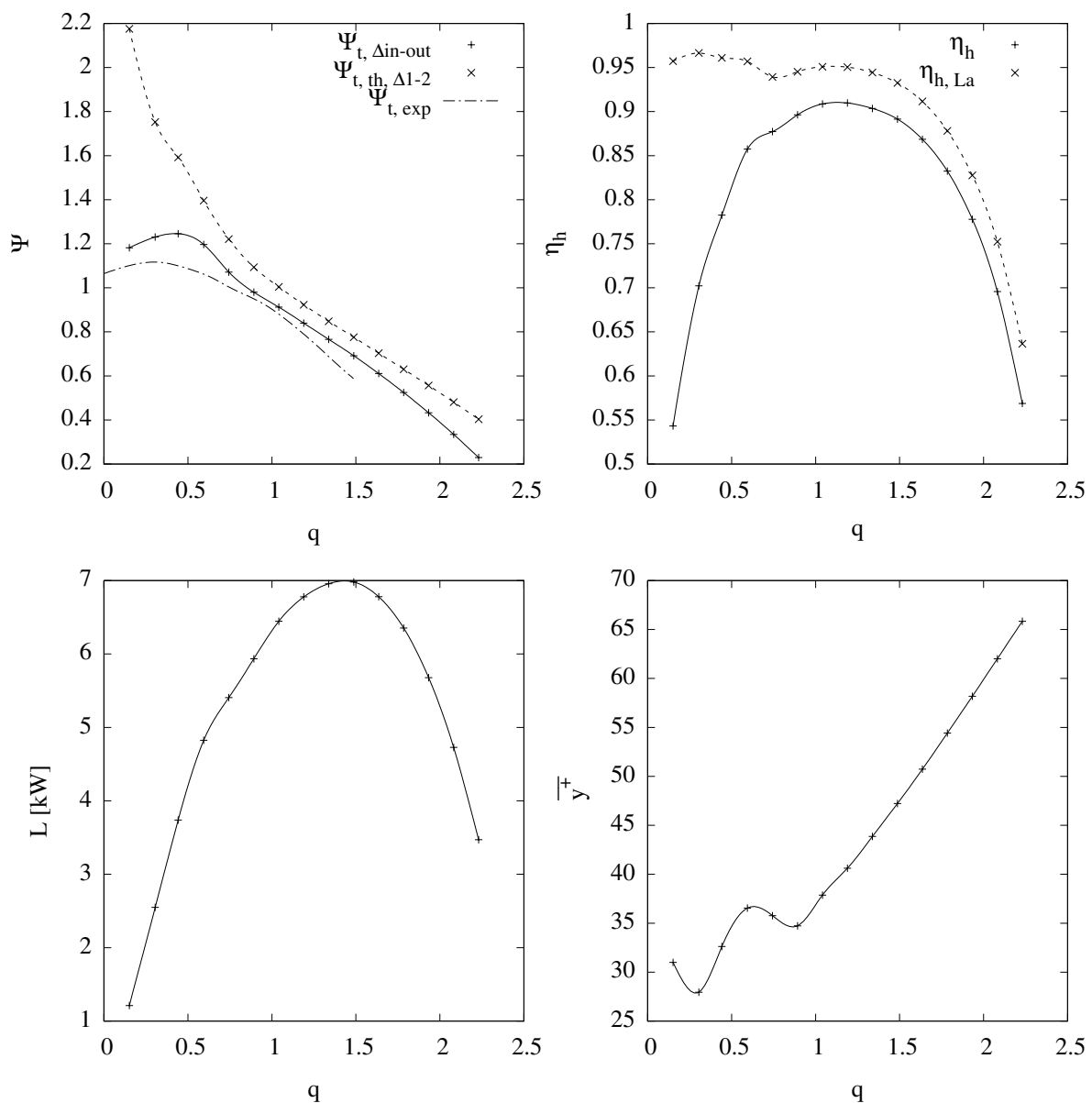
$$\Psi_t = P_{t,out} - P_{t,in} \quad (5.16)$$

$$\eta_{h,La} = \frac{P_{t,2} - P_{t,1}}{\Psi_{t,th}} \quad (5.17)$$

$$\eta_h = \frac{\Psi_t}{\Psi_{t,th}} \quad (5.18)$$

Die ermittelten Charakteristiken der Einzelkanalrechnung sind in Abbildung 5.17 abgebildet. Die  $\overline{y}^+$ -Werte entsprechen den Anforderungen an das Pumpennetz. Der Verlauf der Totaldruckerhöhung liegt deutlich über den Messwerten, da weder das Spiralgehäuse noch die Radseitenräume in der Simulation berücksichtigt wurden. Im Bereich des Auslegungspunkts der Radialpumpe ist der Abstand der in der Einzelkanalrechnung ermittelten und der gemessenen Totaldruckerhöhung am geringsten, was plausibel erscheint, da sowohl Spiralgehäuse wie auch die Radseitenraumgestaltung für diesen Betriebspunkt angepaßt sind.

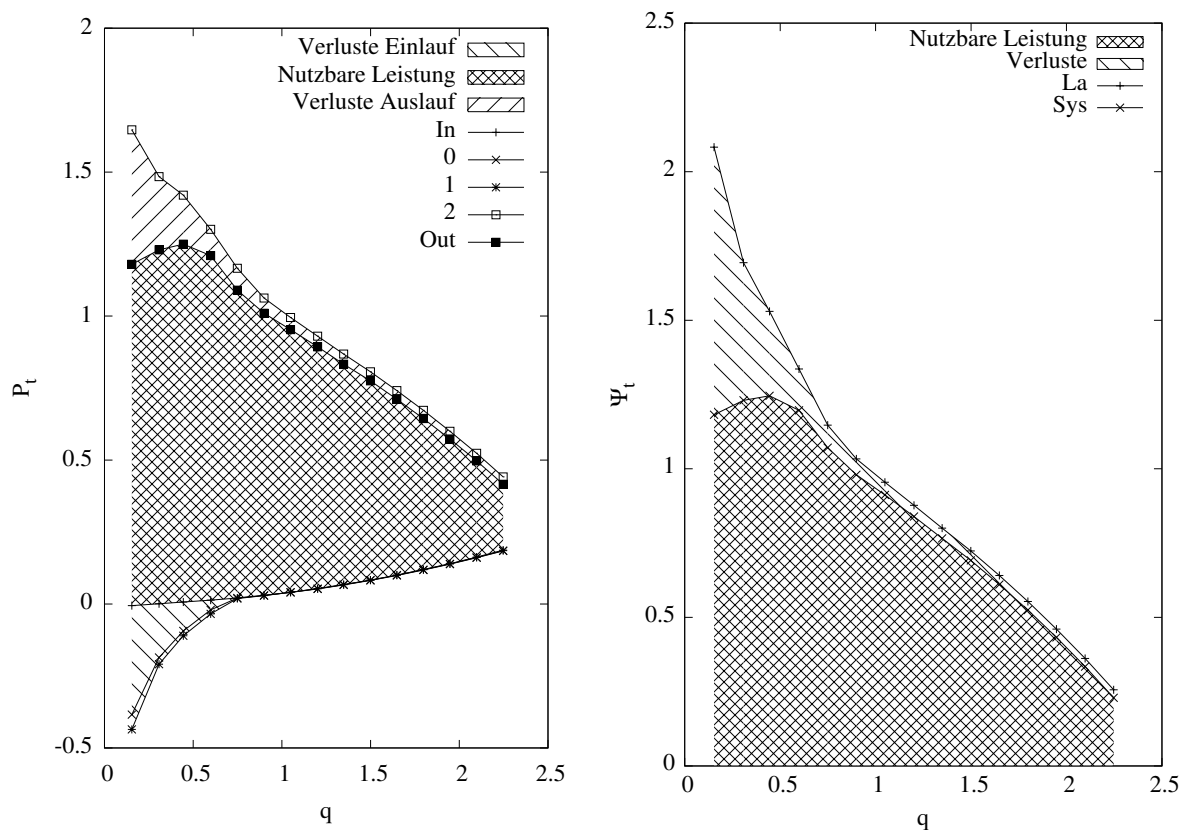
Das Wirkungsgradoptimum wird für die Einzelkanalrechnung ohne Radseitenraum bei  $\phi \approx 0.115$  ermittelt. Der Wirkungsgrad des Laufrads  $\eta_{h,La}$  steigt für  $\phi \leq 0.073$  wieder an, was auf die Lage der einlaßseitigen Bilanzierungsebene zurückzuführen ist. In Abbildung 5.19 ist für die Bilanzierungsebene 0 der Drall  $\Psi_{rcu}$  dargestellt, das Einset-



**Abbildung 5.17:** Kennlinien der Einzelkanalrechnung ohne RSR für die Radialpumpe mit  $n_q = 53 \text{ min}^{-1}$

zen des Teillastwirbels ist deutlich zu erkennen, der in Richtung  $\phi = 0$  stark ansteigt. Der Teillastwirbel trägt den Impulsstrom aus dem Kontrollraum  $\Delta 1 - 2$  stromaufwärts und senkt den Totaldruck auf der Strecke In - 0 stark ab, s. Abbildung 5.18. Der durch den Teillastwirbel erzeugte stromaufgerichtete Impuls- bzw. Energiestrom geht der Förderung verloren und kann somit als Verlust bewertet werden. Der Anstieg des Laufradwirkungsgrads  $\eta_{h,La}$  ist darauf zurückzuführen, dass die Verluste, die stromauf vom Laufrad erzeugt werden, durch die richtungsgebundene Impulsmittelung der Werte als erzeugte Leistung interpretiert werden. Die ermittelten Gesamtwirkungsgrade sind wieder plausibel.

In Abbildung 5.20 wird zur Ergänzung der Unterschied zwischen  $\Psi_{t,th}$  und  $\Psi_{mech}$  exemplarisch aufgeführt. Die Auswirkung auf den Wirkungsgrad ist im Bereich des Teillastwirbels hoch, hier hat die Lage der Bilanzierungsebene für die Ermittlung von  $\Psi_{t,th}$  einen großen Einfluß.



**Abbildung 5.18:** Totaldruckverlauf über dem relativen Volumenstrom  $q$  der Einzelkanalrechnung an den Bilanzierungseben 0, 1 sowie am Einlaß und am Auslaß (links), Totaldruckdifferenzen (rechts)

### Vernetzung für Axialschubberechnung

Anders als bei der Einzelkanalrechnung soll in den Spalten und in der Bohrung die Geschwindigkeitsverteilung in Wandnähe und somit die wirkende Schubspannung vollständig aufgelöst werden, die Modellierung der Wandfunktion soll in diesen Bereichen wegfallen. Wie in Abschnitt 2.1.3 gefordert muß in diesen Bereichen  $y^+ \leq 5$



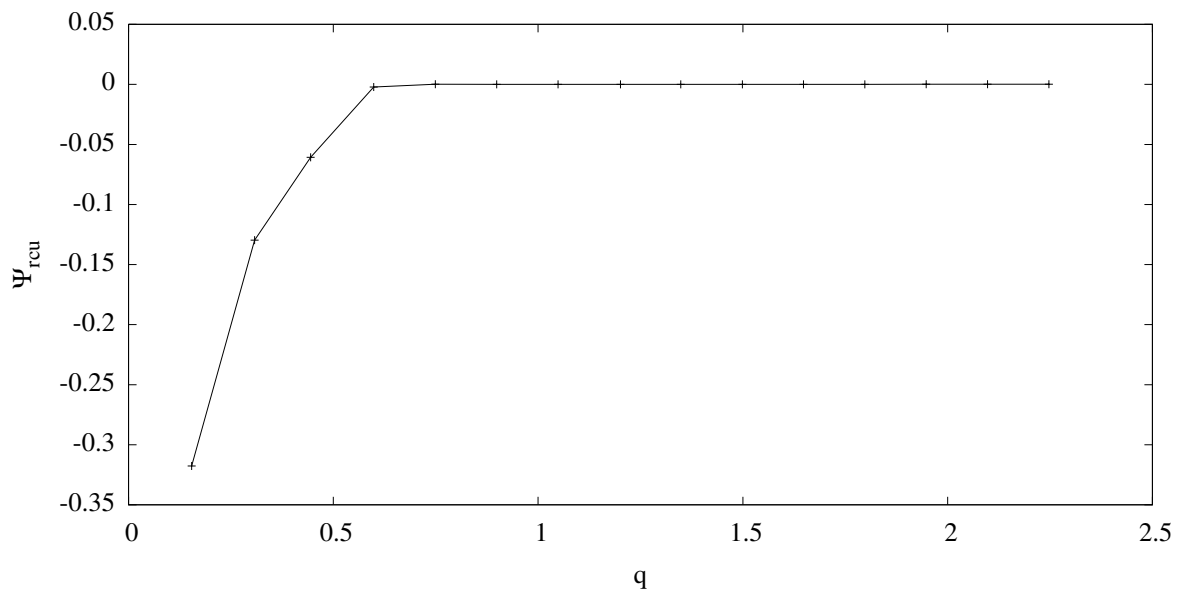


Abbildung 5.19: Dimensionsloser Drall  $\Psi_{rcu}$  an der Bilanzierungsebene 0 in Abhängigkeit vom relativen Volumenstrom  $q$

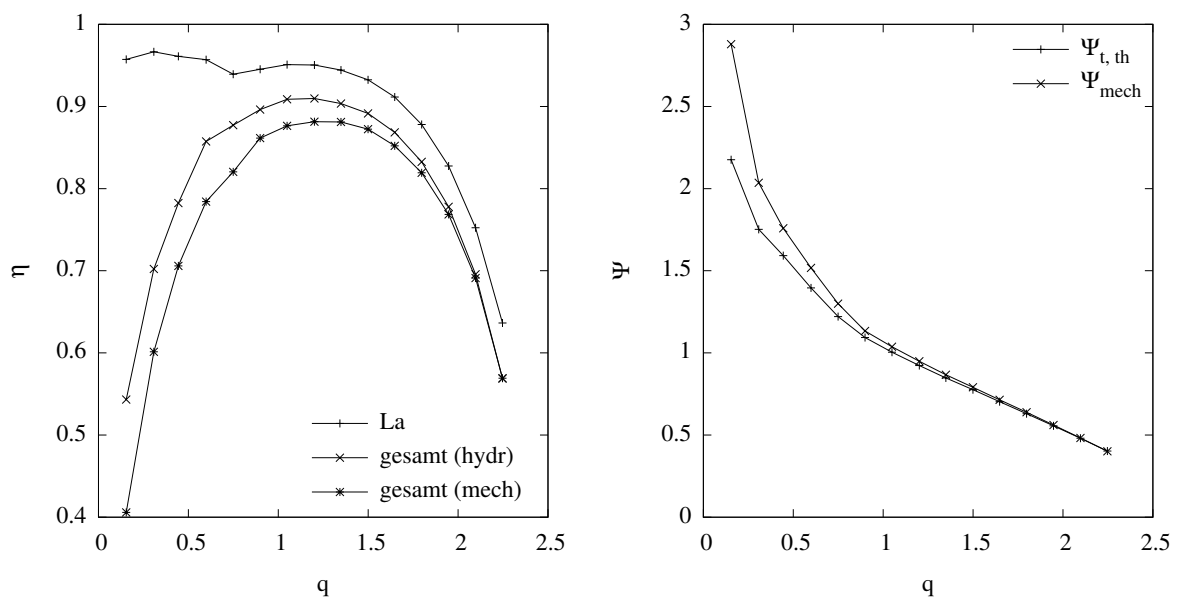


Abbildung 5.20: Ermittelte Wirkungsgrade der Einzelkanalrechnung (links), Unterschied zwischen  $\Psi_{mech}$  und  $\Psi_{t, th}$  (rechts)

gelten, wodurch die Anzahl notwendiger Zellen senkrecht zur Wand recht groß wird. Um die Simulation insgesamt handhabbar zu gestalten, soll an den restlichen Wänden die Wandschubspannung weiterhin über das Universelle Logarithmische Gesetz modelliert werden. Als Turbulenzmodell wurde ein nichtlineares  $k - \varepsilon$ -Modell nach LIEN AT AL. [37] (LCL) in der Implementierung von SKODA [50] gewählt, da es anders als das Standard- $k - \varepsilon$ -Modell den Einfluß der Strombahnkrümmungen berücksichtigt. Zusätzlich konnten EINZINGER ET. AL. [20] bereits feststellen, dass das LCL-Modell für die Ermittlung der Durchströmung der Spalte gut geeignet ist.

Der Abstand zwischen dem saugseitigen Spalt zum Radseitenraum und der Eintrittskante der Beschaukelung ist recht gering, so dass die bislang verwendete Dicke des C-Netzes um die Schaufel von  $0.02 \cdot r_{ref}$  nicht eingehalten werden kann, s. Abbildung 5.21. Der Schaufelbereich mußte mit einem deutlich dünneren und feineren C-Netz versehen werden, so dass die Forderung der Wandfunktion an die  $y^+$ -Werte nicht erfüllt werden konnte. Somit wurde im C-Netz ebenfalls auf die Wandfunktion verzichtet und das Netz so fein gestaltet, dass es das Geschwindigkeitsprofil komplett auflösen kann.

Die Definition des Netzes geschieht hierarchisch wie in Abschnitt 2.6.2 diskutiert. Die folgenden Schritte werden durchlaufen:

1. Definition der Hauptstruktur des Netzes inkl. aller Randbedingungen. Es werden alle rotationssymmetrischen Blöcke und der Schaufelblock definiert, mit Interfaces und Randbedingungen versehen und schließlich mit Netzdimensionen belegt.
2. Verfeinern des druckseitigen Radseitenraum-Blocks, so dass der Anschluß für die Bohrung feiner aufgelöst wird als der restliche Radseitenraum. Dies geschieht in insgesamt vier Schritten für die Umfangsrichtung und die radiale Richtung
3. Hinzufügen der Bohrung zwischen dem Schaufelkanal und dem verfeinerten RSR-Block

Für den ermittelten Optimalpunkt wurde eine Netzstudie durchgeführt, in der der Radseitenraum und die Entlastungsbohrungen in der Strömungssimulation berücksichtigt wurden. Dabei lag das Augenmerk neben der notwendigen Netzauflösung im Radseitenraum, den Spalten und in der Entlastungsbohrung auf der notwendigen Anzahl von Netzebenen in Hub-Shroud-Richtung. Auswahlkriterium war hier die möglichst gute Annäherung an den ermittelten Axialschub für den gewählten Betriebspunkt. Der Axialschub wird nach den Maßgaben des Abschnitts 2.7.2 ermittelt, d.h. über die in axialer Richtung wirkende Druck- und Reibkraft auf die finiten Oberflächenelemente des Impellers.

Die ermittelten Axialschübe aus Tabelle 5.6 zeigen mit zunehmender Zahl von Hub-Shroud-Netzebenen  $N_k$  eine stetige Verringerung der Differenz zwischen der experimentellen Messung und der CFD-Simulation. Es konnte noch beobachtet werden, dass durch die Verfeinerung des Netzes die Unterschiede in der Netzauflösung in den verschiedenen Indexrichtungen etwas abgemildert werden. Diese Verringerung der Expansionsfaktoren stabilisiert das Konvergenzverhalten tendenziell.

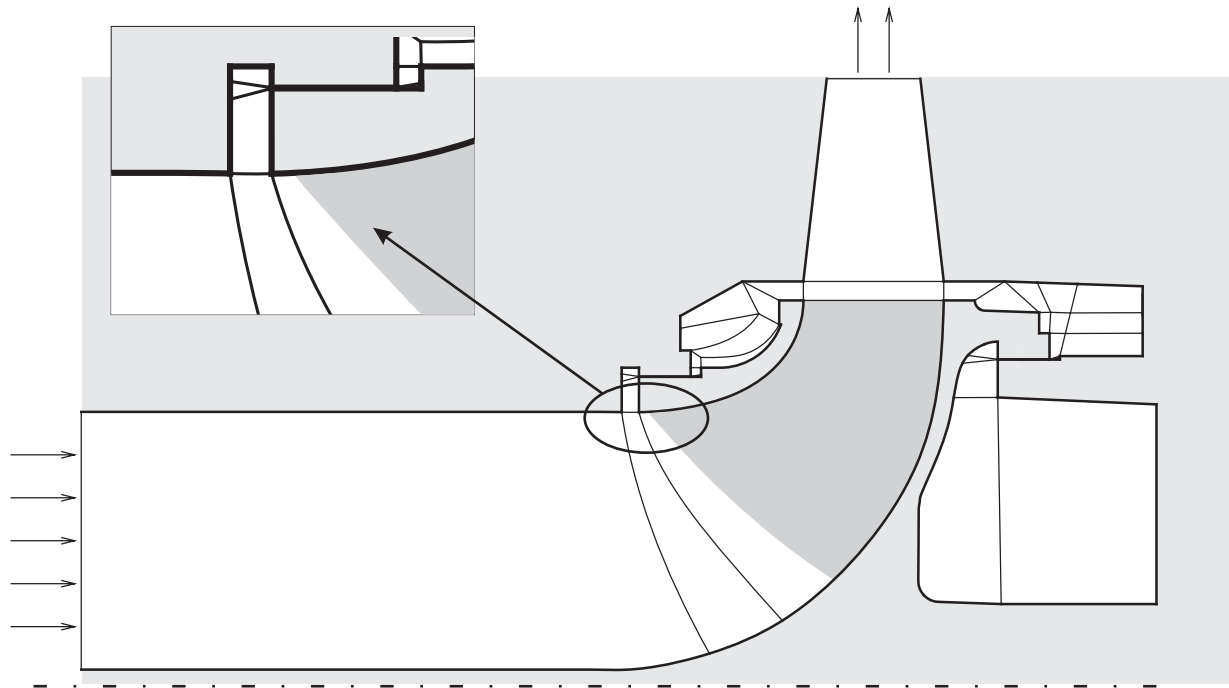


Abbildung 5.21: Meridianansicht der Rechengebiete inkl. Radseitenräumen

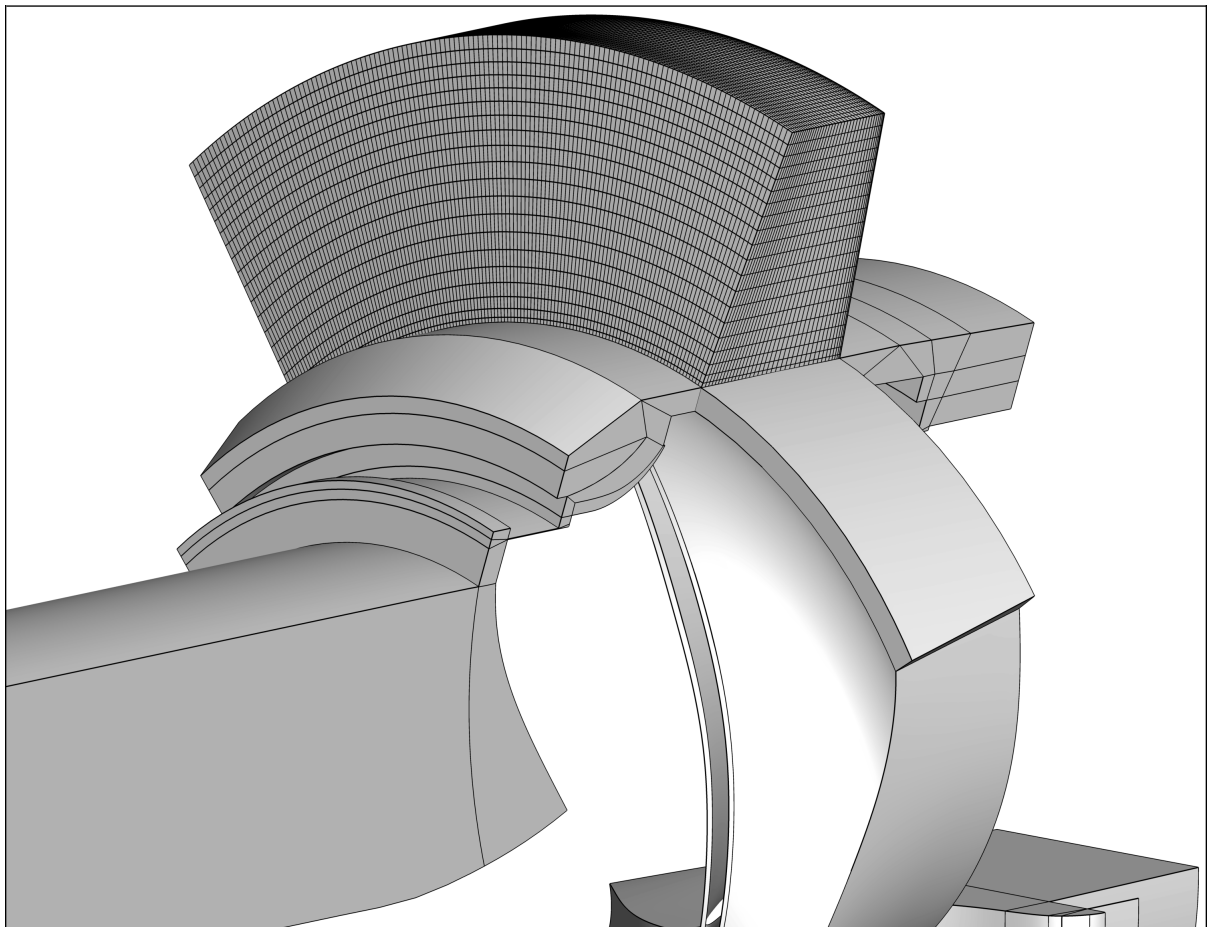


Abbildung 5.22: Laufradauslaß als schmale radiale Düse

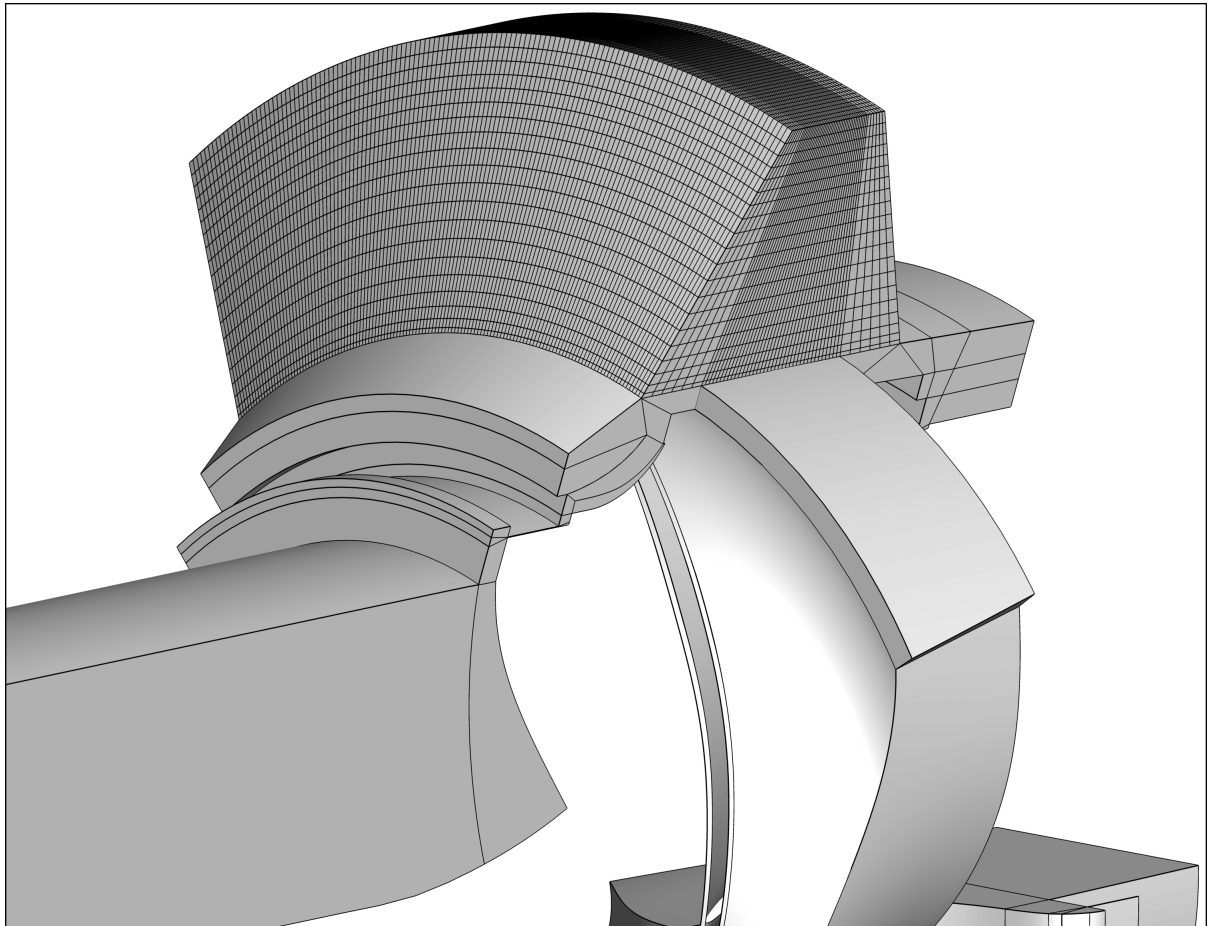


Abbildung 5.23: Laufradauslaß als breite radiale Düse

### Modellierung des Ein- und Auslaßbereichs

Auf Basis der feinsten Hub-Shroud-Auflösung,  $N_k = 39$  wurden Variationen der Auslaßgeometrien simuliert. Dabei wurden drei verschiedene Geometrien betrachtet:

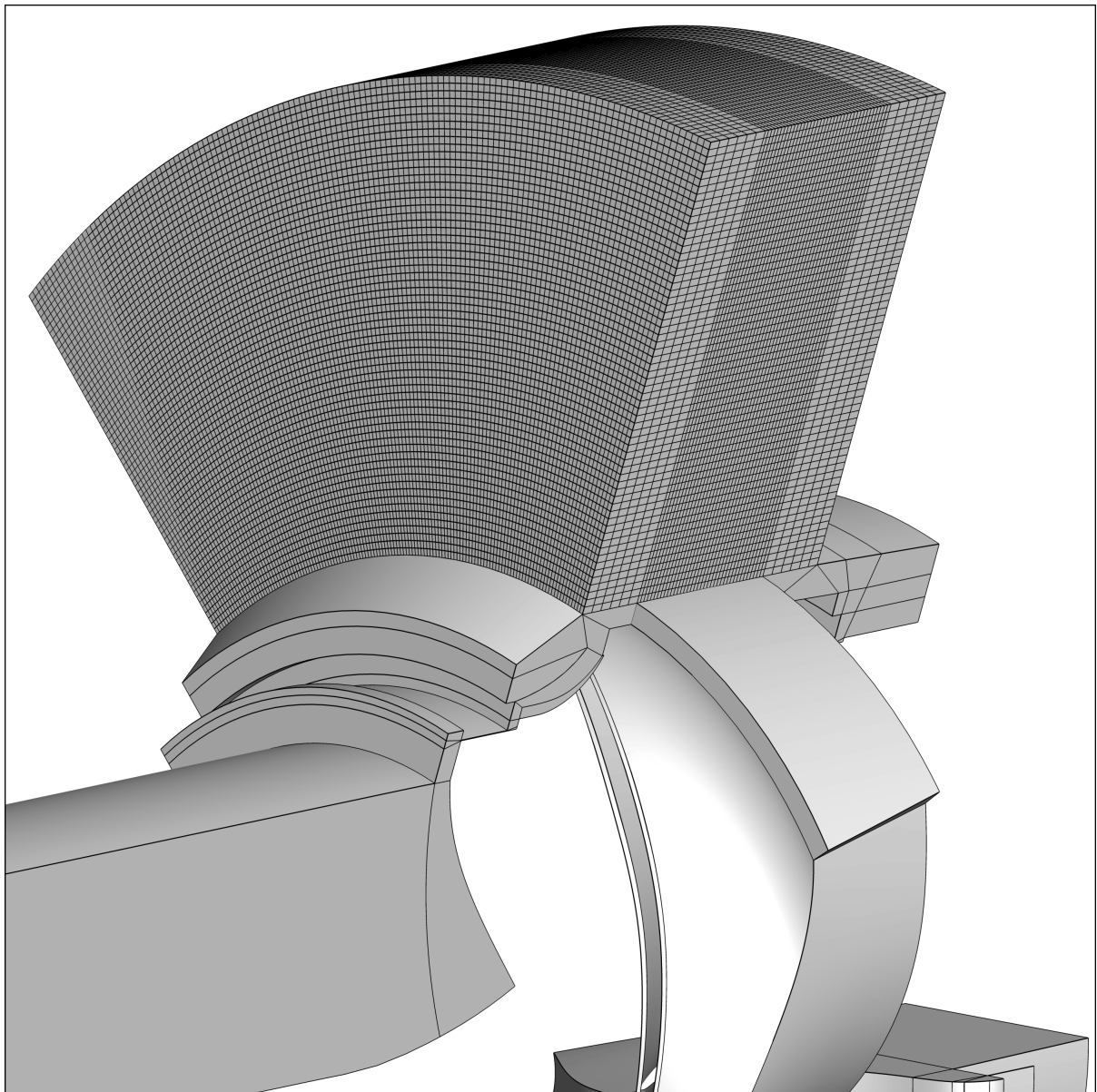
**Auslaßtyp „schmal“:** Der Auslaß wird als radiale Düse modelliert, das Flächenverhältnis ist leicht konvergent. Die anfängliche Breite der Düse entspricht der Breite des Schaufelkanals, s. Abbildung 5.22.

**Auslaßtyp „breit“:** Die radiale Düse wird über die gesamte Breite des Kontaktfläche zum Spiralgehäuse erzeugt. Das Flächenverhältnis ist ebenfalls leicht konvergent, s. Abbildung 5.23.

**Auslaßtyp „Spiralteil“:** Es wird keine radiale Düse modelliert, sondern ein logarithmisches Spiralteil, so dass das Fluid tangential ausströmen kann. Die Anschlußbreite entspricht der Breite der Spirale, s. Abbildung 5.24.

Die Ergebnisse der einzelnen Simulationen sind in Tabelle 5.6 zusammengefaßt. Die Meridianansicht der genutzten Bilanzierungsebenen für die integralen Werte des Laufschaufelkanals, Index  $La$ , sowie des Laufrads, Index  $Imp$ , sind in Abbildung 5.25 dargestellt.

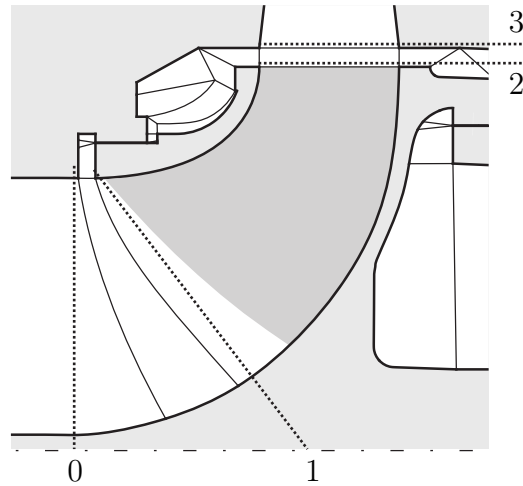
Der Einlaßbereich wurde zuerst mit einer Länge von 157 mm und konstantem Querschnitt simuliert, Einlaßvariante „kurz“. Da der entstehende Teillastwirbel bei dieser Konfiguration bis an die Einlaßfläche des Rechengebiets reicht, wurde ein verlängerter Einlaßbereich von 907 mm und konstantem Querschnitt erzeugt, Einlaßvariante „lang“ .



**Abbildung 5.24:** Laufradauslaß als Spiralteil, kombiniert periodisch/tangentialer Auslaß

MINMOD	0	0.4	0	0	0	0	0	0	0	0	0	0	0
$N_k$	19	19	25	25	31	35	39	39	39	39	39	39	39
Dicke C-Netz [mm]	1.2	1.2	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
Turb	LCL	LCL	LCL	LCL	LCL	LCL	LCL	LCL	LCL	LCL	LCL	LCL	LCL
Auslass	schmal	schmal	schmal	schmal	schmal	schmal	schmal	schmal	schmal	schmal	schmal	breit	Spiralteil
$\Psi_{t,La}$	0.858	0.852	0.840	0.845	0.839	0.843	0.840	0.845	0.843	0.840	0.840	0.831	0.841
$\Psi_{t,th,La}$	0.887	0.884	0.874	0.875	0.873	0.870	0.872	0.875	0.870	0.872	0.872	0.869	0.867
$\eta_{La}$	0.967	0.964	0.961	0.966	0.961	0.969	0.963	0.966	0.969	0.963	0.969	0.956	0.970
$\Psi_{t,Imp}$	0.842	0.839	0.831	0.835	0.830	0.835	0.832	0.835	0.835	0.832	0.833	0.822	0.843
$\Psi_{t,th,Imp}$	0.882	0.881	0.876	0.875	0.875	0.873	0.875	0.875	0.873	0.875	0.872	0.870	0.880
$\eta_{Imp}$	0.949	0.949	0.951	0.954	0.951	0.960	0.954	0.954	0.960	0.954	0.960	0.946	0.958
$H_{t,Imp}$ [m]	9.960	9.925	9.833	9.870	9.822	9.875	9.838	9.870	9.875	9.838	9.850	9.723	9.971
$L_{mech}$ [kW]	7.543	7.416	7.431	7.448	7.467	7.380	7.391	7.448	7.380	7.391	7.380	7.559	7.354
$L_{sys}$ [kW]	6.659	6.645	6.602	6.615	6.603	6.639	6.614	6.615	6.639	6.614	6.570	6.494	6.412
$\eta_{sys}$	0.883	0.896	0.888	0.888	0.884	0.900	0.895	0.888	0.900	0.895	0.890	0.859	0.8719
$H_{t,sys}$ [m]	9.832	9.813	9.732	9.745	9.728	9.780	9.743	9.745	9.780	9.743	9.691	9.585	9.480
$F_{ax,CFD}$ [N]	-64	-44	-49	-63	-61	-84	-86	-63	-84	-86	-58	-59	-112
$F_{ax,exp}$ [N]	-88	-88	-88	-88	-88	-88	-88	-88	-88	-88	-88	-88	-88
$\Delta F$ [N]	-24	-44	-39	-25	-27	-4	-2	-25	-4	-2	-30	-29	24
				n. konv.	n. konv.								

Tabelle 5.6: Integrale Daten der Netzvariationen für die Volumenzahl  $\phi = 0.115$



**Abbildung 5.25:** Bilanzierungsebenen für Impeller (0 – 3) und Laufschaufelkanal (1 – 2) der Radialpumpe

### Konvergenzverhalten

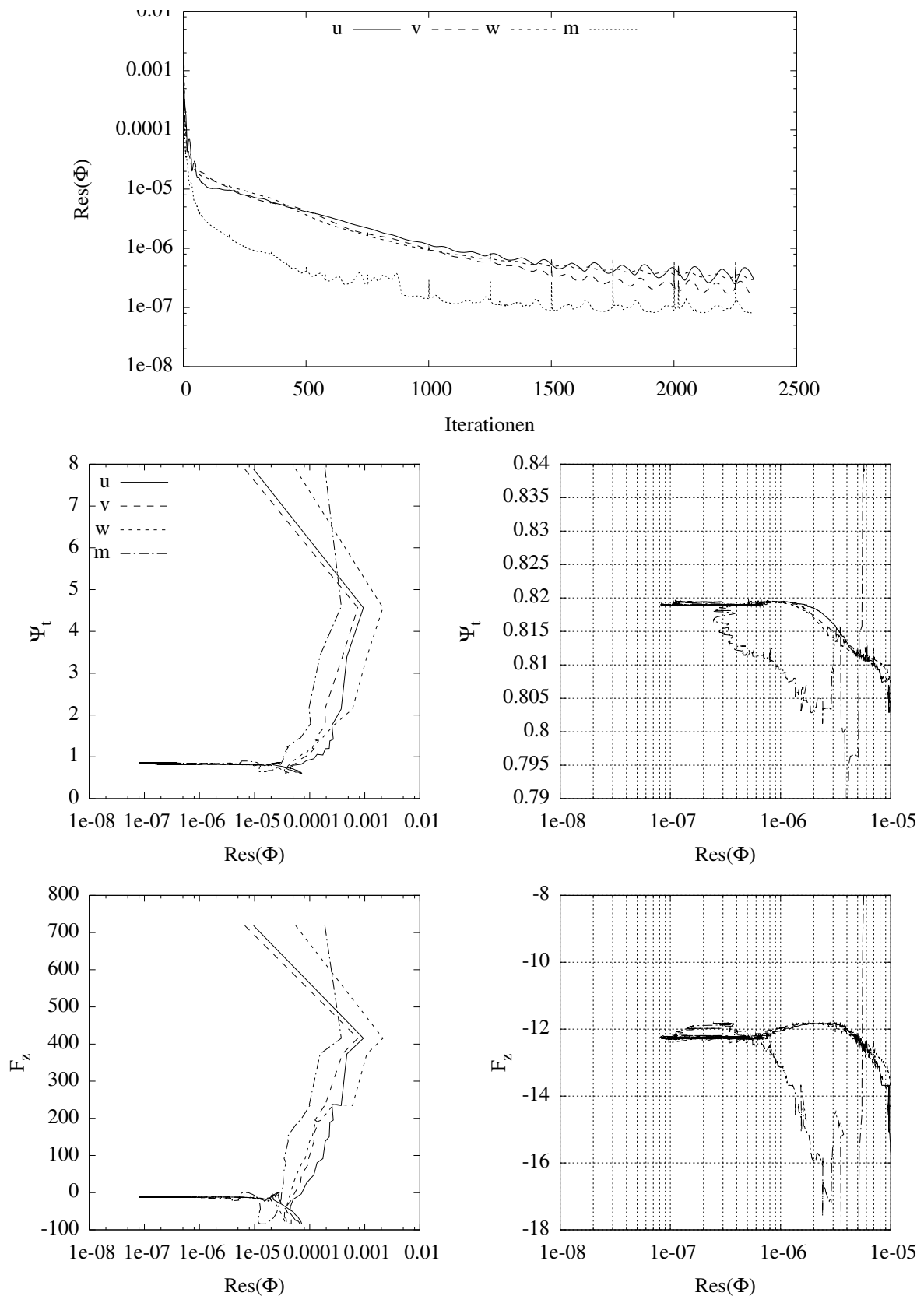
In Abbildung 5.26 ist die Entwicklung der Residuen  $Res$  des Gleichungssystems der einzelnen Erhaltungsgleichungen für die Masse und den Impuls, d.h. für die vier Strömungsgrößen  $u$ ,  $v$ ,  $w$  und  $p$  über die äußeren Iterationen abgedruckt, sowie die Entwicklung der integralen Größen Totaldruckerhöhung  $\Psi_{t,sys}$  und der integrale Axial Schub  $F_z$  auf das Laufrad. Die Residuen starten beim NS3D typischerweise bei  $Res \approx 10^{-2}$ . FERZIGER UND PERIC [21] empfehlen für eine stabile Lösung des CFD-Simulation einen Residuenabfall von insgesamt 4 Größenordnungen. Für die Totaldruckerhöhung ist dies zutreffend, jedoch ist der Axial Schub erst bei  $Res = 10^{-7}$  stabil. Als Kompromiss zwischen Laufzeitverhalten, Stabilität und stabilen integralen Werten für den Axial Schub wird für das maximale Residuum gefordert:

$$Res_{max} \stackrel{!}{\leq} 3 \cdot 10^{-7} . \quad (5.19)$$

Dieses strenge Konvergenzkriterium verlängerte die Rechenzeiten deutlich. Viele Betriebspunkte konnten die normalerweise geforderten 4 Größenordnungen Residuenabfall ohne große Benutzereingriffe erreichen. Das Unterschreiten des maximalen Residuums von  $3 \cdot 10^{-7}$  erforderte in vielen Fällen umfangreiche Benutzereingriffe und starke Unterrelaxation der einzelnen Gleichungen. Das Residuenkriterium wurde nur an die Impulsgrößen sowie an die Massenerhaltung angelegt. Die turbulenten Größen  $k$  und  $\varepsilon$  konvergierten in einigen Fällen nicht unter  $Res \leq 3 \cdot 10^{-7}$ , teilweise blieb insbesondere das Residuum von  $\varepsilon$  bei  $Res(\varepsilon) \approx 10^{-6}$ .

### 5.3.2 Charakteristiken

Die Netzstudie zeigt, dass die Variante „schmal“ bei  $N_k = 39$  die beste Übereinstimmung mit dem gemessenen Axial Schub aufweist. Somit wurde diese Variante als Referenzvariante für die Simulation ausgewählt. Während die Differenzen der integralen



**Abbildung 5.26:** Konvergenzverhalten der Größe  $\Delta P_t$  und  $F_z$  in Abhängigkeit des Residuums  $r$ , linke Spalte: global, rechte Spalte: Teilausschnitt



	$N_k$	yPMax [mm]	MINMOD	Auslaß	Einlaß	$N$
I	15	1.2	-	schmal	kurz	129 593
II	15	1.2	0.4	schmal	kurz	129 593
III	15	1.2	-	breit	kurz	139 626
IV	15	1.2	-	Spiralteil	kurz	190 594
V	25	0.9	-	schmal	kurz	308 598
VI	39	0.9	-	schmal	lang	531 362
VII	39	0.9	-	Spiralteil	lang	451 186

**Tabelle 5.7:** Netzvarianten für die Einzelkanalrechnung

Werte der drei Auslaßtypen in der Größenordnung  $10^{-3}$  variieren, ist die Größenordnung der Abweichung der Axialschubdifferenz die gleiche wie der gemessene Wert.

Neben dieser Referenzvariante werden noch weitere Netzvarianten für die Berechnung der Charakteristiken herangezogen, s. Tabelle 5.7. Die grob vernetzten Varianten I-V sollen Aufschluß geben, inwiefern der Verlauf des Axialschubs über die Betriebspunkte mit etwas größeren Netzen wiedergegeben werden kann. Die Varianten VI stellt die Referenzauflösung der Netzstudie dar, Variante VII dient der zusätzlichen Begutachtung des Auslaßtyps „Spiralteil“.

Die Varianten des Auslaßtyps „schmal“ – I, II, V und VI – konvergieren weitgehend stabil. Im Teillbereich  $\phi \leq 0.08$  sind teilweise sehr starke Unterrelaxationen notwendig, um das geforderte Residuum von  $3 \cdot 10^{-7}$  zu erreichen. Die Variante „breiter Auslaß“ III ist in diesen Fällen sehr instabil. Hier muß auch bei höheren Volumenströmen mit starken Unterrelaxationsfaktoren gearbeitet werden. Die Varianten des Auslaßtyps „Spiralteil“ sind ähnlich instabil: für die Variante VII, Betriebspunkte  $\phi \leq 0.06$  wird das Residuenkriterium gar nicht erfüllt. In Variante IV mußte teilweise die Laufradstellung geändert werden, was die Gesamtkurve tendenziell verfälscht. Das sonst geforderte Residuum von  $10^{-6}$  wird in den meisten Fällen ohne größere Eingriffe erreicht.

Die ermittelten Wirkungsgradverläufe aus Abbildung 5.27 zeigen ein mehr oder weniger ausgeprägtes untypisches Verhalten im Bereich des Optimalpunkts. Für die Fälle I bis IV zeigen die Graphen für  $\eta$  für alle verschiedenen Bilanzierungsweisen – nur Laufschaufelkanal, Impeller oder über das Gesamtsystem – im Bereich von  $0.1 \leq \phi \leq 0.14$  kein ausgeprägtes Maximum, sie schwingen in einem breiten Band von ca.  $\pm 0.5\%$ -Punkten. Erst mit deutlich feinerer Auflösung in Hub-Shroud-Richtung wird das Verhalten in diesem Band stabiler.

Alle Betriebspunkte wurden abgesehen von der Hub-Shroud-Richtung mit einem einheitlichem Rechennetz gerechnet. Somit wurde insbesondere die Auflösung des C-Netzes um die Schaufel beibehalten und nicht an den Betriebspunkt angepaßt. Da hier keine Wandfunktion verwendet wurde, die über einen breiten  $y^+$ -Bereich Gültigkeit besitzt, konnten nicht für jeden Betriebspunkt und jede Zelle ein gültiger  $y^+$ -Wert erzielt werden, wodurch die wirkende Wandschubspannung nicht überall richtig aufgelöst wird. Dies hat wiederum Auswirkung auf die ermittelte Wandreibung und somit auf

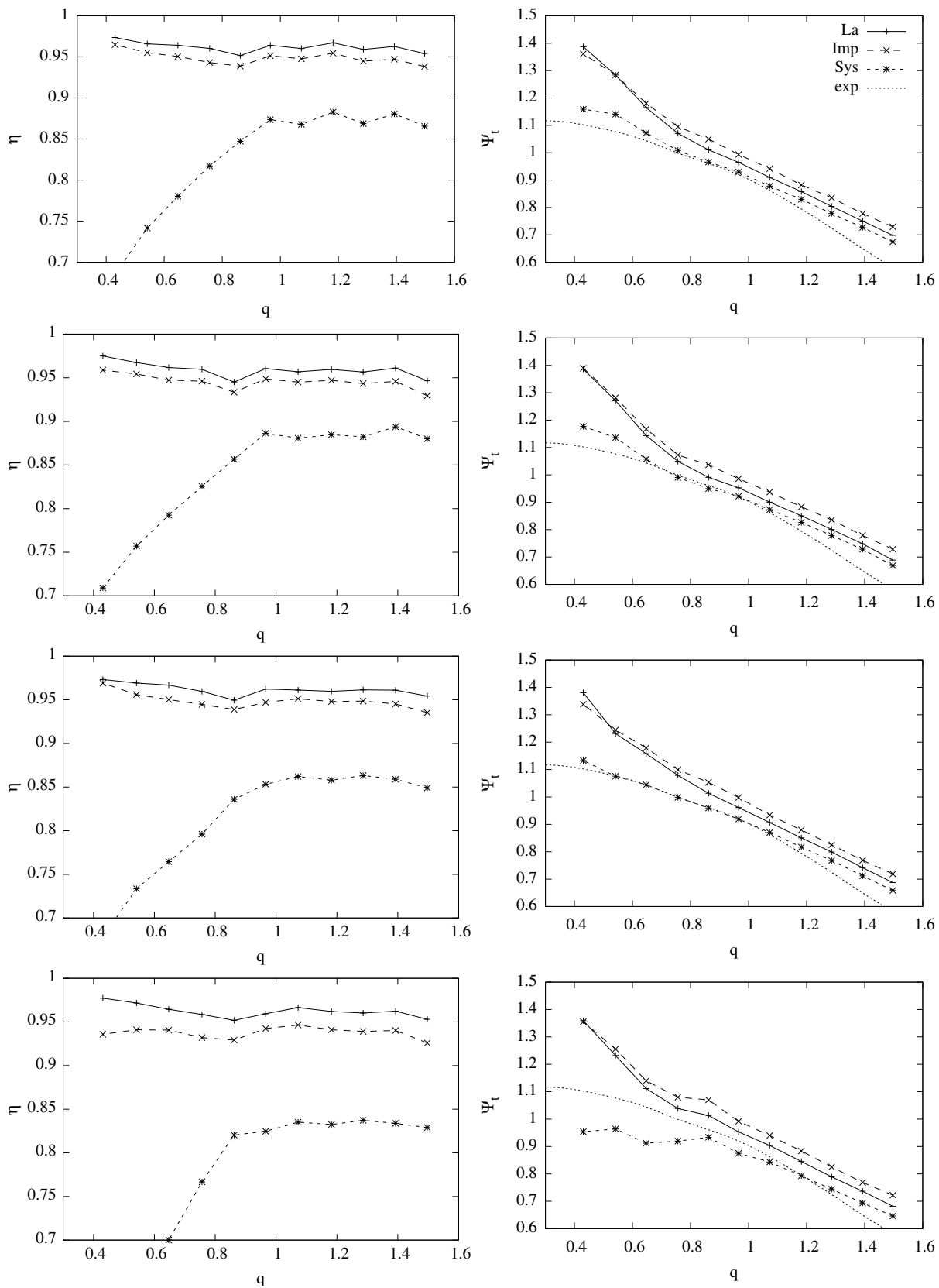


Abbildung 5.27: Integrale Werte der Varianten I, II, III und IV (von oben nach unten)

die ermittelten Verluste. Abhilfe können bei dieser Herangehensweise nur Rechenetze schaffen, die für jeden Betriebspunkt individuell angepaßt werden.

### 5.3.3 Axialkraft

Die ermittelten Axialschubbeiwerte aus Abbildung 5.29 zeigen eine globale Übereinstimmung mit den gemessenen Werten, solange man die Meßtoleranz mit einbezieht. Bezogen auf den Einzelwert sind die Abweichungen in der Größenordnung des Meßwerts. Die Steigung der Charakteristik aus der Simulation entspricht nicht der der Meßkurve. Die ermittelten Axialschübe der grob aufgelösten Varianten I bis IV wei-

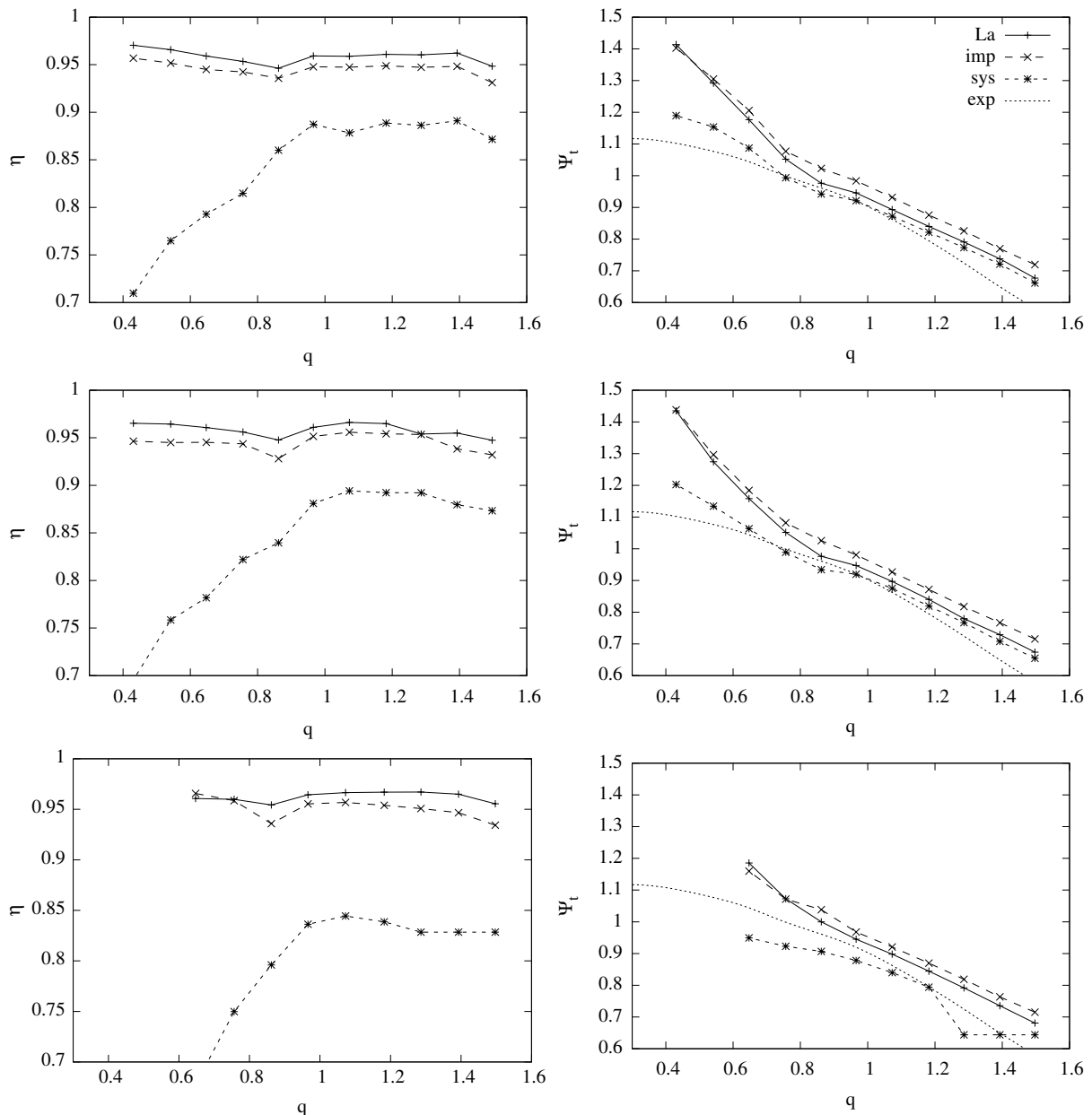


Abbildung 5.28: Integrale Werte der Varianten V, VI und VII (von oben nach unten)

sen große Abweichungen auf, der Ort des betragsmäßigen Schubminimums wird nicht korrekt wiedergegeben. Das Globalverhalten scheint von der Variante V am ehesten wiedergegeben, die nominale Variante VI ist hingegen nur in 3 Betriebspunkten außerhalb der Meßtoleranz und gibt den Axial Schub somit am besten wieder. Die Varianten mit Spiralteil IV und VII spiegeln die Axialschubbeiwerte im Bereich  $\phi > 0.09$  recht gut wider, im Teillastbereich  $\phi < 0.09$  sind die ermittelten Axialschübe – analog zur Instabilität der Simulation – deutlich schlechter.

Ein Einflußfaktor für die Druckverteilung im Radseitenraum und somit den Axial Schub ist der Spaltvolumenstrom  $\phi_{sp}$  durch den Radseitenraum, vgl. SCHILLING[45] und GÜLICH [26]. Der Spaltvolumenstrom durch den druckseitigen Radseitenraum unterliegt in allen Varianten großen Schwankungen, wobei er einen Globalverlauf aufweist, der umgekehrt proportional zu Volumenstrom und proportional zur Totaldruckerhöhung verläuft: Die Druckdifferenz zwischen dem Laufradaustritt und der Bohrung ist die treibende Kraft für die Durchströmung. Der Spaltvolumenstrom durch den saugseitigen Radseitenraum zeigt für Variante VI einen glatten Verlauf. Für  $\phi \leq 0.084$  ist er proportional zum Volumenstrom. Darüberhinaus ist er nach einem starken Anstieg, analog zum Spaltvolumenstrom durch den druckseitigen Radseitenraum, umgekehrt proportional zum Volumenstrom. Diese Umkehrung fällt zusammen mit dem Einsetzen des Teillastwirbels, vgl. Abbildung 5.31. Mit Einsetzen des Teillastwirbels wird die Druckdifferenz zwischen Saugmund und Laufradaustritt geändert: Statt eines tendenziell konstanten Drucks über der gesamten Saugmundfläche stellt sich eine Druckschichtung ein, bei der der statische Druck im Aussenbereich deutlich höher ist als der mittlere Druck auf dieser Bilanzierungsebene, s. Abbildung 5.32. Abhängig von der Intensität des Teillastwirbels wird die Druckdifferenz zwischen Laufradaustritt und dem Radialspalt des saugseitigen Radseitenraums somit verringert. Der Vergleich der Spaltvolumenströme der Varianten VI und VII zeigt eine unterschiedliche Steigung im Teillastbereich. Hier zeigt sich die unterschiedliche Modellierung des Auslasses am Laufradaustritt, der Ort des Spaltvolumenstrommaximums und des maximalen Axialschubs fallen auch hier zusammen.

### Eindeutigkeit der Lösung

Bei der Durchführung der Simulationen konnte beobachtet werden, dass der Axial Schub eine Abhängigkeit hinsichtlich der Reihenfolge der Simulationen aufweist, s. Tabelle 5.8. Die integralen Werte schwanken abhängig von der initialen Lösung. Dabei wurde das Feld mit einem Startwert  $\Phi_i = 0$  und alternativ mit der auskonvergierten Lösung eines benachbarten Betriebspunkts vorbelegt. Die ermittelten integralen Werte streuen in diesem Beispiel im Bereich von  $\delta < 1.0\%$ . Der Axial Schub als Differenz dreier Werte zeigt allerdings eine deutliche größere Streuung: Die ermittelte Differenz ist in der gleichen Größenordnung angesiedelt wie die Differenzen infolge der Änderung des Lösungswegs. Die ermittelten Werte sind jeweils UDS gerechnet. Es ist zu erwarten, dass mit Erhöhung der Diskretisierungsordnung sich die integralen Werte unabhängig von der initialen Belegung stärker annähern, da sie i.d.R. mit der Startlösung einer auskonvergierten UDS-Lösung ermittelt werden.

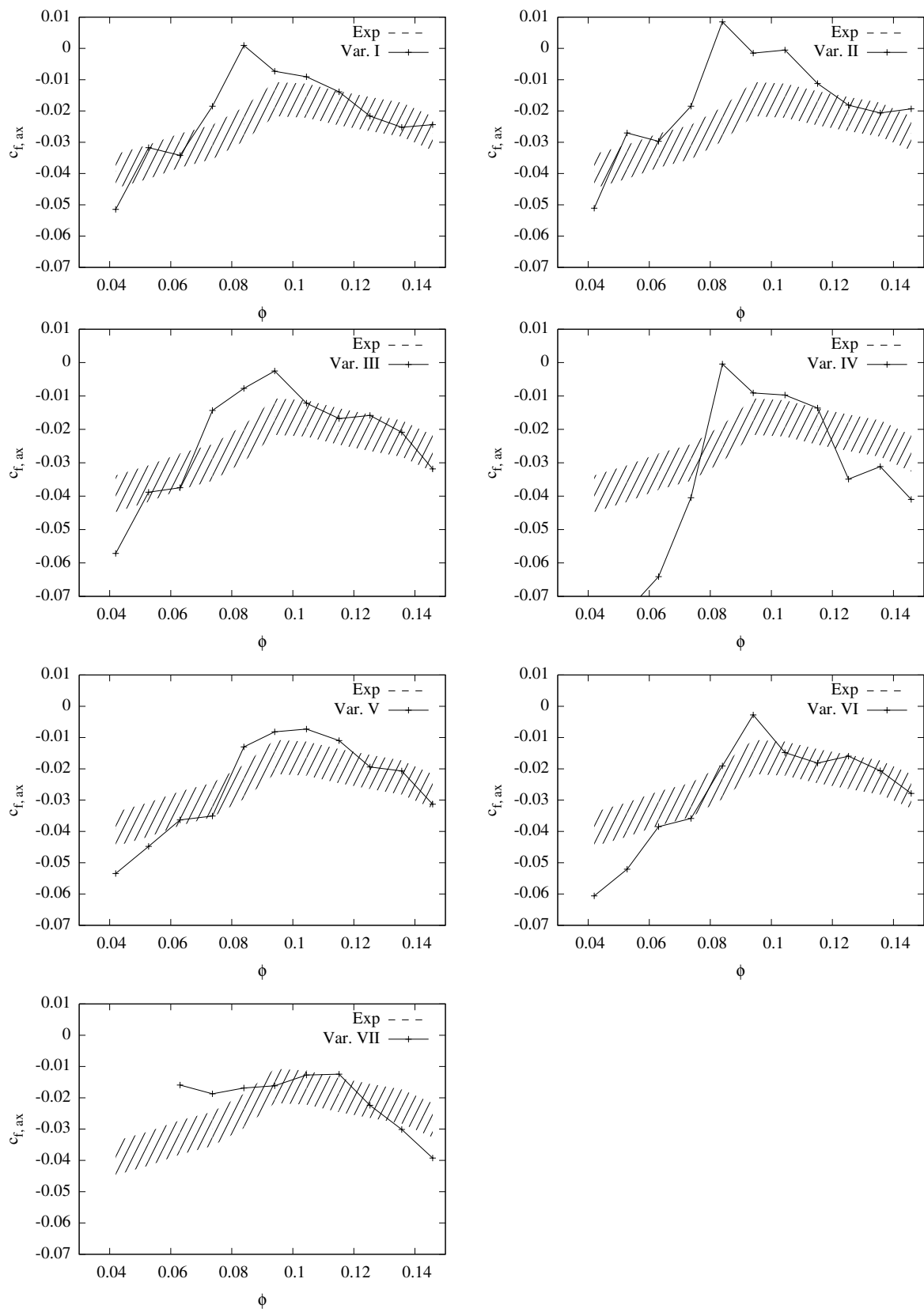
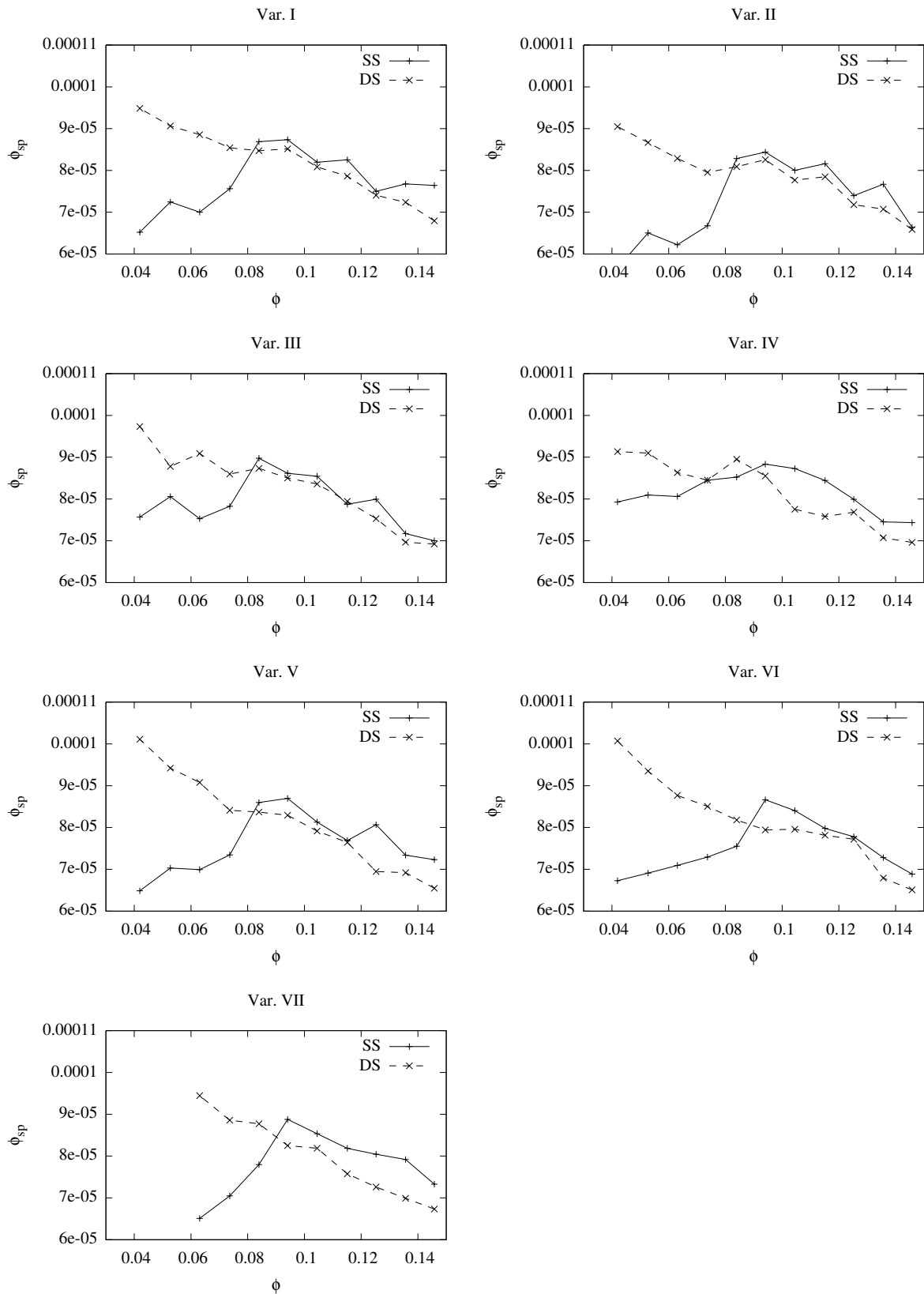


Abbildung 5.29: Axialschubbeiwerte der Einzelkanalrechnungen



**Abbildung 5.30:** Spaltvolumenströme der Radseitenräume, ermittelt am saugseitigen Gehäusespalt (SS) und in der Entlastungsbohrung (DS)

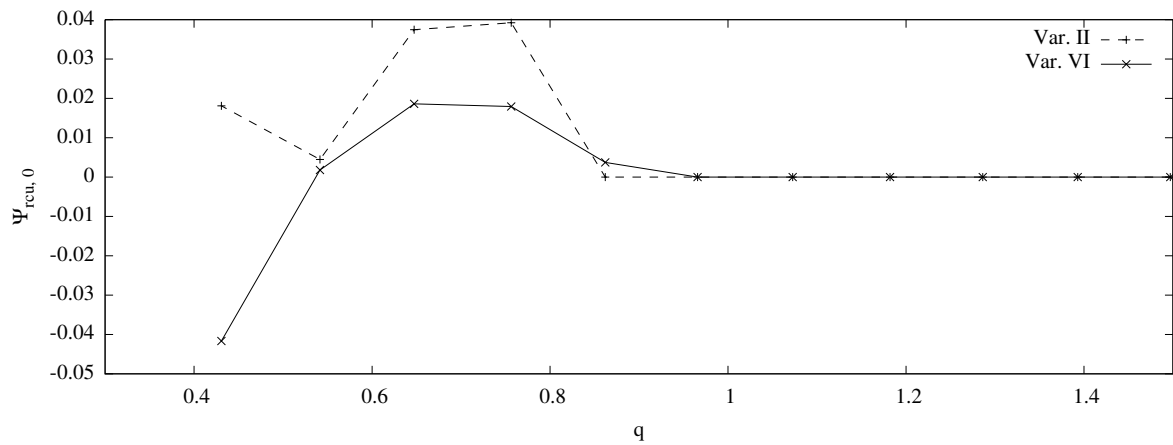


Abbildung 5.31: Dimensionsloser Drall  $\Psi_{rcu}$  in der Bilanzierungsebene 0, Varianten II und VI

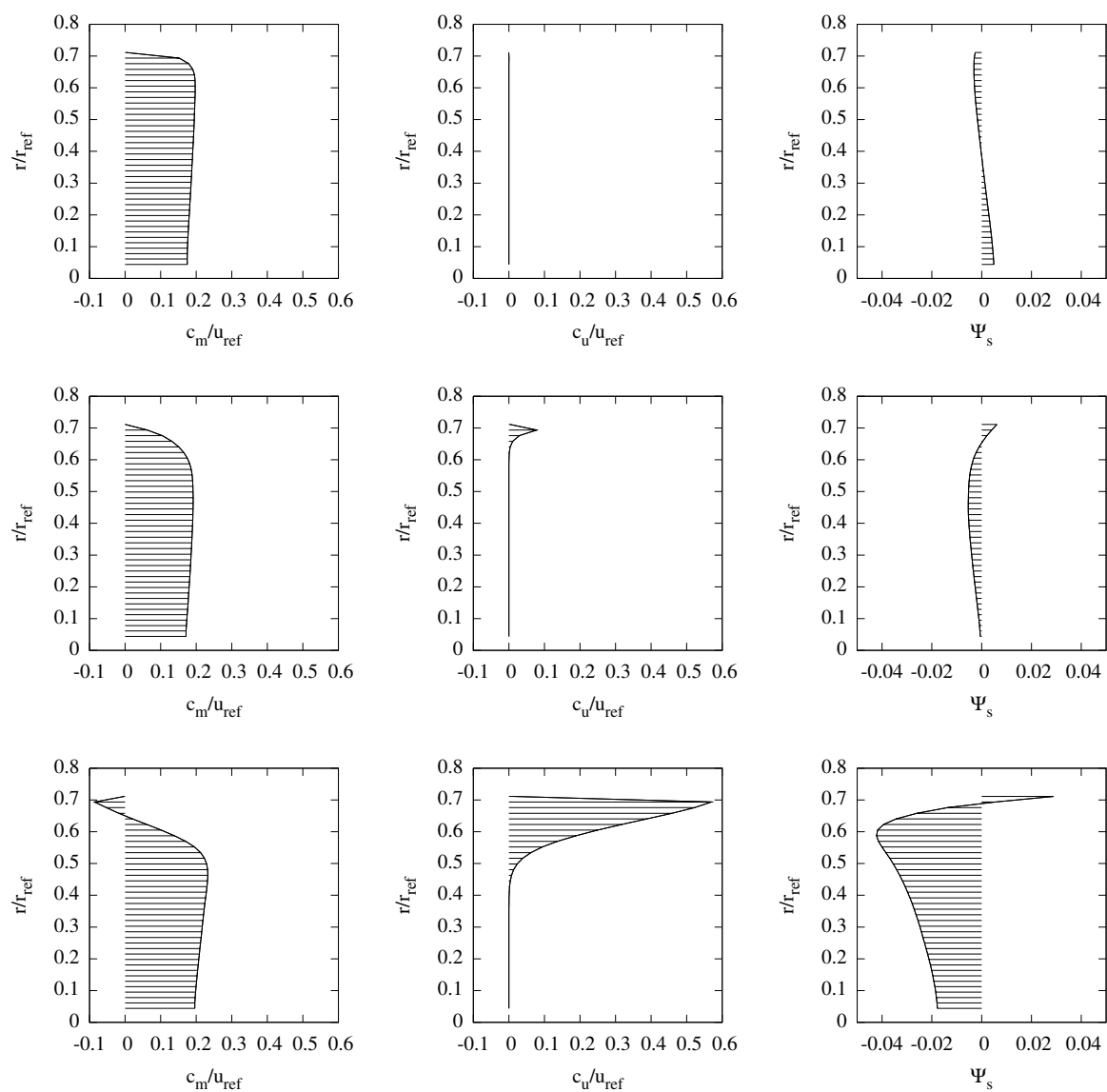


Abbildung 5.32: Umfanggemittelte Größen der Variante VI in der Bilanzierungsebene 0 für  $\phi = 0.094$  (oben),  $\phi = 0.084$  (mitte) und  $\phi = 0.074$  (unten)

Startwert	$\Psi_{t,imp}$ [-]	$\Psi_{t,th,imp}$ [-]	$\eta_{imp}$ [-]	$P_{mech}$ [kW]	$P_{fluid}$ [kW]	$\eta_{sys}$ [-]
Restart	0.776	0.820	0.947	7.541	6.686	0.887
$\Phi_i = 0$	0.779	0.818	0.953	7.522	6.712	0.892
$\delta$	-0.4%	0.3%	0.7%	0.3%	-0.4%	-0.6%
	$Q_{sp,SS}$ [m <sup>3</sup> /s]	$Q_{sp,DS}$ [m <sup>3</sup> /s]	$F_{ax}$ [N]	$\Delta F_{ax}$ [N]	$c_{f,ax}$ [-]	
Restart	$2.115 \cdot 10^{-5}$	$1.978 \cdot 10^{-5}$	-66.6	29.5	$-1.455 \cdot 10^{-2}$	
$\Phi_i = 0$	$2.158 \cdot 10^{-5}$	$2.141 \cdot 10^{-5}$	-73.2	23.0	$-1.597 \cdot 10^{-2}$	
$\delta$	-2.0%	-7.6%	-8.9%	28.3%	-8.9%	

**Tabelle 5.8:** Unterschiede der integralen Werte der Variante VI bei  $\phi = 0.1253$  in Abhängigkeit des Lösungswegs

### 5.3.4 Frozen-Rotor-Rechnung

Basierend auf dem Rechennetz mit  $N_k = 39$  wurde ein Netz für das komplette Laufrad mit langem Einlauf erzeugt. Die Spirale wurde als benetztes Volumen aus dem CAD exportiert und mittels des Spiralnetzgenerators aus Abschnitt 2.6.8 vernetzt. Die verwendete tangentielle Auflösung beträgt dabei ca. 360 Knoten, d.h. eine Netzzelle hat einen durchschnittlichen Öffnungswinkel von  $1^\circ$ . Um einen Einfluß der Austrittsfläche auf die Strömung innerhalb der Pumpe weitestgehend ausschließen zu können, wurde ein Auslaßbereich als Rohr von 1 000 mm Länge auf den Druckstutzen der Spirale aufgesetzt. Das erzeugte Rechennetz hat eine Größe von  $N \approx 3.8 \cdot 10^6$  Zellen. Das verwendete Rechennetz, s. Abbildung 5.33, entspricht der Einbausituation im Prüfstand. Da die Frozen-Rotor-Rechnung stellungsabhängig ist, wurden insgesamt 6 Laufradstellungen durchgerechnet, die sich um jeweils  $10^\circ$  unterscheiden. Die in Abbildung 5.36 abgebildeten integralen Werte ergeben sich aus der Mittelung der integralen Werte der einzelnen Positionen. Wegen des stark gestiegenen Rechenaufwands wurden folgende Anpassungen vorgenommen:

- Aus Stabilitäts- und Laufzeitgründen wurde das Residuenkriterium auf  $Res_{max} \leq 10^{-6}$  erhöht.
- Es wurden nur der Teilbereich von  $0.084 \leq \phi \leq 0.115$  betrachtet.
- Es wurde neben der UDS-Simulation das MINMOD-Diskretisierungsschema mit einem Blending von 0.2 errechnet.

Somit wurden insgesamt 48 Simulationen durchgeführt. Die Rechenzeit einer Einzelsimulation bewegt sich in der Größenordnung von 2-5 Tagen auf einem Rechner wie aus Tabelle 4.2. Da die einzelnen Simulationen nur gering voneinander abhängen, läßt sich dieses Experiment durch Verwenden mehrerer Rechner beschleunigen.



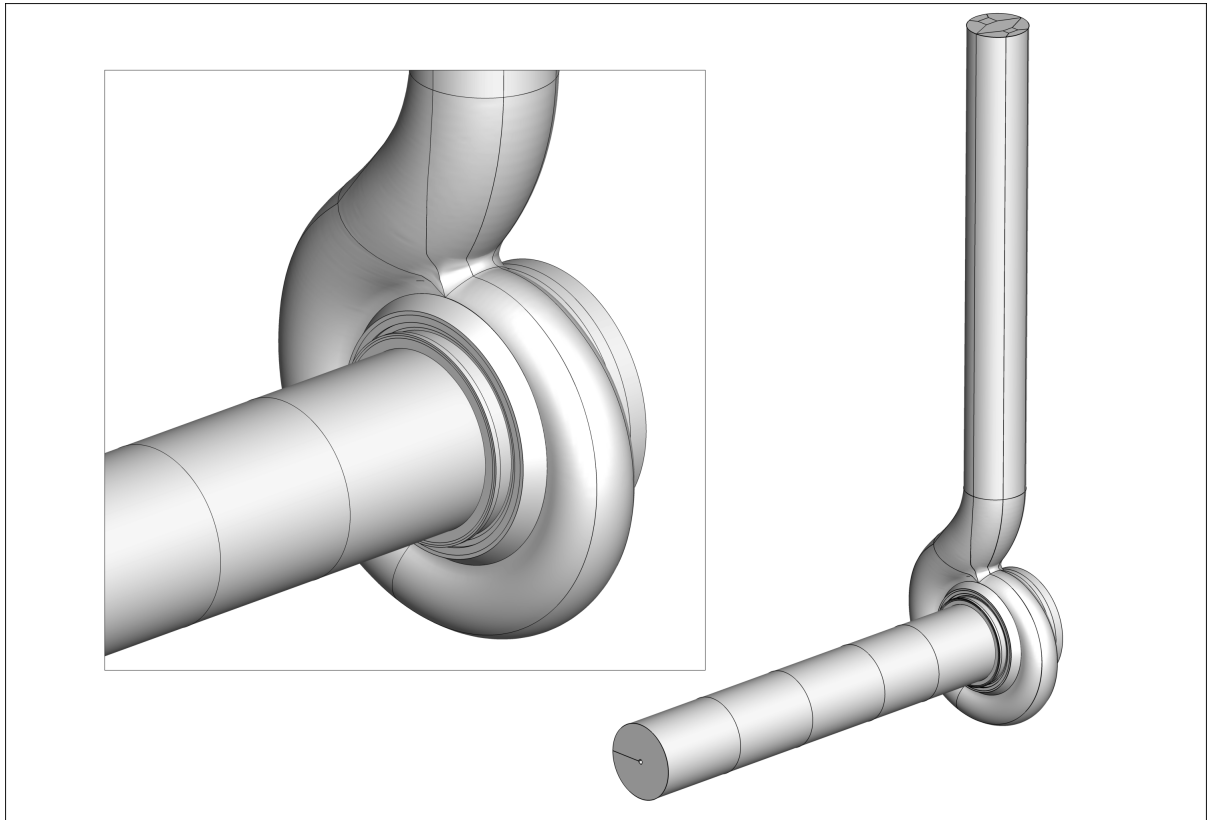


Abbildung 5.33: Vernetzung für die Frozen-Rotor-Simulation

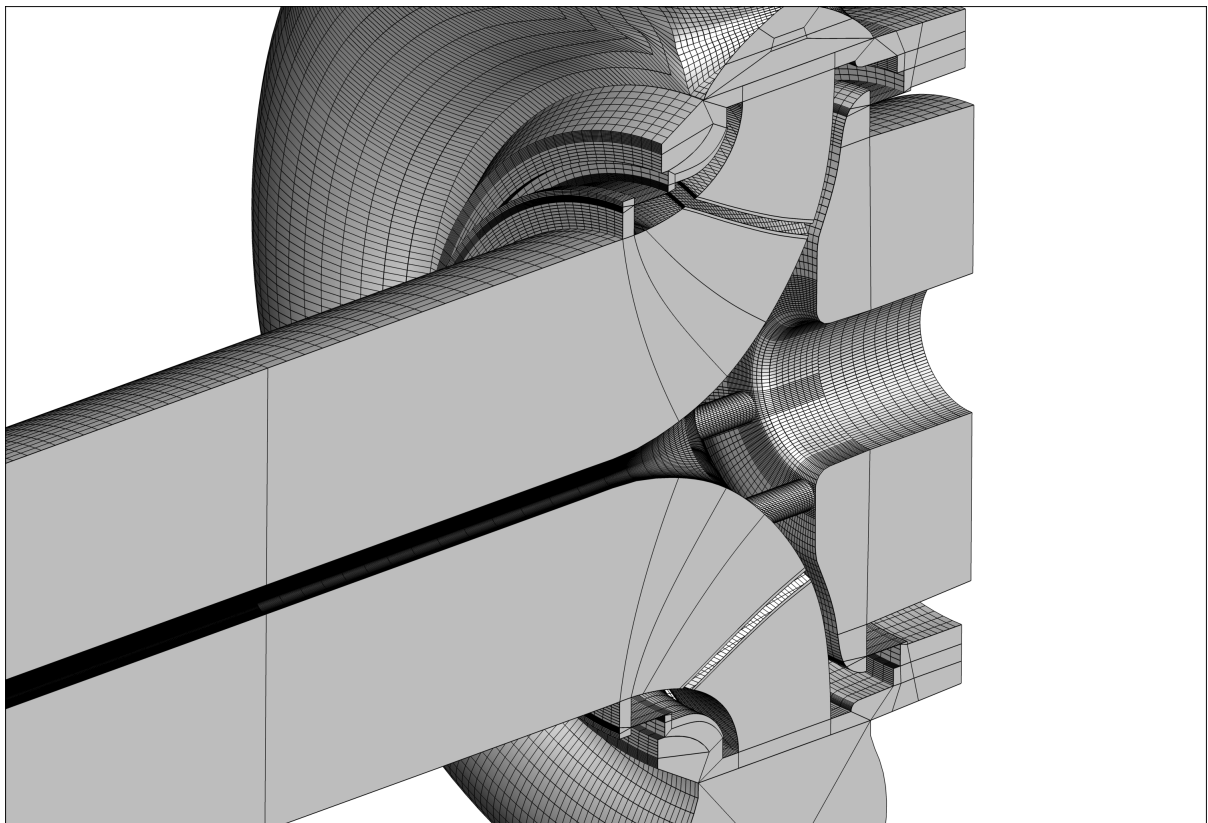
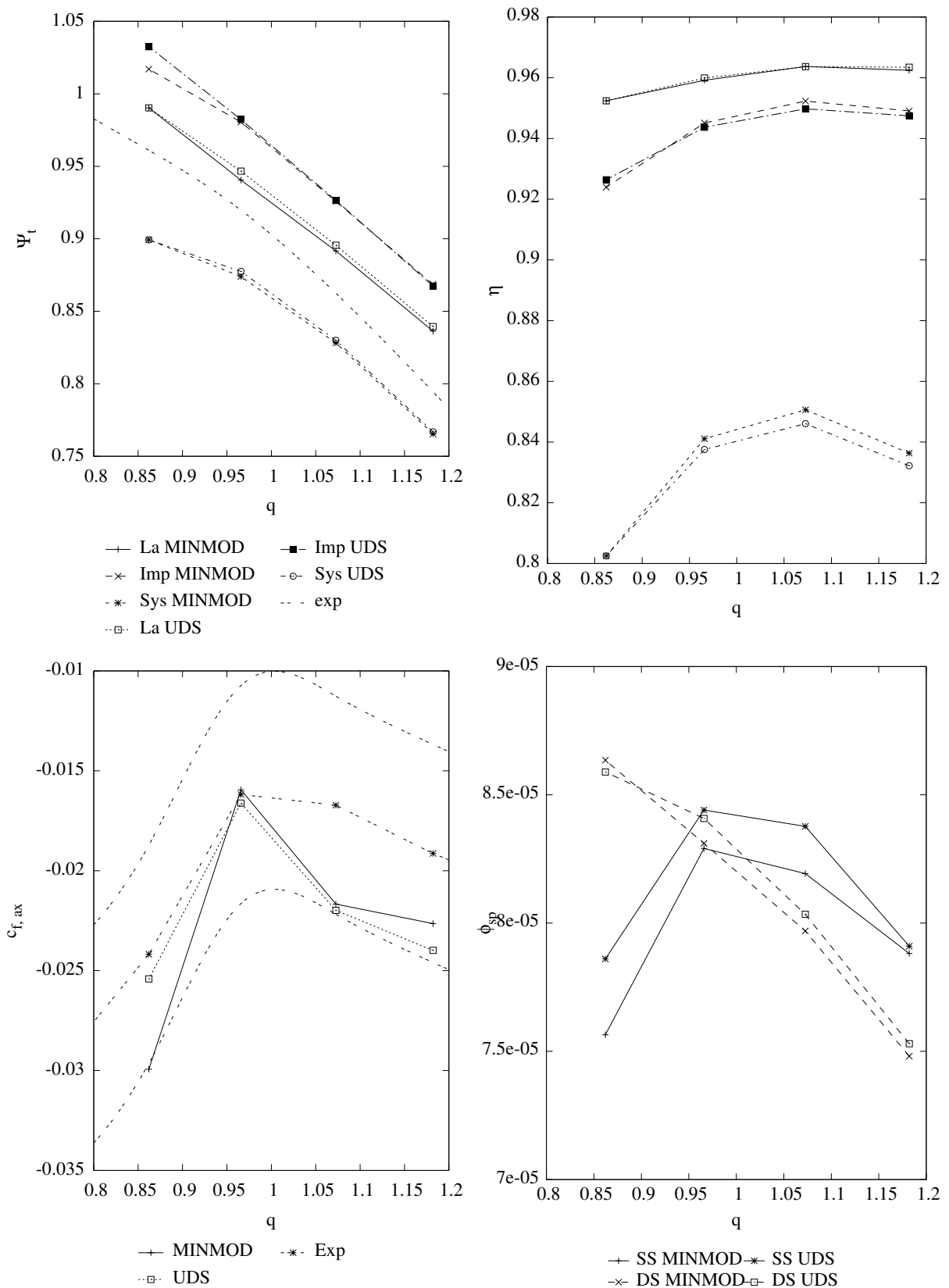
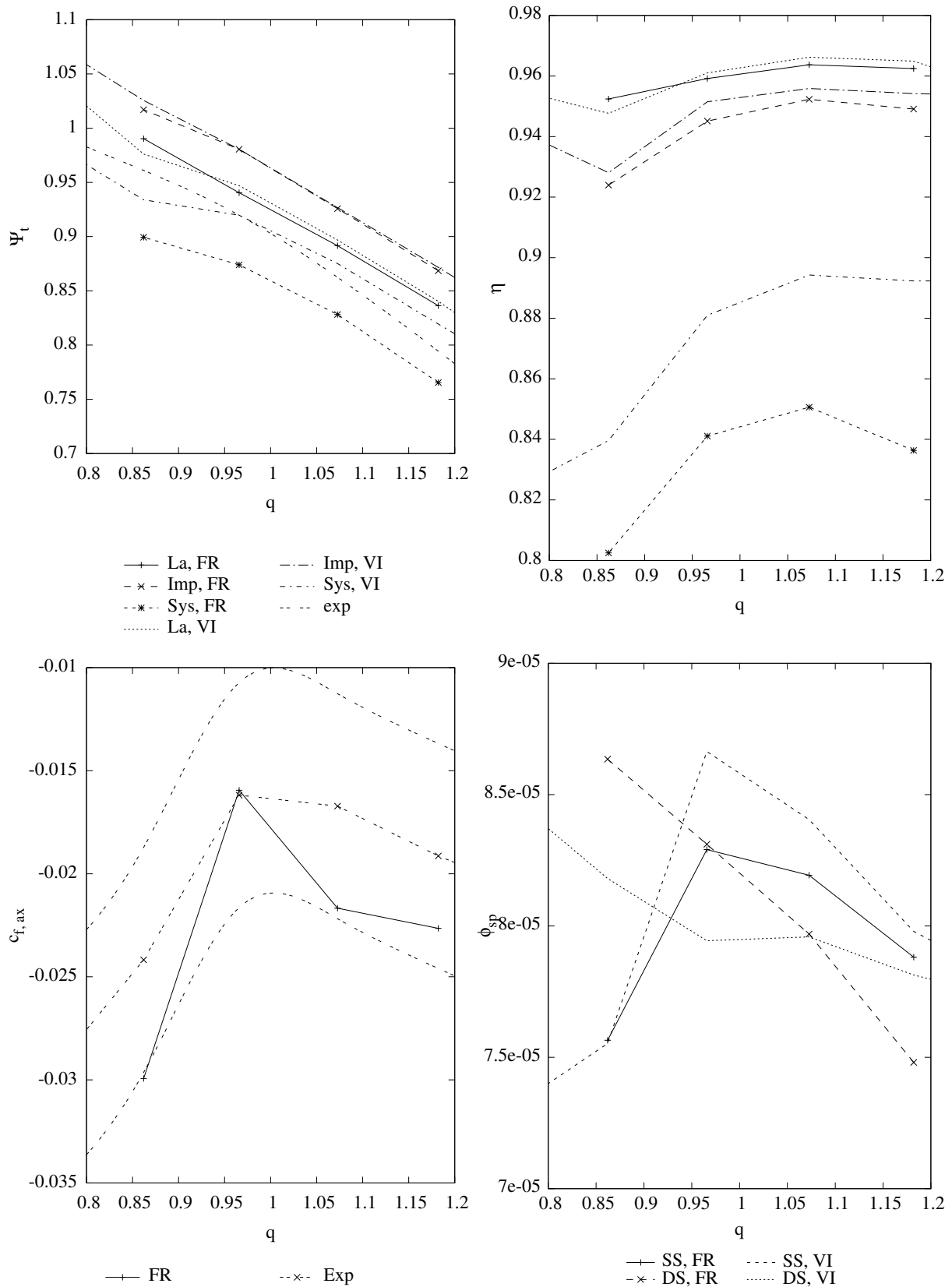


Abbildung 5.34: Achsschnitt der Vernetzung für die Frozen-Rotor-Simulation



**Abbildung 5.35:** Charakteristiken aus der Frozen-Rotor-Simulation über dem relativen Volumenstrom  $q$ , Unterschiede zwischen MINMOD und UDS



**Abbildung 5.36:** Charakteristiken aus der Frozen-Rotor-Simulation über dem relativen Volumenstrom  $q$ , Frozen Rotor (FR) und Einzelkanalrechnung Variante VI (VI)

In Abbildung 5.35 sind die Unterschiede der Ergebnisse zwischen dem Diskretisierungsverfahren UDS und MINMOD aufgeführt. Die integralen Werte wurden für die jeweils sechs Rotorstellungen ermittelt und dann gemittelt. Der Einfluß des Diskretisierungsschema auf die integralen Ergebnisse ist gering, der Axialschubbeiwert hingegen variiert je nach Betriebspunkt deutlich. Dies ist vor allem auf die unterschiedlichen Volumenströme in den Spalten zurückzuführen. Die Durchströmung der druckseitigen Spalte variiert etwas geringer mit dem Diskretisierungsschema als der saugseitige. Da im Ergebnis der Axialschub mit genauerer Berechnung des numerischen Modells stärker abweicht, ist wahrscheinlich der saugseitige Radseitenraum oder Spalt nicht ausreichend genau modelliert, sei es hinsichtlich der räumlichen Auflösung oder des verwendeten Turbulenzmodells.

In Abbildung 5.36 sind die integralen Werte der Frozen-Rotor-Simulationen mit dem MINMOD-Diskretisierungsschema zusammengefaßt und – soweit sinnvoll – den Ergebnissen der Variante VI gegenübergestellt. Für die verschiedenen Wirkungsgrade und die Druckzahlen ergeben sich nur geringe Unterschiede, die auf das unterschiedliche Geometriemodell zurückzuführen sind. Die große Differenz bei den über das Gesamtsystem ermittelten integralen Werten ist den Spiralverlusten und den Verlusten im Druckrohr geschuldet, die bei keiner der Einzelkanalrechnungen berücksichtigt werden. Die Unterschiede in den Spaltvolumenströmen sind deutlich größer: die Frozen-Rotor-Simulation weist ein asymmetrisches Druckprofil in Umfangsrichtung auf, wodurch sich die Durchströmung der Radseitenräume über den Umfang stark verändert. Die ermittelten integralen Axialschubbeiwerte  $c_{f,ax}$  treffen das Band der Meßunsicherheit deutlich besser als die Variante VI. Der Betriebspunkt  $\phi = 0.094$  wird von der Frozen-Rotor-Simulation deutlich besser vorhergesagt als bei der Einzelkanalsimulation, Variante VI.

### 5.3.5 Bewertung

Mit dem vorgestellten Programmsystem ist es möglich, ein Rechengitter für die Ermittlung des Axialschubs zu erstellen, die Simulationen durchzuführen und die Ergebnisse auszuwerten. Durch die Verwendung hybrider Rechengitter mit lokal unterschiedlicher Wandbehandlung kann die Gesamtgröße des Rechnernetzes begrenzt werden. Die damit verbundenen Sprünge in den Zellgrößen verringern die Stabilität der Simulation. Die Güte des ermittelten Achsschubs steigt erwartungsgemäß mit der Netzauflösung. Eine Einzelkanalrechnung ist unabhängig von der verwendeten Auslaßgeometrie nicht ausreichend, da die unsymmetrischen Druckprofile in der Spirale nicht abgebildet werden können. Mit der Frozen-Rotor-Rechnung kann über mehrere Stellungen gemittelt ein plausibler Achsschub ermittelt werden. Die in der Simulation ermittelten integralen Größen weisen die gleichen Richtungstendenzen der Messung auf. Der Verlauf des Axialschubs wird hinsichtlich der Lage des betragsmäßigen Maximums richtig abgebildet, die Steigung wird jedoch falsch wiedergegeben.

Die Rechenzeiten der Frozen-Rotor-Rechnung sind vertretbar, allerdings müssen die Simulationen überwacht werden, damit bei Bedarf Unterrelaxationsfaktoren angepaßt werden können. Die Stabilität der Simulation ist noch verbesserungswürdig.

Angesichts der großen Streubreite der Axialschubwerte muß die Zuverlässigkeit des Verfahrens auf dem gegebenen Stand als fragwürdig angesehen werden. Die hier vorgestellten Simulationen sind ein geeigneter Anknüpfungspunkt, um Guidelines für die

Auflösung von Rechengittern für die Axialschubermittlung zu erstellen und in Zukunft verlässlichere Axialschübe aus der Simulation ermitteln zu können.



# Kapitel 6

## Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde ein Programmsystem zur Nachrechnung und Optimierung von hydraulischen Turbomaschinen radialer Bauart vorgestellt und seine Leistungsfähigkeit anhand von praxisbezogenen Beispielen aufgezeigt. Dabei wurde für die schnelle Voro Optimierung das QNS-Verfahren betrachtet und mit dem RANS-Verfahren verglichen. Es wurde nachgewiesen, dass das QNS-Verfahren bei deutlich geringerer Laufzeit ähnliche Ergebnisänderungen bei gegebenen Geometrieänderungen aufweist wie das RANS-Verfahren. Das QNS-Verfahren konnte erfolgreich für die Optimierung zweier radialer Pumpenlaufräder genutzt werden, indem es mit einem frei verfügbaren Genetischen Algorithmus gekoppelt wurde. Die Optimierung geschah vollautomatisch ohne Benutzereingaben und hatte eine deutlich verbesserte Laufschaufelgeometrie als Ergebnis.

Das CFD-Experiment konnte weitestgehend automatisiert werden, wobei sowohl die Vernetzung, die Modellierung der Wandreibung sowie die Konfiguration des CFD-Solver NS3D betrachtet worden sind. Das automatisierte CFD-Experiment wurde erfolgreich genutzt, um für eine gegebene Reihe von Radialpumpen  $10 \text{ min}^{-1} \leq n_q \leq 90 \text{ min}^{-1}$  Kennlinien zu ermitteln. Die ermittelten Kennlinien zeigen für  $n_q \geq 30 \text{ min}^{-1}$  einen plausiblen Verlauf, für kleinere  $n_q$  ergeben sich jedoch noch zu klärende Tendenzen, die der experimentellen Erfahrung widersprechen.

Für eine Francis-Turbine mit dem  $n_q = 44 \text{ min}^{-1}$  konnte das System erfolgreich genutzt werden, um ein Kennfeld basierend auf einer Einzelkanalrechnung zu ermitteln. Die Ähnlichkeit mit einem gemessenen Kennfeld einer Turbine mit gleichen  $n_q$  ist bezogen auf das einfache Modell und die nicht simulierten Maschinenteile gut.

Mit der Ermittlung des Axialschubs für eine Radialpumpe mit dem  $n_q = 53 \text{ min}^{-1}$  konnte die Leistungsfähigkeit des Systems bei komplexeren geometrischen Modellen nachgewiesen werden. Dabei wurden die Radseitenräume, Spalte, Bohrungen, Spiralgehäuse und Zu- und Ablauf im System modelliert, simuliert und ausgewertet. Die in der Simulation ermittelten integralen Größen weisen die gleichen Richtungstendenzen der Messung auf.

Für zukünftige Arbeiten bestehen folgende Anknüpfungspunkte:

- Die Erweiterung des Programmsystems um die Simulation der Struktur der Turbomaschine, die mit der Simulation der Strömung gekoppelt wird, also der Fluid-

Struktur-Interaktion. Die Strömungsrechnung zeigt die Belastungen auf, die auf die Struktur der Turbomaschine wirken. Neben Fragestellungen der Festigkeit eines Bauteils können mit der Simulation der Struktur auch vibroakustische Probleme behandelt werden. Ein Ansatz von EINZINGER [19] liegt bereits vor.

- Die Vernetzungsstrategien für blockstrukturierte Netze stoßen bei der Simulation einer ganzen Turbomaschine an ihre Grenzen. Gerade bei Pumpen existieren Bauformen – wie etwa einschaulige, offene Laufräder oder mehrstufige Pumpen mit Rückführstufen – die mit den vorgestellten Strategien gar nicht vernetzt werden können. Die Erweiterung der Netzgenerierung und des CFD-Codes auf unstrukturierte bzw. hybride, d.h. gemischt blockstrukturierte und polyedrische, Netze ist spätestens bei der Berücksichtigung der Fluid-Struktur-Interaktion unabweichlich.
- Erarbeiten von Richtlinien zur erfolgreichen Ermittlung des Axialschubs. Die Simulationen haben deutlich aufgezeigt, dass für die korrekte Ermittlung des Axialschubs erst Frozen-Rotor-Rechnungen verlässliche Ergebnisse liefern. Damit verbunden sind große Rechengitter und lange Laufzeiten. Es sind Vernetzungsrichtlinien zu erarbeiten, damit nicht bei jeder neuen Problemstellung Netzstudien durchgeführt werden müssen. Die Rechenzeiten sind zusätzlich zu verringern, um CFD-Simulationen für die Ermittlung des Axialschubs als Werkzeug für die Praxis etablieren zu können.
- Die Zuverlässigkeit des NS3D für Radialpumpen mit niedriger Schnellläufigkeit  $n_q$  muß erhöht werden. Dabei ist die Modellierung der Reibung zu verbessern.

Einige dieser aufgezählten Punkte sind Aufgaben, die in kommerziell verfügbaren Programmsystemen bereits realisiert sind, insbesondere die polyedrischen Netze sind anerkannter Stand der Technik. Punkte wie Verbesserung der Reibungsmodelle bleiben tendenziell länger bestehen, solange der wirkende Mechanismus nicht komplett in der Simulation wiedergegeben werden kann.

Die CFD-Simulation wird seit langem erfolgreich für Optimierungsaufgaben herangezogen, da ihre relativen Aussagen über „bessere“ und „schlechtere“ Geometrien zuverlässig sind. Das höchste Ziel der Simulationstechnik muß aber sein, zuverlässig absolute Ergebnisse zu liefern, um die Anzahl von zeit- und kostenintensiven Modellversuchen entscheidend verringern zu können.



# Literaturverzeichnis

- [1] *Eclipse Rich Client Platform*. [http://wiki.eclipse.org/index.php/Rich\\\_Client\\\_Platform](http://wiki.eclipse.org/index.php/Rich\_Client\_Platform), Abruf: 16. Februar 2010
- [2] *Apache License, Version 2.0*. <http://www.apache.org/licenses/LICENSE-2.0>. Version: 2004, Abruf: 16. Februar 2010
- [3] *NetBeans IDE Dual License Header and License Notice*. <http://netbeans.org/cddl-gplv2.html>. Version: 2004, Abruf: 16. Februar 2010
- [4] PumpLinX: A Modelling Tool for Pump Designers. In: *Int. Journal of Fluid Power* 8, No. 1 (2007), March
- [5] <http://www.industrie-energieeffizienz.de/technologien.html>
- [6] <http://www.cfturbo.de/cft.html>
- [7] *JDK 7 Features*. <http://openjdk.java.net/projects/jdk7/features>. Version: Februar 2010, Abruf: 9. Januar 2010
- [8] *Netbeans Platform*. <http://platform.netbeans.org/>. Version: 2010, Abruf: 16. Februar 2010
- [9] ADAMS, Nikolaus A.: *Skriptum zur Vorlesung Fluidmechanik I: Einführung in die Dynamik der Fluide*. 2009
- [10] BÄCK, T.: *Evolutionary Algorithms in Theory and Praxis*. New York, NY : Oxford University Press, Inc., 1996
- [11] BADER, Robert: *Simulation kompressibler und inkompressibler Strömungen in Turbomaschinen*, Technische Universität München, Diss., 2006
- [12] BÖCK, Heiko: *NetBeans Platform 6 – Rich-Client-Entwicklung mit Java*. Galileo Computing, Bonn, 2008
- [13] BOUDREAU, Tim ; TULACH, Jaroslav ; WIELENGA, Gertjan: *Rich Client Programming: Plugging into the Netbeans Platform*. Prentice Hall, Upper Saddle River, 2007
- [14] BOUSSINESQ, J.: Essai sur La Théorie Des Eaux Courantes. In: *Mem. Présenté Acad. Sci.* 23 (1877), S. 46. – Paris
- [15] BURNS, Marshall: *Automated Fabrication: Improving productivity in manufacturing*. Englewood Cliffs, NJ : Prentice Hall, 1993

- [16] COMELLAS, Francesc ; SAPENA, Emili: A multiagent algorithm for graph partitioning. In: *Applications of Evolutionary Computing* Bd. 3907/2006. Berlin / Heidelberg : Springer, 2006. – ISBN 978-3-540-33237-4, S. 279 – 281
- [17] COWARD, Danny ; MULLER, Hans: *JSR 296: Swing Application Framework*. <http://jcp.org/en/jsr/detail?id=296>. Version: 2006, Abruf: 16. Februar 2010
- [18] DURBIN, Paul A. ; PETTERSSON REIF, Björn A.: *Statistical theory and modelling for turbulent flows*. Wiley, New York, 2000
- [19] EINZINGER, Johannes: *Numerische Simulation der Fluid-Struktur-Interaktion in Turbomaschinen*, Technische Universität München, Diss., 2003
- [20] EINZINGER, Johannes ; WUNDERER, Roland ; SCHILLING, Rudolf: Abschlußbericht zum AiF-Forschungsvorhaben Nr. 12921 N/1 Numerische Simulation des stationären und instationären Betriebsverhaltens sowie der Strömungsphänomene in Kreiselpumpen beliebiger spezifischer Drehzahl - Numerischer Prüfstand / Technische Universität München. 2005 (AIF12921 N/1). – Forschungsbericht
- [21] FERZIGER, J. H. ; PERIĆ, M.: *Computational methods for fluid dynamics*. Springer, Berlin, 1997
- [22] FIEREDER, Rainer: Quasi-Euler-Verfahren - Erstausslegung und Optimierung von Strömungsmaschinen ohne Turbulenzmodellierung / TU München. 2007. – interner Bericht
- [23] FOWLER, Martin: *Inversion of Control Containers and the Dependency Injection pattern*. <http://martinfowler.com/articles/injection.html>. Version: 23. Januar 2004
- [24] FROBENIUS, Moritz: *Numerische Simulation kavitierender Strömungen in hydraulischen Strömungsmaschinen*. München, Technische Universität München, Diss., 2004
- [25] GOLDBERG, D. E.: *Genetic Algorithms in search, Optimisation and machine learning*. Reading, Mass. : Addison-Wesley, 2003
- [26] GUELICH, Johann F.: *Kreiselpumpen*. Berlin Heidelberg New York : Springer Verlag, 2004
- [27] HARTEN, A.: High resolution schemes for hyperbolic conservation laws. In: *Journal of Computational Physics* 49 (1983), Nr. 3, S. 357–393
- [28] HERRMANN, Stefan: Allgemeine Blockinterfaces / Technische Universität München. 2008. – interner Bericht
- [29] HOSCHEK, Josef ; LASSER, Dieter: *Grundlagen der geometrischen Datenverarbeitung*. Stuttgart : B.G. Teubner, 1992
- [30] HUNTS, Andy ; THOMAS, Dave ; CUNNINGHAM, Ward: *The Pragmatic Programmer*. 1. Auflage. Addison-Wesley, 2000. – ISBN 978-0-201-61622-4

- [31] KARYPIS, George ; KUMAR, Vipin: Parallel Multilevel k-way Partitioning Scheme for Irregular Graphs. In: *SC Conference 0* (1996), S. 35. <http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/SC.1996.32>. – DOI <http://doi.ieeecomputersociety.org/10.1109/SC.1996.32>. ISBN 0–89791–854–1
- [32] KRÄMER, Stefan: *Entwurf von Beschaukelungen hydraulischer Maschinen mit Hilfe neuronaler Netze*, Technische Universität München, Diss., 2006
- [33] KRONSCHNABL, Florian: *Algebraische Generierung von Strombahnen für Strömungsmaschinen*. 2003 interner Bericht
- [34] KRONSCHNABL, Florian: *Numerische Strömungssimulation von Horizontalachs-Windturbinen*, Technische Universität München, Diss., 2009
- [35] LAUNDER, B. E. ; SPALDING, D. B.: The numerical computation of turbulent flows. In: *Computer Methods in applied mechanics and engineering* 3 (1974), S. 269–289
- [36] LEPACH, Thomas: *Entwurf und Optimierung von Francis Turbinen*, Technische Universität München, Diss., 2006
- [37] LIEN, F. S. ; CHEN, W. L. ; LESCHZINER, M. A.: Low-Reynolds-number eddy-viscosity modelling based on non-linear stress-strain/vorticity relations. In: RODI, W. et a. (Hrsg.): *Proceedings of the 3<sup>rd</sup> Int. Symposium on Engineering Turbulence Modelling and Experiments, Kreta*. Elsevier, Amsterdam u.a., 1996, S. 91–100
- [38] MEFFERT, Klaus: *Java Genetic Algorithms Package*. <http://jgap.sourceforge.net/>
- [39] O’CONNOR, John: Using the Swing Application Framework (JSR 296). (2007). <http://java.sun.com/developer/technicalArticles/javase/swingappfr/>. – zuletzt besucht am 16. Februar 2010
- [40] PFLEIDERER, Carl ; PETERMANN, Hartwig: *Strömungsmaschinen*. Springer, Berlin, 2005
- [41] PIEGL, Les ; TILLER, Wayne: *The NURBS book*. 2. Auflage. Berlin Heidelberg New York : Springer, 1997. – ISBN 3–540–61545–8
- [42] RAABE, Joachim: *Hydraulische Maschinen und Anlagen*. Düsseldorf : VDI Verlag, 1989
- [43] RECHENBERG, I.: *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart : Frommann-Holzboog, 1994
- [44] ROGERS, David F.: *An introduction to NURBS with historical perspective*. San Diego : Academic Press, 2001. – ISBN 1–55860–669–6
- [45] SCHILLING, R.: Strömung in Radseitenräumen von Kreiselpumpen. In: *Strömungsmechanik und Strömungsmaschinen* 27 (1979), S. 21–86
- [46] SCHILLING, R. ; THUM, S. ; MÜLLER, N. ; KRÄMER, S. ; RIEDEL, N. ; MOSER, W.: Design optimisation of hydraulic machinery blading by multi level CFD technique. In: *Proceedings of the 21<sup>th</sup> IAHR Symposium*. Lausanne, 2002

- [47] SCHILLING, Rudolf: *Skriptum zur Vorlesung Wasserturbinen*
- [48] SCHILLING, Rudolf ; FLURL, Benedikt ; JANTZEN, Andreas ; SCHMALHORST, Carsten: Development of a Numerical Test Rig for Hydraulic Turbomachinery. In: *Proceedings of the International Rotating Equipment Conference, Pumps and Compressors, Düsseldorf Oct. 28-29 2008*, 2008
- [49] SCHMALHORST, Carsten ; FLURL, Benedikt ; SCHILLING, Rudolf: Development of an Integrated Design System (IDS) for Hydraulic Machinery. In: *Proceedings of the 9th ASME Conference on Engineering Systems Design and Analysis, July 7-9, 2008, Haifa, Israel*, 2008
- [50] SKODA, Romuald: *Numerische Simulation abgelöster transitionaler Strömungen in Turbomaschinen*, Technische Universität München, Diss., 2003
- [51] TENNEKES, H. ; LUMLEY, J. L.: *A first course in turbulence*. MIT Press, Cambridge, 1976
- [52] THOMPSON, Joe F. (Hrsg.) ; SONI, Bharat (Hrsg.) ; WEATHERILL, Nigel (Hrsg.): *Handbook of Grid Generation*. Boca Raton : CRC Press LLC, 1999. – ISBN 0-8493-2687-7
- [53] THUM, Susanne: *Optimalauslegung und numerische Optimierung strömungsführender Bauteile in Francis Spiralturbinen*, Technische Universität München, Diss., 2006
- [54] TRUCKENBRODT, E.: *Stömungsmechanik*. Springer, Berlin, 1968
- [55] WILL, Björn-Christian ; BENRA, Friedrich-Karl ; DOHMEN, Hans-Josef: Numerical and experimental investigations of the Flow in the Side Cavities of a Centrifugal Pump. In: *Proceedings of the 13<sup>th</sup> International Symposium on Transport Phenomena and Dynamics of Rotating Machinery, Honolulu, Hawaii, April 4-7*, 2010
- [56] WUNDERER, Roland: *Netzgenerierung von Spiralgehäusen für den NS3D*. Interner Bericht, 2004
- [57] ZIEREP, J.: *Grundzüge der Strömungslehre*. C. Braun, Karlsruhe, 1990