

Abschlussbericht

der Technischen Universität Dresden im Projekt Darwin-TUD

LuFo VI-1

Darwin-TUD

TUD Beitrag zu „Digitaler Faden“-basierte Triebwerksentwicklung mit eingebetteter Künstlicher Intelligenz

Dresden, Juni 2025

Technische Universität Dresden

Institut für Strömungsmechanik und Center for Interdisciplinary Digital Sciences

01062 Dresden

Verfasser: Christian Linné, Maximilian Sander
Sebastian Strönisch, Ilias Vasilopoulos
Arbeitsgruppenleiter: Dr.-Ing. Matthias Voigt, Dr.-Ing. Matthias Lieber
Lehrstuhlinhaber: Prof. Dr.-Ing. habil. Ronald Mailach
Prof. Dr. rer. nat. Wolfgang E. Nagel

Die Untersuchungen wurden im Rahmen des Projektes LuFo VI durchgeführt und mit Mitteln des Bundesministeriums für Wirtschaft und Klimaschutz (BMWK) aufgrund eines Beschlusses des Deutschen Bundestages unter dem Förderkennzeichen 20D1911C gefördert. Die Autoren danken Rolls-Royce Deutschland Ltd & Co KG für die Unterstützung und die Erlaubnis zur Publikation. Die Verantwortung für den Inhalt liegt allein bei den Autoren.

Gefördert durch:



Bundesministerium
für Wirtschaft
und Klimaschutz

aufgrund eines Beschlusses
des Deutschen Bundestages

Inhaltsverzeichnis

Abbildungsverzeichnis	6
Tabellenverzeichnis	10
Nomenklatur	11
Abkürzungsverzeichnis	16
1 Zielstellung	17
1.1 Bezug des Vorhabens zu förderpolitischen Zielen	17
1.2 Gesamtziel des Vorhabens	17
1.3 Wissenschaftliche und technische Arbeitsziele des Vorhabens	18
2 Einleitung	21
2.1 Motivation	21
2.2 Aufbau des Berichts	23
3 Teilbericht AP1.4	24
3.1 Einführung AP1.4	25
3.2 Ausgewählte Grundlagen	25
3.2.1 JSON-LD: JSON für verlinkte Daten	26
3.2.2 Datenweitergabe durch den Linker	27
3.2.3 Der Performance Engineering Zyklus auf Basis von Score-P	27
3.2.4 Grundlagen und Nutzung von Score-P	28
3.3 Score-P Metadaten Schema auf JSON-LD Basis	30
3.3.1 Metadatenschema Definition	30
3.3.2 Übersicht über Metadatenimplementierungsentwürfe	31
3.3.3 Umsetzung mit RDF/JSON-LD	32
3.4 Hydra Performance Engineering mit Score-P	33
3.4.1 Score-P Integration in Hydra	34
3.4.2 Score-P Instrumentierung von ParaView Catalyst	34
3.4.3 Erweiterung und Optimierung der Mess- und Analyseinfrastruktur	36

3.4.4	Performance Evaluierung	38
3.4.5	Performance Optimierungen	42
3.5	Zusammenfassung und Ausblick	43
3.5.1	Zusammenfassung	43
3.5.2	Ausblick	44
4	Teilbericht AP2.1	45
4.1	Ausgewählte Grundlagen (AP2.1)	46
4.1.1	Messverfahren zur Digitalisierung der Hochdruckturbinenschaufeln . . .	46
4.1.2	Probabilistische Grundlagen	51
4.1.3	Numerische Modellierung	57
4.2	Parametrisierung der Variabilität von Hochdruckturbinenschaufeln (AP2.1) . . .	71
4.2.1	Parametrisierung des Schaufelblatts	71
4.2.2	Parametrisierung des Kühlluftkanals	74
4.3	Prozesskette (AP2.1)	79
4.3.1	Messdaten	80
4.3.2	Turbulatorentfernung	80
4.3.3	Messdatenausrichtung	84
4.3.4	Statistische Beschreibung der Eingangsgrößen	87
4.3.5	gpds-fc Prozesskette und Dateneinbindung	92
4.4	Zusammenfassung und Ausblick (AP2.1)	104
4.4.1	Zusammenfassung	104
4.4.2	Ausblick	104
5	Teilbericht AP2.3	106
5.1	Introduction AP2.3	107
5.2	Roughness Measurements	108
5.2.1	Literature Review	108
5.2.2	Measuring System and Automated Process	109
5.2.3	Validation of Measuring Device	113
5.2.4	Demonstration of Detailed Roughness Measurements for Rotor Blades .	114
5.2.5	Optimization of Measuring Point Distribution	117
5.2.6	Summary of Roughness Measurements for Entire HPC	120
5.3	Roughness Parameterization	122
5.3.1	Roughness Stripe Measurements and Definition of Parameters	122
5.4	Simulation-based Modelling of Roughness	123
5.4.1	Literature Review	123
5.4.2	CFD Grid Generation for Channel Flow Simulations	124
5.4.3	Validation of CFD Setup for Smooth Channel	126
5.4.4	Comparison between RANS and Wall-Resolved LES of Rough Channels .	129
5.4.5	Database Construction for Modelling	131
5.4.6	New Roughness Function Model	132
5.5	Summary and Conclusions (AP2.3)	137

6	Teilbericht AP3.1	139
6.1	Einführung	140
6.2	Ausgewählte Grundlagen	140
6.2.1	RDF Grundlagen	140
6.2.2	SPARQL und Triplestores	141
6.3	Anforderungen durch Daten und Metadaten	142
6.4	Repositorium und Zugriffsschnittstellen	143
6.4.1	Infrastruktur — Dataverse & ToMeDa	143
6.4.2	Infrastruktur — RDF & scorep-db	144
6.4.3	scorep-db	144
6.4.4	Verfügbare RDF-Triplestore-Backends	145
6.4.5	Beispielnutzung von scorep-db	145
6.5	Zusammenfassung und Ausblick	147
6.5.1	Zusammenfassung	147
6.5.2	Ausblick	148
7	Teilbericht AP. 3.4.8 - 3.4.11	149
7.1	Related Work - Graph Neural Networks for Flow Field Predictions	149
7.1.1	Fundamental Concepts of Graph Neural Networks	150
7.1.2	Graph Neural Networks for Flow Field Predictions	151
7.2	Distributed Multiscale Graph Machine Learning Model	152
7.2.1	Description of Learning Tasks and Evaluation Modes	152
7.2.2	Model Selection	155
7.2.3	Strategies for Distributed Training	159
7.2.4	Conclusion	160
7.3	Prediction of Turbomachinery Flow Using GNNs	160
7.3.1	Turbine Test Case	161
7.3.2	LT1: Time-Averaged Volume Quantities	163
7.3.3	LT2: Unsteady Volume Quantities	166
7.3.4	LT3: Time-Averaged Surface Quantities	168
7.3.5	LT4: Unsteady Surface Quantities	169
7.3.6	Training and Inference Costs	173
7.3.7	Conclusion	174
7.4	Summary and Outlook	175
8	Zusammenfassung und Ausblick	177
9	Liste der Veröffentlichungen	181
	Literaturverzeichnis	182

Abbildungsverzeichnis

3.1	Performance Engineering Zyklus	28
3.2	Arbeitsablauf Instrumentierung	29
3.3	Schematische Übersicht von Metadaten und deren Struktur	33
3.4	Darstellung der Hydra Bibliotheksabhängigkeiten	34
3.5	Kommunikationsmuster von zwei gekoppelten Simulationen samt Beendigungs- prozedur mittels MPI_Ibarrier (grün). Die Nutzung dieser dient zur Zustands- synchronisation und signalisiert dem Kopplungsprozess, dass beide Simulationen (Sessions) beendet sind, und dass sich jener auch beenden kann.	36
3.6	Konzeptuelle Analyse einzelner Lösesitzungen innerhalb des gekoppelten Strö- mungslösers	37
3.7	Kopplungsoperationen in Hydra (dargestellt in Vampir). Schwarze Linien markie- ren MPI-Kommunikation der Koppler. Die ersten drei Master-Threads dienen als Koppler; die folgenden 29.0 Threads verteilen sich auf drei Strömungssitzungen. Messfenster: 350 ms; 208 Ränge auf zwei Knoten.	39
3.8	Annotierter Vampir Screenshot einer Top-Down Analyse. Interessante Bereiche sind markiert.	40
4.1	Graustufenbild mit ISO-Filter	47
4.2	Schema Triangulationsverfahren	49
4.3	Streifenlichtprojektionsverfahren mittels räumlich-zeitlicher Kodierung	49
4.4	Schematische Darstellung der Messung eines Kühlluftlochs mittels vertikaler Fokusvariation	51
4.5	Schematische Ablauf der Monte-Carlo-Simulation	54
4.6	Systematik des Latin Hypercube Sampling und Methodik zur Unterteilung der Intervalle des Extended Latin Hypercube Samplings	56
4.7	Elementtopologie	58
4.8	Dimensionsloses Geschwindigkeitsprofil der Grenzschicht	63
4.9	Dimensionsloses Temperaturprofil der Grenzschicht	64
4.10	Profilschnitte einer vermessenen Turbinenschaufel	72
4.11	Parametrisierung einer Turbinenschaufel nach [68]]	73

4.12	Schwerpunktverschiebung des Kühlluftkanals durch Turbulatoren	75
4.13	Schwerpunktverschiebung bei Anpassung der Anzahl diskreter Punkten	75
4.14	Variation der Reproduktionsgüte bei unterschiedlichen Parametrisierungsansätzen	78
4.15	Übersicht der Prozesskette	79
4.16	GOM-Messungen mit CT Schnittebenen	80
4.17	Prozesskette zur Turbulatorbereinigung	81
4.18	Methoden der Turbulatordetektierung	83
4.19	Bereinigte Kühlkanäle	84
4.20	Messdatenausrichtung anhand des CAD-Modells	85
4.21	Messdatenausrichtung anhand des Mittelwertnetzes	86
4.22	Ausrichtungsabweichung der CT-Messungen	87
4.23	Wahrscheinlichkeitsdichtefunktion zur Oberflächenabweichung zwischen CT- und GOM-Messung nach dem Alignment	87
4.24	Verteilungsidentifikation für c_{max}	89
4.25	Q-Q-Plots für die Verschiebung des Kerns 1	89
4.26	Korrelationsmatrix und zugehöriges Konfidenzintervall des Datensatzes	90
4.27	Abweichung der Korrelationsstruktur zwischen Messdaten und Realisierungen .	91
4.28	Vernetzte Fluiddomain mit Rotor	94
4.29	y^+ -Wert und Vernetzung von Kühlluftlöcher	96
4.30	Temperaturfeld der CHT Berechnung	98
4.31	Randbedingungen des FE-Modell	101
4.32	Vergleich der Von Mises Spannung zwischen Star-CCM+ und SC03	103
5.1	Processes for roughness and geometry measurements	108
5.2	Simulation-based evaluation of roughness	108
5.3	Roughness measuring system at TU Dresden	110
5.4	Workflow from measurement to roughness computation	112
5.5	Example of profile versus surface roughness evaluation	113
5.6	Validation of Alicona sensor against reference NanoFocus device	114
5.7	Measurement drid exemplary shown for Rotor 2 and difference between 10x and 20x lens	115
5.8	Rotor 2 suction side images of the two engines and comparison against contour plot of wall shear stress	116
5.9	Rotor 2 pressure side images of the two engines and comparison against contour plot of wall shear stress	116
5.10	Detailed roughness distribution over the suction side of Rotor 2 blades	117
5.11	Detailed roughness distribution over the pressure side of Rotor 2 blades	118
5.12	Approximate roughness distribution over the suction side of Rotor 2 blades. Red points indicate exact measurements used for RBF-based interpolation for the rest	119
5.13	Approximate roughness distribution over the pressure side of Rotor 2 blades. Red points indicate exact measurements used for RBF-based interpolation for the rest	120

5.14 Mean roughness distribution over the suction side through the entire HPC for both engines (before and after cleaning)	121
5.15 Mean roughness distribution over the pressure side through the entire HPC for both engines (before and after cleaning)	121
5.16 Roughness stripe measurement exemplary shown for the mid-span of a Rotor 2 blade	122
5.17 Example of CFD grid of rough bottom wall generated in ANSA	125
5.18 Volume CFD domain of smooth channel generated in ANSA	127
5.19 Validation of HYDRA LES of smooth channel flow ($Re_{\tau} = 540$) against reference LES and DNS data from the literature [46, 48]	129
5.20 Non-dimensional mean velocity profiles for HYDRA RANS and LES of two rough channel flow test-cases ($Re_{\tau} = 540$). Results of RANS with HYDRA roughness function implementation are also shown	130
5.21 Database of the 14 measured roughness geometries used for training the roughness function model	132
5.22 Non-dimensional mean velocity profiles corresponding to the first 6 cases of Fig. 4.21 for two different Reynolds numbers	133
5.23 Scatter plots of roughness parameters (input) and ΔU^+ (output) corresponding to the database of Fig. 4.21	134
5.24 Prediction of ΔU^+ using a model based on two roughness parameters	135
5.25 Approximate relation between ΔU^+ and Nikuradse's k_s^+ covering both transitional and fully rough regimes	136
5.26 Prediction of ΔU^+ using a new model based on five roughness parameters	137
6.1 Vereinfachter Überblick über scorep-db. Weitere scorep-db Befehle sind in Tabelle 6.2 aufgelistet.	145
7.1 Single Frame of a Cylinderflow trajectory	154
7.2 BSMGN architecture	155
7.3 Pairplot of sampling strategy (left), computational setup with visualization planes (top) and two stator samples (right), from [163]	161
7.4 Turbine stage efficiency over reduced massflow of train (black) and test (red) samples normalized to nominal case (sample A)	162
7.5 Comparison of radial cuts through Time-Averaged Stator Flowfield Predictions	164
7.6 Comparison of radial cuts through Time-Averaged Rotor Flowfield Predictions	165
7.7 Comparison of relative test set error for time-averaged full domain predictions (LT1) with median (dashed) and 5 and 95% quantiles (whiskers).	166
7.8 50% cuts through rollout of rotor domain predictions from model Vol-U for sample A	167
7.9 Nodewise relative Error for Surface Pressure Prediction of Test set with whiskers at 5th and 95th percentiles and dashed median.	169
7.10 Surface Pressure Distribution at 50% radial span for samples A and C.	170
7.11 Stator Surface Pressure prediction of sample A from model Surf-AV. Note different color scale for relative error.	171

7.12 Stator Surface Pressure prediction of sample C from model Surf-AV. Note different color scale for relative error.	172
7.13 Comparison of point with highest amplitude over time from sample A and C. . .	173

Tabellenverzeichnis

3.1	Liste relevanter Metadaten	30
3.2	Auflistungen der für die Instrumentierung von Hydra gelösten Score-P und Hydra Probleme	35
4.1	Objektivspezifische Merkmale	51
4.2	Profilparameter Turbine	74
4.3	Zellgrößen in verschiedenen Bereichen des CHT-Modells	95
4.4	Berechnungszeiten SC03	99
4.5	Zellgrößen in unterschiedlichen Bereichen des FEM Modells	100
5.1	Technical specifications of Alicona SensorR25	111
5.2	Differences between Alicona lenses and reference NanoFocus device	114
5.3	Information on the measured hardware	115
6.1	Vom Dataverse/ <i>ToMeDa</i> -Ansatz zur RDF-Lösung mit scorep-db.	144
6.2	Übersicht der scorep-db Befehle	147
6.3	Übersicht der verfügbaren scorep-db Daten-Backends	147
7.1	Features of Dist-BSMGN models	161
7.2	Hardware specifications from [185]	163
7.3	Mean RMSE and standard error for the turbine test set and training time for full domain time-averaged predictions (LT1).	166
7.4	Comparison of training time, model parameters and mean and standard error of RMSE over the turbine test set for LT2 models	168
7.5	Mean and standard error of RMSE and training time over the turbine test set for time-averaged surface predictions.	168
7.6	Training Costs and Average Inference Timings	174

Nomenklatur

Lateinische Buchstaben

Symbol	Einheit	Bezeichnung
A	m^2	Fläche
AR	–	Total-to-projected area ratio
B	–	Konstante
b		Zufallsvariable
b, c	m	Strecke
$\mathbf{b}$	–	Stichprobe
$c(x)$	m	Wölbungsverteilung
c_p	$\frac{J}{kg \cdot K}$	spezifische Wärmekapazität
C_1, C_2	$[m, m, m]$	Kamera
Cov	–	Kovarianz
d		Zero-plane displacement
$\mathbf{D}$		Materialmatrix
E	$\frac{J}{kg}$	Spezifische Gesamtenergie
E	$\frac{N}{m^2}$	Elastizitätsmodul
Err_{RMS}	–	RMS error of interpolation model for $\overline{Sa}$
ES		Effective slope parameter
$f(b)$	–	Wahrscheinlichkeitsdichtefunktion
f_b	$\frac{N}{m^3}$	Volumenkräfte
$F(b)$	–	Verteilungsfunktion
G	$\frac{N}{m^2}$	Schubmodul
h	–	lokaler Wärmeübertragungskoeffizient
I	–	Intervall
$\mathbf{I}$	–	Einheitsmatrix
IQR	–	Interquartilbreite, (engl.: Inter Quartile Range)
k_s		Equivalent sand-grain roughness parameter
K_u		Kurtosis parameter
L_r		Sensor lateral resolution parameter

R^2		Bestimmtheitsmaß (engl.: Coefficient of determination)
Ra		Mean profile roughness parameter
Re_τ		Reynolds number based on u_τ
S_a		Mean surface roughness parameter
$\bar{S}_a$		Average S_a over 5 blades
S_k		Skewness parameter
S_p		Maximum peak height
S_q		RMS roughness height
S_v		Maximum valley depth
O	[m, m, m]	Messpunkt
P	[m, m, m]	Projektor
P	–	Wahrscheinlichkeit
$\dot{Q}_E$	$\frac{W}{m^3}$	Energiequelle oder Energiesenke
Q_α	–	α – Quantil
U_0		Uniform inlet velocity
V_r		Sensor vertical resolution parameter
(x, y, z)		Axial, lateral, and vertical directions
p	Pa	statischer Druck
l	m	Länge
k	$\frac{W}{m^2}$	Wärmeleitfähigkeit
m	m	meridionale Koordinate
n	–	Anzahl
r	m	Radius
r	m	radiale Koordinate
r_k	–	Korrelationskoeffizient nach PEARSON
$\tilde{r}$	–	Korrelationskoeffizient nach SPEARMAN
pos	m	Position
$\dot{q}$	W	Wärmestrom
$\dot{q}''$	$\frac{W}{m^2}$	Wärmestromdichte
s	–	Realisierungen
s	–	relative Schaufelhöhe
s_i	–	Normalspannung
s_σ	–	empirische Standardabweichung
T	–	Zielkorrelationsmatrix
t	–	Laufvariable
t	m	Dicke
$t(x)$	m	Dickenverteilung
T	K	Temperatur
I	Pa	Spannungstensor
u^+		Normalized mean axial velocity
(u'^+, v'^+, w'^+)		Normalized RMS velocity fluctuations
u, v	–	relative Koordinaten

u, v	$\frac{m}{s}$	Fluidgeschwindigkeit
u_T	$\frac{m}{s}$	Schubspannungsgeschwindigkeit
$u\tau$		Friction velocity
u	$[m, m, m]$	relative Verschiebung
v	$[\frac{m}{s}, \frac{m}{s}, \frac{m}{s}]$	Geschwindigkeit
Var	–	Varianz
X	$[m, m, m]$	Punkt
x	–	relative Position
x, y, z	m	kartesische Koordinaten
y	m	Wandabstand

Griechische Buchstaben

Symbol	Einheit	Bezeichnung
α	K^{-1}	linearer thermischer Ausdehnungskoeffizient
α	°	Vorderkantenwinkel
α, β	°	Winkel
γ	°	Staffelungswinkel
ΔU^+		Downward shift of u^+ / Roughness function
ε	–	Dehnung
ε_T	–	thermische Dehnung
θ	°	Umfangskoordinate
κ	–	Krümmung
κ	–	Von-Karman-Konstante
μ		Mittelwert
μ	Pa s	dynamische Viskosität
ρ	$\frac{kg}{m^3}$	Dichte
σ	Pa	mechanische Spannung
σ		Standardabweichung
σ		Spannungstensor
$\dot{Q}_E$	$\frac{W}{m^3}$	Energiequelle oder Energiesenke
Q_α	–	α – Quantil
$\dot{Q}_E$	$\frac{W}{m^3}$	Energiequelle oder Energiesenke
Q_α	–	α – Quantil
σ_{Mises}	Pa	v.-Mises-Spannung
τ	$\frac{N}{m^2}$	Schubspannung
τ_w		Wall shear stress magnitude
K		Von Kármán constant

Indizes

Indexverzeichnis

Index	Bezeichnung
$\square_{ax}$	axial
$\square_b$	Eingangsgrößen
$\square_c$	Wölbung (engl.: camber)
$\square_{ch}$	Sehne (engl.: chord)
$\square_{CP}$	Flächenschwerpunkt (engl.: Center Point)
$\square_{dim}$	Dimensionen
$\square_h$	Nabe (engl.: hub)
$\square_i$	Zählvariable
$\square_f$	Fluid
$\square_{max}$	maximal
$\square_{mod}$	modifiziert
$\square_{ref}$	Referenz
$\square_s$	Gehäuse (engl.: shroud)
$\square_{sim}$	Realisierungen
$\square_{sec}$	Sektion
$\square_t$	Dicke (engl.: thickness)
$\square_{tan}$	tangential
$\square_{TE}$	Hinterkante (engl.: Trailing Edge)
$\square_w$	wandnah
$\bar{\square}$	arithmetische Mittelwert
$\square^+$	entdimensionierte Variable
$\hat{\square}$	Schwankungsanteil
$\dot{\square}$	erste Ableitung
$\ddot{\square}$	zweite Ableitung

Abkürzungen

Abkürzung	Bezeichnung
2D, 3D	zweidimensional, dreidimensional
CAD	Rechnergestützte Design (engl.: Computer-Aided Design)
CFD	Numerische Strömungsmechanik (engl.: Computational Fluid Dynamics)
CHT	Gekoppelte Wärmeübertragung (engl.: Conjugate Heat Transfer)
CLDA	gekrümmter Liniendetektor (engl.: Curved Linear Diode Array)
Col	Wichtigkeitskoeffizient (engl.: Coefficient of Importance)
DNS	Direct Numerical Simulations
eLHS	extended Latin Hypercube Sampling
FEM	Finite Elemente Methode
FVM	Finite Volumen Methode
gpds-fc	Gas Path Design System for Fans, Compressors and Turbines
HPC	High-Pressure Compressor
ICP	Nächste Punkt Verfahren (engl.: Iterative Closest Point)
LDA	Liniendetektor (engl.: Linear Diode Array)
LE	Vorderkante (engl.: Leading Edge)
LES	engl. Large Eddy Simulation
LHS	Latin Hypercube Sampling
MCS	Monte-Carlo-Simulation
PDE	partielle Differentialgleichung (engl.: Partial Differential Equation)
PS	Druckseite (engl.: Pressure Side)
RANS	Reynolds-gemittelten Navier-Stokes-Gleichungen
RBF	Radial Basis Function
RMS	Root-Mean-Square
SS	Saugseite (engl.: Suction Side)
SST	Shear Stress Transport
TE	Hinterkante (engl.: Trailing Edge)

Abkürzungsverzeichnis

Abkürzung	Bedeutung
AD-Test	ANDERSON-DARLING-Test
Col	Wichtigkeitsmaß (engl. <i>Coefficient of Importance</i>)
CSN	Zyklenzahl (engl. <i>Cycles Since New</i>)
eLHS	extendend Latin Hypercube Sampling
HDW	Hochdruckwelle
HPC	Hochdruckverdichter (engl. <i>High Pressure Compressor</i>)
HPT	Hochdruckturbine (engl. <i>High Pressure Turbine</i>)
KS-Test	KOLMOGOROV-SMIRNOV-Test
LHS	Latin Hypercube Sampling
LPC	Fan (engl. <i>Low Pressure Compressor</i>)
LPT	Niederdruckturbine (engl. <i>Low Pressure Turbine</i>)
mCol	modifiziertes Wichtigkeitsmaß (engl. <i>modified Col</i>)
MCS	Monte-Carlo-Simulation
MRO	Maintenance, Repair, and Overhaul
RP	Restricted Pairing
SCR	Sample-to-Coefficient Ratio
SRS	Simple Random Sampling

1 Zielstellung

1.1 Bezug des Vorhabens zu förderpolitischen Zielen

Die Luftfahrtindustrie steht vor der Herausforderung, ökonomische und ökologische Anforderungen gegeneinander abzuwägen. Die Weiterentwicklung der Triebwerkskomponenten spielt dabei eine zentrale Rolle, da mit steigenden Leistungsanforderungen an die Triebwerke sowohl die aerodynamischen als auch die strukturellen Belastungen stetig zunehmen.

Innerhalb des Verbundvorhabens DARWIN adressiert das Teilvorhaben der TU Dresden das Förderziel „leistungsfähige und effiziente Luftfahrt“ in der Programmlinie Industrie 4.0/Künstliche Intelligenz. Durch Einsatz von KI Verfahren bei der Analyse grosser Datenmengen im Rahmen von multidisziplinären Simulationen von Triebwerkskomponenten werden Ansätze entwickelt und validiert, um zukünftig den Test- und Verifizierungsaufwand zu reduzieren. Die Performance- und Skalierbarkeits-Optimierungen dienen der Effizienzsteigerung und sind eine wesentliche Voraussetzung, um besonders umfangreiche und detaillierte Simulationen überhaupt durchführbar zu machen. Das gilt insbesondere für die Kopplung mehrerer Simulationsdisziplinen. Hochgenaue Simulationsmodelle bilden den Grundbaustein für das Konzept des Digitalen Zwillings und die Vermessung von Verschleiß und Oberflächen-Rauheit ermöglicht eine lebenszyklus-übergreifende Bewertung dieser Effekte. Diese Erkenntnisse sind wegbereitend für die Optimierung der Komponenten und tragen zur Sicherung der Wettbewerbsfähigkeit und Verbesserung von Effizienz und Leistungsfähigkeit des Triebwerksentwurfs bei.

Das Zentrum für Informationsdienste und Hochleistungsrechnen (ZIH) und die Professur für Turbomaschinen und Flugantriebe (TFA) der TU Dresden leisten im Verbundprojekt DARWIN mit Rolls-Royce Deutschland und weiteren Projektpartnern einen wesentlichen Beitrag zur Weiterentwicklung innovativer Entwurfsverfahren und zur Anwendung von Methoden der Künstlichen Intelligenz in der Triebwerksentwicklung.

1.2 Gesamtziel des Vorhabens

Das Verbundvorhaben DARWIN beinhaltet die Weiterentwicklung, Anwendung und Validierung von innovativen digitalen Simulations- und Entwurfsverfahren sowie Methoden der Künstlichen

Intelligenz zur Verbesserung des fachübergreifenden Verständnisses von Triebwerkssystemen in drei Themenschwerpunkten:

1. Hochgenaue Simulation des multidisziplinären Verhaltens von Triebwerkssystemen mittels hybrider und skalen-übergreifender Methoden sowie die Performance- und Skalierbarkeitsverbesserung hochparalleler Simulationsläufe und insbesondere der Lastbalancierung gekoppelter Simulationen.
2. Erstellung digitaler Zwillinge von Triebwerkskomponenten und Einbindung der Daten von Entwicklungs-, Validierungs- und Betriebsphase des Produktlebenszyklus in das Konzept des „Digitalen Fadens“
3. Anwendung und Weiterentwicklung von Methoden der Künstlichen Intelligenz und des Maschinellen Lernens zur Beschleunigung der Design-Prozesse sowie zur Verbesserung der Qualität und Genauigkeit von Simulationsergebnissen

1.3 Wissenschaftliche und technische Arbeitsziele des Vorhabens

Das Zentrum für Informationsdienste und Hochleistungsrechnen (ZIH) arbeitete in AP1.4 an der Verbesserung des Last-Balancing von hochparallelisierten, gekoppelten Simulationscodes. Die gleichmäßige Lastverteilung war eine wichtige Voraussetzung für die optimale Ausnutzung der Rechenleitung von Hochleistungsrechnern. Besondere Anforderungen ergaben sich dabei, wenn Simulationsmodelle mit Kopplung unterschiedlicher Disziplinen vorlagen. Hierfür wurden einschlägige Software-Werkzeuge (Score-P) für die parallele Performance-Analyse weiterentwickelt und zur Anwendbarkeit auf gekoppelte parallel Codes zu erweitern. Diese Software-Werkzeuge wurden für die kontinuierliche Aufzeichnung von Metadaten zu Simulationsläufen und essentiellen Performance-Messdaten erweitert. Durch deren Anwendung auf den Rolls-Royce Code HYDRA wurden im Laufe des Vorhabens große Mengen Metadaten von Simulationen auf der TU Dresden Infrastruktur gesammelt.

Im Rahmen von AP2.1 wurden durch die Professur für Turbomaschinen und Flugantriebe (TFA) Methoden zum robusten und automatisierten Aufbau digitaler Zwillinge von Triebwerkskomponenten aus optisch vermessenen Geometriedaten (GOM-Scans) erforscht. Eine wichtige Voraussetzung für die Umsetzung des Digitalen Fadens war dabei die Verknüpfung aller Softwaretools vom GOM-Scan über die CAD-Geometrie zum Simulations-Code. Abschließend wurde am Beispiel einer Turbinenschaufel eine probabilistische, aerothermale Untersuchung von Produktionsvariationen durchgeführt.

In AP 2.3 wurde durch TFA die Vermessung von Rauheiten und Abnutzungseffekten und deren Rückführung in Simulationsmodelle adressiert. Durch die Vermessung der Oberflächenrauheit von Verdichterschaufeln mit bekanntem Lebenszyklus wurden Erkenntnisse über die typische Verschmutzung gewonnen. Der Vergleich mit gereinigten Schaufeln ergab darüber hinaus Aussagen zur dauerhaften Abnutzung und dem Effekt der Reinigung. Parallel wurden Oberflächengeometrien vermessen, um Ablagerungen und grösserskalige Geometrieänderungen zu detektieren. Eine Berücksichtigung von Oberflächenrauheit ist in derzeitigen Strömungssimulationen nicht gegeben. Daher wurden skalenauflösende Large Eddy Simulationen (LES)

durchgeführt, um Aufschluss über das Grenzschichtverhalten der Profilumströmung bei typischen Rauheiten zu erhalten.

Mit Methoden des Maschinellen Lernens (ML) für die Bereitstellung von Ersatz-Modellen wurden in HAP3 durch das ZIH in Kooperation mit TFA einerseits die Vorhersage von Rauheits-Parametern in Abhängigkeit des Lebenszyklus und andererseits des Grenzschichtverhaltens basierend auf den LES Daten untersucht. Eine Grundvoraussetzung dafür war im Design und Aufbau eines Daten- und Metadaten-Repositorys für die koordinierte und zentrale Sammlung von Daten und Metadaten insbesondere für Trainings- und Evaluations-Daten für die ML-Verfahren geschaffen.

2 Einleitung

2.1 Motivation

Ein zentrales Problem bei der Auslegung von moderner Flugtriebwerken ist die Komplexität dieser Flugtriebwerke, die Ingenieurinnen und Ingenieure zunehmend an die technischen Grenzen der Leistungsfähigkeit von Turbomaschinenkomponenten führt. Um diese Herausforderung zu bewältigen, wird in der Triebwerksentwicklung ein Modell des Triebwerks erstellt, das alle Triebwerksbauteile umfasst und das Betriebsverhalten des Triebwerks unter verschiedenen Umgebungsbedingungen beschreiben soll. Die Triebwerkskomponenten liegen während der Entwicklung zunächst als nominales Design vor, das in den jeweiligen Betriebspunkten die Anforderungen optimal erfüllen soll. Bereits geringfügige Abweichungen in der Konstruktion können erhebliche Auswirkungen auf die Einzelkomponente und in der Folge auf das Gesamtsystem haben. Technische bedingte Einschränkungen in der Fertigungsgenauigkeit und wirtschaftliche Abwägungen machen es unmöglich, diese Abweichungen vollständig zu vermeiden. Um diese Unsicherheiten zu kompensieren, werden im Modell Sicherheitsfaktoren berücksichtigt.

In der Triebwerksindustrie werden zur Berücksichtigung der Geometrieviationen bereits vielfach probabilistische Methoden eingesetzt, um den Einfluss von Fertigungsstreuungen auf die Schaufelaerodynamik und die Schaufelstruktur zu bestimmen. Ein wesentlicher Vorteil dieser Methoden liegt darin, dass bei Kenntnis der Fertigungsstreuungen von bereits im Einsatz befindlichen Triebwerken eine erste Abschätzung der Produktionsstreuungen auf neue Schaufelentwicklungen mit geringem Aufwand möglich ist. Durch die zunehmend Erfassung der Außen- und Innengeometrie von Turbinenschaufeln kann eine Datengrundlage zur Validierung und Verbesserung der initialen Annahmen erstellt werden. Die stetig steigende Rechenleistung eröffnet zudem das Feld neuartiger Berechnungsmethoden, die immer umfangreichere und genauere numerische Berechnungen erlauben.

Rolls-Royce Deutschland digitalisiert im Rahmen der Qualitätssicherung standardmäßig die gefertigten Turbinenschaufeln innerhalb ihrer Produktionskette. Dabei kommen neuartige Messverfahren zum Einsatz, die eine flächenhafte Vermessung der Außengeometrie sowie eine zerstörungsfreie Vermessung der Innengeometrie von Turbomaschinenschaufeln ermöglichen. Diese Daten eignen sich neben der Qualitätssicherung auch für die Entwicklung neuer Methoden zur Optimierung und Weiterentwicklung bestehender und zukünftiger Prozessket-

ten. Vor diesem Hintergrund werden im Arbeitspaket (AP) 2.1 die verfügbaren optischen und CT-Messdaten aus der Fertigung der Trent1000-Prozesskette analysiert und auf die BR000-Prozesskette übertragen, die die Grundlage für zukünftige Turbinenschaufeln bildet. Hierbei wird eine Monte-Carlo-Simulation mit Extended Latin Hypercube Sampling verwendet. Dabei erfolgt auf Grundlage der verarbeiteten und analysierten Messdaten eine parametrische Rekonstruktion der äusseren und inneren Schaufelgeometrie auf Basis der zuvor entwickelten Parametrisierungsmethoden. Diese Rekonstruktion bildet die Grundlage für eine anschließende CFD-Simulation mit aerothermaler Kopplung, um den Einfluss der Geometrievariation auf die Schaufeltemperatur und die Spannungsverteilung innerhalb der Schaufel zu ermitteln.

Die Arbeiten im AP 2.3 besteht hauptsächlich aus zwei Teilen und befasst sich mit Rauheitsuntersuchungen an in Betrieb befindlichen Hochdruckkompressoren (HPC)-Blättern, sowohl in Bezug auf Messungen als auch auf Simulationen. Im 1. Teil wurde ein automatisiertes Verfahren zur Durchführung detaillierter Oberflächenrauheitsmessungen der Schaufeln entwickelt. Insbesondere wurde ein hochgenauer Alicona-Sensor (in der Lage Rauheitswerte bis zu $0,1\text{ }\mu\text{m}$ zu erfassen) mit einem Pick-and-Place-Roboterarm kombiniert, und das System wurde darauf trainiert, Rauheitsmessungen an allen verschiedenen Rotorblättern eines 10-stufigen HPC durchzuführen. Zunächst wird das Alicona-Messgerät mit einem NanoFocus-Gerät ähnlicher Genauigkeit validiert. Anschließend wird eine detaillierte Messung der Rauheit auf der Saug- und Druckseite von Rotor 2 unter Verwendung eines 100-Punkte-Gitters demonstriert. Dieser Prozess wird durch die Verwendung einer reduzierten Anzahl von Messpunkten weiter beschleunigt. Die Lage dieser Punkte wurde durch numerische Optimierung bestimmt, die darauf abzielt, den Fehler zwischen einem RBF-basierten Näherungsmodell und der detaillierten gemessenen Rauheitsverteilung zu minimieren. Schließlich wird eine Zusammenfassung der Rauheitsmessungen durch das gesamte HPC gegeben (nur Rotoren; vor und nach der Reinigung der Schaufeln in einem Ultraschallbad) und endet mit einer Diskussion der Ergebnisse. Im zweiten Teil der Arbeiten im AP 2.3 werden zunächst Streifenmessungen der Saugseite der Schaufeln der Oberflächenrauheit durchgeführt, wobei erneut das hochgenaue Alicona-Messgerät verwendet wird. Dann werden diese Rauheitsverteilungen verwendet, um die Wände von Nulldruckgradienten, voll turbulenten Kanalflossimulationen zu konstruieren. Es werden körperangepasste unstrukturierte Gitter von bis zu 80 Millionen Knoten erzeugt, auf denen wand aufgelöste LES- sowie RANS-Simulationen mit dem $k-\omega$ SST-Turbulenzmodell durchgeführt werden. Das CFD-Setup wird zunächst in einem reibungslosen Kanalreferenzfall anhand von LES- und DNS-Daten aus der relevanten Literatur validiert. Darüber hinaus wird der Einfluss der Reynolds-Zahl auf mehrere grobe Kanalflossimulationen unter Verwendung von zwei verschiedenen Setups bei $Re=540$ und $Re=880$ untersucht. Schließlich wird nach einer Identifizierung der wichtigsten Rauheitsparameter (angegeben der relativ begrenzten Datenbank) ein neues Rauheitsfunktionsmodell vorgeschlagen, das die Vorhersage der Strömung über eine raue Oberfläche ermöglichen würde, ohne dass die Rauheitsskalen geometrisch aufgelöst werden müssen.

AP 1.4 erweitert die Messinfrastruktur Score-P um eine FAIR-konforme Metadatenerfassung und wendet sie am gekoppelten Strömungslöser Hydra an, um Performance transparent zu analysieren und zielgerichtete Optimierungen abzuleiten. Großskalige, gekoppelte Simulationen liefern enorme Mengen an Laufzeitdaten, deren Nutzen ohne Kontext begrenzt ist. Für

belastbare Analysen und Reproduzierbarkeit müssen neben Profilen/Traces auch maschinen-, build- und laufzeitbezogene Metadaten systematisch erfasst werden. Score-P ist ein etablierter Messstandard in HPC-Anwendungen, es fehlte jedoch eine integrierte, FAIR-konforme Metadatenschicht. Daher wird in diesem Projekt eine Lösung zur Nachverfolgbarkeit von Performance Daten erforscht und zur Analyse der Hydra-Lastbalancierung eingesetzt.

AP3.1 schafft die Grundlage für ein gemeinsames Daten- und Metadaten-Repository zur Performanceanalyse von Hydra-Simulationen. Ohne eine einheitliche, maschinenlesbare Erfassung von Konfigurations-, Laufzeit- und Analyseinformationen bleiben Ergebnisse schwer vergleichbar und kaum wiederverwendbar. Die wachsende Heterogenität von HPC-Systemen und Toolchains erzwingt eine Lösung, die sowohl lokal als auch online zuverlässig funktioniert. Ein Ansatz zur Lösung dieser Herausforderungen wird in diesem Projekt entwickelt und genutzt.

Und die Arbeiten in AP3.4 erkunden das Potential von datengetriebenen Ersatzmodellen zur Vorhersage des gesamten Strömungsfeldes. Die effiziente Vorhersage von Strömungsfeldern in Turbomaschinen stellt eine zentrale Herausforderung dar, da komplexe Geometrien und stark variierende physikalische Größen eine feine räumliche Auflösung und damit hohe Rechenressourcen erfordern. Herkömmliche Ersatzmodelle wie Modelle der Dimensionsreduktion oder empirische Korrelationen bieten zwar schnelle Vorhersagen, beschränken sich jedoch meist auf globale Kenngrößen oder wenige Oberflächenwerte. Datengetriebene Modelle auf Basis von Graph neuronalen Netzwerken (GNNs) können hier einen vielversprechenden Ansatz darstellen: Sie ermöglichen eine direkte Verarbeitung der numerischen Gitterstruktur und erlauben die Modellierung lokaler und globaler physikalischer Abhängigkeiten, ohne die zugrunde liegende Auflösung vollständig aufzugeben. Dadurch eröffnen sich neue Möglichkeiten für eine schnelle, präzise und flexible Strömungsvorhersage in der Turbomaschinenauslegung. Die Anwendung eines GNN-Ansatzes auf numerische Netze von industrieller Relevanz wird hier in diesem Projekt untersucht und die Herausforderungen diskutiert.

2.2 Aufbau des Berichts

Dieser Bericht fasst die Arbeiten in HAP1 an AP1.4, in HAP2 an AP 2.1 und 2.3, sowie in HAP3 in AP 3.1, 3.2 und 3.4. der TU Dresden zusammen. Insgesamt vier wissenschaftlichen Mitarbeitern bearbeiteten diese Arbeitspakete.

3 Teilbericht AP1.4

3.1 Einführung AP1.4

In AP1.4 wird die Messinfrastruktur Score-P um die Fähigkeit der Metadaten Sammlung erweitert. Zudem wird der gekoppelte Strömungslöser Hydra mittels Score-P analysiert und optimiert. In Abschnitt 3.2 werden Grundlagen über die genutzten Technologien vorgestellt. Dies umfasst die Nutzung von Score-P sowie die Voraussetzungen für eine korrekte Anwendung. Auch wird kurz in das Konzept von JSON-LD eingeführt, welches die Grundlage für die Metadaten-basierten Arbeitspakete in diesem Arbeitspaket und auch die Grundlage für die Arbeiten in AP3.1 aus Kapitel 6 bildet.

Abschnitt 3.3 umfasst die Arbeiten der folgenden Arbeitspakete:

- *AP1.4.5* Definition von Metadaten zu Code-Versionen, Build-Parametern, Input-Daten und Laufzeit-Parametern sowie der Ausführungs-Umgebung
- *AP1.4.6* Erweiterung der Score-P Code-Instrumentierung um die Sammlung von Metadaten zu Code-Versionen und Build-Parametern
- *AP1.4.7* Erweiterung der Score-P Laufzeit-Messumgebung um die Sammlung von Metadaten zu Anwendungsläufen

Die Ergebnisse der folgenden Arbeitspakete sind in Abschnitt 3.4 beschrieben.

- *AP1.4.1* Instrumentierung und Annotierung für Kopplungs-Operationen mittels Score-P
- *AP1.4.2* Vereinigung von Profilen bzw. Traces gekoppelter Anwendungen, Matching der Kopplungs-Operationen
- *AP1.4.3* Gemeinsame Darstellung des parallelen Laufzeitverhaltens gekoppelter Anwendungen, Hervorhebung der Kopplungs-Operationen
- *AP1.4.4* Anwendung auf Hydra, Performance-Analyse und Ableitung von Tuning-Empfehlungen, Umsetzung und Evaluierung der Empfehlungen
- *AP1.4.8* Analyse der Lastverteilung von Simulationen großer gekoppelter Modelle (installationär, inklusive In-situ Postprocessing)

In Abschnitt 3.5 werden die Ergebnisse zusammengefasst.

3.2 Ausgewählte Grundlagen

Dieses Kapitel beschreibt ausgewählte Grundlagen, welche für das Verständnis der Arbeitspaket-beschreibungen notwendig sind. Es ist unterteilt in die Grundlagen der Metadatenbeschreibung mittels JSON-LD und die Grundlagen der Performance-Analyse und von Score-P.

```

1 {
2   "@context": {
3     "type": "@type",
4     "name": "http://schema.org/name",
5     "homepage": {
6       "@id": "http://schema.org/url",
7       "@type": "@id"
8     },
9     "Person": "http://schema.org/Person"
10  },
11  "type": "Person",
12  "name": "Alice Beispiel",
13  "homepage": "https://alice.example.org"
14 }

```

Quelltext 3.1: Beispielhafte JSON-LD Auflistung. Durch den @context Block wird eine reguläre JSON Struktur zu einer JSON-LD Struktur erweitert.

```

1 {
2   "@type": "http://schema.org/Person",
3   "http://schema.org/name": "Alice Beispiel",
4   "http://schema.org/url": {
5     "@id": "https://alice.example.org"
6   }
7 }

```

Quelltext 3.2: Kompakte JSON-LD Darstellung des obigen Beispiels

3.2.1 JSON-LD: JSON für verlinkte Daten

Das JSON-LD (JSON for Linked Data) [159] Format wird in diesem Projekt für die Beschreibung von Metadaten genutzt. Es ist eine Notationsart von RDF, was in Abschnitt 6.2.1 näher beschrieben wird.

JSON (JavaScript Object Notation) [22] ist ein weit verbreitetes Datenformat zur strukturierten Darstellung von Informationen in Schlüssel-Wert-Paaren. JSON-LD (JSON for Linked Data) ist eine Erweiterung von JSON, die es ermöglicht, Daten mit Bedeutung zu versehen und in größere Wissensnetzwerke einzubetten. Der zentrale Mechanismus dabei ist der @context-Block, der Begriffe aus dem JSON mit standardisierten Bedeutungen verknüpft. Ein gewöhnliches JSON-Dokument muss keine JSON-LD-spezifischen Elemente wie @context, @id oder @type enthalten – es bleibt damit ein reguläres JSON. Erst durch Hinzufügen dieser Elemente wird es zu einem gültigen JSON-LD-Dokument. In Listing 3.1 ist ein valides JSON mit @context gezeigt; Listing 3.2 zeigt die kompakte Darstellung. JSON-LD ermöglicht so eine nahtlose Integration in bestehende JSON-Workflows, ohne dass grundlegende Strukturen geändert werden müssen, erlaubt allerdings ebenso die Integration in RDF-basierte Workflows. Diesen Umstand macht sich die Metadatenerweiterung von Score-P zunutze (siehe Abschnitt 3.3), um in der Wahl der Postprozessoren flexibel zu bleiben.

3.2.2 Datenweitergabe durch den Linker

Der Linker übernimmt im Softwarebauprozess die Aufgabe, den vom Compiler in Objekte übersetzten Quellcode in eine ausführbare Datei zusammenzuführen. Derselbe Mechanismus kann genutzt werden, um beliebige sogenannte Sektionen (Sections, Teil einer Objektdatei) aus den verschiedenen Objekten zusammenzuführen. Dazu müssen die Sektionen denselben Namen haben, dann werden die beliebigen Inhalte (mit `\0` terminiert) aneinandergereiht. In diesem Projekt wird ein JSON-String, der Metadaten enthält, an jedes Objekt angefügt. Der Linker übernimmt dann das Zusammenführen der Strings in einen einzelnen String. Diese Eigenschaft garantiert, dass nur die Metadaten genutzter Objektdateien in der ausführbaren Datei (Executable) vorliegen. Dies beugt Fehlern vor, durch eine Nachimplementierung von Kompilier- und Linkervorgang entstehen könnten.

3.2.3 Der Performance Engineering Zyklus auf Basis von Score-P

Score-P [92] ist eine hochskalierbare Messinfrastruktur für parallele Applikationen. Es wird sowohl für *Profiling* als auch *Tracing* genutzt und bildet so die Messinfrastruktur in diesem Projekt.

Score-P ist Bestandteil des OTF2 (OpenTraceFormat2) [46] Ökosystems. OTF2 stellt das gemeinsame Dateiformat für Trace Event Daten dar und bildet so die Basis für, unter anderem, folgende Performance Tools: lo2s [78], Scalasca [57] und Vampir [91], im weiteren Umfeld aber auch Cube [144] und Extra-P [28].

Diese Tools können im Kontext eines **Performance Engineering Zyklus** genutzt werden, um iterativ Performance Probleme zu erkennen, zu analysieren und zu beheben (Abb. 3.1). Weitere Informationen sind auf der Score-P Website ¹ zu finden.

Für die oben genannten Tools erzeugt Score-P die Datengrundlage in Form von OTF2 oder Cubex Dateien, wobei erstere Trace Event Daten und letztere Profil-Daten enthalten. Trace Event Daten enthalten ENTER und LEAVE Events, welche das Betreten und Verlassen von Programm- und Kommunikationsprozeduren samt Zeitstempel mitschreiben, und so den Programmablauf beschreiben, und Kommunikationsevents wie z.B. MPI_Send oder MPI_Recv, welche die Kommunikation auf semantischer Ebene beschreiben. Aus diesen Informationen lässt sich ein Programm- und Kommunikationsablauf rekonstruieren, welcher sich mit den Tools Scalasca oder Vampir weiter analysieren lässt.

Profil-Daten lassen sich aus Trace Event Daten ableiten, indem durch Akkumulation die zeitliche Dimension aufgehoben wird. Das Ergebnis ist die Ausführungsdauer und Anzahl der Prozeduraufrufe, nicht jedoch die genauen Zeitpunkte dieser.

Somit stellen beide Datenarten eine wichtige Informationsquelle für die Laufzeitdaten dar, die für das iterative Vorgehen im Performance Engineering Zyklus genutzt werden.

Sowohl Trace Event Daten als auch Profil-Daten können durch zwei Methoden gewonnen werden: Durch Instrumentierung und durch Sampling.

Bei der Instrumentierung wird bei Prozeduraufruf und -rückkehr eine Score-P Funktion aufgerufen, welche die zuvor genannten, zeitaufgelösten ENTER und LEAVE Events erzeugt. Die

¹<https://perftools.pages.jsc.fz-juelich.de/cicd/scorep/tags/latest/html/index.html>

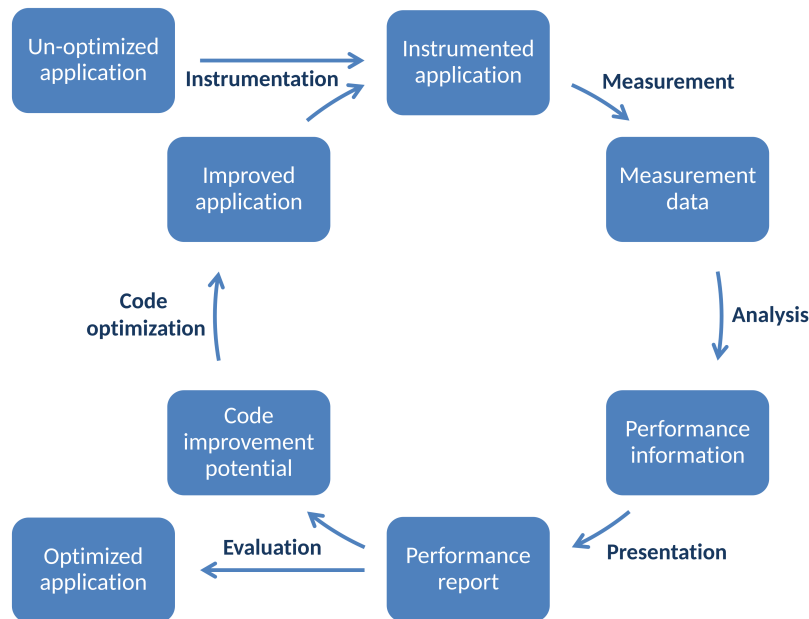


Abbildung 3.1: Performance Engineering Zyklus

Instrumentierung geschieht entweder automatisch durch ein Compiler-Plugin oder durch händische Annotation.

Anders ist es beim Sampling. Hier wird ein externes Signal genutzt, um das Programm zu unterbrechen und die aktuell ausgeführte Prozedur aufzuzeichnen. Dieses externe Signal kann beispielsweise eine vorgegebene Frequenz oder das Überschreiten eines zuvor definierten Grenzwertes einer gewählten Metrik, wie etwa die Anzahl abgearbeiteter Instruktionen, sein. Abbildung 3.7 zeigt ein Beispiel OTF2 Trace, das mithilfe von Instrumentierung aufgezeichnet wurde. Zudem dient die Abbildung als Einführung in das zur Analyse genutzte Tool Vampir. Dort sind in der Hauptansicht (Master Timeline) längs der x-Achse die Zeit aufgetragen, und die verschiedenen MPI Prozesse längs der y-Achse. Auf der rechten Seite sind eine Funktionszusammenfassung und eine Übersicht von Festplattenzugriffen und deren Art abgebildet.

3.2.4 Grundlagen und Nutzung von Score-P

Nutzerfreundlichkeit hat bei der Score-P Entwicklung einen hohen Stellenwert. Dies beinhaltet unter anderem, dass die Nutzung von Score-P für den Nutzer transparent, d.h. ohne Modifikation des Programmquelltextes erfolgen kann. Um dies zu gewährleisten, muss Score-P sich selbst bei Programmstart initialisieren und bei Programmende automatisch finalisieren und die gesammelten Profil oder Trace Event Daten zur späteren Analyse speichern. In der Startphase werden die Subsysteme initialisiert, welche MPI, Threads, CUDA oder auch I/O Aktivitäten aufzeichnen.

Damit dieses Konstrukt funktioniert, muss eine Applikation vollumfänglich instrumentiert werden, d.h. Score-P muss in jeden Compiler- und Linkeraufruf involviert werden. Dies ist durch

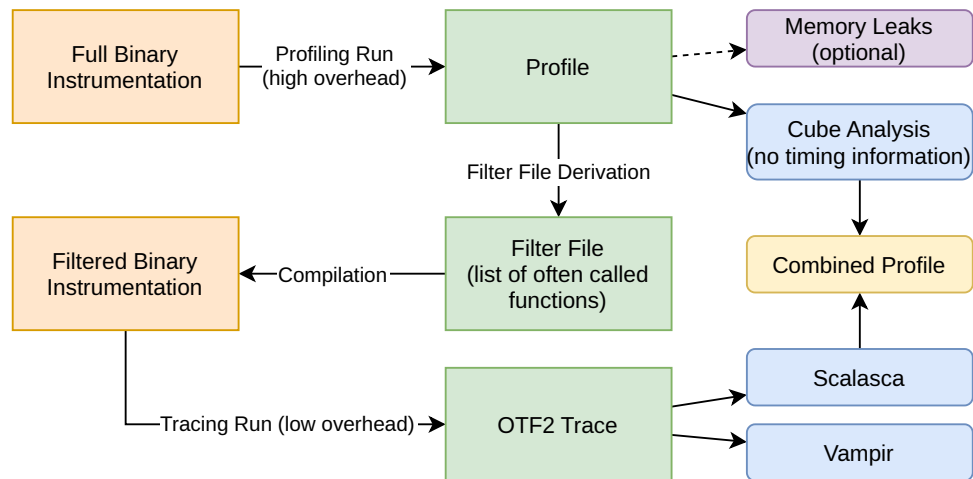


Abbildung 3.2: Arbeitsablauf Instrumentierung

das Setzen des Compiler-Wrappers für die jeweilige Sprache möglich, wie beispielsweise mit

```
CC=scorep-gcc CXX=scorep-g++ FC=scorep-gfortran make
```

für ein C/C++/Fortran Programm. Da die Compiler häufig auch für das Linken genutzt werden, ist das Setzen der Umgebungsvariablen häufig die einzige Änderung, welche zur Nutzung von Score-P notwendig ist. Score-P wird ebenfalls durch Umgebungsvariablen wie `SCOREP_WRAPPER_INSTRUMENTER_FLAGS` konfiguriert. Damit können, wie zuvor beschrieben, eine Funktionsfilterdatei angegeben werden, welche für eine effektive Nutzung vonnöten ist.

Die Kommunikationsevents von z.B. MPI werden in den Postprocessing Tools wie Scalasca oder Vampir abgeglichen und zusammengesetzt. Daher muss das instrumentierte Programm diese standardkonform nutzen, denn sonst schlägt dieser Schritt aufgrund eines semantischen Fehlers fehl.

Mit den Aufzeichnungen von Score-P beginnt der Performance Engineering Zyklus. Die Aufzeichnung erfolgt mithilfe von Sampling oder Instrumentierung. Letzteres wird aufgrund der höheren Genauigkeit vorwiegend verwendet. Nachteil des Aufzeichnens aller Funktionen ist ein erhöhter Overhead, welchen man durch gezielten Ausschluss von Funktionen (Filtern) reduzieren muss, ansonsten sind die Messwerte verzerrt.

Um diese nachteiligen Effekte soweit wie möglich zu reduzieren, kann, mithilfe des Hilfsprogramms `scorep-score`, ein Funktionsfilter erstellt werden, der Funktionen von der automatischen Compilerinstrumentierung ausschließt. Dies wird zum Beispiel für kurze und wiederholt auftretende Funktionen angewendet, die keinen informativen Wert für die Analyse haben.

Der übliche Arbeitsablauf ist in Abb. 3.2 dargestellt. Ausgehend von einer vollständig instrumentierten Applikation wird ein Profil erstellt, das bereits Analysen ohne Zeitbezug ermöglicht (z.B. eine Untersuchung auf Speicherlecks). Aus dem Profil wird ein Filter erstellt, der in der erneuten Instrumentierung zu einer Overhead-armen Messung führt. Diese erlaubt auch Analysen des zeitlichen Ablaufs mittels Scalasca oder Vampir und kann im Ergebnis mit anderen Messungen kombiniert werden.

3.3 Score-P Metadaten Schema auf JSON-LD Basis

Zur Analyse und Reproduzierbarkeit von Performance-Daten ist es entscheidend, nicht nur die Messdaten selbst, sondern auch deren Entstehungskontext zu dokumentieren. Daher ist Score-P um Metadaten erweitert worden. Die bereits existierenden Ausgabedaten von Score-P befinden sich in einem gemeinsamen Ordner. Dort befinden sich bisher eine `Manifest.md` und eine `scorep.cfg` Datei, und zusätzlich, je nach Auswahl, Profil und Trace Event Daten. In der Ersten ist eine Übersicht der Daten zu finden, welche Dateien sich nach Konfiguration in diesem Ordner befinden sollten, in der Letztere befinden sich Schlüssel-Wert-Paare, die den Zustand der Score-P Konfigurationsumgebungsvariablen zum Zeitpunkt der Messung wiedergeben.

Aus diesen Daten ergibt sich der Zustand von Score-P, allerdings nicht der Zustand der vermessenen Applikation. Unbekannt sind so z.B. die Problemgröße, die genutzte Applikationskonfiguration und die genutzte Softwareumgebung, wie beispielsweise MPI Implementierung oder Compilerversion. Diese Informationen sind aber wichtig, um das Gemessene in einen Kontext zu bringen und Vergleiche zu ermöglichen. Zu diesem Zweck wurde ein Mechanismus implementiert, der im Ausgabeordner eine weitere Datei namens `scorep.fair.json` erzeugt. Diese enthält die benötigten Metadaten im JSON Format. Die Metadaten-datei enthält einen `@context` Block, welcher durch Score-P definiert wird, und die JSON Datei so zu einer JSON-LD Datei macht (vgl. Abschnitt 3.2.1).

Im Folgenden werden die Metadatenstruktur als auch die technische Umsetzung beschrieben.

3.3.1 Metadatenschema Definition

Aus einer Analyse wünschenswerter Metadaten ist Tabelle 3.1 entwickelt worden. Mindestens diese gilt es aufzuzeichnen.

Name	Typ	Quelle	Erhebungszeitpunkt	Kommentar
Compiler Version	String	intern	Übersetzen	z.B.: "gcc (Ubuntu 12.3.0 22.04) 12.3.0"
Compiler Flags	String	intern	Übersetzen	z.B.: "-O3 -mavx2"
MPI Version	String	intern	Laufzeit	z.B.: "MPICH 4.2.1"
Software Git Hash	String	extern	Übersetzen	z.B.: "845eebd5"
Abhängigkeiten	–	extern	(Übersetzen/) Laufzeit	z.B. die Applikationskonfiguration
Quelltext-Abhängigkeiten	–	intern	Übersetzen (/Laufzeit)	Der Erhebungszeitpunkt hängt von der Nutzung einer statischen oder dynamischen Bibliothek ab

Tabelle 3.1: Liste relevanter Metadaten

Aus diesen Anforderungen wird ein Schema definiert, um die Daten strukturiert ablegen zu können. Die Entwicklung des Schemas folgt der Empfehlung aus dem Buch *Datenbanksysteme - Eine Einführung* [87].

Integration weiterer Metadaten

Allein aus den Compiler Flags und Compiler (als *intern* in Tabelle 3.1 bezeichnet) wird der Zustand der Applikation nicht vollständig beschrieben. So fehlt beispielsweise die Information über die genutzte Quelltextversion um die gesamten Kompilierungsmetadaten zu erhalten. Diese *externen* Metadaten sind essentiell für die gesamtheitliche Beschreibung.

Um diese Informationen dem Metadatensatz hinzuzufügen, muss eine Score-P Schnittstelle geschaffen werden. Diese Schnittstelle ist in Form der Umgebungsvariable `SCOREP_INSTRUMENTER_PRECIOUS_ENVIRONMENT_VARIABLES` implementiert worden. In dieser speichert man den Namen einer anderen Umgebungsvariable, deren Wert dann gespeichert wird. So ist es beispielsweise sinnvoll, die Quelltextversion in einer Umgebungsvariable zu speichern und deren Name als Wert der zuvor genannten Umgebungsvariable anzugeben.

Auch zur Laufzeit der instrumentierten Programms sind vom Nutzer angegebene Informationen hilfreich, wie möglicherweise überschriebene Umgebungsvariablen. Diese lassen sich mittels der Umgebungsvariable `SCOREP_PRECIOUS_ENVIRONMENT_VARIABLES` spezifizieren. Da die Menge an möglicherweise sinnvollen Umgebungsvariablen groß und für den Nutzer nicht überschaubar ist, definiert Score-P einen Satz von Umgebungsvariablensets, welche vordefinierte Umgebungsvariablen beinhalten. Es existieren Sets mit den Namen `default`, `slurm`, `pbs`, `cuda`, `rocm` und `openmp`. Genaue Informationen können mittels `scorep-info config-vars -full` eingesehen werden. Wie zuvor ist auch die manuelle Angabe von Umgebungsvariablen möglich. Je nach Software lassen sich immer noch nicht alle Informationen speichern, denn wenn eine Applikation über eine Konfigurationsdatei gesteuert werden kann, fehlen diese Verhaltensändernden Informationen bei der Analyse der Score-P Ausgabe. Diese Verbindung wird durch RDF hergestellt, was in Abschnitt 6.4.3 näher beleuchtet wird. Durch die Nutzung des RDF Ansatzes lassen sich weitere RDF Daten einfach zu der Beschreibung hinzu fügen, indem die Solverkonfiguration mit einem nutzerdefinierten Prädikat mit der `scorep:ExperimentRun` Entität verlinkt wird. Diese lässt sich mittels `scorep-info show-metadata experiment-id` ausgeben, und kann vom Nutzer frei verwendet werden.

Die vom Nutzer spezifizierten Metadaten gelten in diesem Kontext als *extern*.

3.3.2 Übersicht über Metadatenimplementierungsentwürfe

Aufbauend auf dem Schema muss ein Datenformat ausgewählt oder gegebenenfalls entwickelt werden, welches diese Daten speichern kann und ggf. noch weiteren Anforderungen standhält. Folgende Anforderungen haben Einfluss auf die Wahl des zu nutzenden Datenformates.

- Direkte Nutzbarkeit
- Umsetzung mit geringem technischen Aufwand
- Nutzung in höheren Abstraktionsebenen für Interoperabilität

Je nach Gewichtung dieser Anforderungen ergeben sich verschiedene Lösungsmöglichkeiten. Der letzte Punkt wurde erst während der Bearbeitung als Aufgabe des Dateiformates identifiziert. In diesem Projekt wurden zwei Ansätze implementiert und im Anschluss einer ausgewählt. Dabei ist *Entwurf 2* auf Basis von JSON-LD letztendlich ausgewählt worden. Der Vollständigkeit halber ist hier auch *Entwurf 1* aufgeführt.

Entwurf 1 – Eigenes Datenformat

Die erste Implementierung priorisierte geringen Aufwand und direkte Nutzbarkeit; eine höhere Abstraktionsebene war nicht vorgesehen. Dafür wurde ein eigenes, an `git fast-import` angelehntes und in ABNF definiertes Format entworfen, dessen Metadatenschnipsel per Linker in der Sektion `.scorep.instrumenter.metadata` verkettet und per Python-Skript nach JSON übersetzt wurden. Der Ansatz wurde verworfen, weil die Konvertierung im Nachverarbeitungsschritt konzeptionell falsch ist (enge Kopplung, falsche Zuständigkeit) und ein tragfähiges Konzept für Vergleichbarkeit über mehrere Datensätze fehlte.

Entwurf 2 – JSON-LD basierend

Basierend auf den Erfahrungen von Entwurf 1 wurde im zweiten Entwurf die Nutzbarkeit priorisiert und auch Erkenntnisse hinsicht der Implementierung berücksichtigt.

Zur Erhöhung der direkten Nutzbarkeit erzeugt Score-P nun eine JSON Datei mit den Metadaten, wie bereits beschrieben, welche direkt, ohne Konvertierungsschritt, als JSON eingelesen werden kann. Zur Nutzung in einem abstrakten Kontext kann diese JSON Datei als JSON-LD [159] interpretiert werden. Score-P definiert dazu den nötigen Score-P Kontext (@context Block). Die JSON-LD Datei kann anschließend in eine RDF Datenbank, auch Triple-Store genannt, importiert werden, und steht dort zur experimentübergreifenden Nutzung zur Verfügung.

Bei der technischen Umsetzung wird wie bereits in Entwurf 1 auf einen MetadatenString gesetzt, der durch den Linker aneinandergereiht wird. Der Unterschied besteht darin, dass der String nun ein JSON Objekt darstellt. Diese Darstellung lässt sich allerdings nicht verketteten, welcher in der Implementierung beachtet werden muss. Somit muss ein Korrekturschritt nach dem Linken vorgenommen werden, welche die JSON Objekte der einzelnen Objekte in eine Liste von Objekten umwandelt, und so wieder ein valides JSON konstruiert.

Das Auslesen und die Nutzung der Metadaten wird in Abschnitt 6.4.3 weiter beleuchtet.

Aufgrund der leichteren Anwendbarkeit und nur kleineren technischen Hürden, wurde dieser Entwurf 2 umgesetzt.

3.3.3 Umsetzung mit RDF/JSON-LD

Ausgehend von den Anforderungen zur Umsetzung wurde ein Metadatenschema auf RDF-Basis entwickelt. Abb. 3.3 zeigt das Metadatenschema, welches die in Tabelle 3.1 identifizierten Metadaten enthält.

Insbesondere bei der Nutzung von RDF ist es wichtig, Entitäten genau zu identifizieren. Die Identifikation eines Objektes erfolgt über die IRI (Internationalized Resource Identifier), welche die internationalisierte Form der URI (Uniform Resource Identifier) darstellt. Die RDF-Subjekte

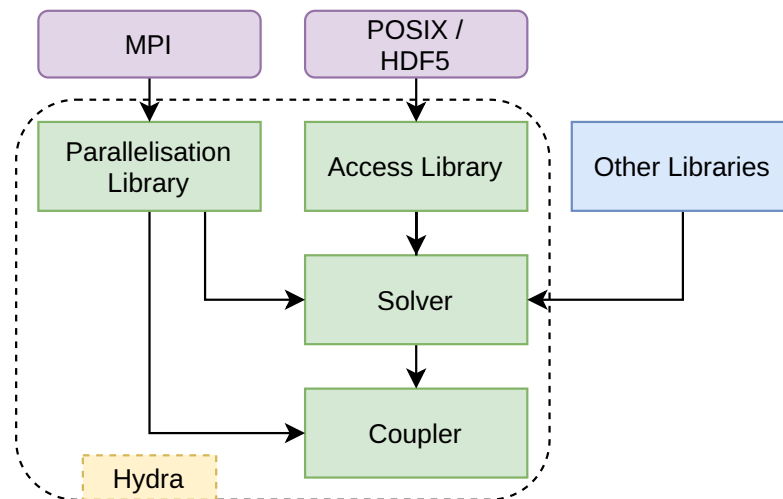


Abbildung 3.4: Darstellung der Hydra Bibliotheksabhängigkeiten

Strömungslöser mit Score-P instrumentiert und analysiert, Verbesserungen vorgeschlagen und diese teilweise implementiert.

3.4.1 Score-P Integration in Hydra

Nachdem zuvor sowohl Score-P, das dazugehörige Ökosystem als auch die Integration der Metadatenerhebung beschrieben wurden, wird in diesem Kapitel die konkrete Implementierung in Hydra beschrieben.

Wie zuvor beschrieben, stellt Score-P einen Compiler-Wrapper zur Verfügung, welcher den ursprünglichen Compiler um die Score-P Funktionalität erweitert. Dieser Wrapper soll zur einfacheren Nutzung in den Hydra-Kompilierungsvorgang integriert werden.

Hierzu muss die Softwarestruktur des Strömungslösers verstanden werden - samt Bibliotheksabhängigkeiten - um sicherzustellen, dass der Score-P Wrapper an allen benötigten Stellen zum Einsatz kommt. Diese Abhängigkeit ist in Abb. 3.4 abgebildet. Der Hydra Kompilierungsvorgang basiert auf einem Bash-Skript, welches die verschiedenen Bibliotheken ansteuert und dort mittels make kompiliert. Durch die Nutzung von Hilfsfunktionen innerhalb dieses Bash-Skripts lässt sich die Instrumentierung auf Bibliotheksebene ein- und auch ausschalten. Somit sind auf Hydra-Buildsystemebene nur minimale Änderungen notwendig.

Konzeptionell ist die Instrumentierung damit möglich. In der Praxis traten jedoch Probleme auf (siehe Tabelle 3.2).

Hervorzuheben ist das Einführen einer Beendigungsprozedur, welche die gekoppelte Simulation korrekt im Sinne des MPI Standards herunterfährt. Weitere Informationen sind in der Bildunterschrift von Abb. 3.5 zu finden.

3.4.2 Score-P Instrumentierung von ParaView Catalyst

Score-P kann mithilfe eines Plugins auch Python-Programme profilieren und tracieren. Obwohl Hydra in Fortran und zentrale Bibliotheken in C oder C++ geschrieben sind, ist auch das Tracieren

Problem	Lösung
Objektdateien enden nicht mit ".o"	Namenskonvention geändert, damit diese auf ".o" enden
__FILE__ Makro wird durch Score-P zu lang für "Fixed Form Fortran" expandiert, da Score-P den absoluten Pfad nutzt	Nutzen der <code>-ffixed-line-length-none</code> Compiler Option für gfortran
Definition von Funktionen mit Namen <code>open</code> und <code>close</code> , die zudem mit <code>extern</code> versehen sind. Verhindert die Nutzung von Score-P I/O Tracing	Definition dieser Funktionen als <code>static</code> .
Deadlock von Score-P in <code>MPI_Wait</code> Funktion	Debugging mit DDT: Identifikation und Lösung eines 12 Jahre alten Fehlers der Fortran MPI Modul von Score-P
Abbruch der Simulation aufgrund von <code>MPI_Invalid_Tag</code> durch Fehler im Score-P <code>MPI_Comm_get_attr</code> Fortran Wrapper Routine und durch manuelle Nutzung eines <code>MPI_TAG</code> als Signal mit einem Wert außerhalb des vom Standard definierten Bereichs	Verzicht auf <code>MPI_Comm_get_attr</code> und Setzen der <code>MPI_TAG</code> Werte auf einen validen Bereich
Nicht MPI-konforme Beendigungsprozedur	Einführen einer eleganten Lösung mittels <code>MPI_Ibarrier</code> (siehe Abb. 3.5)
Nicht MPI-konforme Nutzung von <code>MPI_Irecv</code> beim optionalen Empfangen für Nachrichten	Einführen einer Lösung basierend auf <code>MPI_Iprobe</code>

Tabelle 3.2: Auflistungen der für die Instrumentierung von Hydra gelösten Score-P und Hydra Probleme

von Python wichtig, denn Python wird für Konfiguration des Catalyst In-Situ Postprocessings verwendet.

Die Nutzung des Score-P Plugins innerhalb eines Catalyst-Skripts stellt eine besondere Herausforderung dar. Üblicherweise greifen die Score-P Python Bindings auf das Score-P Modul zurück, welches beim Starten des Python-Skripts mit angegeben wird

```
python3 -m scorep <my_script.py>
```

Da ParaView allerdings Python integriert, und dieses aus Catalyst-Komponente, also dem laufenden C/C++/Fortran Programm, aufgerufen wird, ist dieser Ansatz nicht möglich. Stattdessen muss auf die bereits initialisierte Score-P Komponente direkt zugegriffen werden, ohne Initialisierung über das Python-Modul. Um die Instrumentierung ein- bzw. auszuschalten, wird direkt auf die Instrumenter-API zurückgegriffen. Die Nutzung ist in Listing 3.3 abgebildet.

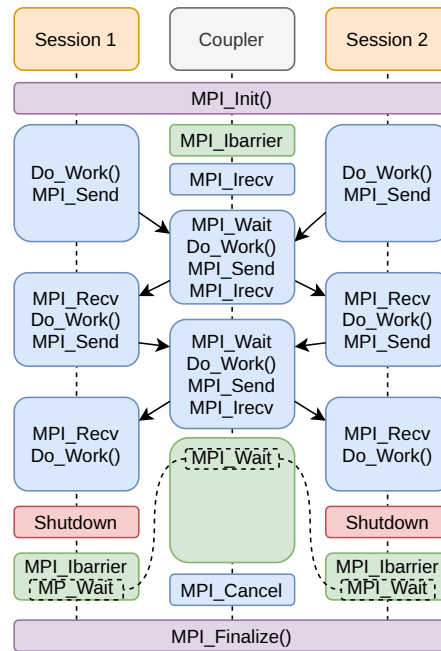


Abbildung 3.5: Kommunikationsmuster von zwei gekoppelten Simulationen samt Beendigungsprozedur mittels MPI_Ibarrier (grün). Die Nutzung dieser dient zur Zustandsynchronisation und signalisiert dem Kopplungsprozess, dass beide Simulationen (Sessions) beendet sind, und dass sich jener auch beenden kann.

3.4.3 Erweiterung und Optimierung der Mess- und Analyseinfrastruktur

Score-P soll im Rahmen dieses Projektes genutzt werden, um die Lastverteilung eines gekoppelten Strömungslösers zu bestimmen. Für die Performance Analyse eines gekoppelten Löser muss dieser mit Score-P instrumentiert werden können. Ob dies der Fall ist, hängt von der konkreten Implementierungsart der Kopplung ab. Der im Projekt verwendete Strömungslöser wird als monolithische Applikation gestartet. Dieser Ansatz ist ein übliches MPI Verwendungsmuster und benötigt auf der Datenaufnahme keine Änderungen von Score-P. Allerdings müssen Fehler sowohl auf Score-P als auch Strömungslöserseite behoben werden (siehe Tabelle 3.2).

Funktionserweiterung CubeLib

Für die Analyse von gekoppelten Anwendungen wie in Abb. 3.5 gezeigt, sind Änderungen an CubeLib notwendig. CubeLib ist eine Bibliothek für die Interaktion mit Profiles und wird gemeinsam mit Score-P installiert.

CubeLib verfügt über mehrere Programme zur Nachbearbeitung/Exploration von Profiles. So müssen Profile der monolithischen, intern task-parallelen Applikation in ihre Sitzungen unterteilt werden, um die Eigenschaften einer jeder einzelnen Sitzung untersuchen zu können.

```

1 from paraview.simple import *
2 try:
3     import scorep.instrumenter
4     scorep.instrumenter.get_instrumenter(True, "profile").register()
5 [...]
6 <ParaView Catalyst Python Code>
7 [...]
8 try:
9     scorep.instrumenter.get_instrumenter(False, "profile").unregister()

```

Quelltext 3.3: Manuelles Ein- und Ausschalten für die Python-Instrumentierung innerhalb eines ParaView Catalyst Skripts

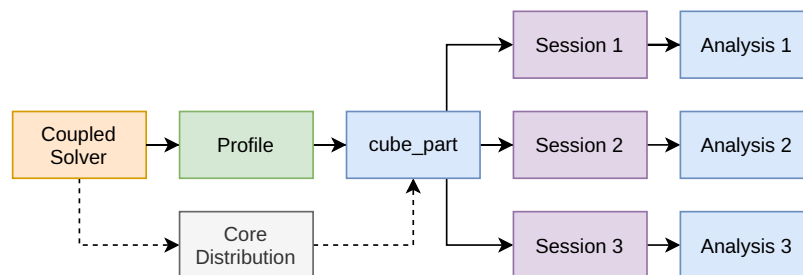


Abbildung 3.6: Konzeptuelle Analyse einzelner Lösesitzungen innerhalb des gekoppelten Strömungslösers

Die aktuelle CubeLib Implementierung macht einen Fehler mit Profiles, welche durch die Nutzung von `cube_part` nur einen Teil der Prozessorkerne beinhalten, wie beispielsweise nur die Prozessorkerne für eine alleinstehende Strömungslösesitzung innerhalb des gekoppelten Programmes. Logisch sind die Informationen der abgeschnittenen MPI Ranks vorhanden, allerdings mit Werten die auf 0 gesetzt sind. Sämtliche Berechnungen wie beispielsweise mittlere Laufzeit und parallele Effizienz sind im Ergebnis falsch, weil diese abgeschnittenen MPI Ranks mit Wert 0 in die Berechnung eingehen.

Dies wurde durch einen `cube_part -F` Optionsparameter korrigiert, welcher nun die abgeschnittenen Ranks vollends entfernt.

Funktionserweiterung Score-P

In diesem Rahmen sind noch weitere Verbesserungen in Score-P vorgenommen worden.

Um wiederholtes Kompilieren zu beschleunigen, kann `ccache` genutzt werden. Dies ist nicht mit der Nutzung eines Score-P Compiletime-Filter kompatibel, denn der Inhalt des Filter kann sich ändern, während sich die für `ccache` sichtbare Kommandozeile nicht ändert. Das sorgt dafür, dass bei Änderung des Filters diese Änderung nicht detektiert wird – und so unerwartete Ergebnisse erzeugt werden. Das Problem wurde identifiziert und gelöst: Score-P setzt nun die Filterdatei als `CCACHE_EXTRAFILES` Umgebungsvariable, um so `ccache` über eine Änderung zu informieren.

Zudem wurde bei der Instrumentierung von ParaView folgende Fehler entdeckt und behoben. In dem unüblichen Fall, dass ein Compiler stdout als Ausgabe nutzt (z.B. via `gcc -E`), ist Score-P aufgrund einer fehlenden Ausgabedatei abgestürzt und hat den Instrumentierungsvorgang abgebrochen. Der Fehler wird nun abgefangen, indem die Ausgabe automatisch in eine Datei umgeleitet und anschließend mittels `cat` ausgegeben wird, und so die Instrumentierung von ParaView ermöglicht. Für die erfolgreiche Instrumentierung von ParaView müssen allerdings noch einige Komponenten von der Instrumentierung ausgenommen und einer Filterdatei zugefügt werden, wie zum Beispiel die ParaView-Komponenten, die die GUI implementieren, denn diese sind nicht instrumentierbar. Dies hat keinen Einfluss auf die Analyse von In-Situ Applikationen, denn die Berechnungsroutinen sind instrumentiert.

Außerdem wurde festgestellt, dass der Score-P Fortran Wrapper von `MPI_Comm_get_attr` je nach MPI Implementierung falsche Ergebnisse zurückliefert. Im Hydra Code wird dieser Funktionsaufruf dazu genutzt, um `MPI_TAG_UB` abzufragen, welcher für das automatische taggen von MPI Nachrichten verwendet wird. Der Fehler führte dazu, dass entweder ungültige Werte zurückgeliefert wurden, oder auf unterschiedlichen Ranks verschiedene Werte, was zu einem Deadlock führt. Das Problem wurde identifiziert, allerdings noch nicht behoben - dies ist in einem zukünftigen Score-P Release geplant. Als Workaround wurde dieser Funktionsaufruf im Quellcode durch die vom Standard garantierte Obergrenze von 32767 ersetzt.

Ein weiteres Feature erweitert das essenziellen Score-P Tool `scorep-score`, welches zur Filtererstellung genutzt wird. Es ist nun möglich, gewisse Gruppen von der Größenberechnung auszuschließen. Das ist für den in Abschnitt 3.2.4 beschriebenen Arbeitsablauf wichtig, um die Events der Gruppe `MEMORY`, welche nur in dem initialen Profil aufgezeichnet werden, von der Speicherplatzabschätzung eines Traces ausschließen zu können.

Des Weiteren hat Score-P stets neue Dateihandles für jedes Öffnen und Schließen einer Datei angelegt. Durch die große Menge an Operationen, die Hydra durchführt, führte dies zu Problemen in der Ausführungsgeschwindigkeit von Vampir. Um diese zu lösen, werden die Handles nun wiederverwendet, was die Anzahl deutlich senkt.

3.4.4 Performance Evaluierung

Im Folgenden wird die Visualisierung der Kopplungsoperationen, eine Analyse der Laufzeit sowie die Lastverteilung untersucht.

Kopplungsvisualisierung

Hydra startet als monolithisches Programm, welches sich während der Initialisierungsphase in die einzelnen Simulationsdomänen unterteilt und von da an als gekoppelte Simulation agiert. Durch die Nutzung dieses üblichen Schemas zur Kopplung von Applikationen, ist eine Änderung der Score-P Messinfrastruktur für die Aufnahme gekoppelter Hydra Simulation nicht notwendig. Allerdings wurden Kommunikationsmuster gefunden, die nicht MPI-konform sind und daher ein semantisch falsches Score-P Trace erzeugt haben, welches keine Analyse erlaubt.

So wurden Fehler in der Beendigungsprozedur gefunden und mittels `MPI_Ibarrier` gelöst (Abb. 3.5).

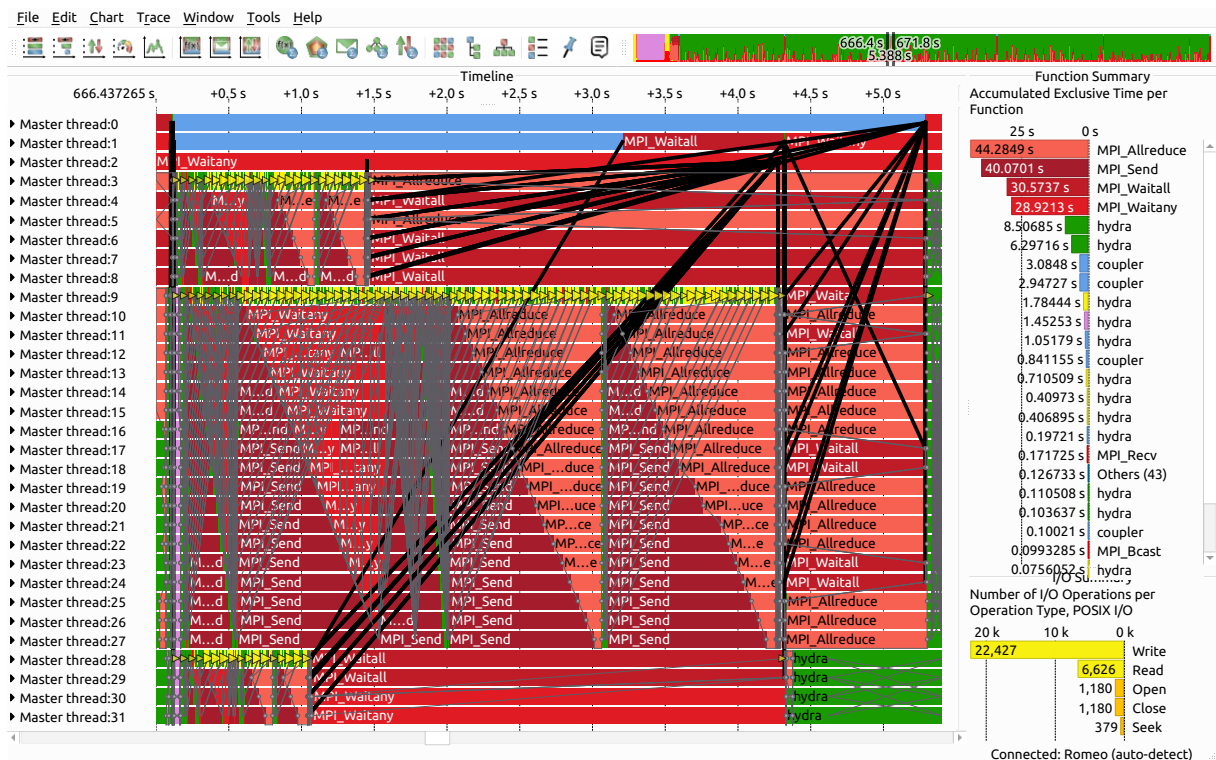


Abbildung 3.7: Kopplungsoperationen in Hydra (dargestellt in Vampir). Schwarze Linien markieren MPI-Kommunikation der Koppler. Die ersten drei Master-Threads dienen als Koppler; die folgenden 29.0 Threads verteilen sich auf drei Strömungssitzungen. Messfenster: 350 ms; 208 Ränge auf zwei Knoten.

Zur besseren Identifikation von Kommunikationsmustern wird jedem MPI Kommunikator ein eindeutiger Name gegeben. Routinen zur Kommunikatormodifikation (solche, die z.B. `MPI_Comm_split` aufrufen) werden um eine Operation erweitert, welche den Kommunikatornamen modifiziert, mit dem Ziel, den Entstehungsweg des Kommunikators eindeutig nachvollziehen zu können.

Für die gemeinsame Darstellung sind keine Änderungen an der Infrastruktur notwendig, denn Hydra nutzt, wie bereits oben erläutert, einen üblichen und bereits verfügbaren Ansatz zur Applikationskopplung.

Die Hervorhebung der Kopplungsoperationen ist hingegen nicht ohne Weiteres möglich. Eine Möglichkeit ist das Einfärben der Kommunikation mit verschiedenen Farben. Dazu muss ein Kriterium gefunden werden, welches die Kopplungsoperationen gemein haben, wie beispielsweise den Kommunikator. Alle Kopplungsoperationen nutzen absichtlich den Standardkommunikator `MPI_COMM_WORLD`. Weitere Analyse ergibt, dass die Kopplungsoperationen die einzigen Kommunikationen sind, welche `MPI_COMM_WORLD` nutzen. Diese Eigenschaft ermöglicht das Eingrenzen der gesamten Löserkommunikationen auf die vorhandenen Kopplungsoperationen.

Dies ermöglicht die Visualisierung in Vampir, wie in Abb. 3.7 zu sehen.

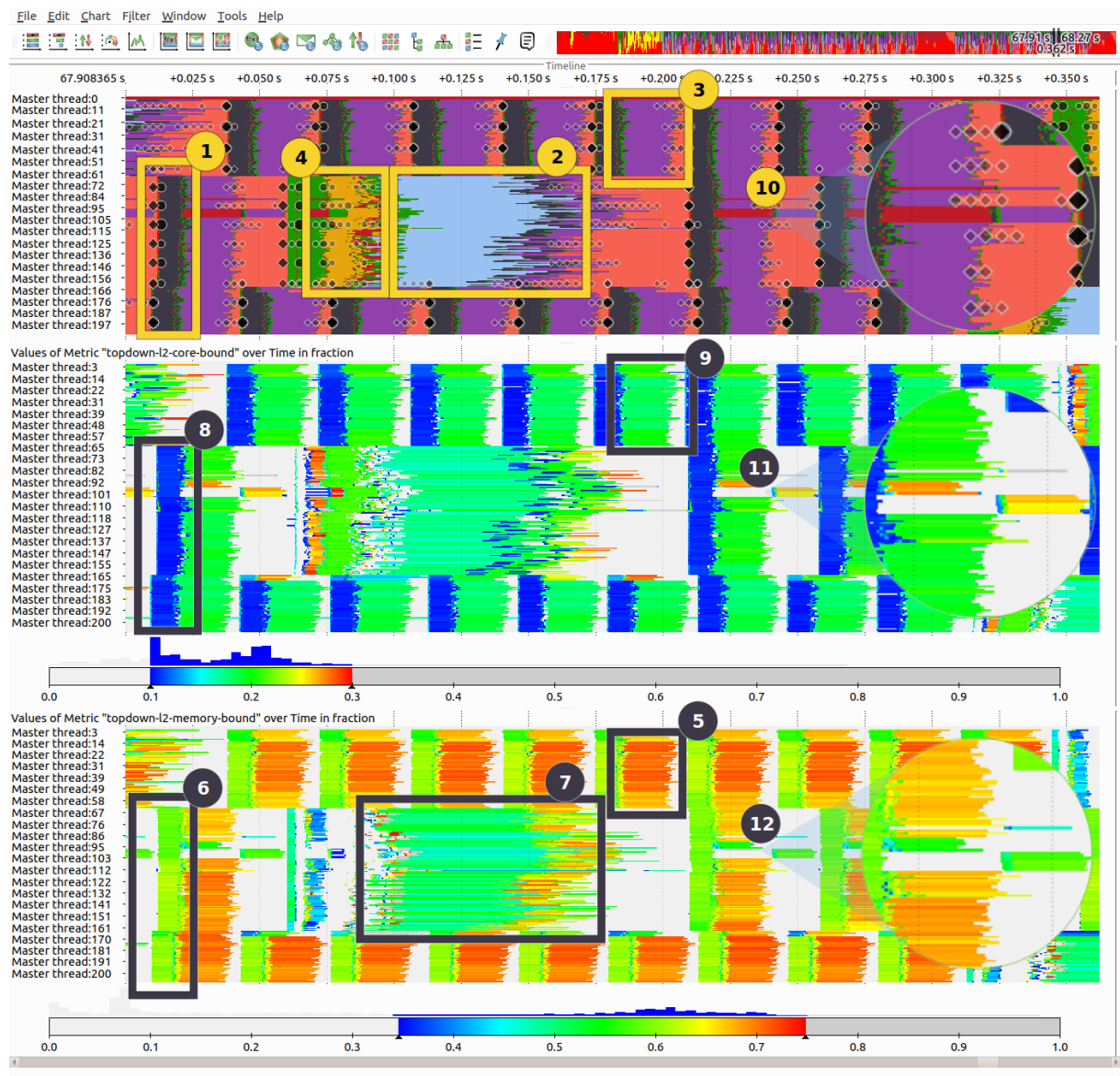


Abbildung 3.8: Annotierter Vampir Screenshot einer Top-Down Analyse. Interessante Bereiche sind markiert.

Top-Down Analyse mit Score-P von Hydra

In diesem Kapitel wird die Hardwareausnutzung von Hydra mittels Score-P *Top-down microarchitecture analysis method Plugin* ermittelt.

Abbildung 3.8 zeigt die *Top-Down microarchitecture analysis* für Hydra. Analysiert wurde die gekoppelte Hydra-Version [2] mit 208 MPI-Rängen (zwei voll belegte Knoten): zwei Prozesse als Koppler, 63.0 Prozesse für Sitzung 1, 100 für Sitzung 2 (über beide Knoten verteilt) und 43.0 für Sitzung 3 auf Knoten 2.

Der obere Teil von Abbildung 3.8 zeigt rund 350ms eines Solver-Iterationsschritts dieser gekoppelten Hydra-Simulation. Die gemessene Laufzeit ist jedoch nicht repräsentativ, da der Open MPI Laufzeitmechanismus eine Serialisierung erzwingt. Diese Serialisierung und ihre Auswirkungen auf die Top-Down-Metriken werden im Folgenden noch genauer diskutiert.

Abbildung 3.8 beleuchtet die Engpässe der Rechen-Routinen, indem sie deren Core- und Memory-Boundedness getrennt ausweist. Von den vier zeitintensivsten Routinen sind `calculate_bmax_prod_a` (Annotation 1), `lu_decomposition` (Annotation 2) und `lu_substitution` (Annotation 3) klar speichergebunden, während `vflux_compute` (Annotation 4) vorwiegend rechenwerksgebunden ist. Die roten MPI-Aufrufe sind hingegen fetch-bandwidth-limitiert und erscheinen daher in keiner der Heatmaps. In der unteren Heatmap wird deutlich, dass die Memory-Boundedness zwischen den Routinen variiert: `lu_substitution` erreicht hier mit etwa 60 – 65% den höchsten Wert (Annotation 5), gefolgt von `calculate_bmax_prod_a` mit 55 – 60% (Annotation 6) und `lu_decomposition` mit rund 45 – 50% (Annotation 7). Parallel dazu liegt der Anteil core-gebundener Instruktionen nur zwischen 10% und 30% und unterscheidet sich merklich: `calculate_bmax_prod_a` ist zu etwa 10% core-gebunden (Annotation 8), `lu_substitution` zu rund 20% (Annotation 9).

Ein Sonderfall zeigt sich in Simulationsitzung 2: Vor und nach einem Rechen-Abschnitt wird asynchrone MPI-Kommunikation über `MPI_Isend/MPI_Irecv` initiiert, echter Fortschritt erfolgt jedoch erst während `MPI_Waitany`. Diese MPI-Semantik führt dazu, dass die Rechen-Phasen auf den Rängen 97–103 auseinanderlaufen (Annotationen 10–12), weil diese Ränge auf Daten des nicht lokal gelegenen Rangs 111 warten müssen. Da die Ränge 91–103 denselben Memory-Controller teilen, steht den aktiv rechnenden Rängen 91–96 pro Rang mehr Speicherbandbreite zur Verfügung, was ihre Memory-Boundedness senkt, ihre Core-Boundedness erhöht und ihre Routinenlaufzeiten verkürzt – obwohl die Gesamtlaufzeit wegen der Verzögerung der Ränge 97–103 letztlich steigt. Aus den Beobachtungen an den Annotationen 10–12 lässt sich schließen, dass eine höhere verfügbare Speicherbandbreite die Memory-Boundedness insbesondere der `lu_substitution`-Region verringern, die Core-Auslastung steigern und die Laufzeit des Codes insgesamt verkürzen würde. Da die meisten Rechen-Regionen überwiegend speichergebunden sind, dürfte dieser Effekt auch andere Abschnitte betreffen. Folglich erscheinen Rechensysteme mit hoher Bandbreite als vielversprechende Option, um die Performance des gesamten Programms zu steigern.

Diese Erkenntnisse sind in [142] publiziert.

Für das beobachtete Problem der Serialisierung in Simulationsitzung 2 wurde eine Lösung implementiert, die allerdings nicht für alle Simulationskonfigurationen die Serialisierung zuverlässig verhindert. Eine Lösung, die in allen Fällen zu einer kürzeren Laufzeit führt, erfordert

umfangreiche Änderungen an der Kommunikationslogik und wird gegebenenfalls in einem Folgeprojekt vertieft untersucht.

Lastverteilung gekoppelter Modelle

Die Lastverteilung kann mittels Wartezeiten innerhalb von blockierenden kollektiven MPI Funktionsaufrufen ermittelt werden. Die gekoppelten Simulationssitzungen haben eine eigene Laufzeit, welche auf eine sequentielle Laufzeit, welche dem Gesamtarbeitsaufwand einer Simulation entspricht, normiert werden. Diese Normierung erfolgt nicht linear, sondern genügt Amdahlsches Gesetz, welches den Zusammenhang zwischen Laufzeit und Ressourcenaufwand unter Berücksichtigung der Parallelisierbarkeit berücksichtigt. Somit wird ausgehend von den gemessenen Laufzeiten eine fiktive sequentielle Laufzeit mit

$$t_{n=1}^k = \frac{t_n^k}{(1 - p^k) + \frac{p^k}{n^k}}$$

ermittelt. Dabei stellt t die Rechenzeit für Simulationssitzung k dar, welche auf n Prozessorkernen ausgeführt wurde, unter der Annahme eines parallelen Anteils p . Anschließend wird iterativ für jede Sitzung eine projizierte Laufzeit unter Zuhilfenahme eines zusätzlichen Prozessorkerns ermittelt. Diese Laufzeitprojektion folgt ebenfalls nach Amdahlsches Gesetz. Der Sitzung mit der längsten projizierten Laufzeit wird dann tatsächlich ein Prozessorkern zugewiesen. Dieser Zuweisungsvorgang eines Prozessorkerns zu der langsamsten Simulationssitzung wird wiederholt, bis alle verfügbaren Ressourcen Sitzungen zugewiesen sind.

Die Berechnung einer projizierten Laufzeit benötigt einen parallelen Anteil (Skalierbarkeit) dieser Sitzung. Im Versuch wurde diese Skalierung zu 99.7% ermittelt.

In der beobachteten Konfiguration konnte kein Vorteil gegenüber der analytischen Lastverteilungsheuristik erzielt werden. Dies lässt sich erklären durch die Nutzung eines einzelnen Rechenknotens, die in der Konfiguration identischen Berechnungen pro Sitzung (auf unterschiedlichen Domänen) sowie eine gut funktionierende Heuristik.

3.4.5 Performance Optimierungen

Mit den zuvor beschriebenen Änderungen an Messinfrastruktur und Strömungslöser lässt sich der Programmablauf nachvollziehen und langsame oder schlecht skalierbare Sektionen identifizieren. Während die Rechenphasen gut optimiert sind und keine offensichtlichen Performance-Probleme ermittelbar sind (Abschnitt 3.4.4), gibt es Optimierungspotential im Umgang mit Festplatten-I/O.

Monitor I/O

Bei Betrachtung der I/O Phase (Abb. 3.7) fällt auf, dass diese sehr viele kleine I/O Operationen ausführt. Bei genauerer Analyse fällt auf, dass viele Metadatenoperationen (`open`, `seek`, `close`) ausgeführt werden im Vergleich zu tatsächlich ausgeführten `write` Operationen. Genauere Analyse im Quelltext offenbart ein Muster, welches vereinfacht in Listing 3.4 dargestellt ist.

```

1 void write_monitor(char* msg[], int nrows, char* fileName)
2 {
3     for( int idx = 0; idx < nrows; ++idx)
4     {
5         FILE* fp = fopen(fileName, "a");
6         fwrite(msg[idx], ROWLEN, 1, fp);
7         fclose(fp);
8     }
9 }

```

Quelltext 3.4: Identifizierte I/O Performance Schwäche durch Metadatenoperationen

```

1 void write_monitor(char* msg[], int nrows, char* fileName)
2 {
3     FILE* fp = fopen(fileName, "a");
4     for( int idx = 0; idx < nrows; ++idx)
5     {
6         fwrite(msg[idx], ROWLEN, 1, fp);
7     }
8     fclose(fp);
9 }

```

Quelltext 3.5: Reduktion von Metadaten-Operationen (open/seek/close) durch das Öffnen außerhalb der Schleife.

Lösung ist das Verschieben des Dateiöffnens und -schließens vor beziehungsweise hinter die Schleife wie in Listing 3.5 dargestellt.

Diese Optimierung konnte im untersuchten Testfall die Laufzeit um etwa 5% reduzieren. Zudem ist diese gesamtheitlich besonders effektiv, denn die Reduktion von I/O Operationen reduziert die Last des gesamten geteilten Dateisystems und beschleunigt somit sämtliche auf dem Cluster zeitgleich laufende I/O-Operationen.

3.5 Zusammenfassung und Ausblick

3.5.1 Zusammenfassung

In diesem Arbeitspaket wurde Score-P um die Möglichkeit erweitert, Metadaten sowohl zur Compile- als auch zur Laufzeit zu erheben und zu speichern. Es wurden mehrere Ansätze verfolgt, und letztendlich ein RDF/JSON-LD basierter Lösungsansatz verfolgt. Weiterhin wurde Score-P genutzt, um gekoppelte Strömungssimulationen zu instrumentieren und sowohl Profil als auch Trace Event Daten zu erzeugen.

Gekoppelte Hydra-Traces lassen sich darstellen, und ermöglichen so eine Performance Analyse des gesamten Programmablaufs. Auch die Analyse von In-Situ Programmabläufen wurde ermöglicht, inklusive der Programmabläufe innerhalb des ParaView-Catalyst Python Skriptes.

Bei der Performanceanalyse wurden langsame I/O-Zugriffsmuster identifiziert, welche durch eine Optimierung beschleunigt wurde und so im betrachteten Testfall die Laufzeit um rund 5% reduzieren konnte. Weiterhin wurde die Lastverteilung beobachtet. Diese war in der analysierten Konfiguration gut verteilt. Um sich dennoch auf unterschiedlich gut skalierende Simulations-sitzungen anpassen zu können, wurde ein Algorithmus implementiert, der die Skalierbarkeit jener berücksichtigt.

Während eines einwöchigen Workshops beim Projektpartner wurden die Anwendungsentwickler in die Performancetools eingeführt, und können so eigenständig Analysen durchführen.

Während dieser Arbeiten wurden Korrekturen sowohl an der Messinfrastruktur Score-P als auch am Strömungslöser Hydra implementiert. So wurde die Softwarequalität beider Programme gesteigert, und beide können so als Grundstein für die Nutzung in weiteren Projekten dienen.

3.5.2 Ausblick

Die hier entwickelten Erweiterungen sind grundlegend für das gesamtheitliche Performan-
ceverständnis von HPC-Applikationen. Die Score-P Metadaten können als Datengrundlage für weitreichende Analysen genutzt werden, wie im Ausblick des AP3.1 erläutert wird (Abschnitt 6.5.2). Weiterhin sind Analysen der gekoppelter Anwendungen samt ParaView-Catalyst In-Situ Komponente möglich, was einen gesamtheitlichen Blick auf die Applikation ermöglicht. In Zukunft lässt sich dies auf In-Transit-Anwendungsfälle ausweiten.

Aufbauend auf den in diesem Arbeitspaket entwickelten Erweiterungen ergeben sich zahlreiche Perspektiven für zukünftige Arbeiten. Insbesondere die Kombination aus detaillierter Programminstrumentierung, strukturierter Metadatenbeschreibung und der Möglichkeit zur Analyse gekoppelter Anwendungen eröffnet Potenziale für eine automatisierte Bewertung und Optimierung komplexer Simulationsabläufe.

Darüber hinaus ermöglicht die RDF-basierte Repräsentation eine semantische Verknüpfung mit externen Datenquellen und Ontologien, etwa zur automatisierten Kontextualisierung von Laufzeitphänomenen mit domänenspezifischem Wissen. Dies könnte insbesondere im Rahmen von langfristigen Analysevorhaben oder Benchmarks über verschiedene Cluster hinweg hilfreich sein.

Langfristig lässt sich die entwickelte Infrastruktur auch für energieeffiziente HPC-Szenarien nutzen, indem neben Performance- auch Energie- und Ressourcendaten integriert und ausgewertet werden. In Kombination mit In-Transit-Technologien könnten so selbstlernende Steuerkomponenten realisiert werden, die das Verhalten großskaliger Simulationen zur Laufzeit optimieren.

4 Teilbericht AP2.1

4.1 Ausgewählte Grundlagen (AP2.1)

4.1.1 Messverfahren zur Digitalisierung der Hochdruckturbinenschaufeln

Bedingt durch Faktoren wie Werkzeugverschleiss, variable Fertigungstemperaturen und inhomogene Materialeigenschaften kommt es bei der Produktion von Triebwerkskomponenten zu Fertigungsvariationen, die zu Produktionsstreuungen und damit zu Abweichungen der realen Geometrie vom nominalen Modell führen. Daher ist es notwendig einen digitalen Zwilling zu erzeugen, welcher eine möglich realitätsnahe Quantifizierung der auftretenden Geometriestreuungen mit anschliessende CFD und FEM Simulationen beinhaltet. Hierfür kommen in der Industrie unterschiedliche Methoden zum Einsatz, welche sich in taktile und optische Messverfahren unterteilen lassen. Die Wahl des verwendeten Messsystems orientiert sich an der Zugänglichkeit der jeweiligen Messposition und der Messgenauigkeit des Messverfahren. Es ist darauf zu achten, dass die Messgenauigkeit des Messsystems unterhalb der zu bestimmenden geometrischen Streuung des Bauteils liegt. Da im Rahmen dieses Projekts nur CT-Messungen zur Erfassung der inneren Strukturen und optische Messverfahren für die Datenerhebung verwendet wurden, wird das Thema taktiler Messverfahren nur grob umrissen, und der Schwerpunkt liegt auf den eingesetzten CT- und optischen Verfahren. Zu den verwendeten Verfahren gehören schnittbasierte CT-Messungen zur Bestimmung der Schaufelinnengeometrie sowie GOM-Messungen zur Bestimmung der Schaufelaussengeometrie und die Alicona-Messung zur Schaufelfeaturemessung.

Taktile Messung

Beim taktilen Messen werden Messsysteme eingesetzt, die die Oberfläche des Messobjekts mit einem berührend arbeitenden Tastelement antasten. Als Tastsystem wird in der Regel ein kugelförmiger Tastkopf verwendet, der an einem Taststift befestigt ist. Zur Oberflächenerfassung werden zuvor definierte Oberflächenmesspunkte angefahren und bei Erfüllung einer Antastbedingung, wie z. B. einer Antastkraft, der Mittelpunkt des Tastelements ermittelt. Gängige Messsysteme erlauben unterschiedliche Arten der Messpunkterfassung. Zum einen können diskrete Messpunkte angefahren werden, wodurch jeder Messposition ein 1D-Wert zugeordnet werden kann. Zum anderen können vorgegebene 2D- oder 3D-Konturen abgefahren werden, wodurch kontinuierlich Werte über die vorgegebene Messstrecke ermittelt werden können. Die Messwerterfassung erfolgt je nach Messsystem entweder durch Einzelpunktantastung oder durch das Scanningverfahren. Bei der Einzelpunktantastung wird nach jedem Antasten ein Wert ermittelt und der Kontakt zum Messobjekt wieder gelöst. Das Scanningverfahren beruht dagegen auf der Erfassung mehrerer Messpunkte pro Zeiteinheit. Koordinatenmessgeräte weisen in der Regel eine sehr hohe Messgenauigkeit auf, die je nach Messsystem unter $1.00\text{ }\mu\text{m}$ liegen kann, siehe MARXER [106] und HEINZE [66]. Da es sich bei taktilen Messverfahren nicht um im weitesten Sinne flächenhafte Verfahren handelt, ist eine vollständige Erfassung komplexer Geometrien sehr zeitaufwändig bis unmöglich. Aus diesem Grund können im Vergleich zu optischen Messverfahren nur geringe Datenmengen pro Messung erfasst werden.

CT-Messung

Die Computertomographie hat sich in den vergangenen Jahren zu einem etablierten Prüfverfahren in der Luftfahrt entwickelt. Diese Technologie ermöglicht die zerstörungsfreie Untersuchung von Bauteilen. Ein wesentlicher Vorteil der CT-Messung ist die Möglichkeit, die Innengeometrie komplexer Bauteile zu untersuchen, wenn diese von aussen nur schwer zugänglich sind. Bei Turbinenschaufeln ist dies beispielsweise eine vielversprechende Methode, um die Geometrie der Kühlluftkanäle abzubilden. Bei der CT-Messung wird ein Objekt mit Röntgenstrahlen durchleuchtet, wobei Schnittbilder entstehen, die eine genaue Darstellung des Inneren des Objektes ermöglichen. Beim Durchdringen des Bauteils wird ein Teil der Röntgenstrahlung absorbiert oder gestreut, wobei diese Phänomene je nach Stärke der Strahlungsquelle und des verwendeten Messverfahrens unterschiedlich stark ausgeprägt sind. Im Detail nachzulesen in der entsprechenden Fachliteratur [73, 76, 137]. Anhand dessen können CT-Messsysteme grundsätzlich in zwei Typen unterschieden werden:

Flachdetektoren mit konischem Messstrahl:

Bei diesem Verfahren wird auf einem Flächendetektor eine 2D-Projektion des zu messenden Bauteils erzeugt. Durch Umfahren des Bauteils im Winkel 360° wird eine Serie von Graustufenbildern aufgenommen, die anschliessend mit geeigneter Software, beispielsweise VGStudio Max, in eine 3D-Geometrie umgewandelt werden können.

Zeilendetektoren mit Fächerstrahl:

Bei dieser Methode wird eine 1D-Projektion auf einen Liniendetektor (LDA) oder einen gekrümmten Liniendetektor (CLDA) mit einer Zeile oder mehreren Zeilen projiziert. Auch hier wird das Bauteil unter einem Winkel von 360° umkreist. In der Regel wird zunächst ein Sinogramm der gemessenen Schnittebene erstellt, das dann mit Hilfe moderner Messtechnik durch die RADON-Transformation automatisch in ein repräsentatives Grauwertbild zurückgeführt wird [138]. Um ein 3D-Abbild des Messobjektes zu erhalten, muss dementsprechend das gesamte Bauteil in diskreten Abständen der z-Richtung mehrfach vermessen werden.



Abbildung 4.1: Graustufenbild mit ISO-Filter

Die Rückführung der Grauwertbilder in ein Oberflächennetz kann ebenfalls in VGStudio Max erfolgen. In den Grauwertbildern ist ein gradueller Übergang zwischen Umgebung und Schaufelmaterial erkennbar. Die Auflösung der CT-Scanner begrenzt die Möglichkeit, scharfe Kanten

darzustellen. Streueffekte und Strahlaufhärtungseffekte beeinflussen den Übergangsbereich zusätzlich. Ebenso ist der Einfluss der Messunsicherheiten der vorgestellten Methoden zu berücksichtigen. Zur Definition der Materialoberfläche muss daher ein ISO-Filter verwendet werden, siehe Abbildung 4.1.

In einem früheren Projekt am Lehrstuhl für Turbomaschinen und Flugantriebe wurde der Messfehler mit verschiedenen Messsystemen und Leistungskonfigurationen quantifiziert. Dabei wurde durch HÖGNER [76] festgestellt, dass der Flachdetektoren mit etwa 100 µm die höchste Messunsicherheit aufweist. Mit dem CLDA-Verfahren können dagegen Messgenauigkeiten von etwa 33.0 µm erreicht werden, wobei im Bereich der Turbinenschaufelhinterkante aufgrund der zuvor genannten Effekte nur Messgenauigkeiten von 105 µm für die untersuchte Turbinenschaufel erreicht werden. Da eigene Analysen der CT-Graustufenbilder mit VGStudio Max keinen signifikanten Unterschied zu den von Rolls-Royce zur Verfügung gestellten rekonstruierten Oberflächennetzen ergaben, werden in diesem Projekt die von Rolls-Royce zur Verfügung gestellten Oberflächennetze verwendet.

Optische Messung

Optische Messverfahren ermöglichen die berührungslose und in der Regel flächenhafte Vermessung von Bauteilen. Dabei gibt es verschiedene Möglichkeiten der Oberflächenerfassung. In diesem Abschnitt soll auf zwei Verfahren eingegangen werden, die im Rahmen dieses Projektes zur Datenerfassung eingesetzt wurden. Unter anderem wurde hierfür die Scanbox 5108 der Firma GOM eingesetzt, welche die automatisierte Vermessung von Bauteilen mittels Streifenprojektionsverfahren ermöglicht. Das System verwendet dazu einen ATOS Triple Scan Sensor 16M mit einer Kameraauflösung von 16 Megapixel. Zum anderen wurde der InfiniteFocus SensorR25 der Firma Alicona eingesetzt, der mittels Fokusvariationsverfahren eine hochauflösende Vermessung von Merkmalen der Schaufelgeometrie ermöglicht.

GOM

Wie bereits erwähnt, verwendet das GOM-Messgerät ein Streifenprojektionsverfahren zur flächenhaften Objektbestimmung mittels Triangulationsmethode. Das Triangulationsverfahren soll im Folgenden kurz anhand des in Abbildung 4.2 dargestellten Schemas erläutert werden. Bei diesem Verfahren beobachten eine oder mehrere Kameras die auf das zu beobachtende Objekt projizierte Profillinie. Zwischen einem beliebigen Messpunkt O auf der Profillinie und der Kamera 1 C_1 sowie dem Projektor P kann ein zweidimensionales Dreieck konstruiert werden. Die Basis der Triangulation c sowie die angrenzenden Winkel α und β sind bekannt. Unter Berücksichtigung der geometrischen Beziehung im allgemeinen Dreieck kann mit Hilfe des Innenwinkelsatzes und des Sinussatzes folgende Formel aufgestellt werden.

$$b = \frac{c}{\sin(\pi - \alpha - \beta)} \cdot \sin(\beta) \quad (4.1)$$

Damit kann der Abstand b zwischen Objekt und Kamera bestimmt werden [72].

Beim Streifenprojektionsverfahren werden im Gegensatz zum Lichtschnittverfahren mehrere Messlinien gleichzeitig auf das Messobjekt projiziert, wie in Abbildung 4.3 schematisch darge-

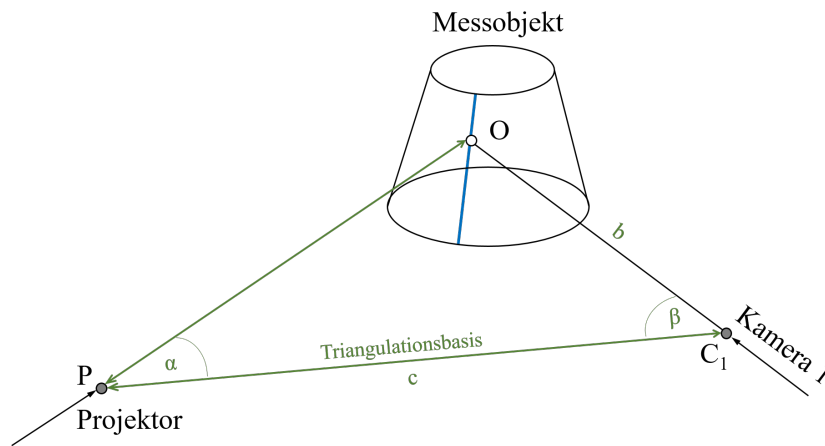


Abbildung 4.2: Schema Triangulationsverfahren, nach HOLZHAUSEN [72]

stellt. Hier wird eine zweite Kamera C_2 eingesetzt. Die Genauigkeit des Ergebnisses kann durch den Einsatz eines sogenannten Stereokamerasystems verbessert werden.

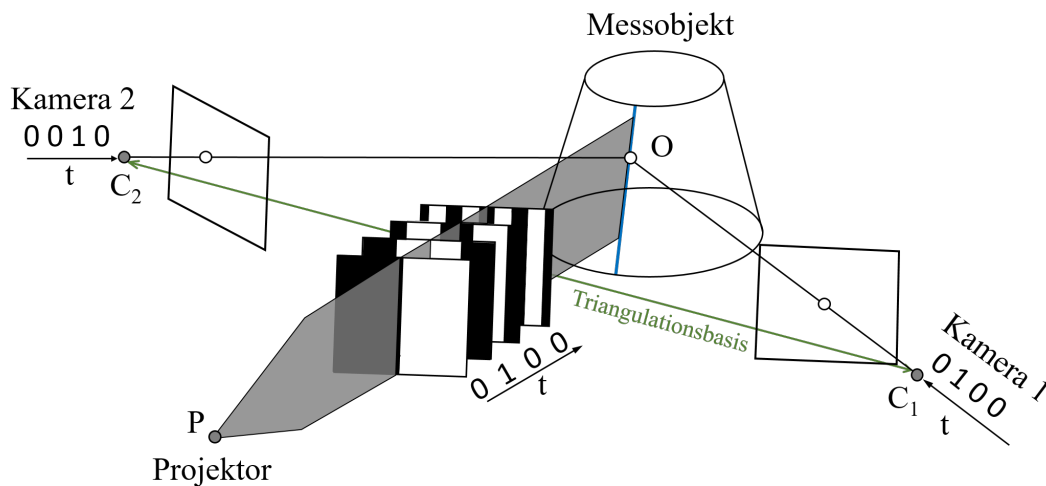


Abbildung 4.3: Streifenlichtprojektionsverfahren mittels räumlich-zeitlicher Kodierung, nach HOLZHAUSEN [72] und BACKHAUS [9]

Mit dem Triple Scan Prinzip können mit der Stereokamera insgesamt 3 Triangulationsdreiecke aufgespannt werden. Zum einen zwischen den Kameras $C_1 - P - O$ und $C_2 - P - O$ und zum anderen zwischen den beiden Kameras $C_1 - C_2 - O$. Bei den oben genannten Verfahren handelt es sich um sogenannte aktive Projektionsverfahren. Die sogenannte Stereoskopie benötigt keine aktive Beleuchtungsquelle und kann daher auch passiv durchgeführt werden. Ausgehend von einer bekannten Orientierung der beiden Kameras zueinander kann der Messpunkt O anhand bestimmter Objektmerkmale, die in beiden Aufnahmen vorhanden sein müssen, eindeutig bestimmt werden. Die spezifischen Objektmerkmale können Kanten, Bohrungen oder am Messobjekt angebrachte codierte Marker sein. Zur Vereinfachung der Korrespondenzfindung verwendet das GOM-Messgerät zeitkodierte Streifenlicht, um den Einfluss von Fremdlichteinflüssen zu minimieren. Diese sind in der Abbildung 4.3 als unterschiedliche Muster ausgehend vom Projektor symbolisiert und werden in zeitlicher Abhängigkeit von den Kameras

detektiert [149]. Da ein Erfassen der gesamten Oberfläche des Messobjekts mit einer einzigen Messposition nicht möglich ist, ist es insbesondere bei komplexen Geometrien erforderlich, Messaufnahmen aus unterschiedlichen Winkeln vorzunehmen. Diese werden anschliessend mittels zuvor aufgebrachter Referenzmarken miteinander verrechnet. Überlappungsbereiche der Einzelmessungen und redundante Daten werden dabei automatisch eliminiert [67, 149].

Basierend auf bereits am Lehrstuhl durchgeführten Untersuchung kann von einer Messgenauigkeit von $10\ \mu\text{m}$ für leicht gekrümmte Flächen wie beispielsweise Saugseiten- (SS) und Druckseitenbereich (DS) der Turbinenschaufel ausgegangen werden. Im Bereich grosser Oberflächenkrümmung wie beispielsweise an den Vorderkanten- (LE) und Hinterkantenbereich (TE) der Turbinenschaufel konnte eine Messgenauigkeit von $20\ \mu\text{m}$ unter den vorliegenden Bedingungen nachgewiesen werden [76].

Alicona

Die Alicona-Messgeräte findet verbreitet Anwendung bei der Detektion von Oberflächenfehlern, wie Rissen und Kerben, sowie der Erfassung von Oberflächenmerkmalen, beispielsweise der Rauigkeit. Weiterhin ist die Untersuchung kleinster Geometrien möglich. Eine Untersuchung von ZANGL [180] hat ergeben, dass sich das Messgerät ebenfalls für die Vermessung der äusseren Kühlluftlochgeometrie eignet.

Das verwendete Alicona-Messgerät arbeitet auf dem Prinzip der vertikalen Fokusvariation, wodurch eine detaillierte Erfassung sowohl topografischer als auch farblicher Informationen des Bauteils gewährleistet wird. Zu diesem Zweck wird eine Optik mit geringer Tiefenschärfe verwendet, sodass lediglich ein kleiner Bereich des zu vermessenden Objektes scharf in der Aufnahme dargestellt wird. Durch eine Verschiebung der Optik parallel zur optischen Messachse erfolgt eine Verschiebung des Fokus des Aufnahmebildes. Die Erfassung der Daten erfolgt dabei kontinuierlich, bis jeder Bereich des Messobjektes scharf abgebildet wurde. Die erfassten Sensordaten können unter Verwendung diverser Algorithmen als Fokuginformation entlang der Verschiebungsachse bestimmt werden, wie in Abbildung 4.4 schematisch dargestellt.

In der Regel lässt sich festhalten, dass ein Objektpunkt schärfer fokussiert ist, wenn die benachbarten Punkte eine grössere Variation der Bildintensität aufweisen. Daher kann die Fokuginformation anhand der Standardabweichung der Objektpunkte quantifiziert werden. Die Entfernung zum Messpunkt kann anschliessend durch die Position des maximalen Fokus berechnet werden. Für die Bestimmung des Maximums werden unterschiedliche Verfahren verwendet, welche in der DIN EN ISO 25178-606 detailliert beschrieben werden [79]. Wie bereits im Kontext des GOM-Messgeräts dargelegt, wird auch bei diesem Gerät eine künstliche Lichtquelle zum Beleuchten des Messobjekts verwendet, um eine Unabhängigkeit von äusseren Lichteinflüssen zu gewährleisten. Das Ringlicht, welches sich um das Objektiv befindet, gewährleistet eine möglichst gleichmässige Ausleuchtung der Oberfläche mit Weisslicht. Ringlicht und Objektiv sind in einem wechselbaren Messsondenkopf integriert. Am Lehrstuhl stehen die Objektive 10xAX und 20xAX für die Vermessung zur Verfügung. Eine Zusammenfassung der objektivspezifischen Merkmale findet sich in Tabelle 4.1. Im Gegensatz zu dem von ZANGL [180] verwendet Sensor verfügt der InfiniteFocus SensorR25 über keine koaxiale Beleuchtung, weshalb die Ausleuchtung von Kühlluftlöchern nur eingeschränkt möglich ist.

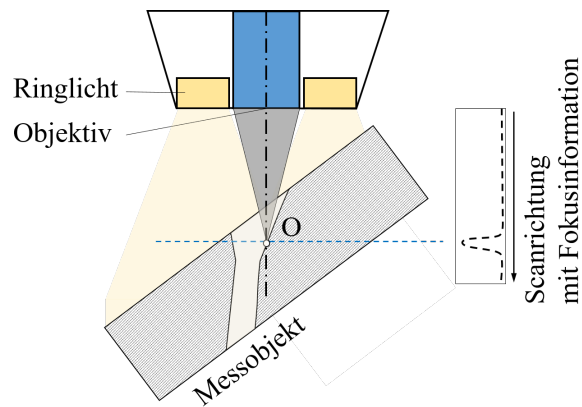


Abbildung 4.4: Schematische Darstellung der Messung eines Kühlluftlochs mittels vertikaler Fokusvariation, nach ZANGL [180]

Tabelle 4.1: Objektivspezifische Merkmale

Objektiv	10xAX	20xAX	Einheit
Arbeitsabstand	33.5	20	mm
Messbereich	2x2	1x1	mm x mm
Feinste laterale topologische Auflösung	2	1	μm
Vertikale Auflösung	0.130	0.070	μm
Min. messbarer Radius	5	3	mm
Min. messbarer Keilwinkel	20	20	°

Wie bereits zuvor im Rahmen der Erläuterung des GOM-Messverfahrens dargelegt, erfolgt die Generierung des Polygonnetzes unter Zuhilfenahme der von Alicona bereitgestellten Messsoftware. Zu Beginn erfolgt eine Bereinigung der Datenpunkte von Messrauschen. Die hohe mögliche Dichte an Messpunkten führt zu einem relativ grossen Speicherbedarf der dabei erzeugten Polygonnetze im Vergleich zu GOM-Messdaten. Daher ist es in der Regel sinnvoll im Hinblick auf die weiter Verwendung der Daten für die Untersuchung der Kühlluftlochgeometrie, das erzeugte Polygonnetz in Bereichen geringer lokaler Krümmung zu reduzieren. Hierfür geeignete Funktionen sind in der zuvor beschriebenen GOM-Software zu finden, welche eine starke Reduzierung der Dateigrösse ermöglichen, jedoch einhergehend mit den Verlust der Rauheitsinformationen.

4.1.2 Probabilistische Grundlagen

Die deterministische Auslegung von Bauteilen basiert auf der Ermittlung einer, für die betrachteten Grössen, eindeutig definierten Lösung. Jedoch unterliegen alle messbaren Grössen und Parameter Schwankungen, deren Variationsraum in der deterministischen Auslegung nicht berücksichtigt werden kann. Für die direkte Berücksichtigung des Variationsraums können probabilistische Methoden verwendet werden. Die Probabilistik ist ein Teilgebiet der Stochastik und stellt Ansätze zur quantitativen Analyse von zufälligen Vorgängen bereit. So wird z. B. in

einer Monte-Carlo-Simulation (MCS) das Streuverhalten der Eingangsgrößen berücksichtigt und lässt die Beurteilung deren Auswirkungen auf die Streuung der Ergebnisgrößen zu.

Im Folgenden werden die Grundlagen probabilistischer Untersuchungen erläutert. Diese umfassen in Abschnitt 4.1.2 Zufallsvariablen und statistische Masse zur Charakterisierung von Stichproben. Weitere Ausführungen befassen sich mit der Anwendung einer MCS in Abschnitt 4.1.2 sowie mit Samplingverfahren in Abschnitt 4.1.2

Zufallsvariablen

Eine Zufallsvariable b repräsentiert eine streuende Grösse, die z. B. infolge von zufälligen Ereignissen ein variable Werte annimmt. Eine Unterteilung der Zufallsvariablen erfolgt anhand ihrer Abzählbarkeit in diskrete und stetige Zufallsvariablen. Stetige Zufallsvariablen können überabzählbar unendlich viele Werte annehmen. Zur Beschreibung von stetigen Zufallsvariablen dienen die Verteilungsfunktion $F(b)$ und die zugehörige Wahrscheinlichkeitsdichtefunktion $f(b)$. Die Verteilungsfunktion an der Stelle a drückt die Wahrscheinlichkeit aus, dass die Zufallsvariable einen Wert von $P(b \leq a)$ annimmt. Es ergibt sich ein Wertebereich $F(b) \in [0.0; 1.0]$, wobei die Funktion monoton steigend ist. Die Verteilungsfunktion ist dabei wie folgt mit der Wahrscheinlichkeitsdichtefunktion verknüpft:

$$F(a) = P(b \leq a) = \int_{-\infty}^a f(b') db' . \quad (4.2)$$

Stichproben

Typischerweise werden bei statistischen Analysen Informationen über die Merkmale einer Grundgesamtheit gesucht, z. B. die geometrischen Streuungen der Innen- und Aussengeometrie aller Rotorschaukeln aus einer Produktionsreihe. Da die Erhebung der Grundgesamtheit aller produzierten Rotorschaukeln in der Regel mit einem hohen Kosten- und Zeitaufwand verbunden ist, wird auf Stichproben zurückgegriffen. Diese repräsentieren eine zufällig Teilmenge der produzierten Schaukeln und dienen zur Schätzung der statistischen Eigenschaften der Grundgesamtheit. Dieses Vorgehen ist mit einer Unsicherheit behaftet, die umso geringer ausfällt, je grösser der Stichprobenumfang ist.

Die folgenden Definitionen statistischer Maße sind daher für eine Stichprobe

$$\mathbf{b} = (b_1; b_2; \dots; b_{n_{\text{sim}}})^T \quad (4.3)$$

formuliert, mit dem Stichprobenumfang n_{sim} . Die erläuterten statistischen Maße sind als Schätzung der Maße der Grundgesamtheit zu verstehen. Die Unsicherheit, die mit dieser Schätzung verbunden ist, kann mit Hilfe von Konfidenzintervalle quantifiziert werden.

Statistische Maße einer Stichprobe

Zur Charakterisierung einer Stichprobe werden Lage- und Streumasse verwendet. Zu den Lagemaßen gehören z. B. der Median, der die nach ihrer Grösse sortierten Stichprobe in zwei gleich große Teilmengen teilt, der Modus, d. h. der am häufigsten auftretende Wert, oder

der arithmetische Mittelwert $\bar{b}$. Der arithmetische Mittelwert einer Stichprobe kann mittels folgender Gleichung berechnet werden

$$\bar{b} = \frac{1}{n_{\text{sim}}} \sum_{i=1}^{n_{\text{sim}}} b_i. \quad (4.4)$$

Zu den Streumaßen gehört die empirische Varianz, die die mittlere quadratische Abweichung der Realisierungen von ihrem arithmetischem Mittel beschreibt. Der erwartungstreue Schätzer ist

$$\text{Var}(\mathbf{b}) = \frac{1}{n_{\text{sim}} - 1} \sum_{i=1}^{n_{\text{sim}}} (b_i - \bar{b})^2. \quad (4.5)$$

Daraus ergibt sich als Quadratwurzel die empirische Standardabweichung zu

$$s_o(\mathbf{b}) = \sqrt{\text{Var}(\mathbf{b})}. \quad (4.6)$$

Als weitere Möglichkeit zur Quantifizierung der Streuung ist die Interquartilbreite IQR zu nennen. Diese ist definiert als

$$IQR = Q_{q=0.75} - Q_{q=0.25} \quad (4.7)$$

mit dem Quantil Q_q . Das Quantil Q_q teilt die Stichprobe so, dass der Anteil $q \in [0.0; 1.0]$ unterhalb und der Anteil $1 - q$ oberhalb von Q_q liegt.

Neben Lage- und Streumaßen sind ebenfalls die Zusammenhänge zwischen zwei Variablen einer Stichprobe von Bedeutung. Diese Zusammenhänge lassen sich durch die Kovarianz oder Korrelationskoeffizienten beschreiben. Die Stichprobenkovarianz ist ein Zusammenhangsmaß für die gemeinsame Streuung zweier Zufallsvariablen $\mathbf{b}_1$ und $\mathbf{b}_2$. Sie kann mittels der Gleichung 4.8 berechnet werden

$$\text{Cov}(\mathbf{b}_1; \mathbf{b}_2) = \frac{1}{n_{\text{sim}} - 1} \sum_{i=1}^{n_{\text{sim}}} (b_{1,i} - \bar{b}_1) (b_{2,i} - \bar{b}_2). \quad (4.8)$$

Positive bzw. negative Werte zeigen einen bestehenden positiven bzw. negativen Zusammenhang der beiden Zufallsvariablen an. Für einen Wert von $\text{Cov} = 0$ liegt kein linearer Zusammenhang der betrachteten Größen vor. Allerdings hängt die Kovarianz auch von der Streuungen der einzelnen Zufallsvariablen ab, so dass die Vergleichbarkeit zweier Zusammenhänge über die Kovarianz nicht zweckmässig ist. Daher erfolgt die Normierung über die Varianzen der betrachteten Größen. Der lineare Korrelationskoeffizient nach PEARSON [128] berechnet sich zu

$$r_k(\mathbf{b}_1; \mathbf{b}_2) = \frac{\text{Cov}(\mathbf{b}_1; \mathbf{b}_2)}{\sqrt{\text{Var}(\mathbf{b}_1)} \sqrt{\text{Var}(\mathbf{b}_2)}} \in [-1.0; 1.0]. \quad (4.9)$$

Der lineare Korrelationskoeffizient dient der Beschreibung linearer Zusammenhänge. Der betragsmässige Wert von $|r_k| = 1$ zeigt einen perfekten positiven oder negativen linearen Zusammenhang der betrachteten Zufallsvariablen an. Analog zur Kovarianz weisen die Zufallsvariablen bei einem Wert von $r_k = 0$ keinen linearen Zusammenhang auf.

Zur Detektion monotoner Zusammenhänge zwischen zwei Zufallsvariablen eignet sich der Rangkorrelationskoeffizient $\tilde{r}$ nach SPEARMAN [158]. Dieser baut auf der Bestimmung der linearen Korrelationskoeffizienten auf. Für die Berechnung werden anstelle der Zufallsvektoren $\mathbf{b}_1$

und $\mathbf{b}_2$ deren Ränge $\text{rg}(\mathbf{b}_1)$ und $\text{rg}(\mathbf{b}_2)$ betrachtet. Der Rangkorrelationskoeffizient lässt durch Gleichung 4.10 berechnen.

$$\tilde{r}(\mathbf{b}_1; \mathbf{b}_2) = \frac{\text{Cov}(\text{rg}(\mathbf{b}_1); \text{rg}(\mathbf{b}_2))}{\sqrt{\text{Var}(\text{rg}(\mathbf{b}_1))} \sqrt{\text{Var}(\text{rg}(\mathbf{b}_2))}} = r_k(\text{rg}(\mathbf{b}_1); \text{rg}(\mathbf{b}_2)) \quad (4.10)$$

Ein Vorteil gegenüber des Korrelationskoeffizient nach PEARSON ist hervorzuheben, daß so auch monotone, nichtlineare Zusammenhänge erkannt werden können. Ausserdem ist der Rangkorrelationskoeffizient robuster gegenüber Ausreißern. Daher wird im Folgenden bei der Diskussion von Korrelationskoeffizienten stets der Rangkorrelationskoeffizient nach SPEARMAN [158] verwendet.

Monte-Carlo-Simulation

Die MCS ist ein Verfahren der Stochastik zur Analyse und Modellierung von Systemzusammenhängen hoher Komplexität.

Der grundlegende Ablauf einer MCS ist in Abbildung 4.5 dargestellt. Zunächst werden n_{sim} Realisierungen der zuvor definierten Eingangsgrößen erzeugt. Dies erfolgt auf Basis der Verteilungsfunktionen der Eingangsgrößen und der Korrelationsstruktur zwischen den Eingangsgrößen. Die Generierung der Realisierungen erfolgt mithilfe eines geeigneten Samplingverfahrens und einem Verfahren zur Korrelationskontrolle. Im nächsten Schritt wird für jede Realisierung das deterministische Modell hinsichtlich seiner Ergebnisgrößen analysiert, woraus sich die Sensitivitäten bezüglich der probabilistischen Eingangsgrößen für alle Realisierungen ermitteln lassen.

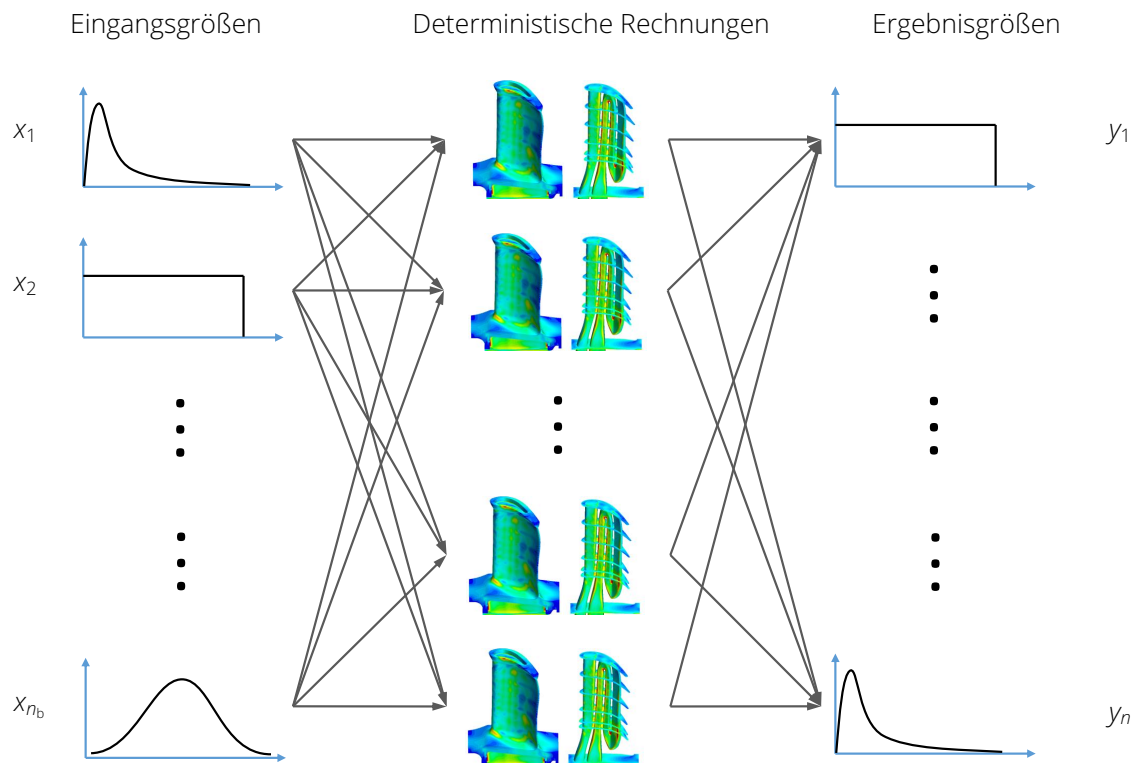


Abbildung 4.5: Schematische Ablauf der Monte-Carlo-Simulation

Weiterhin ist für die MCS ein valides deterministisches Modell mit Gültigkeit im gesamten Definitionsbereich der vorgegeben Eingangsgrößen erforderlich. Ebenfalls sollten auch die Verteilungsfunktion der Eingangsgrößen hinreichend genau bekannt sein. Wird eine MCS ohne ausreichende Kenntnis der Eingangsgrößenverteilungen durchgeführt, ist zu beachten, dass Schätzfehler der angenommen Verteilungsfunktion einen signifikanten Einfluss auf die Verteilung der Ergebnisgrößen haben [132]. Zunächst wird auf Samplingverfahren eingegangen, bevor die Ersatzmodelle näher beschrieben werden.

Samplingverfahren

Die Generierung von Realisierungen erfolgt durch den Einsatz von Samplingverfahren. Ziel ist es, die Verteilung der Eingangsgrößen möglichst genau abzubilden. In der vorliegenden Arbeit wird dazu das Latin Hypercube Sampling (LHS) und das extended LHS (eLHS) verwendet, welche im Folgenden kurz erläutert werden sollen.

Latin Hypercube Sampling

Die Methode des LHS nach McKay et al. [107] basiert auf der Zerlegung des Wertebereichs der Verteilungsfunktion in n_{sim} Intervalle I^* gleicher Wahrscheinlichkeit beschreibbar durch

$$\Delta P = \frac{1}{n_{\text{sim}}}. \quad (4.11)$$

Die Methodik ist in Abbildung 4.6a dargestellt. Die Intervallgrenzen sind als grau gestrichelte Linien markiert. Für jedes Intervall wird mithilfe eines Zufallszahlengenerators eine Realisierung im Intervall der Verteilungsfunktion erzeugt und in den Wertebereich der Eingangsgrösse überführt. Die resultierenden Realisierungen sind in Abbildung 4.6a als schwarze Punkte gekennzeichnet. Abschliessend werden Methoden der Korrelationskontrolle zum Einstellen der gewünschten Zielkorrelationsmatrix eingesetzt.

In der vorliegenden Arbeit kommt das Restricted-Pairing [52, 33] zur Korrelationskontrolle zum Einsatz. Der iterative Prozess stellt eine beliebige Rangkorrelationsstruktur in einer Datenmatrix mit n_{sim} Realisierungen und n_b Eingangsgrößen ein. Vorausgesetzt wird, dass die Zielkorrelationsmatrix $\mathbf{T} \in \mathbb{R}^{n_b \times n_b}$ positiv definit ist. Ebenso muss $n_{\text{sim}} > n_b$ gewählt werden.

Als Eigenschaften des LHS sind aufgrund der systematischen Generierung der Realisierungen die vergleichsweise genaue Abbildung der vorgegebenen Verteilungsfunktion auch bei kleiner Realisierungsanzahl hervorzuheben.

Extended Latin Hypercube Sampling

Häufig ist die notwendige Anzahl an Realisierungen zur Gewährleistung einer ausreichenden statistischen Sicherheit zu Beginn der MCS schwer abschätzbar. Hier schafft das eLHS nach Schmidt et al. [148] eine Möglichkeit die bestehenden Realisierungen um ein Vielfaches des ursprünglichen gewählten Realisierungsanzahl n_{sim} zu erweitern. Dabei baut die Methode auf die Arbeiten von Sallaberry [139], sowie Fleming und Mantoufel [130] auf.

Bei dem eLHS werden die ursprünglichen Intervalle I^* eines zuvor erstellten LHS $\mathcal{S}_0$ anhand der Wahrscheinlichkeiten $\frac{\Delta P}{2}$ halbiert. Dies ist in Abbildung 4.6b als rot gestrichelte Linien

dargestellt. In nur einem Teilintervall I_0 des ursprünglichen Intervalls I^* befindet sich bereits eine Realisierung. Die übrigen Intervalle I_1 werden analog der zuvor beschriebenen Vorgehensweise aufgefüllt. Die resultierenden Realisierungen sind in Abbildung 4.6b als rote Punkte markiert. Sie stellen die Realisierungen $\mathcal{S}_1$ dar. Für die zweite Erweiterung $\mathcal{S}_2$ werden die Intervalle I_0 und I_1 erneut halbiert, nun wird eines (I_2) der zwei freien Teilintervalle von I^* aufgefüllt. Wenn mehr als ein freies Teilintervall vorliegt, erfolgt die Auswahl der zu besetzenden Intervalle anhand des Vergleichs von Zielverteilungsfunktion und der diskreten Verteilungsfunktion der erzeugten Realisierungen. Dies erfolgt in Anlehnung an den KS-Test. Für jedes I^* wird die Lage des größten Abstandes zwischen Zielverteilungsfunktion und diskreter Verteilungsfunktion ermittelt. Somit werden die Bereiche identifiziert, in denen die Abbildung der Zielverteilungsfunktion aufgrund fehlender Realisierungen weniger genau erfolgt. Die entsprechenden Realisierungen sind in Abbildung 4.6b mit grünen Punkten gekennzeichnet. In der dritten Erweiterung $\mathcal{S}_3$ werden die verbleibenden freien Intervalle (I_3) aufgefüllt, was in Abbildung 4.6b mit blauen Punkten gekennzeichnet ist. Für weitere Erweiterungen werden die resultierenden Intervalle $I_0 - I_3$ erneut geteilt und entsprechend der beschriebenen Methodik aufgefüllt.

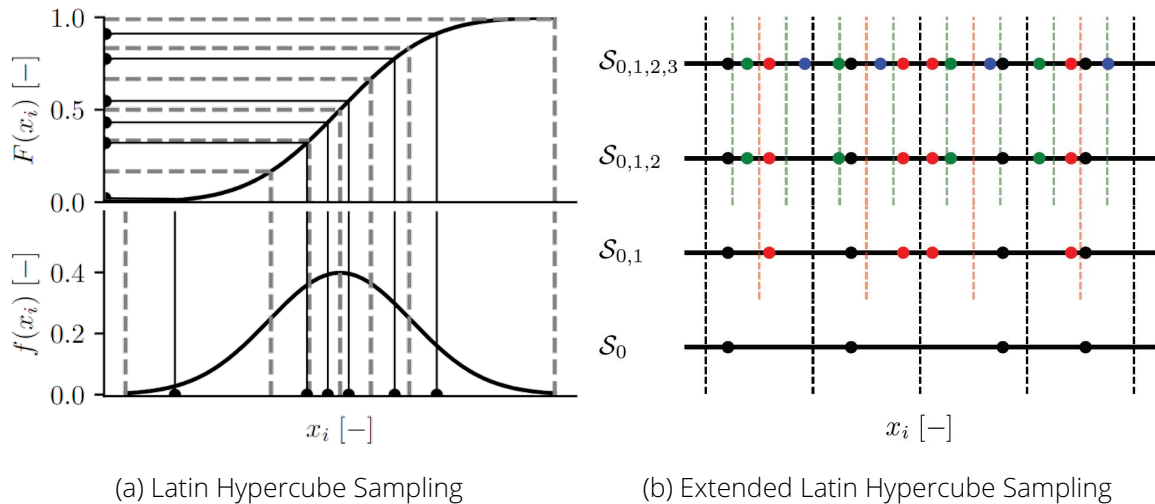


Abbildung 4.6: Systematik des Latin Hypercube Sampling und Methodik zur Unterteilung der Intervalle des Extended Latin Hypercube Samplings

Bei Vorliegen freier Teilintervalle ist mit höheren Abweichungen von der gewünschten Verteilung zu rechnen, als wenn alle Teilintervalle besetzt sind.

Die Korrelationsstruktur wird abschließend für jede Erweiterung mit dem Restricted-Pairing [52, 33] eingestellt. Im Rahmen des eLHS nach SCHMIDT et al. [148] erfolgt eine iterative Anwendung des Restricted-Pairings für eine verbesserte Korrelationskontrolle der gesamten Stichprobe.

In Hinblick auf die Erweiterung der MCS mittels eLHS ist die Initialisierung des Systems mit einer möglichst kleinen Anzahl von initialen Realisierungen empfehlenswert. Dies ermöglicht insbesondere bei rechenzeitintensiven deterministischen Modellen eine flexiblere Erweiterung der Realisierungen in kleinen Umfängen.

4.1.3 Numerische Modellierung

In zahlreichen ingenieurtechnischen Anwendungen stehen komplexe physikalische Problemstellungen wie Strömungsvorgänge, mechanische Spannungszustände und Wärmeübergänge im Fokus. Diese Phänomene werden häufig durch partielle Differentialgleichungen (PDE) beschrieben, deren exakte Lösungen in der Praxis jedoch selten möglich sind. Um diese Probleme rechnergestützt zu lösen, werden numerische Diskretisierungsmethoden eingesetzt. Diese Verfahren transformieren kontinuierliche Probleme in diskrete Modelle, die mithilfe moderner Rechentechnik analysiert werden können. In diesem Kapitel werden die zentralen numerischen Methoden vorgestellt, die zur Modellierung und Analyse der betrachteten Problemstellungen verwendet wurden. Im Mittelpunkt stehen die Finite-Elemente-Methode (FEM) und die Finite-Volumen-Methode (FVM), die für die mechanische Spannungsanalyse bzw. die Strömungsmechanik eingesetzt werden. Darüber hinaus wird die konjugierte Wärmeübertragung (CHT) betrachtet, die eine Kopplung zwischen numerischer Strömungsmechanik (CFD) und Festkörpermechanik ermöglicht. Im weiteren Verlauf werden zunächst die Grundlagen der verwendeten CFD-Methoden beschrieben, wobei der Wärmeaustausch an der Grenzfläche zwischen Fluid und Festkörper berücksichtigt wird. Für eine detailliertere Darstellung wird auf die Buchreihe von ZIENKIEWICZ et al. [182, 183, 184] verwiesen. Abschliessend werden die Aspekte der mechanischen Spannungsanalyse mittels Finite-Elemente-Modellierung erläutert, die in der Arbeit von KRYSL [95] beschrieben sind.

Elementtypen in der Diskretisierung

Für die Diskretisierung des Rechengebiets kommen verschiedene Elementtypen zum Einsatz, die sich hinsichtlich ihrer Funktionalität und Eignung je nach Anwendungsgebiet unterscheiden. Im Folgenden werden einige der Elemente vorgestellt, die in dieser Arbeit verwendet wurden. Die Geometrie des Simulationsmodells wird bei der Diskretisierung in eine endliche Anzahl von Elementen unterteilt. Diese Elemente können unterschiedliche Formen haben, die je nach Anwendungsgebiet und Lösungsmethode variieren. Zu den im Rahmen dieser Arbeit verwendeten Elementtypen zählen die Tetraeder, Hexaeder und Polyeder. Im einfachsten Fall besitzt jedes Element nur Knotenpunkte auf den Ecken des Elements, wie in den Abbildungen 4.7a, 4.7b, 4.7c in rot dargestellt. Es ist zudem möglich Mittelpunkt-Knoten während der Initialisierung des Rechengebiets auf den Kanten des Elements einzufügen. Diese in blau dargestellten Knoten spielen eine entscheidende Rolle, da sie häufig verwendet werden, um die Genauigkeit der Berechnungen zu erhöhen, insbesondere in Elementen höherer Ordnung. Durch die Integration von Mittelpunkt-Knoten in das Netz wird die Interpolation zwischen den benachbarten Elementen verbessert, was zu einer präziseren Darstellung der physikalischen Felder führt. Die Wahl des richtigen Elementtyps und die Implementierung von Mittelpunkt-Knoten sind entscheidend für die Qualität der Simulationsergebnisse. Insbesondere bei komplexen Geometrien und physikalischen Phänomenen, bei denen hohe Genauigkeit erforderlich ist, tragen diese Techniken dazu bei, numerische Instabilitäten zu vermeiden und die Konvergenz des Lösungsverfahrens zu fördern. Im Folgenden wird kurz auf die verwendeten Elemente eingegangen, wie sie unter anderem in Star-CCM+ [153] verwendet werden.

Tetraeder

Tetraeder sind vierseitige Elemente, die häufig in der Finite-Elemente-Methode (FEM) für dreidimensionale Spannungsanalysen eingesetzt werden. Sie bieten Flexibilität bei der Netzgenerierung für komplexe Geometrien. In STAR-CCM+ werden hierfür die in Abbildung 4.7a dargestellten C3D10-Elemente verwendet.

Hexaeder

Hexaeder sind sechsseitige Elemente, die vor allem in der Finite-Volumen-Methode (FVM) bevorzugt werden. Sie ermöglichen eine effiziente Berechnung durch ihre regelmässige Struktur und werden oft in CFD-Berechnungen insbesondere im wandnahen Bereich verwendet. Die geometrische Einfachheit des in Abbildung 4.7b dargestellten Elements, erlaubt eine kompakte Darstellung komplexer Strömungsfelder und verbessert die Stabilität der numerischen Lösung insbesondere im wandnahen Bereich.

Polyeder

Polyeder sind vielseitige Elemente, die sowohl in der FEM als auch in der FVM verwendet werden können. Ein Element ist in Abbildung 4.7c am Beispiel eines Dodekaeders dargestellt. Die Vielseitigkeit der Elemente erlaubt eine bessere Anpassung an unregelmässige Geometrien. Da ein Polyeder aus mehreren Tetraedern erstellt werden kann, werden diese Elemente häufig zur Reduzierung der Netzgröße verwendet. Abhängig von der Geometrie lässt sich dadurch die Elementanzahl um den Faktor 4 bis 5 reduzieren [165]. Zudem führt die Verwendung einer Polyederstruktur in der CFD in der Regel zu einer Verbesserung der Konvergenzgeschwindigkeit, wie von Sosnowski et al. [154] gezeigt. In STAR-CCM+ ist die Verwendung von Polyeder-Elementen in Verbindung mit dem Solid Stress Löser nicht möglich.

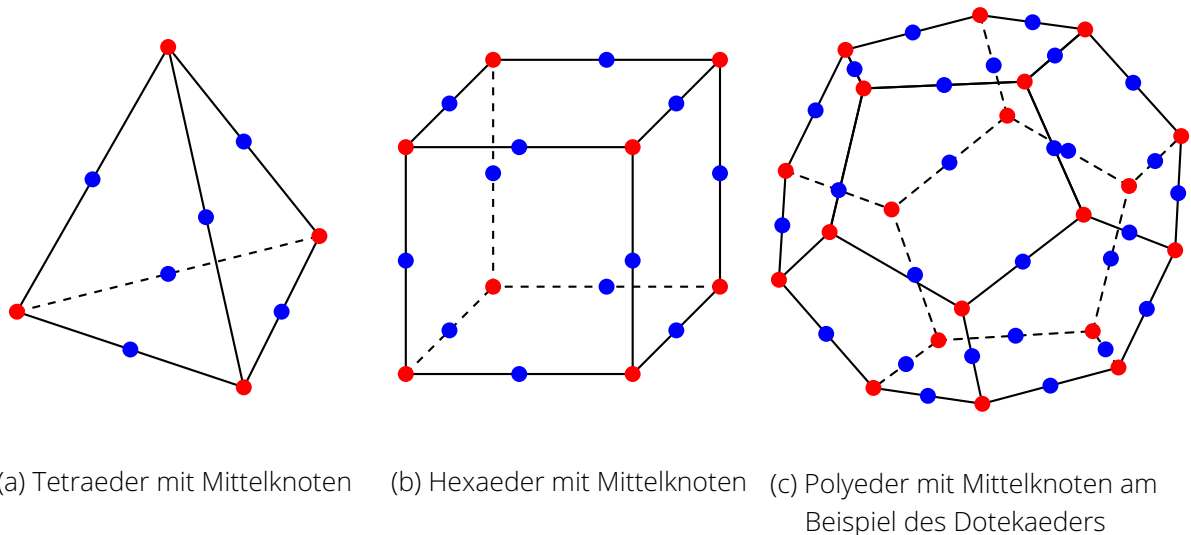


Abbildung 4.7: Elementtopologie

Numerische Strömungsmechanik (CFD) mit Finite-Volumen-Methoden (FVM)

Eine weit verbreitete Methode zur Lösung der zugrunde liegenden partiellen Differentialgleichungen in der numerischen Strömungsmechanik ist die Finite-Volumen-Methode (FVM). Sie

wird als konservative Berechnungsmethode bezeichnet, weil sie die Erhaltung von physikalischen Größen wie Masse, Impuls und Energie in diskreter Form sicherstellt. Dies geschieht durch die Integration der Erhaltungsgleichungen über Kontrollvolumina unter Berücksichtigung der Flüsse an den Grenzen dieser Volumina [44]. Die Definition eines Kontrollvolumens kann entweder nach dem Lagrange-Ansatz oder dem Euler-Ansatz erfolgen. Bei dem Lagrange-Ansatz stellt das Kontrollvolumen einen Teil des Materials dar, der sich während der Berechnung durch den Raum bewegt. Dieser Ansatz wird in der Regel für Partikelbeobachtungen im Fluid verwendet. Der Euler-Ansatz hingegen definiert ein feststehendes Volumen im Medium, durch das das Fluid hindurchströmen kann. In den meisten praktischen Anwendungen hat sich die Verwendung der eulerschen Beschreibung weitgehend durchgesetzt [44].

Erhaltungsgleichungen

Die bereits erwähnten Erhaltungsgleichungen für Masse, Impuls und Energie bilden die Grundlage für die numerische Strömungssimulation. Sie können in der einschlägigen Literatur wie beispielsweise [44, 93, 127] nachgelesen werden und sind durch Star-CCM+ [153] wie folgt definiert.

Massenerhaltung

Die Massenerhaltung, auch als Kontinuitätsgleichung bekannt, beschreibt, dass die Gesamtmasse innerhalb eines Kontrollvolumens erhalten bleibt. Dies bedeutet, dass jede zeitliche Änderung der Masse innerhalb des Kontrollvolumens durch die Differenz der eintretenden und austretenden Massenströme ausgeglichen werden muss. In differentieller Form lässt sich dies durch folgende Gleichung ausdrücken:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (4.12)$$

Dabei beschreibt der erste Term $\frac{\partial \rho}{\partial t}$ die lokale zeitliche Änderung der Dichte ρ , während der zweite Term $\nabla \cdot (\rho \mathbf{v})$ die Divergenz des Massenflusses berücksichtigt. Diese Gleichung stellt sicher, dass jede Veränderung der Dichte innerhalb des Kontrollvolumens durch Massentransport kompensiert wird. Hierbei ist $\mathbf{v}$ die Geschwindigkeit des Fluids und ρ die Dichte.

Impulserhaltung

Die Impulserhaltung besagt, dass die zeitliche Änderung des Impulses der Summe der auf das Kontrollvolumen angreifenden Kräfte entspricht. Die differentielle Form der Impulsgleichung 4.13 wird auch als Navier-Stokes Gleichung bezeichnet. Die Navier-Stokes Gleichungen (NSG) sind ein nichtlineares partielles Differenzialgleichungssystem zweiter Ordnung und lautet wie folgt:

$$\frac{\partial(\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v}) = -\nabla \cdot (p \mathbf{I}) + \nabla \cdot \mathbf{T} + \mathbf{f}_b \quad (4.13)$$

Der Term $\frac{\partial(\rho \mathbf{v})}{\partial t}$ beschreibt die zeitliche Änderung des Impulses durch die Beschleunigung des Fluids aufgrund von Druckgradienten, Reibung und äusseren Kräften. Der Term $\nabla \cdot (\rho \mathbf{v} \mathbf{v})$ beschreibt den Transport des Impulses durch die Strömung. Die auf die Oberfläche des Volumens wirkenden Oberflächenkräfte werden durch den Druckgradienten $-\nabla \cdot (p \mathbf{I})$ und den viskosen Spannungssensoren $\nabla \cdot \mathbf{T}$ berücksichtigt. Alle weiteren auf das Volumen angreifenden Kräfte, wie Schwerkraft und Zentrifugalkräfte, werden in $\mathbf{f}_b$ zusammengefasst.

Energieerhaltung

Bei Berücksichtigung des ersten Hauptsatzes der Thermodynamik auf das Kontrollvolumen, kann die differentielle Energieerhaltungsgleichung 4.14 wie folgt ausgedrückt werden:

$$\frac{\partial(\rho E)}{\partial t} + \nabla \cdot (\rho E \mathbf{v}) = -\nabla \cdot (\mathbf{v} p \mathbf{I}) + \nabla \cdot (\mathbf{v} \mathbf{T}) - \nabla \cdot \dot{\mathbf{q}} + \mathbf{f}_b \cdot \mathbf{v} + \dot{Q}_E \quad (4.14)$$

Hierbei ist E die spezifische Gesamtenergie, welche sich aus der kinetischen Energie und der inneren Energie des Volumens zusammensetzt. $\frac{\partial(\rho E)}{\partial t}$ berücksichtigt die zeitliche Änderung der spezifischen Gesamtenergie. $\nabla \cdot (\rho E \mathbf{v})$ und $-\nabla \cdot (\mathbf{v} p \mathbf{I})$ beschreibt den konvektiven Energietransport über das Kontrollvolumen. $\nabla \cdot (\mathbf{v} \mathbf{T})$ beschreibt die Umwandlung von mechanischen Energie in Wärme durch den viskosen Spannungstensor $\mathbf{T}$. Der Wärmestrom $\dot{\mathbf{q}}$ beschreibt die Wärmeleitung innerhalb des Elements und $\dot{Q}_E$ berücksichtigt zusätzlich definierte Energiequellen oder Energiesenken innerhalb des Kontrollvolumens.

Die FVM diskretisiert das Rechengebiet in eine endliche Anzahl von Kontrollvolumina. Für jedes dieser Volumina werden die oben genannten Erhaltungsgleichungen integriert. Die Werte der physikalischen Größen wie zum Beispiel Druck und Geschwindigkeit an den Zellgrenzen werden durch Interpolation aus den Werten der benachbarten Zellen ermittelt.

Turbulenzmodellierung

Turbulente Strömungen sind in ingenieurtechnischen Anwendungen weit verbreitet und stellen eine Herausforderung für die numerische Strömungssimulation dar. Der Großteil technisch relevanter Strömungen, insbesondere in Turbomaschinen, liegt in turbulenter Form vor. Das Strömungsfeld ist durch unregelmässige, wirbelbehafteten und instationären Charakteristik mit dreidimensionale Bewegung gekennzeichnet. Dabei werden die Mittelwerte der Strömungsgrößen wie Druck, Geschwindigkeit und Temperatur durch statistisch beschreibbare Fluktuationen überlagert. Diese Turbulenz verbessert die Mischung in der Strömung, erhöht jedoch auch die Wandreibung und den Druckverlust.

Für die Beschreibung turbulenter Strömungen kommen in der CFD verschiedene Ansätze zum Einsatz. Eine Möglichkeit ist die direkte numerische Simulation (DNS), bei der alle relevanten turbulenten Skalen direkt gelöst werden. Diese Methode erfordert jedoch eine hohe Rechenleistung aufgrund der Notwendigkeit fein aufgelöster Gitter und kleiner Zeitschritte, um auch kleinste turbulente Strukturen erfassen zu können. Die DNS ist mit einem sehr hohen numerischen Aufwand verbunden, der für technische Anwendungen zumeist nicht praktikabel ist. Dies trifft bei großen Berechnungsgebieten auch auf Methoden der Large Eddy Simulation (LES) zu, die die großskaligen Effekte der Turbulenz auflösen und die kleinskaligen Effekte modellieren. Um den Rechenaufwand zu reduzieren, wird derzeit häufig eine Modellierung mit Zerlegung der Strömung in einen zeitlichen Mittelwert und einem zeitabhängigen Schwankungsanteil verwendet. Durch Verwendung der zeitlich gemittelten Geschwindigkeit $\bar{\mathbf{v}}$ und des zeitlich gemittelten Druck $\bar{p}(t)$, sowie der zeitabhängigen Schwankungsanteile für Geschwindigkeit $\hat{\mathbf{v}}(t)$ und Druck $\hat{p}(t)$, kann die turbulente Charakteristik der Strömung durch die Fluktuation abgebildet werden.

$$\begin{aligned} \mathbf{v}(t) &= \bar{\mathbf{v}} + \hat{\mathbf{v}}(t) \\ p(t) &= \bar{p} + \hat{p}(t) \end{aligned} \quad (4.15)$$

Durch das Einsetzen der Summe aus Fluktuation und Mittelwert in die NSG und eine zeitliche Mittelung ergibt sich die Reynolds-gemittelten Navier-Stokes-Gleichungen (RANS). Die resultierenden Gleichungen für die Mittelwerte sind im wesentlichen identisch mit der ursprünglichen NSG. Die RANS-Gleichungen 4.16 enthalten zusätzliche Terme, wie den Reynoldsspannungstensor T_{RANS} , der den Einfluss der Turbulenz auf das mittlere Strömungsfeld beschreibt.

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \bar{\mathbf{v}}) &= 0 \\ \frac{\partial}{\partial t}(\rho \bar{\mathbf{v}}) + \nabla \cdot (\rho \bar{\mathbf{v}} \bar{\mathbf{v}}) &= -\nabla \cdot \bar{p}_{\text{mod}} \mathbf{I} + \nabla \cdot (\bar{\mathbf{T}} + T_{RANS}) + \mathbf{f}_b \\ \frac{\partial}{\partial t}(\rho \bar{E}) + \nabla \cdot (\rho \bar{E} \bar{\mathbf{v}}) &= -\nabla \cdot \bar{p}_{\text{mod}} \bar{\mathbf{v}} + \nabla \cdot (\bar{\mathbf{T}} + T_{RANS}) \bar{\mathbf{v}} - \nabla \cdot \bar{\mathbf{q}} + \mathbf{f}_b \bar{\mathbf{v}} \end{aligned} \quad (4.16)$$

mit

$$\bar{p}_{\text{mod}} = \bar{p} + \frac{2}{3} \rho k \quad (4.17)$$

und

$$T_{RANS} = -\rho \begin{pmatrix} \overline{v'_1 v'_1} & \overline{v'_1 v'_2} & \overline{v'_1 v'_3} \\ \overline{v'_1 v'_2} & \overline{v'_2 v'_2} & \overline{v'_2 v'_3} \\ \overline{v'_1 v'_3} & \overline{v'_2 v'_3} & \overline{v'_3 v'_3} \end{pmatrix} + \frac{2}{3} \rho k \mathbf{I} \quad (4.18)$$

Um das resultierende unterbestimmte Differentialgleichungssystem zu schliessen, müssen Turbulenzmodelle verwendet werden. Eine gängige Methode ist die Anwendung von Wirbelviskositätsmodellen, bei denen die Reynoldsschen Spannungen in Abhängigkeit von den mittleren Geschwindigkeitsgradienten und einem Proportionalitätsfaktor beschrieben werden. Diese turbulente Wirbelviskosität berücksichtigt die Erhöhung der Viskosität aufgrund turbulenter Schwankungsbewegungen. Es gibt verschiedene Ansätze zur Berechnung der Wirbelviskosität, die in Null-, Ein- und Zweigleichungsmodelle unterteilt werden. Nullgleichungsmodelle, wie beispielsweise das Turbulenzmodell nach BALDWIN und LOMAX [11] oder das Turbulenzmodell nach CEBECI und SMITH, verwenden empirische Formeln zur Schliessung. Eingleichungsmodelle hingegen benötigen eine zusätzliche Differentialgleichung zur Beschreibung der turbulenten Viskosität. Zu nennen ist hier das weit verbreitete Turbulenzmodell nach SPALART und ALLMARAS [155]. Zweigleichungsmodelle hingegen lösen zwei zusätzliche Differentialgleichungen zur Bestimmung der Wirbelviskosität. Zweigleichungsturbulenzmodelle hingegen sind Schliessungsansätze, die zwei gekoppelte Transportgleichungen für die Modellierung der Turbulenzgrössen verwenden. Häufig genutzte Modelle sind das $k - \varepsilon$ -Turbulenzmodell und das $k - \omega$ -Turbulenzmodell. In diesen Modellen wird die turbulente Viskosität mit der turbulenten kinetischen Energie k und der turbulenten Dissipation ε bzw. der turbulenten Frequenz ω verknüpft. Das $k - \varepsilon$ -Turbulenzmodell hat sich insbesondere für Strömungen ohne Wandnähe bewährt. Hingegen ist das $k - \omega$ -Turbulenzmodell gut für die Modellierung in wandnahen Bereichen geeignet. Um die Vorteile beider Modelle zu kombinieren wurde das ssT-Modell (engl. Shear Stress Transport Modell) nach MENTER [108] entwickelt, welches im wandnahen Bereich das $k - \omega$ -Modell und im Bereich der Hauptströmung das $k - \varepsilon$ -Modell verwendet. Daher hat sich diese Modell insbesondere für die numerische Strömungsberechnung von Turbinen durchgesetzt.

Grenzschichtmodellierung

Bezug nehmend auf das Kapitel 4.1.3 spielt die Turbulenzmodellierung ebenfalls eine wichtige Rolle in der Grenzschichtmodellierung. In der Strömungsmechanik beschreibt die Grenzschicht die dünne Schicht, in der die Strömungsgeschwindigkeit von Null an der Wand auf den Wert der freien Strömung ansteigt. Diese Schicht ist stark von der Viskosität des Fluids beeinflusst und spielt eine entscheidende Rolle für die Effizienz und Leistung von Turbinen, da sie direkt die Strömungsbedingungen an den Schaufeln beeinflusst. Die komplexen geometrischen Strukturen von Turbinenrotoren, insbesondere in Verbindung mit Mehrstrom-Kühlsystemen und Filmkühlungsbohrungen, erfordern eine präzise Turbulenzmodellierung im wandnahen Bereich. Hierbei ist zu beachten, dass die Strömung in den meisten Bereichen des Rotors aufgrund hoher Geschwindigkeiten und Druckgradienten grundsätzlich turbulent anzunehmen ist. Dennoch ist es möglich, dass lokal begrenzt Bereiche mit Grenzschicht in laminarer Form vorliegen. Dies wird aufgrund der sehr komplexen dreidimensionalen Strömungsmechanismen insbesondere im Bereich des laminar-turbulenten-Umschlags in 3D Simulationen häufig vernachlässigt, um den numerischen Aufwand und numerische Instabilitäten zu reduzieren. Diese Annahme ist gerechtfertigt, da die Strömung bereits durch die Systemrandbedingungen turbulent aufgeprägt werden.

Die korrekte Abbildung der Turbulenz im Grenzschichtbereich erfordert eine sorgfältige Wahl der Netzauflösung, die stark von dem verwendeten Turbulenzmodell abhängt. Aufgrund der Wandhaftbedingung bildet sich nahe der Wand ein ausgeprägter Geschwindigkeitsgradient aus. Daher ist die Netzauflösung in Wandnormalenrichtung wichtig um die dämpfende Wirkung der Turbulenzgrößen in der Grenzschicht korrekt abzubilden. Um den Geschwindigkeitsgradienten nahe der Wand ausreichend genau diskretisieren zu können ist es notwendig die Zellhöhe immer weiter zu verkleinern. Die damit einhergehende Vergrößerung der Zellanzahl ist aus numerischer Sicht nicht tragbar. Daher muss ein Modell für die Berechnung im wandnahen Bereich gefunden werden. Die Anforderung an die Netzauflösung unterscheidet sich in Abhängigkeit vom verwendeten Turbulenzmodell. Ein hierfür häufig verwendetes Modell zur Grenzschichtmodellierung wird durch FERZIGER [49] beschrieben. Es wird zunächst der dimensionslose Wandabstand y^+ durch die Gleichung 4.19 eingeführt:

$$y^+ = \frac{y \rho u_T}{\mu} \quad (4.19)$$

Dieser berechnet sich aus dem Abstand y normal zur Wand, der Fluidichte ρ , der dynamischen Viskosität μ und der Schubspannungsgeschwindigkeit u_T berechnet. Die Geschwindigkeit u wird entsprechend durch die Gleichung

$$u^+ = \frac{u}{u_T} \quad (4.20)$$

normiert.

Wie in FERZIGER [49] beschrieben, liegen die skalierten Geschwindigkeitsprofile für unterschiedliche Reynolds-Zahlen übereinander und unterscheiden sich nur in der äußeren Schicht der Grenzschicht. Der Verlauf der skalierten Geschwindigkeit ist in Abbildung 4.8 als blauer Graph dargestellt.

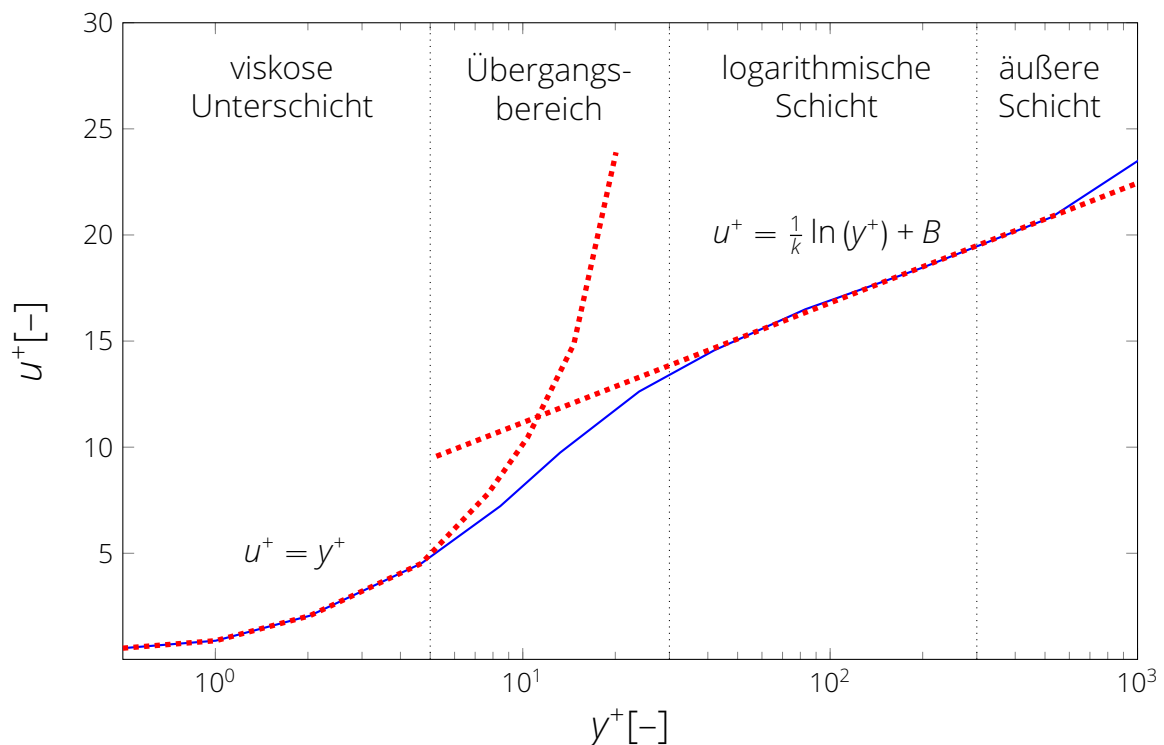


Abbildung 4.8: Dimensionsloses Geschwindigkeitsprofil der Grenzschicht, nach FERZIGER [49]

Das Geschwindigkeitsprofil lässt sich in Abhängigkeit vom dimensionslosen Wandabstand in die folgende Bereiche unterteilen:

Viskose Unterschicht

Die viskose Unterschicht wird in ihrem Stoff-, Energie- und Impulsaustausch durch ihre dynamische Viskosität dominiert und kann aufgrund der geringen Reynolds-Zahl als weitgehend laminar betrachtet werden. Hier gilt vereinfacht die Annahme $u^+ = y^+$ bis zu $y^+ < 5$.

Übergangsbereich

Der Übergangsbereich bildet den Übergang zwischen der viskosen Unterschicht und der logarithmischen Schicht, in welcher beide der genannten Schichten einen Einfluß haben.

Logarithmische Schicht

Die logarithmische Schicht beschreibt den Bereich ab $y^+ > 30$. Sie ist überwiegend durch Turbulenzeffekte geprägt und kann durch Gleichungen der Form

$$u^+ = \frac{1}{\kappa} \ln y^+ + B \quad (4.21)$$

beschrieben werden. Die Von-Karman-Konstante κ und der Wert für die Konstante B wurde mehrfach sowohl experimentell und durch DNS-Rechnungen bestimmt und wird je nach Simulationsmodell unterschiedlich definiert.

Äußere Schicht

Die äußere Schicht bildet den Übergang zur freien Strömung und wird abhängig von den Anforderungen an die Simulation und den Rechenaufwand unterschiedlich definiert. Durch die Modellierung der turbulenten Transportphänomene kann für glatte Wände der Rechenaufwand

mit ausreichender Genauigkeit gegenüber einer vollständig aufgelösten Grenzschicht um ein vielfaches reduziert werden.

Analog zur Modellierung der Geschwindigkeit im wandnahen Bereich erfolgt ebenfalls die Modellierung der Temperatur im wandnahen Bereich mittels analytischer Gleichungen nach KADER [84]. Hierfür die dimensionslose Temperatur T^+ eingeführt zu

$$T^+ = \rho c_p u_T \frac{T - T_w}{\dot{q}''} \quad (4.22)$$

wobei $T - T_w$ die Temperaturdifferenz zwischen Fluid und Wand, sowie c_p die spezifische Wärmekapazität des Fluids ist. $\dot{q}''$ wird als Wärmeflussdichte bezeichnet. In der Grenzschicht ist die Wärmeleitung (Konduktion) an der Wand von zentraler Bedeutung, da sie den Wärmefluss von der Wand in das angrenzende Fluid bestimmt. Dieser Prozess wird durch den Temperaturgradienten an der Wand beeinflusst, wobei eine hohe Temperaturdifferenz zwischen Wand und Fluid zu einer erhöhten Wärmeübertragung führt. Die Fourier'sche Wärmeleitungsgleichung beschreibt diesen Mechanismus und zeigt, wie die Wärme durch molekulare Interaktionen übertragen wird. Parallel dazu spielt auch die Konvektion eine entscheidende Rolle in der Grenzschichtströmung. Hierbei wird die Wärme durch die Bewegung des Fluids transportiert, was besonders in turbulenten Strömungen von Bedeutung ist. Die Grenzschicht kann für die Modellierung wie zuvor in 4 Bereiche unterteilt werden, wie in Abbildung 4.9 dargestellt.

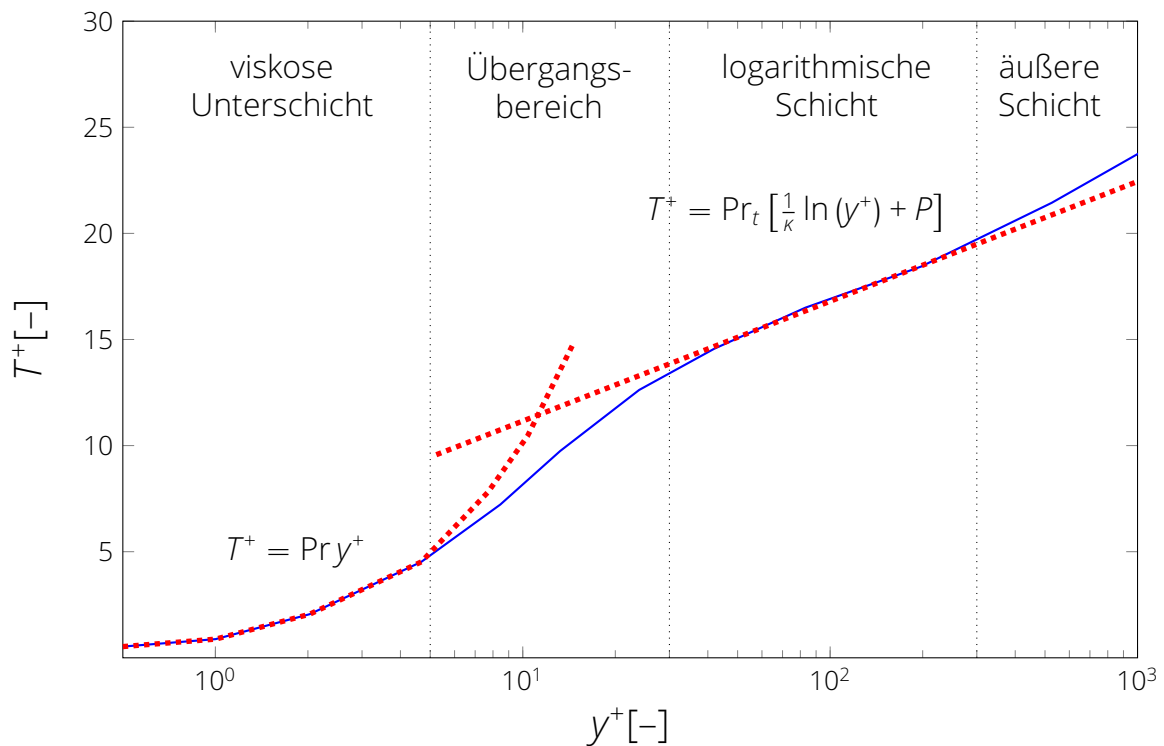


Abbildung 4.9: Dimensionsloses Temperaturprofil der Grenzschicht, vgl. [153]

In der viskosen Unterschicht strebt der turbulente Transport in Richtung Wand gegen 0 und der molekulare Wärmetransport überwiegt. Dieser Bereich kann durch

$$T^+ = \text{Pr} y^+ \quad (4.23)$$

beschrieben werden. Die Prandtl-Zahl Pr ist dabei ein dimensionsloses Mass, das das Verhältnis von kinematischer Viskosität zu thermischer Diffusivität beschreibt. In turbulenten Strömungen verbessert sich der Wärmetransport erheblich durch das turbulente Mischungsverhalten. Die logarithmische Schicht wird entsprechend durch die Gleichung 4.24 modelliert.

$$T^+ = Pr_t \left(\frac{1}{K} \ln(y^+) + P \right) \quad (4.24)$$

Pr_t ist die turbulente Prandtl-Zahl, gültig in der logarithmischen Grenzschicht. P ist eine Konstante, die die spezifischen Eigenschaften der Wand und der Strömung widerspiegelt.

Konjugierte Wärmeübertragung (CHT)

Die konjugierte Wärmeübertragung (CHT) beschreibt sowohl die Wärmeleitung in festen Materialien als auch die Wärmeübertragung durch das strömende Fluid. Für ein grundlegendes Verständnis der Wärmeübertragung sei an dieser Stelle auf den Wärmeatlas [175] verwiesen. Für ein fundamentales Verständnis der Simulationsmethode bietet HERWIG [69] einen guten Überblick. Im Weiteren wird auf die Modellierung der Wärmeübertragung zwischen Fluid und Festkörper eingegangen, welche bei der gekoppelten Simulation verwendet wird. Je nach Problemstellung kann die Kopplung zwischen den betrachteten Bereichen monodirektional oder bidirektional erfolgen. In der bidirektionalen Simulation wird die Wärmeübertragung zwischen dem Fluid und den festen Oberflächen in beider Richtungen berechnet, wodurch die Interaktion zwischen Feststoff und Fluid vollständig berücksichtigt wird. Die dabei wirkenden Hauptmechanismen sind die Wärmeleitung (Konduktion) und die Konvektion.

Die Wärmeleitung ist der Prozess, durch den Wärmeenergie durch direkte Wechselwirkungen zwischen den Teilchen eines Materials übertragen wird. Energie fließt dabei entsprechend des ersten Hauptsatzes der Thermodynamik stets von Bereichen höherer Temperatur zu Bereichen niedrigerer Temperatur. Wärmeleitung findet sowohl im Fluid, als auch im Festkörper statt. Im Fluid erfolgt die Wärmeübertragung durch molekulare Interaktionen und der Diffusion wie in Kapitel 4.1.3 zur Grenzschichtmodellierung bereits hergeleitet wurde. In Festkörpern geschieht die Wärmeleitung hauptsächlich durch Vibrationen der Atome im Gitter. Da in Festkörpern der Atomabstand wesentlich geringer ist als in Gasen, ist in diesen Bereichen die Wärmeleitung um ein vielfaches grösser. In der differentiellen Form beschreibt das Fourier'sche Gesetz in Gleichung 4.25 den lokalen Wärmefluss $\dot{q}''$.

$$\dot{q}'' = -k \nabla T \quad (4.25)$$

k ist die thermische Leitfähigkeit des Materials und ist in metallischen Festkörpern wesentlich grösser als in Gasen. ∇T ist dabei der Temperaturgradient im Material. Dementsprechend ist die folgende Energiegleichung 4.26 von zentraler Bedeutung für die Beschreibung der Wärmetransports im Festkörper.

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + Q_E \quad (4.26)$$

wobei die zeitliche Änderung der Temperatur $\frac{\partial T}{\partial t}$ durch die Dichte ρ , die spezifische Wärmekapazität c_p und die thermische Leitfähigkeit k des Festkörpers beeinflusst werden. Q_E berücksichtigt etwaige Wärmequellen im Körper.

Die Konvektion kann in zwei Hauptarten unterteilt werden. Zum einen die frei Konvektion bei der natürliche Auftriebskräfte die Bewegung des Fluids antreiben und zum anderen die erzwungene Konvektion bei der externe Kräfte das Fluid bewegen. Die konvektive Wärmeübertragung an den Schaufeln wird durch das Newtonsche Kühlgesetz beschrieben:

$$\dot{q}_s'' = h(T_w - T_{\text{ref}})$$

Hierbei ist $\dot{q}_s''$ der lokale Oberflächenwärmestrom (Leistung pro Flächeneinheit), h der lokale konvektive Wärmeübergangskoeffizient, T_w die Temperatur an der Schaufeloberfläche und T_{ref} eine charakteristische Temperatur des über die Schaufel strömenden Fluids. Der Wärmeübergangskoeffizient hängt stark von den spezifischen Strömungsbedingungen ab und kann einen stark nichtlinearen Charakter aufweisen. Die Charakterisierung der Strömung kann anhand folgenden Ausdruck vorgenommen werden:

$$\begin{aligned} \frac{Ra}{PrRe^2} &\gg 1 && : \text{frei Konvektion dominant} \\ \frac{Ra}{PrRe^2} &\ll 1 && : \text{erzwungene Konvektion dominant} \\ \frac{Ra}{PrRe^2} &\approx 1 && : \text{gemischte Konvektion} \end{aligned} \quad (4.27)$$

Dabei ist die Rayleigh-Zahl Ra ist ein dimensionsloses Mass, das das Verhältnis von Auftriebskräften zu viskosen Kräften beschreibt. Die Prandtl-Zahl Pr beschreibt das Verhältnis von Impuls- zu Wärmeübertragung und die Reynolds-Zahl Re ist ein Mass für das Verhältnis von Trägheits- zu viskosen Kräften. Da dieser Wert für Turbinenschaufeln in der freien Strömung Werte kleiner 1 annimmt, kann in weiten Teilen der Strömung von einer erzwungenen Konvektion ausgegangen werden. Entsprechend der Ausführungen im Kapitel 4.1.3 wird fluidseitig hierfür eine Modellierung von h vorgenommen, welche von den Strömungsparametern abhängig ist nach [151]:

$$h = \frac{\rho_f C_{p,f} U_T}{T^+ (y^+)}$$

Da basierend auf der Energieerhaltung der Gesamtwärmestrom über ein Wandelement zum Fluidelement bzw. vom Fluidelement zum Wandelement erhalten bleibt, ergibt sich folgende Gleichung für die konjugierte Wärmeübertragung welche es zu lösen gilt:

$$\dot{q}_f + \dot{q}_w = -\dot{Q}_E \quad (4.28)$$

Dabei ist $\dot{q}_f$ der Wärmestrom vom Fluid und $\dot{q}_w$ der Wärmestrom von der Wand die gemeinsame Kontaktfläche. Wobei Q_E eine definierbare Wärmequelle darstellt.

Spannungsberechnung mit Finite-Element-Methode (FEM)

Die Finite-Elemente-Methode (FEM) ist ein leistungsfähiges numerisches Verfahren zur Lösung partieller Differentialgleichungen und findet insbesondere in der Strukturmechanik zur Spannungsanalyse Anwendung. Bei der FEM wird das kontinuierliche physikalische Problem in eine endliche Anzahl von finiten Elementen unterteilt. Innerhalb jedes Elements werden lokale Gleichungen verwendet, die sich aus den globalen Grundgleichungen des gesamten Systems ableiten und durch spezielle Formfunktionen die Feldgrößen approximieren. Jedes Element trägt mit diesen lokal definierten Gleichungen zum gesamten System bei. Durch die

Verbindung dieser Elemente zu einem globalen Gleichungssystem wird die Lösung für das gesamte Problem ermittelt. Diese Methode ermöglicht es, komplexe physikalische Phänomene präzise zu modellieren und zu analysieren. Insbesondere wird die FEM bevorzugt im Bereich der Festkörpermechanik verwendet, um das Verhalten eines Festkörpers als Reaktion auf einwirkende Lasten zu berechnen. Hierbei handelt es sich meist um Körperkräfte, Oberflächenlasten, Punktkräfte oder thermische Lasten, die berücksichtigt werden müssen. Eine umfassende Beschreibung der FEM im Bereich der Spannungsanalyse findet sich in der Literatur, beispielsweise bei KRYS [95], KLEIN [90] und STEINKE [160].

Die Zerlegung der Geometrie erfolgt im Bereich der FEM entsprechend der geometrischen Randbedingungen in eine endliche Anzahl diskreter Elemente, wie in Kapitel 4.1.3 beschrieben. Ziel der Analyse ist es, die Verformung und die resultierenden Spannungen der Elemente und somit im Bauteil zu beschreiben. Hierfür wird die Verformung $\mathbf{x}$ als Kombination aus starrer Körperbewegung $\mathbf{X}$ und relativen Verschiebungen $\mathbf{u} = (u_1 \ u_2 \ u_3)^T$ der Knotenpunkte beschrieben.

$$\mathbf{x} = \mathbf{X} + \mathbf{u} \quad (4.29)$$

Für Modelle mit kleinen Verschiebungen und kleinen Dehnungen kann für die infinitesimale Verzerrung folgender Zusammenhang angenommen werden:

$$\boldsymbol{\varepsilon} = \frac{1}{2} \left(\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \left(\frac{\partial \mathbf{u}}{\partial \mathbf{x}} \right)^T \right) \quad (4.30)$$

woraus Verzerrung und Dehnung an jedem Knoten im Körper vollständig mit einem Verzerrungstensor $\boldsymbol{\varepsilon}$ nach folgender Gleichung beschrieben werden kann:

$$\boldsymbol{\varepsilon} = \begin{pmatrix} \frac{\partial u_{11}}{\partial x_1} & \frac{1}{2} \left(\frac{\partial u_{11}}{\partial x_2} + \frac{\partial u_{22}}{\partial x_1} \right) & \frac{1}{2} \left(\frac{\partial u_{33}}{\partial x_1} + \frac{\partial u_{11}}{\partial x_3} \right) \\ \frac{1}{2} \left(\frac{\partial u_{11}}{\partial x_2} + \frac{\partial u_{22}}{\partial x_1} \right) & \frac{\partial u_{22}}{\partial x_2} & \frac{1}{2} \left(\frac{\partial u_{33}}{\partial x_2} + \frac{\partial u_{22}}{\partial x_3} \right) \\ \frac{1}{2} \left(\frac{\partial u_{33}}{\partial x_1} + \frac{\partial u_{11}}{\partial x_3} \right) & \frac{1}{2} \left(\frac{\partial u_{22}}{\partial x_3} + \frac{\partial u_{22}}{\partial x_3} \right) & \frac{\partial u_{33}}{\partial x_3} \end{pmatrix} \quad (4.31)$$

Ausgehend von der Verzerrungstensor in Gleichung 4.31 kann nun bei linear-elastischer Betrachtung der Spannungstensor $\boldsymbol{\sigma}$ in Cauchy-Notation wie folgt angegeben werden:

$$\boldsymbol{\sigma} = E \boldsymbol{\varepsilon} \quad (4.32)$$

mit

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{11} & \tau_{12} & \tau_{13} \\ \tau_{21} & \sigma_{22} & \tau_{23} \\ \tau_{31} & \tau_{32} & \sigma_{33} \end{pmatrix} E = \begin{pmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33} \end{pmatrix}. \quad (4.33)$$

Somit steht die Spannung im direkten Zusammenhang mit den Dehnungen und Verzerrungen durch das Elastizitätsmodul E .

Da moderne Turbinenschaufel in der Regel aus einem einkristallinen Material bestehen, weisen diese ein nicht-isotropes Verhalten auf. Genauer beschrieben werden können diese mit orthotropen Materialien mit unabhängige Eigenschaften in drei zueinander orthogonalen Richtungen. Dies ist durch die Materialmatrix $\mathbf{D}$ zu berücksichtigen, wobei die Gleichung 4.32 zunächst in die Voigt-Notation überführt wird:

$$\boldsymbol{\sigma} = \mathbf{D} \boldsymbol{\varepsilon} \quad (4.34)$$

$$\begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{23} \\ \sigma_{13} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & c_{13} & 0 & 0 & 0 \\ & c_{22} & c_{23} & 0 & 0 & 0 \\ & & c_{33} & 0 & 0 & 0 \\ & & & G_{12} & 0 & 0 \\ & & & & G_{23} & 0 \\ & & & & & G_{13} \end{pmatrix} \begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{12} \\ 2\varepsilon_{23} \\ 2\varepsilon_{13} \end{pmatrix} \quad (4.35)$$

Dabei sind die Einträge G_{ij} die Schermodule des Materials. Die Konstanten c_{ij} sind abhängig vom Elastizitätsmodule E_i und den Querkontraktionszahlen ν_i in den entsprechenden Hauptrichtungen. Diese Materialparameter sind untereinander abhängig von der Materialtemperatur am jeweiligen Knoten und werden mittels Datenbanken interpoliert. Bei Berechnungen mit CHT ist es unerlässlich, die Wechselwirkungen zwischen Wärmeübertragung und mechanischen Deformationen zu berücksichtigen. Diese Temperaturänderungen können zusätzliche interne Spannungsfelder erzeugen, die zu einer Verformung der Struktur führen. Entsprechend wird die thermische Dehnung ε_T nach Gleichung 4.36 definiert:

$$\varepsilon_T = \alpha(T - T_{\text{ref}}) \quad (4.36)$$

wobei die Referenztemperatur T_{ref} als Temperatur ohne thermische Beanspruchung angenommen wird. Die thermischen Ausdehnungskoeffizienten α_i entlang der Hauptachsen werden zu einem Vektor der thermischen Ausdehnungskoeffizienten α zusammengefasst. Damit folgt aus Gleichung 4.34

$$\sigma = D(\varepsilon - \varepsilon_T) \quad (4.37)$$

für die Spannungsberechnung mit thermischer Randbedingung.

Wie bereits zuvor angeführt erfolgt die Diskretisierung der Geometrie mittels einer endlichen Anzahl von Elementen, welche in Abbildung 4.7 dargestellt sind. Zu den typischerweise verwendeten Elementen gehören hierbei Hexaeder und Tetraeder. Dabei finden im Bereich der FEM Hexaederelemente vorzugsweise bei Geometrien mit regelmässiger Form Verwendung, da quadratische Hexaederelemente bei gleicher Anzahl von Freiheitsgraden präzisere Ergebnisse als Tetraederelement liefern [40]. Jedoch ist eine automatische Vernetzung mit Hexaeder in der Regel fehleranfälliger. Für die automatische Vernetzung von Bauteilen werden Tetraederelemente verwendet, da diese sich besser an komplexe Geometrien anpassen. Bei allen Elementen werde in der Regel Seitenmittelnknoten verwendet, da diese die Steifigkeit im System besser abbilden, was jedoch durch eine erhöhte Knotenanzahl mit einem erhöhten numerischen Aufwand einhergehen. Das physikalische Verhalten der Elemente wird durch mathematische Ansatzfunktionen beschrieben, welche in der Regele erster oder zweiter Ordnung sind. Im Detail nachzulesen in dem Buch von KLEIN [90]. Ansatzfunktionen dienen dazu, Feldgrößen (z.B. Temperatur, Spannung, Verschiebung) zwischen den Knotenpunkten eines Elements zu interpolieren. Die ermittelten Verschiebungen eines Elements werden durch gemeinsame Knoten auf benachbarte Elemente übertragen. Die Genauigkeit der auf den Knoten berechneten Ergebnisgrößen hängt stark von der Qualität der für die Diskretisierung verwendeten Elemente ab. Der in Star-CCM+ [153] integrierte Vernetzer, bieten hierfür die geläufigen Metriken an um die Identifizierung schlecht geformter Elemente zu ermöglichen:

- Die **Winkelschiefe** beschreibt die Abweichung eines Elements von idealen Winkelverhältnissen. Dabei liegt der ideale Winkel beispielsweise für dreieckige Flächen bei 60.0° und für viereckige Flächen bei 90.0° . Ein Wert von 1 stellt eine ideale Zelle dar, während Werte nahe 0 auf eine schlechte Qualität hinweisen.
- Das **Seitenverhältnis** bewertet das Verhältnis der längsten zur kürzesten Kante eines Elements. Dabei haben gleichseitigen Element einen Wert von 1 und sind als ideal zu betrachten. Hingegen weisen Elemente mit einem Wert grösser 10 auf eine schlechte Qualität hin.
- Die **Mittelpunktabweichung** beurteilt die Position von Mittelknoten im Verhältnis zu den Eckknoten eines Elements. Diese ist insbesondere für Elemente mit Mittelknoten wie quadratischen Elementen relevant. Ideale Elemente haben einen Wert 0 und Werte kleiner 0.25 sind akzeptabel.
- Die **Formabweichung** basiert auf der Jacobi-Determinante der Jacobi-Matrix. Die Jacobi-Determinante beschreibt wie sich das Volumen oder die Fläche eines idealen Elements bei der Transformation an die Geometrie verändert. Die Formabweichungsmetrik wird aus dem grössten und kleinsten Wert der Jacobi-Determinante gebildet. Idealer Elemente haben einen Wert von 1, wobei zulässige Elemente einen Wert im Bereich $[0,1]$ liegen und Werte von -1 auf eine fehlerhafte Zelle hinweisen.
- Der **Flächenverzug** beschreibt den Abstand des Flächenscheitelpunkts von der aufgespannten Flächenebene bezogen auf der mittleren Kantenlänge der betrachteten Seitenfläche. Werte von 0 entsprechend dem idealen Element und Elemente sind bis zu einem Wert von 0.5 zulässig.

Werden diese Qualitätsmerkmale für die FEM-Vernetzung nicht eingehalten kann dies zur fehlerhaften Simulationsergebnissen führen.

Bei erfolgreicher Berechnung des Verschiebungs- und damit des Spannungszustandes im Bauteil kann anhand verschiedener Spannungsdefinitionen eine Auswertung am Bauteil erfolgen, wobei im Folgenden auf die wichtigsten eingegangen werden soll.

Cauchy-Spannung

Der wahre Spannungszustand wurde bereits in Gleichung 4.32 definiert und wird auch als Cauchy-Spannung bezeichnet. Die Cauchy-Spannung ist ein direktes Mass für die Spannung an einem beliebigen Punkt im Bauteil, der sowohl Normalspannungen als auch Scherspannungen umfasst.

Hauptspannungen

Die Hauptspannungen sind die minimal Normalspannung s_1 , die mittlere Normalspannung s_2 und maximale Normalspannung s_3 , welche sich im Material unter multi-direktionalen Spannungsbedingungen einstellen. Hierfür lassen sich in jedem Punkt des Materials Schnittebenen bestimmen, in welchen die Schubspannung zu Null wird. Berechnen lassen sich die Hauptspannungen s_i durch folgende Gleichung:

$$\det(\boldsymbol{\sigma} - s_i \mathbf{I}) = 0 \quad (4.38)$$

Per Konvention wird eine positive Spannung als Zug und eine negative Spannung als Druck im Bauteil interpretiert. Die Hauptspannungen stellen die maximalen und minimalen Normalspannungen dar, die im Material auftreten können.

Von Mises Spannung

Die von-Mises-Spannung beruht auf der Verzerrungsenergie-Theorie. Diese Theorie geht davon aus, dass das Material versagt, wenn die Verzerrungsenergie, die durch komplexe Spannungszustände erzeugt wird, einen kritischen Wert überschreitet. Die von-Mises-Spannung wird aus den Hauptspannungen berechnet, mittels der folgenden Gleichung:

$$\sigma_{\text{Mises}} = \sqrt{\frac{1}{2} \left((s_1 - s_2)^2 + (s_1 - s_3)^2 + (s_2 - s_3)^2 \right)} \quad (4.39)$$

Dabei handelt es sich um eine Vergleichsspannung, mit dem Ziel, den komplexen Spannungszustand in eine vergleichbare Zahl umzuwandeln. Die von-Mises-Spannung stellt jedoch keine tatsächliche Spannung im Material dar, sondern ist ein Mass für die Verzerrungsenergie, die durch die Kombination von Normal- und Schubspannungen entsteht. Daher wird die von Mises Spannung häufig in der Finite-Elemente-Analyse (FEA) verwendet, um das Risiko von Materialversagen unter komplexen Belastungen zu bewerten.

4.2 Parametrisierung der Variabilität von Hochdruckturbinenschaufeln (AP2.1)

4.2.1 Parametrisierung des Schaufelblatts

Für die Parametrisierung der Turbinenschaufeln in diesem Projekt wird auf bestehende Arbeiten von LANGE [97] zurückgegriffen, die von HÖGNER [77] bereits auf Turbinenschaufeln übertragen wurden. Im Folgenden wird daher das parametrische Modell für das Schaufelblatt in seinen Grundzügen beschrieben. Darüber hinaus wird auf notwendige Anpassungen innerhalb der Parametrisierung eingegangen, die für die Integration der neuen Schaufelgeometrie erforderlich sind.

Die parametrische Beschreibung der Schaufel basiert auf optischen Messdaten, die als Polygonnetz in kartesischen Koordinaten vorliegen. Die Daten wurden mit einer ATOS Scanbox der Firma GOM aufgenommen und mit der Verarbeitungssoftware GOM Inspect aufbereitet. Dies wird im Kapitel 4.1.1 beschrieben. Im Kapitel 4.3.3 wird gezeigt, wie die Daten anhand des Mittelwertmodells über den Heissgasbereich zueinander ausgerichtet wurden. Wie in Abbildung 4.10 durch schwarze Linien dargestellt, erfolgt die Extraktion der Profilschnitte auf äquidistant angeordneten Umfangsschnitten auf konstanter relativer Schaufelhöhe s . Die relative Schaufelhöhe orientiert sich anhand der Nabenkontur r_h und der Wingletkontur r_s , die in orange dargestellt sind.

Die relative Schaufelhöhe kann anhand folgender Gleichung errechnet werden.

$$s = \frac{r - r_h(x)}{r_s(x) - r_h(x)} \text{ für } s \in [0,1] \quad (4.40)$$

Die so aus dem Polygonnetz extrahierten Sektionen, bestehend aus den Punkten $X_i = [x_i, y_i, z_i]$, welche sich bei Durchdringung der Polygonnetzkannten mit der Schnittebene ergeben. Diese werden in ein zweidimensionales meridionales $r\theta - m$ - Koordinatensystem überführt. Das hier verwendete meridionale Koordinatensystem $r\theta$ und Umfangskoordinate m ermöglichen eine längenkonforme Darstellung der dreidimensionalen Schnitte. Die Transformation eines diskreten Punktes der Sektion erfolgt mittels folgender Gleichungen [75]

$$r\theta_i = r_i \cdot \theta_i \quad (4.41)$$

und

$$m_i = \begin{cases} m_{i-1} + \sqrt{(r_i - r_{i-1})^2 + (x_i - x_{i-1})^2} & \text{für } \Delta x_i \geq 0 \\ m_{i-1} - \sqrt{(r_i - r_{i-1})^2 + (x_i - x_{i-1})^2} & \text{für } \Delta x_i < 0 \end{cases} \quad (4.42)$$

wobei sich der Radius r_i und der Winkel θ_i für die betrachtete Sektion wie folgt errechnen lassen.

$$r_i = \sqrt{y_i^2 + z_i^2} \quad (4.43)$$

$$\theta_i = \arctan\left(\frac{y_i}{z_i}\right) \quad (4.44)$$

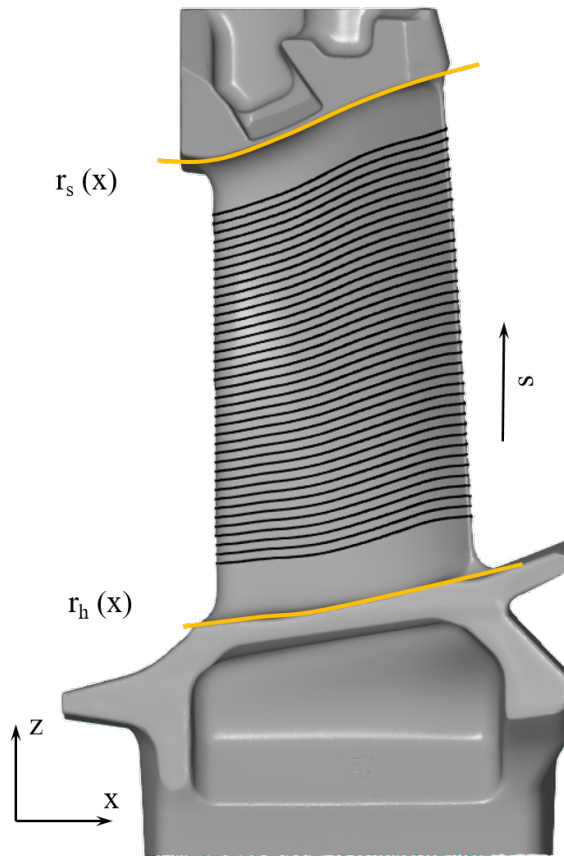


Abbildung 4.10: Profilschnitte einer vermessenen Turbinenschaufel

Anhand der zweidimensionalen Sektion im $r\theta - m$ – Koordinatensystem erfolgt eine inverse Bestimmung der Skelettlinie wofür im Wesentlichen zwei unterschiedliche Methoden zur Verfügung stehen. Zum einen die Rollradienmethoden, der von LANGE [97] verwendet wurde oder zum anderen ein Algorithmus von VORONOI mit DELAUNY-Triangulation, welcher von HEINZE et al. [68] verwendet wurde. Wie in der letzteren Veröffentlichung gezeigt wurde, erweist sich die Skelettlinienbestimmung mittels der VORONOI-Diagramme mit DELAUNY-Triangulation im Bereich der Turbinenschaufelanalyse als deutlich robuster. Ein wesentlicher Grund ist hierfür die vergleichsweise grössere Dicke und Wölbung der Turbinenprofile im Vergleich zu Verdichterschaufeln. Dennoch sind beide Algorithmen nicht in der Lage, die Skelettlinie zuverlässig im LE- und TE-Bereich zu bestimmen. Daher ist es notwendig die Skelettlinie im Bereich des LE mittels eines quadratischen Polynoms zu extrapolieren. Im Bereich des TE wird hingegen eine lineare Extrapolation durchgeführt. Ausgehend von der Skelettlinie erfolgt die Bestimmung der Schaufelparameter, die in Abbildung 4.11a dargestellt sind.

Die Position der Schaufel wird anhand des Durchstoßpunktes der Skelettlinie mit dem Schaufelprofil durch den Punkt (pos_{ax}, pos_{tan}) beschrieben. Unter Einbeziehung des Durchstoßpunktes der Skelettlinie an der Profilhinterkante können weiterhin der Staffelungswinkel θ und die Sehnenlänge l_{ch} bestimmt werden. Zur Bestimmung des Vorderkantenradius r_{LE} wird ausgehend vom Durchstoßpunkt der Skelettlinie ein Kreis mit einem Öffnungswinkel von $\pm 25^\circ$ in den Vorderkantenbereich der Schaufel gefittet, dessen Mittelpunkt sich auf der Skelettlinie

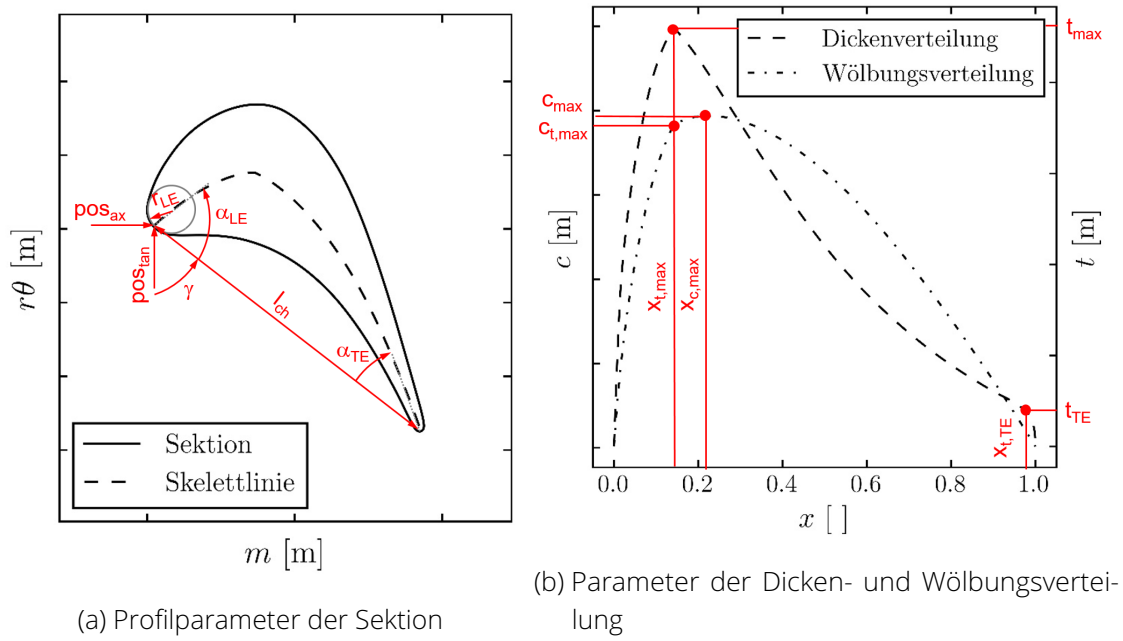


Abbildung 4.11: Parametrisierung einer Turbinenschaufel nach [68]

befindet. Durch Abtragen der Skelettlinie auf der normierten Profilschneide kann die Wölbungsverteilung $c(x)$ und die Dickenverteilung $t(x)$ bestimmt werden. Die Beschreibung der Dicken- und Wölbungsverteilung erfolgt in Anlehnung an die traditionelle Profiltheorie, die häufig auch als NACA-Parameter bezeichnet wird. Nachzulesen in den Büchern von BRÄUNLING [27] und ANDERSON [4]. Die Bestimmung der maximalen Dicke $t_{\max}$ und der maximalen Wölbung $c_{\max}$ erfolgt nicht direkt anhand der zuvor bestimmten diskreten Punkte der Verteilung. Für eine robustere Parametrisierung erfolgt die Bestimmung mit Hilfe eines quadratischen Polynoms im Bereich $\pm 2\%$ der maximalen Dicke bzw. der maximalen Wölbung. Anhand der berechneten Polynomfunktion können nun mehr analytisch die Maxima, sowie die Dickenrücklage $x_{c,\max}$ und Wölbungsrücklage $x_{t,\max}$ der Verteilungen berechnet werden. $c_{t,\max}$ beschreibt die Wölbung an der Position der maximalen Dicke (vgl. 4.11 b). Die Bestimmung der Hinterkantendicke t_{TE} erfolgt aus der Steilheit der Dickenverteilung an der Hinterkante nach folgender Gleichung.

$$\arctan \left(\left| \frac{dt(x)}{dx} \right| \right) \geq 45^\circ \quad (4.45)$$

Daraufhin kann die relative Position der Hinterkantendicke $x_{t,\max}$ auf der Profilschneide angegeben werden. Der Hinterkantenwinkel α_{TE} kann dementsprechend durch Gleichung 4.46 berechnet werden.

$$\alpha_{TE} = \arctan \left(\frac{dc}{dx} \right) \quad (4.46)$$

Analog dazu erfolgt die Bestimmung des Vorderkantenwinkels α_{LE} unter Berücksichtigung des Mittelpunktes des Vorderkantenkreises mit Gleichung 4.47. Da der Vorderkantenkreis durch die Kenntnis des Radius und die zuvor festgelegte Mittelpunktsbedingung auf der Wölbungslinie eindeutig beschrieben ist, kann auf einen weiteren Positionsparameter verzichtet werden.

$$\alpha_{LE} = \arctan \left(\frac{dc}{dx} \right) \quad (4.47)$$

Alle soeben beschreiben Parameter können der nachfolgenden Tabelle entnommen werden.

Tabelle 4.2: Profilparameter Turbine

Symbol	Parameter	Einheit
pos_{ax}	axiale Position	m
pos_{tan}	tangetiale Position	m
l_{ch}	Sehnenlänge	m
γ	Staffelungswinkel	°
α_{LE}	Vorderkantenwinkel	°
α_{TE}	Hinterkantenwinkel	°
r_{LE}	Vorderkantenradius	m
t_{TE}	Hinterkantendicke	m
t_{max}	maximale Dicke	m
c_{max}	maximale Wölbung	m
$c_{t,max}$	Wölbung an Position maximaler Dicke	m
$x_{t,max}$	Dickenrücklage	-
$x_{t,TE}$	Dicke der Hinterkante	-
$x_{c,max}$	Wölbungsrücklage	-

4.2.2 Parametrisierung des Kühlluftkanals

Das Mehrpassagen-Kühlsystem der untersuchten HDT-Schaufeln besteht aus insgesamt zwei Kühlluftpassagen, die sich aus vier Kühlluftkanälen zusammensetzen. Die vordere Passage wird durch den ersten Kühlluftkanal gebildet, der für die Kühlung des LE-Bereiches der Schaufel verantwortlich ist. Die zweite Kühlluftpassage umfasst drei Kühlluftkanäle, die durch U-Biegungen miteinander verbunden sind und den Mittelbereich der Schaufel bis zur Hinterkante kühlen. Die vier Kühlluftkanäle werden im weiteren fortlaufend von der Schaufelvorderkante mit 1 bis zur Schaufelhinterkante mit 4 nummeriert. Die Parametrisierung des Kühlluftkanals basiert auf den bereitgestellten CT-Messungen. Diese wurden automatisch von den Turbulatoren bereinigt. Der hierzu entwickelte Prozess wird in Kapitel 4.3.2 beschrieben. Dies ist notwendig, da sowohl die Höhe, Position und Anzahl der entfernten Rippen einen Einfluss auf den berechneten Schwerpunkt und damit auf die Berechnung der Verschiebung des Kühlluftkanals haben, wie in Abbildung 4.12 dargestellt. Da die Kühlluftlöcher erst in einem späteren Fertigungsschritt hinzugefügt werden, beeinflussen diese die erfasste Innengeometrie nicht.

Für die Parametrisierung des Kühlluftkanals stehen drei CT-Schnitte zur Verfügung, wie in Abbildung 4.10 dargestellt. Analog zur Parametrisierung des Schaufelblatts wird ein sektionen-basierter Ansatz verfolgt, um die geometrische Variation zu beschreiben. Da eine Projektion der Schnittebenen auf die in Kapitel 4.2.1 definierten Profilschnitte zu signifikanten Projektionsfehlern führen würde, erfolgt die Kühlkanalparametrisierung auf den normierten CT-Schnitten auf den z-Ebenen 320 mm, 330 mm und 345 mm.

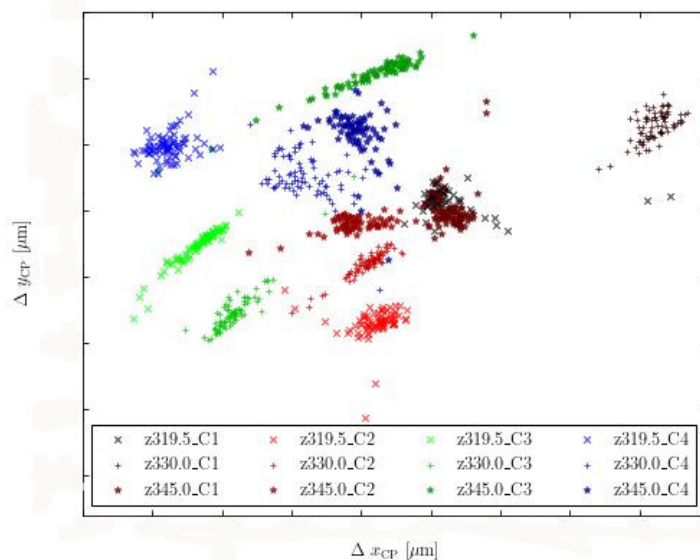


Abbildung 4.12: Schwerpunktverschiebung des Kühlluftkanals durch Turbulatoren

Für die Repräsentation der Kühlkanalgeometrie werden zunächst die Randpunkte der bereinigten CT-Messung für jeden Schnitt extrahiert. Der Randbereich jedes Kühlkanals wird anschliessend durch einen Basis-Spline approximiert. Hierbei ist es notwendig, eine ausreichende Anzahl an Punkten festzulegen, die äquidistant auf dem Basis-Spline verteilt werden. Zur Optimierung dieses Ansatzes wurde eine Konvergenzstudie durchgeführt, bei der die Punktzahl schrittweise reduziert wurde. Wie in Abbildung 4.13 dargestellt, führt bereits eine Repräsentation mit 500 Punkten zu einem Diskretisierungsfehler des Schwerpunktes von weniger als $100 \times 10^{-3} \mu\text{m}$. Die Parametrisierung des Schwerpunktes des Kühlluftkanals basiert

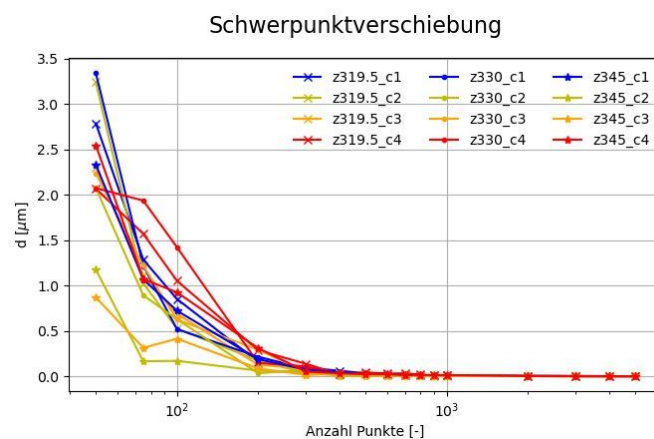


Abbildung 4.13: Schwerpunktverschiebung bei Anpassung der Anzahl diskreter Punkten

auf der Annahme, dass die Formvariationen eines Kühlkanals innerhalb einer CT-Messebene vernachlässigbar gering sind. Wie in Kapitel 4.3.2 erläutert, werden die B-Splines jedes Kühlkanals mittels des ICP-Verfahrens (engl. Iterative Closest Point) zueinander ausgerichtet, um ein Mittelwertmodell für jeden Kanalquerschnitt zu erstellen.

Der minimale Abstand zwischen der Mittelwertgeometrie und den B-Splines aus den CT-Messungen zeigt in den Bereichen der Geometrieradien erhöhte Geometrievationen. Ausserhalb der Radienbereiche liegt die mittlere Geometrieviation unterhalb von 30.0 μm und somit innerhalb der Messgenauigkeit des CLDA-Messsystems, das von HÖGNER [76] untersucht wurde.

In den Radienbereichen treten jedoch durch Strahlaufhärtung und Streueffekte Messfehler von bis zu 100 μm auf, wie ebenfalls von HÖGNER berichtet. Die geringe Geometrieviation der Kühlkanäle lässt sich jedoch auf den Giessprozess des Schaufelkerns zurückführen, wie in den Arbeiten von LEE et al. [101] und MA [104] beschrieben wird.

Abschliessend ist festzustellen, dass die Geometrieviation des Kühlluftkanals innerhalb der Messgenauigkeit des CT-Messsystems liegt und daher vernachlässigt werden kann. Somit ist eine Parametrisierung des Kühlluftkanals über den Flächenschwerpunkt der Geometrie möglich.

Für die Berechnung des Flächenschwerpunktes (x_{CP}, y_{CP}) der Kühlkanalgeometrie werden die folgenden Gleichungen verwendet:

$$x_{CP} = \frac{1}{6A} \sum_{i=0}^{n-1} (x_i + x_{i+1}) (x_i y_{i+1} - x_{i+1} y_i) \quad \text{und} \quad (4.48)$$

$$y_{CP} = \frac{1}{6A} \sum_{i=0}^{n-1} (y_i + y_{i+1}) (x_i y_{i+1} - x_{i+1} y_i) \quad \text{mit} \quad (4.49)$$

$$A = \frac{1}{2} \sum_{i=0}^{n-1} (x_i y_{i+1} - x_{i+1} y_i) \quad . \quad (4.50)$$

Wie von BOURKE [21] beschrieben, kann durch die Gleichungen 4.48 und 4.49 der Flächenschwerpunkt eines beliebigen, geschlossenen und schlaufenlosen Polygons mit n -Eckpunkten bestimmt werden. Die Fläche A der Kanalgeometrie wird mittels der Gauss'schen Trapezformel (Gleichung 4.50) berechnet, wie in KAHMEN [85] beschrieben.

Nach der Berechnung des Schwerpunktes jeder Kühlkanalgeometrie kann mit Hilfe der Gleichungen 4.51 und 4.52 die Schwerpunktverschiebung gegenüber dem Schwerpunkt der Referenzgeometrie ($x_{CP,ref}, y_{CP,ref}$) bestimmt werden:

$$\Delta x_{CP,i} = x_{CP,i} - x_{CP,ref} \quad (4.51)$$

$$\Delta y_{CP,i} = y_{CP,i} - y_{CP,ref} \quad (4.52)$$

Als Referenzmodell dient entweder das nominale CAD-Modell oder ein aus den CT-Messungen berechnetes Mittelwertmodell. Da die Geometrieabweichung der gefertigten Schaufeln vom Nominalmodell nicht vernachlässigt werden kann, wurde für die weiteren Untersuchungen die mittlere Kühlkanalgeometrie als Referenz verwendet.

Zunächst wurde die Korrelation der Schwerpunktverschiebungen der Kühlluftkanäle in den verschiedenen Schnittebenen analysiert. Es zeigte sich eine signifikante Korrelation zwischen den Verschiebungen der Kanäle, die darauf zurückzuführen ist, dass die Gusskerne während

des Giessprozesses durch Stifte miteinander verbunden sind. Diese Stifte bewirken eine über die radiale Höhe ausgeprägte positionsabhängige Kopplung zwischen den Kanälen.

Im Rahmen der Parametrisierung der Kühlkanäle wurden mehrere Ansätze untersucht. In Anlehnung an die Arbeit von HÖGNER [76] wurde neben der Translation der Kanäle auch deren Verkipfung berücksichtigt. Zur Beschreibung der Verkipfung der Kühlluftkanäle sowie zur Untersuchung deren Korrelation mit der Verschiebung wurde ein Polynom ersten Grades durch die Schwerpunkte der drei Schnittebenen gelegt.

Zusammenfassend wurden vier unterschiedliche Parametrisierungsansätze entwickelt, die sich in der Anzahl der benötigten Parameter für den Wiederaufbau der Kühlluftkanäle unterscheiden:

1. **Parametrisierung mit 24 Parametern:** Für jeden Kanal wurde in jeder Schnittebene ein Translationsparameter berücksichtigt.
2. **Parametrisierung mit 16 Parametern:** Für jeden Kanal wurde die Verkipfung durch Parameter für die Schiefe beschrieben, während die Translationsparameter relativ zur untersten Messebene definiert wurden.
3. **Parametrisierung mit 8 Parametern:** Die Translation wurde für jeden Kanal gemittelt und auf die unterste Messebene bezogen.
4. **Parametrisierung mit 2 Parametern:** Die Translation wurde über alle Schnittebenen und Kanäle gemittelt.

Die Auswirkungen der untersuchten Ansätze auf die Geometrieviation und die Rekonstruktionsgenauigkeit des Kühlkanals sind in Abbildung 4.14 dargestellt. Es zeigt sich, dass die Parametrisierung mit 24 Parametern die höchstmögliche Rekonstruktionsgenauigkeit erreicht, jedoch aufgrund der Anzahl der Parameter mit einem hohen Analyseaufwand verbunden ist. Die Berücksichtigung der Kühlkanalschiefe und damit insgesamt 16 Parameter führt zu einer vergleichbaren Rekonstruktionsgenauigkeit. Eine Reduzierung der Parameterzahl auf 8, durch Vernachlässigung der Schiefe und direkte Bestimmung der Kühlkanalverschiebungen in den Schnittebenen, führt jedoch nur zu geringen Einbussen in der Rekonstruktionsgenauigkeit, die weiterhin innerhalb der Messgenauigkeit der CT-Messungen liegt. Im Gegensatz dazu führt die Mittelung der Translationsparameter über alle Schnittebenen und Kühlkanäle (Parametrisierung mit 2 Parametern) zu einer deutlichen Verschlechterung der Rekonstruktionsgenauigkeit.

Aufgrund dieser Ergebnisse wurde für die weitere Modellierung eine Parametrisierung mit 8 Parametern gewählt, da diese einen optimalen Kompromiss zwischen Wiederaufbaugenauigkeit und Parameteranzahl darstellt. Zusammenfassend kann festgestellt werden, dass jede Parametrisierung auf Basis des Mittelwertmodells eine signifikant bessere Wiederaufbaugenauigkeit im Vergleich zur Rekonstruktion auf Basis der nominalen Kühlkanalgeometrie liefert.

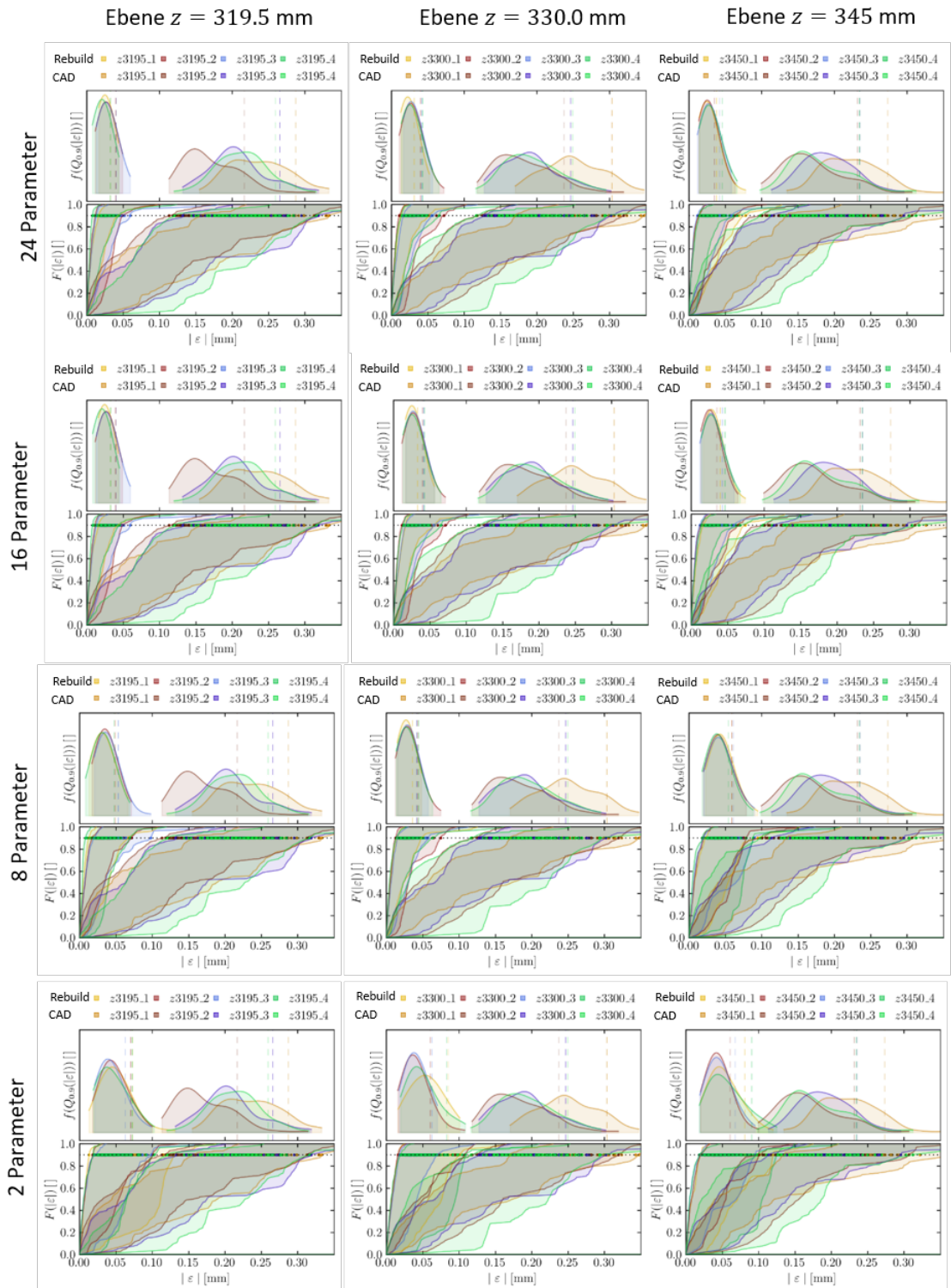


Abbildung 4.14: Variation der Reproduktionsgüte bei unterschiedlichen Parametrisierungsansätzen

4.3 Prozesskette (AP2.1)

Im folgenden Kapitel wird die im Rahmen dieses Projektes entwickelte Prozesskette für die BR000-Schaufel vorgestellt. Eine schematische Darstellung der einzelnen Prozesskomponenten ist in Abbildung 4.15 zu sehen.

Die Prozesskette basiert auf einem von Rolls-Royce zur Verfügung gestellten automatisierten Workflow, der im Laufe des Projektes kontinuierlich erweitert und optimiert wurde. Eine zentrale Komponente ist der "Gas Path Design System for Fans, Compressors und Turbines" Prozess (gpds-fc), der eine vollautomatische Berechnung von Spannungen und Lebensdauern ermöglicht. Dieser Prozess wurde in Zusammenarbeit mit Rolls-Royce speziell angepasst, um reale Turbinenschaufelgeometrien rekonstruieren und analysieren zu können.

Ein wesentlicher Teil der durchgeführten Arbeiten bestand in der parametrischen und statistischen Aufbereitung von GOM- und CT-Messdaten, die als Input für den gpds-fc-Prozess dienen. Neben der Optimierung spezifischer Teilprozesskomponenten lag ein weiterer Schwerpunkt auf der detaillierten Analyse und Bewertung der Ergebnisse des Prozesses.

Da der Fokus dieses Kapitels auf den zentralen Aspekten der Entwicklung und Analyse liegt, werden nicht alle Komponenten der Prozesskette im Detail behandelt.

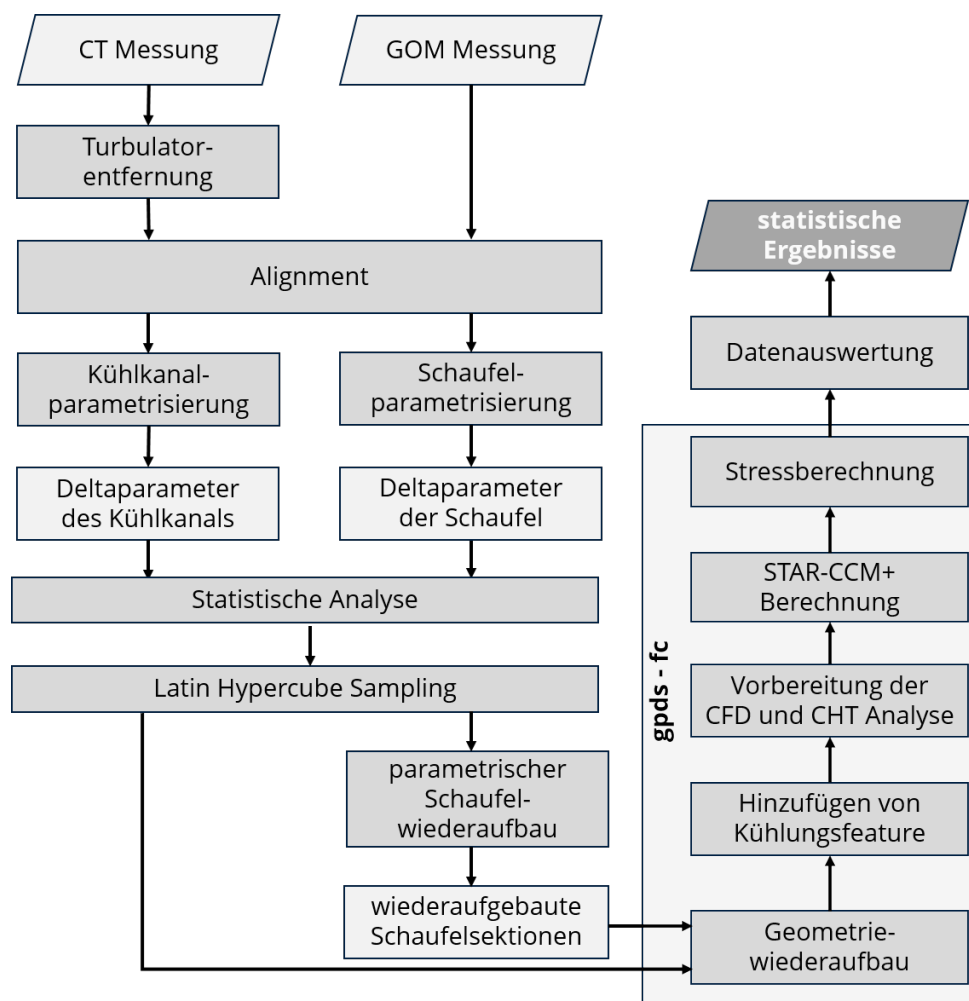


Abbildung 4.15: Übersicht der Prozesskette

4.3.1 Messdaten

Teil des standardisierten Produktionsprozesses bei Rolls-Royce ist die Produktüberwachung in der Fertigung, bevor die Komponenten weiter verarbeitet werden. Im Rahmen der Produktüberwachung werden unter anderem in einem Zwischenschritt u.a. GOM- und CT-Messungen an den Turbinenschaufeln durchgeführt, die zur Erkennung von Produktionsabweichungen dienen. Der von Rolls-Royce für dieses Projekt zur Verfügung gestellte Datensatz umfasst 80 Rotorschaufeln der ersten Hochdruckturbinenstufe eines Trent 1000 Triebwerks. Für jede Schaufel wurde eine GOM-Messung zur Erfassung der Aussengeometrie durchgeführt werden und als unausgerichtetes Polygonnetz zur Verfügung gestellt. Zum einen war zum Zeitpunkt der Messung der Schaufelfuss noch nicht fertig bearbeitet und auch die Endbearbeitung der Schaufeloberfläche stand noch aus, wie im Kapitel 4.3.3 erläutert wird. Jeder GOM-Messung können insgesamt 4 zweidimensionale CT Schnitte zugeordnet werden. Dies ist in Abbildung 4.16 dargestellt. Die Messung erfolgte in Schnittebenen, die durch technische Zeichnungen für die punktuelle Waddickenmessung definiert wurden. Wie zu einem späteren Zeitpunkt gezeigt wird, können hiervon allerdings nur die unteren 3 Schnittebenen verwendet werden, da sich die oberste Schnittebene bereits im Schaufelwinglet befindet. Die Grauwertbilder wurden von Rolls-Royce bereits in planare Polygonnetze mit einer Dicke von $100\mu\text{m}$ umgerechnet.



Abbildung 4.16: GOM-Messungen mit CT Schnittebenen

4.3.2 Turbulatorentfernung

Wie in Abbildung 4.12 zu erkennen ist, haben die Turbulatoren innerhalb eines Kühlkanals einen Einfluss auf den zu berechnenden Flächenschwerpunkt. Daher ist es für die weitere Auswertung des Kanalschwerpunkts notwendig, die Turbulatoren innerhalb der CT-Messungen zu detektieren und zu entfernen. Dies ist unter anderem mit gängigen Design-Tools wie PolyWorks möglich. Jedoch ist dies mit einem erheblichen Zeitaufwand und händischer Bearbeitung

verbunden. Zudem kann eine robuste und reproduzierbare Bereinigung der CT-Messungen auf diese Weise nicht gewährleistet werden.

In diesem Projekt wurde daher ein automatisierter Prozess in Python entwickelt, der die Erkennung und Entfernung von Turbulatoren ermöglicht. Eine wesentliche Herausforderung bei der automatischen Bereinigung ist dabei, dass durch die radiale Verschiebung der Rippen im Kühlluftkanal die Rippen durch die CT-Messungen nicht an der gleichen Stelle geschnitten werden. Dadurch kann zum einen die Anzahl der Rippen für den jeweils betrachteten CT-Schnitt variieren und zum anderen werden Rippen nur teilweise geschnitten. Der Prozess ist in Abbildung 4.17 schematisch dargestellt.

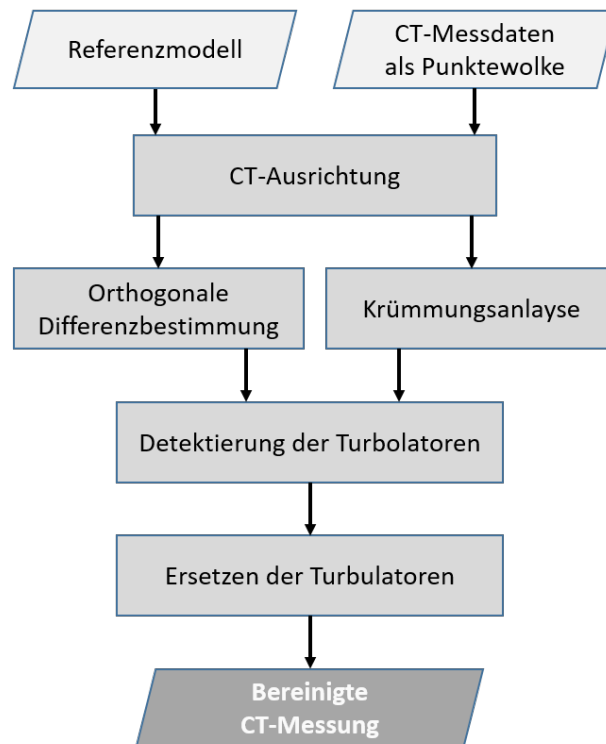


Abbildung 4.17: Prozesskette zur Turbulatorbereinigung

Wie bereits beschrieben liegen die im Prozess verwendeten Messdaten in Form von stl-Netzen vor, die von Rolls-Royce innerhalb der standardisierten Produktionskette erzeugt werden. Mit Hilfe eines PolyWorks-Skripts können daraus sowohl die Schaufelaussengeometrie als auch die Kühlkanalgeometrie in den vorgegebenen CT-Schnittebenen durch Kantendetektion erfasst und als separate Kurven abgelegt werden. PolyWorks bietet mit diversen Bearbeitungstools die Möglichkeit die Kurven händisch von den Turbulatoren zu bereinigen. Es wurde jedoch keine Möglichkeit gefunden, diesen Vorgang zu automatisieren. Da sich der automatische Reinigungsprozess an einer bereits bereinigten Schaufelgeometrie als Referenzwert orientiert, ist es notwendig, die manuelle Bereinigung einmalig durchzuführen. Wie sich bei den weiteren Untersuchungen herausstellte, kann die CAD-Geometrie aufgrund zu grosser Abweichungen zu den CT-Messdaten nicht als Referenzmodell verwendet werden. Daher wurde eine CT-Messung gewählt, die in etwa den Mittelwert aller CT-Messungen widerspiegelt. Die so ermittelten Kurven können dann mit einem Skript als Punktwolke mit einem Punktabstand von $20\mu\text{m}$ exportiert werden.

Sowohl das Referenzmodell als auch die unbereinigten Messdaten können nun zur weiteren Verarbeitung in Python importiert werden. Im ersten Verarbeitungsschritt werden die gemessenen CT-Daten anhand des Referenzmodells ausgerichtet. Die Ausrichtung zweier Punktwolken ist ein in der Literatur häufig untersuchtes Problem, das in der Regel durch Variationen des Iterative Closest Point (ICP) Verfahrens gelöst wird. Ein Beispiel hierfür findet sich in BROENINK et al. [23]. Der Algorithmus geht davon aus, dass die Punktwolken bereits eine gewisse Vororientierung aufweisen. Daher wird jeder Kühlkanal aus der CT-Messung extrahiert und anhand seines Schwerpunktes gemäss den Gleichungen 4.48, 4.49 auf die Referenzgeometrie ausgerichtet. Anschliessend wird ausgehend von der Referenzgeometrie ein Punkt auf der CT-Messung mit dem kleinsten Abstand gesucht, der nicht weiter als $300\mu\text{m}$ entfernt ist. Für die Punktepaare wird iterativ eine Transformationsmatrix gesucht, die den Abstand zwischen den Punkten minimiert. Da hier die Randbereiche der Turbulatoren einen störenden Einfluss auf die Ausrichtgenauigkeit haben, ist es notwendig, auch den maximalen Suchabstand iterativ anzupassen.

Für die Detektierung der Turbulatoren werden zwei Ansätze verfolgt. Im Ansatz 1 wird der orthogonale Abstand zwischen der Referenzgeometrie und der ausgerichteten CT-Messung bestimmt. Dies ist exemplarisch für den Kühlluftkanal 1 in der untersten Schnittebene in der Abbildung 4.18a dargestellt. Die sich hier ergebende Differenz zum Referenzmodell ist in Abbildung 4.18b dargestellt, die für mittig geschnittene Turbulatoren wie T1 einen signifikanten Abstand d zum Referenzmodell aufweist. Dazu hat sich die Suche nach Maxima über einem vorgegebenen Schwellenwert bewährt. Wird die Rippe jedoch wie der Turbulator T4 im Randbereich abgeschnitten, können diese nicht mehr zuverlässig über einen fest vorgegebenen Schwellenwert aus den Messungen herausgefiltert werden. Dazu wird der Schwellenwert so lange adaptiv abgesenkt, bis die detektierten Maxima nicht mehr von der Geometrieabweichung unterschieden werden können. Daher können die Turbulatoren T2 und T3 mit dieser Methode nicht detektiert werden. Die Randbereiche eines Turbulators können mit diesem Ansatz jedoch gut detektiert werden. Im Ansatz 2 zur Detektion von Turbulatoren wird eine Krümmungsanalyse der CT-Messungen durchgeführt. Die Krümmung wird sowohl für das Referenzmodell als auch für die CT Messdaten nach folgender Gleichung 4.53 ermittelt

$$\kappa(t) = \frac{\dot{x}(t)\ddot{y}(t) - \ddot{x}(t)\dot{y}(t)}{\dot{x}(t)^2 + \dot{y}(t)^2}. \quad (4.53)$$

Wie in Abbildung 4.18a zu sehen ist, beginnt die Laufvariable t in beiden Geometrien an der gleichen Position. Der sich daraus berechnete Krümmungsverlauf ist in Abbildung 4.18c dargestellt. Als notwendige Bedingung für einen Turbulator wird das Krümmungsminimum der CT-Messung detektiert, das bei der maximalen Höhe des Turbulators auftritt. Als hinreichende Bedingung darf dieses Minimum im Referenzmodell nicht auftreten, was mit einer hohen Krümmungsdifferenz zwischen Referenzmodell und CT-Messung einhergeht. Im Gegensatz zu Ansatz 1 können mit dieser Methode auch kleine Turbulatoren in der CT-Messung erkannt werden. Allerdings können die Ränder eines Turbulators mit diesem Ansatz nur näherungsweise anhand der auftretenden Krümmungsmaxima abgeschätzt werden. Bei der Analyse der Randbereiche eines Turbulators kommt es zu einer nicht eindeutig trennbaren Überlagerung der Krümmung aus Turbulator und Kanalgeometrie. Zusätzlich wird die Detektion dadurch erschwert, dass die Krümmung aufgrund von Messunsicherheiten relativ grosse Schwankungen

aufweist. Daher wird für die Detektierung der Turbulatoren eine Kombination beider Ansätze

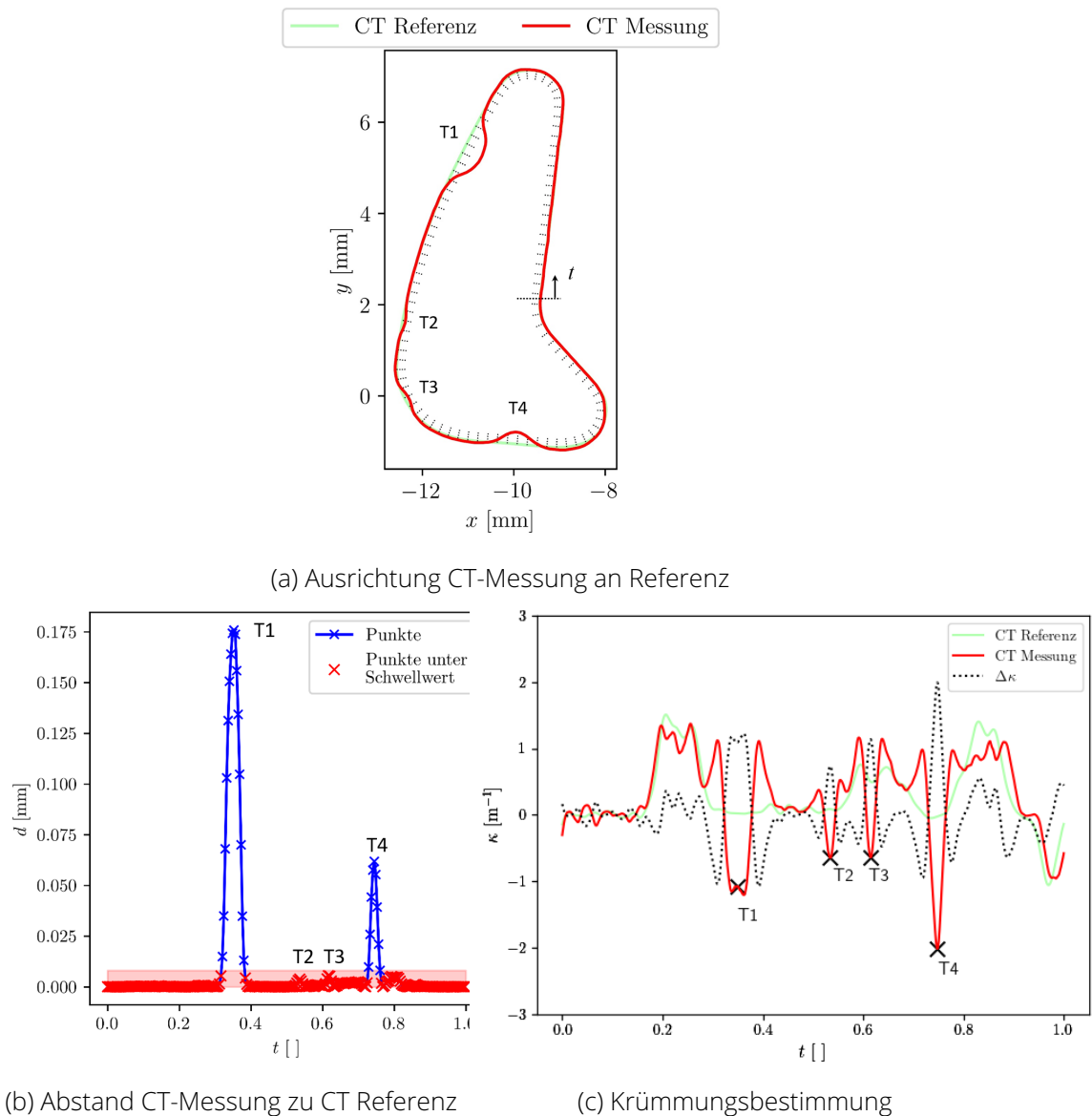


Abbildung 4.18: Methoden der Turbulatordetektierung

verwendet. Dabei werden alle Turbulatoren, die durch den Algorithmus unter Berücksichtigung der Kurvenkrümmung erkannt wurden, aufgelistet. Wenn ein Turbulator an der gleichen Stelle durch Ansatz 1 erkannt wurde, wird der Randbereich des Turbulators entsprechend verbessert, wie in Abbildung 4.19 zu sehen ist. Bei der Bereinigung der Turbulatoren wird wiederum davon ausgegangen, dass sich die CT-Messung nur geringfügig von der Referenzgeometrie unterscheidet. So kann anhand der zuvor detektierten Turbulatoren das entsprechende Segment aus dem bereits bereinigten Referenzmodell an die Stelle des Turbulators gesetzt werden. Abschliessend erfolgte eine Rücktransformation der bereinigten Geometrie in ihre Ausgangslage entsprechend der zu Beginn durchgeführten Ausrichtung.

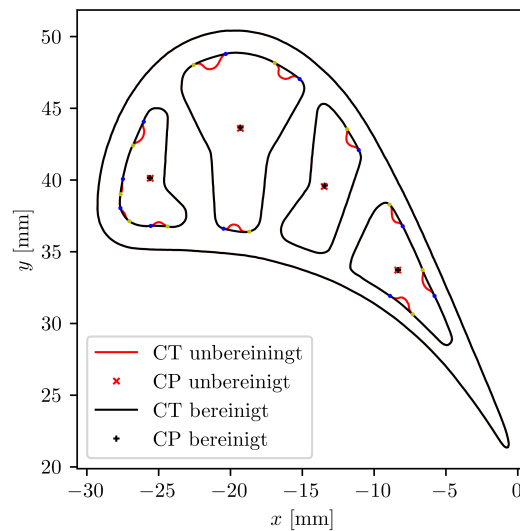


Abbildung 4.19: Bereinigte Kühlkanäle

4.3.3 Messdatenausrichtung

Die im Kapitel 4.2 vorgestellte Parametrisierung der Schaufelaussen- und -innengeometrie basiert auf der Annahme, dass sich die extrahierten Schnitte der Geometrie gegenüber einer gewählten Referenz nur geringfügig ändern. Wie bereits im vorherigen Kapitel erläutert, ist hierfür die Bildung einer Mittelgeometrie erforderlich. Diese wird aus den ausgerichteten Flächen und den CT-Messungen der Turbinenschaufel erzeugt. Dazu wird die Turbinenschaufel im jeweiligen Messsystem an bestimmten Punkten auf der Schaufeloberfläche ausgerichtet, fixiert und anschliessend vermessen. In der Regel werden die Messdaten jedoch in einem nicht ausgerichteten Zustand gespeichert und müssen daher in einem Vorverarbeitungsschritt zueinander ausgerichtet werden. Wie bereits in HÖGNER [74] gezeigt wurde, ist die verwendete Methode zur Schaufelausrichtung relevant für die spätere Qualität des Wiederaufbaus der Turbinenschaufel.

Heissgaspfadalignment

Die Ausrichtung von Turbinenschaufeln erfolgt in der Regel über den Schaufelfuss der Nennweite. Diese Methode kann im vorliegenden Fall jedoch nicht angewendet werden, da der verwendete Datensatz dies nicht zulässt. Es handelt sich um Gusskörper der Turbinenschaufeln, die sich zum Zeitpunkt der Messung noch in einem unbearbeiteten Zustand befinden.

Da die Schaufeln noch nicht den Bearbeitungsprozess des Schaufelfusses durchlaufen haben, sind die Kontaktflächen des Tannenbaumfusses sowie die axiale Anschlagfläche noch nicht durch Fräsbearbeitung hergestellt, wie in Abbildung 4.16 zu sehen ist. Daher ist eine Ausrichtung der GOM-Messung auf den Schaufelfuss des CAD-Modells nach der von HEINZE [68] beschriebenen Vorgehensweise nicht möglich.

Um eine korrekte Ausrichtung der Turbinenschaufel zu gewährleisten, musste eine alternative Methode angewendet werden. Dabei wurde die Tatsache ausgenutzt, dass durch das Fehlen der

Kühlluftbohrungen die gesamte Schaufeloberfläche über die Schaufelgeometrie zur Ausrichtung herangezogen werden konnte.

Zunächst erfolgt eine grobe Vorausrichtung des gemessenen Polygonnetzes auf die CAD-Geometrie. Dieser Schritt ist notwendig, da die Polygonnetze ohne Bezug zu einem räumlichen Koordinatensystem vorliegen. Anschliessend erfolgt eine genauere Ausrichtung der Netze unter Berücksichtigung der gesamten vom Heissgas umströmten Schaufelfläche. Dabei werden auch die Konturen der Naben- und Wingletflächen berücksichtigt, wie in Abbildung 4.20a dargestellt.

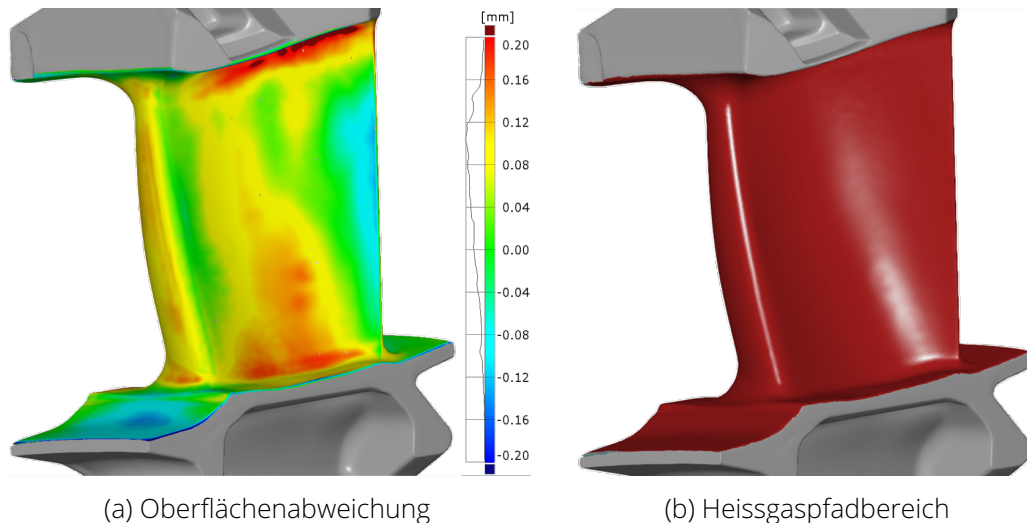


Abbildung 4.20: Messdatenausrichtung anhand des CAD-Modells

Wie in Abbildung 4.20b dargestellt, treten im Vergleich zur CAD-Geometrie signifikante Fertigungsabweichungen zwischen Schaufelblatt und Naben- bzw. Wingletkontur auf. Die Arbeiten von HÖGNER [77] haben bereits gezeigt, dass die Fertigungsabweichungen im Bereich der Endwand bis zu $150\mu\text{m}$ betragen können.

Da auch im Schaufelblattbereich deutliche Abweichungen zur CAD-Geometrie festgestellt wurden, wird zur besseren parametrischen Beschreibung eine erneute Ausrichtung des Polygonnetzes auf Basis des mit GOM gemittelten Netzes durchgeführt.

Weitere Untersuchungen zeigen, dass bei der erneuten Ausrichtung durch Vernachlässigung der Naben- und Wingletflächen im Heissgaspfadbereich eine Verbesserung des Ergebnisses im Schaufelblattbereich erreicht werden kann, wie in Abbildung 4.21 dargestellt.

Ein wesentlicher Nachteil der Parametrisierung auf Basis der über den Heissgaspfad gemittelten Daten ist, dass dabei auch alle Lageparameter der Schaufel durch die Alignment-Prozedur eine Mittelung erfahre. Dies führt zu einer Ungenauigkeit bei der Bestimmung der genauen Lageparameter im Einbauzustand. Bei einer Parametrisierung mittels eines Schaufelfussalignments wäre dieser Effekt nicht gegeben, da der Schaufelfuss eine feste Referenz bietet und somit die Lageparameter unabhängig von der Schaufelgeometrie erhalten bleiben.

Aufgrund der starken geometrischen Verwindung der Turbinenschaufel sowie der unterschiedlichen Schaufelprofilformen in radialer Höhe ist die Berücksichtigung der Naben- und Wingletkonturen für die Ausrichtung der radialen Lage der Messdaten jedoch nicht zwingend erforderlich, um eine radiale Verschiebung der Polygonnetze zu vermeiden.

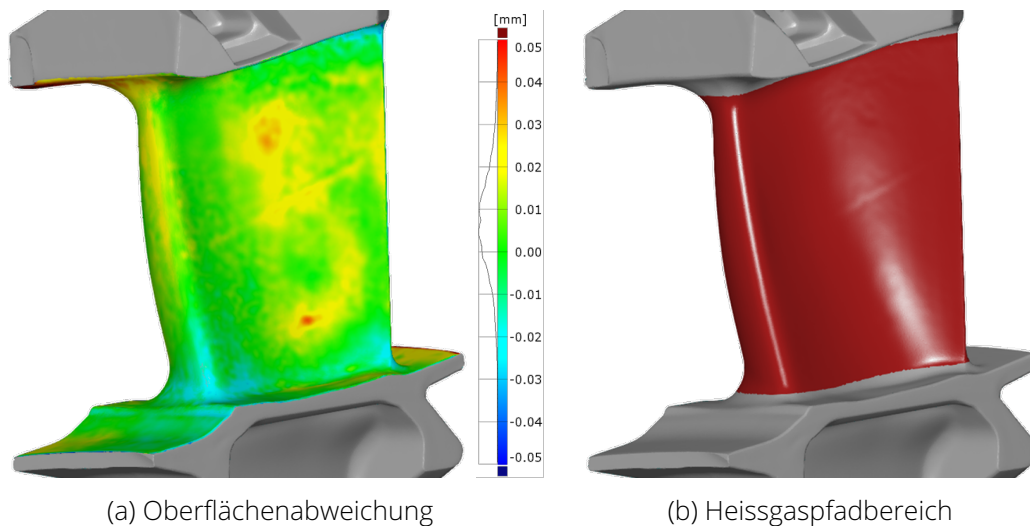


Abbildung 4.21: Messdatenausrichtung anhand des Mittelwertnetzes

CT Alignment

Um die inneren Strukturen der Turbinenschaufeln zu erfassen und die Wandstärken zu validieren, wird bei Rolls-Royce eine schnittbasierte CT-Messung durchgeführt. Diese erfolgt in den Ebenen $z = [320 \text{ mm}, 330 \text{ mm}, 345 \text{ mm}, 357 \text{ mm}]$. Nach der Messung liegen alle Schnitte polygonalisiert in der xy-Ebene vor. Entsprechend der technischen Spezifikation werden die Schnitte zunächst translatorisch in die nominale Schnittebene verschoben. Da die CT-Messung in einer einzigen Messmaschine erfolgt, ist ein Umspannen der Turbinenschaufel zwischen den Schnitten nicht erforderlich. Aufgrund der hohen Genauigkeit der Verfahrenseinrichtung des CT-Scanners bleibt der nominale Abstand zwischen den Messebenen erhalten und auch der Winkelfehler zwischen den Messebenen kann als vernachlässigbar klein angesehen werden.

Die oberste Schnittebene $z = 357 \text{ mm}$ liegt teilweise im Bereich des Winglets, wie in Abbildung 4.16 dargestellt. In diesem Bereich werden im Vergleich zu den anderen Messebenen deutlich unterschiedliche Materialstärken durchdrungen, was sich negativ auf die Qualität der CT-Messung auswirkt. Daher wird die oberste Messebene, wie bereits im Kapitel 4.3.1 erläutert, für die Ausrichtung nicht berücksichtigt. Die Ausrichtung erfolgt daher anhand der verbleibenden drei unteren Schnittebenen. Die Ausrichtung der Schnittebenen erfolgt durch eine Starrkörperverschiebung in der Software GOM Inspect, wodurch die relative Lage und Ausrichtung der CT-Schnitte zueinander erhalten bleibt.

Für die Ausrichtung wurden zwei mögliche Ansätze untersucht. Zunächst wurde das Alignment mit drei translatorischen Freiheitsgraden und einem rotatorischen Freiheitsgrad um die z-Achse getestet, unter der Annahme, dass die Ausrichtung der Schaufel zwischen GOM- und CT-Messung erhalten bleibt. Es wurde jedoch festgestellt, dass die notwendige physikalische Neuausrichtung der Turbinenschaufel beim Wechsel der Messmaschine einen nicht vernachlässigbaren und nicht deterministisch erfassbaren Ausrichtungsfehler verursacht. Daher wurde ein Ausrichtungsverfahren mit drei translatorischen und drei rotatorischen Freiheitsgraden für die Starrkörperverschiebung gewählt. Ein beispielhaftes Ausrichtungsergebnis für die CT-Schnitte ist in Abbildung 4.22 dargestellt.

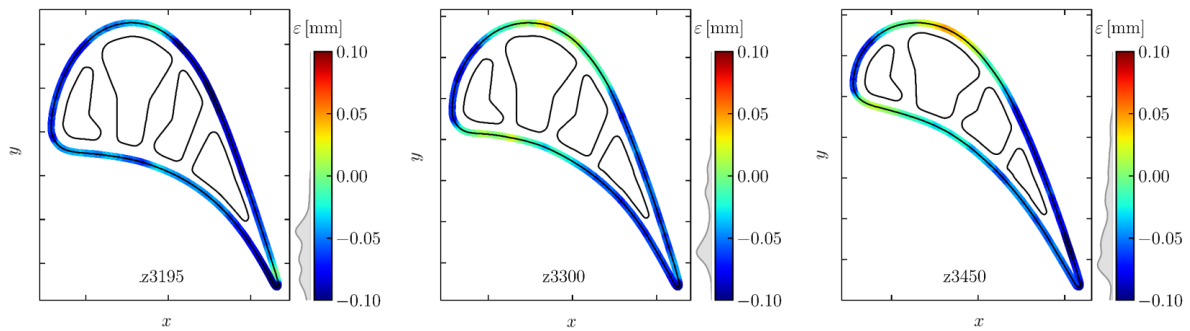


Abbildung 4.22: Ausrichtungsabweichung der CT-Messungen

Wie von HÖGNER [74] festgestellt, sind die Abweichungen zwischen GOM- und CT-Messung im Bereich der Hinterkante am grössten. Es wurde versucht, das Ausrichtungsergebnis durch Vernachlässigung der Bereiche mit hohen Messabweichungen zu verbessern. Dies führte jedoch zu einer Verringerung der für die Ausrichtung zur Verfügung stehenden Fläche und zu einer Verschlechterung des Gesamtergebnisses. Abbildung 4.23 zeigt die Abweichung ε zwischen der gemessenen GOM-Oberfläche und der CT-Messung über den gesamten Datensatz.

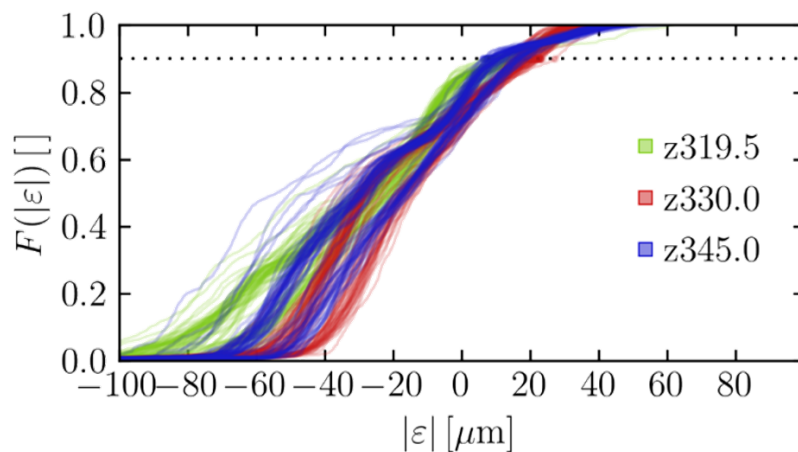


Abbildung 4.23: Wahrscheinlichkeitsdichtefunktion zur Oberflächenabweichung zwischen CT- und GOM-Messung nach dem Alignment

Bei der CT-Messung werden unterschiedliche Materialstärken aus unterschiedlichen Messrichtungen durchstrahlt. Wie in Abbildung 4.22 zu sehen ist, führt dies zu einer Überschätzung der Schaufelprofilstärke im mittleren Schaufelblattbereich um bis zu $20\mu\text{m}$. Im Bereich der Vorder- und Hinterkante hingegen wird die Profilstärke um bis zu $100\mu\text{m}$ unterschätzt.

4.3.4 Statistische Beschreibung der Eingangsgrößen

Aufbauend auf den im Kapitel 4.1.2 beschriebenen probabilistischen Grundlagen erfolgt im folgenden Kapitel die statistische Beschreibung der Messdaten. Dazu wird zunächst auf die Bestimmung der Verteilungsfunktionen der Eingangsparameter eingegangen. Anschliessend erfolgt eine Analyse der Korrelationsstrukturen und es wird kurz auf die probabilistische Modellierung und das Sampling der Daten eingegangen.

Verteilungsfunktion

Im Rahmen der probabilistischen Analyse ist es erforderlich, aus den diskreten Messdaten die kontinuierliche Verteilungs- oder Dichtefunktion abzuleiten. Da die Analyse mit ProSi durchgeführt wird, können hierfür die folgenden Verteilungsfunktionen verwendet werden. Nachzulesen im ProSi-Handbuch von DIERMEIER, VOIGT et al. [42]:

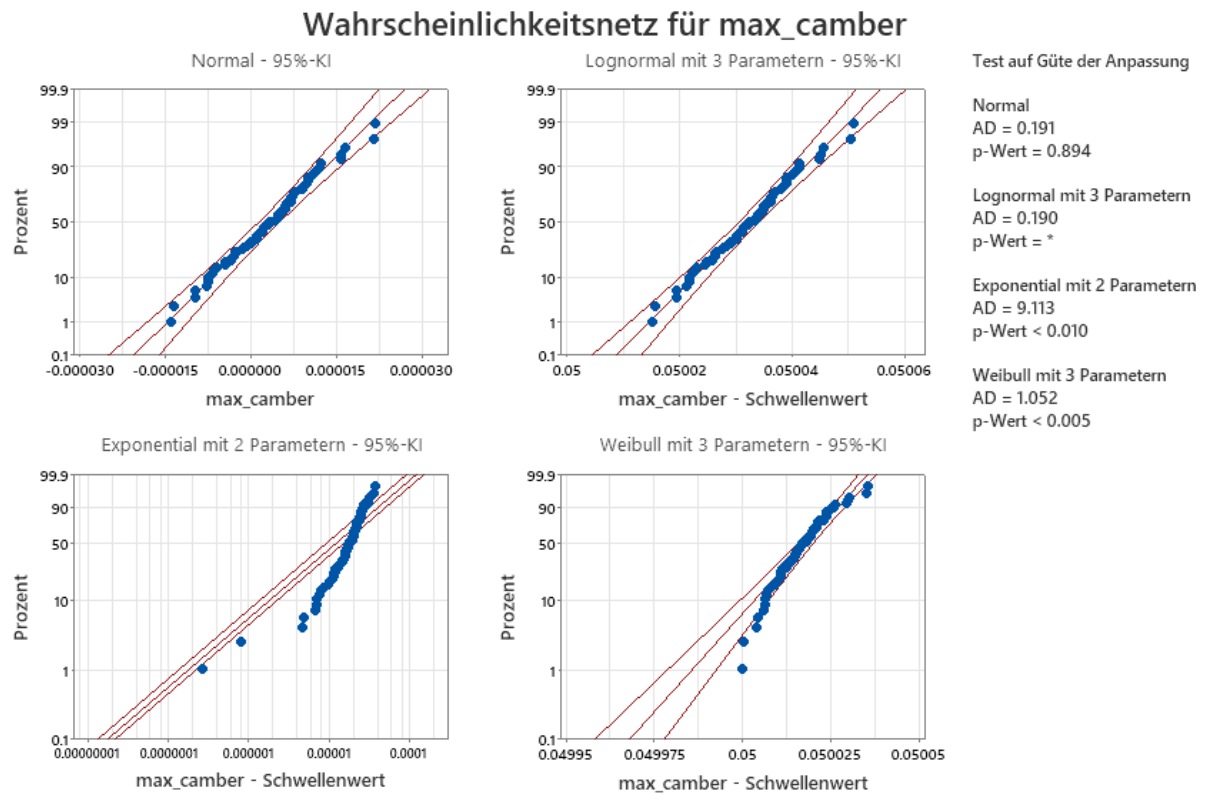
- Normalverteilung,
- Lognormalverteilung,
- Gleichverteilung,
- Dreiecksverteilung,
- Exponentialverteilung und
- Weibullverteilung.

Wie im Folgenden beispielhaft gezeigt wird, kann man sich bei der Beschreibung der zugrunde liegenden Messdaten auf die Normalverteilung beschränken. Der Wertebereich sollte jedoch sinnvoll eingegrenzt werden.

Zur Bestimmung und Überprüfung der Verteilungsfunktionen wurde das Programm Minitab [113] verwendet, welches in Abbildung 4.24 exemplarisch für die maximale Krümmung des Profils dargestellt ist. Dazu werden in einem ersten Schritt die diskreten Messdaten mit Hilfe der Maximum-Likelihood-Methode durch die entsprechenden Verteilungsfunktionen approximiert. Anschliessend wird ein Goodness-of-Fit-Test nach Anderson & Darling [114] durchgeführt. Die Nullhypothese des Tests H_0 besagt, dass die auf den Messdaten basierende diskrete Häufigkeitsverteilung mit der theoretischen Wahrscheinlichkeitsverteilung übereinstimmt. In Minitab wird diese Nullhypothese durch den p-Wert ausgedrückt. Für jede Verteilung wird die modifizierte Anderson-Darling-Statistik AD berechnet, um die Güte der Anpassung zu testen. Wenn die Teststatistik AD den kritischen Wert unterschreitet, der dem gewählten Signifikanzniveau entspricht, wird die Nullhypothese abgelehnt. Dies bedeutet, dass eine signifikante Abweichung von der theoretischen Verteilung vorliegt. Ist AD dagegen grösser als der kritische Wert, kann keine signifikante Abweichung nachgewiesen werden. Für die maximale Krümmung zeigt sich, dass sowohl die Normalverteilung als auch die Lognormalverteilung mit einem AD -Wert von 0,19 nach Anderson-Darling eine ähnlich gute Anpassung aufweisen. Diese Verteilungen sind der Exponential- und der Weibullverteilung vorzuziehen. Da in Minitab sehr kleine p-Werte mit einem Stern gekennzeichnet werden, ist die Normalverteilung aufgrund ihres höheren p-Wertes der Lognormalverteilung vorzuziehen.

Für alle geometrischen Parameter sind die entsprechenden Verteilungsfunktionen entsprechend auszuwerten.

Darüber hinaus kann die Güte der Anpassung der diskreten Häufigkeitsverteilung an die theoretische Wahrscheinlichkeitsverteilung durch einen Q-Q-Plot dargestellt werden, wie in Abbildung 4.25 anhand der Verschiebung des Kühlluftkanals 1 gezeigt. In dieser Darstellung werden die

Abbildung 4.24: Verteilungsidentifikation für c_{max}

Quantile der diskreten Häufigkeitsverteilung mit den Quantilen der theoretischen Wahrscheinlichkeitsverteilung verglichen. In beiden Fällen können in den Randbereichen Messpunkte identifiziert werden, die eine Abweichung vom theoretischen Quantil aufweisen. Der Anpassungstest liefert jedoch auch hier keine ausreichende Begründung für die Ablehnung der Nullhypothese bei einem Signifikanzniveau von 5.00 %, was ebenfalls eine Normalverteilung für die Sehnenlänge unterstützt.

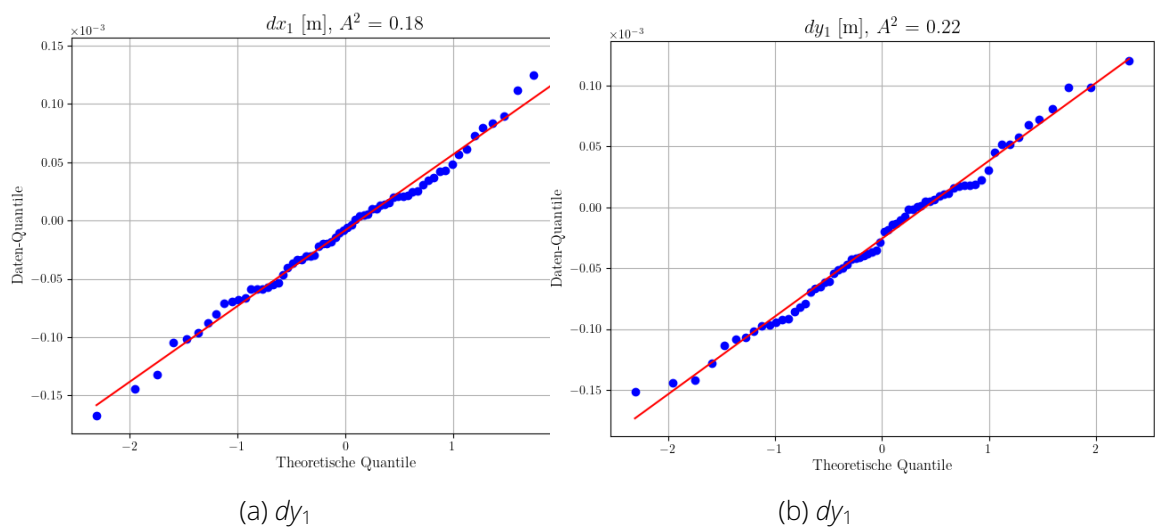


Abbildung 4.25: Q-Q-Plots für die Verschiebung des Kerns 1

Korrelationsstruktur

Im nächsten Schritt wird die Korrelationsstruktur der Eingangsparameter untersucht. Ziel dieser Analyse ist es, Abhängigkeiten zwischen den Parametern zu identifizieren, die durch Korrelationskoeffizienten quantifiziert werden. Dabei werden sowohl der Pearson-Koeffizient als auch der Spearman-Rangkorrelationskoeffizient verwendet. Diese Analyse trägt zu einem besseren Verständnis der den Daten zugrunde liegenden Beziehungen bei und ermöglicht es, realistische Annahmen in das probabilistische Modell zu integrieren.

Für die Erstellung der Korrelationsmatrix in Abbildung 4.26 wurde der Rangkorrelationskoeffizient nach Spearman verwendet. Dieser Koeffizient erlaubt, wie im Grundlagenkapitel beschrieben, sowohl lineare als auch monotone Zusammenhänge zu erfassen.

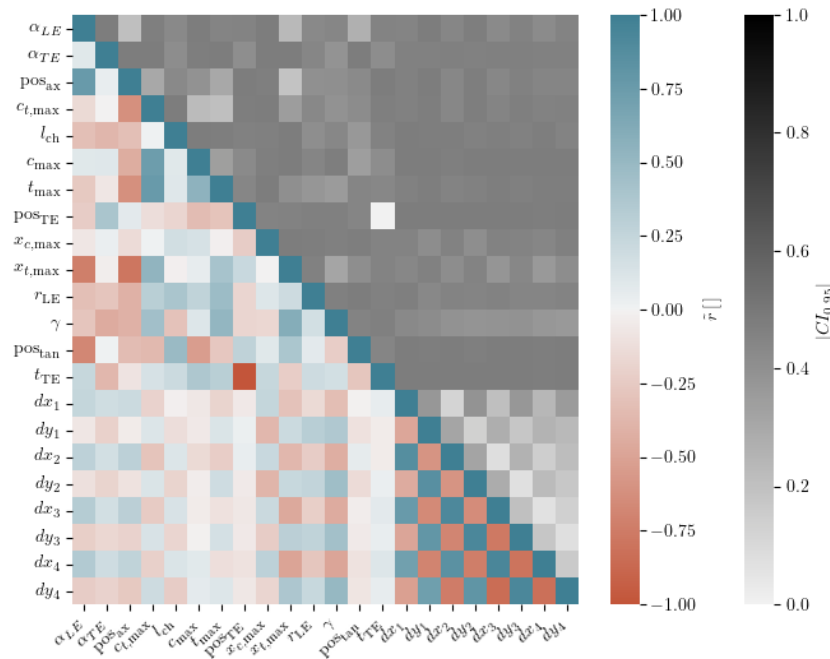


Abbildung 4.26: Korrelationsmatrix und zugehöriges Konfidenzintervall des Datensatzes

Die Korrelationsmatrix zeigt, dass die Position des TE, pos_{TE} , stark mit der Dicke des TE, t_{TE} , korreliert. Andere Parameter, die zur Beschreibung der Schaufelaussengeometrie verwendet werden, weisen dagegen nur geringe Korrelationen auf. Die Parameter zur Beschreibung der Schaufelaussengeometrie sind nahezu unkorreliert mit der Kühlkanalverschiebung, der Korrelationskoeffizient liegt unter 0.22. Dies deutet darauf hin, dass eine Verschiebung der Kühlluftkanäle innerhalb der Schaufelstruktur plausibel ist. Ebenso weist die Position der maximalen Schaufeldicke $x_{t,\text{max}}$ und der maximalen Schaufelwölbung $x_{c,\text{max}}$ nur eine leicht erhöhte Korrelation von etwa 0.38 zur Kühlluftkanalverschiebung auf. Unter Berücksichtigung des durch die Fisher-Z-Transformation [50] berechneten Konfidenzintervalls $|CI_{0.95}|$ ist insgesamt eine geringe Korrelation zwischen den genannten Parametern festzustellen.

Im Gegensatz dazu zeigt die Kühlkanalverschiebung eine starke Korrelation zwischen den Kanälen mit Werten im Bereich von 0.5 bis 0.85. Dieser Zusammenhang lässt sich durch die Art der Kühlkanalherstellung erklären, die im Wesentlichen durch die Verschiebung der Gesamtheit aller Kühlkanäle innerhalb der Turbinenschaufel bestimmt wird, wie bereits im

Kapitel 4.2.2 erläutert wurde. Aufgrund der zum Teil starken Korrelationen ist es notwendig, diese Abhängigkeiten bei der Erstellung der Realisierungen zu berücksichtigen.

Probabilistische Modellierung und Sampling

Abschliessend wird auf die probabilistische Modellierung der Messdaten eingegangen, die als Grundlage für das Sampling dient. Hierbei werden Methoden wie Latin-Hypercube-Sampling eingesetzt, um repräsentative Realisierungen zu generieren. Diese Realisierungen ermöglichen es, die Unsicherheiten und Variationen der Eingangsparameter in den weiteren Analysen der CHT-Turbinenschaufel umfassend zu berücksichtigen.

Im Rahmen dieses Berichtes wird die Monte-Carlo-Simulation wie in KROESE et al. [94] beschreiben zur Berücksichtigung von Geometrievariationen eingesetzt, um eine Sensitivitätsanalyse durchzuführen. Dazu ist es erforderlich, die geometrischen Parameter so zu modellieren, dass die Variation korrekt dargestellt wird. Dies wird durch die Verwendung approximierter Verteilungsfunktionen erreicht, wobei die Korrelationsstruktur der Parameter in die Stichprobenziehung einbezogen wird. Für diese Aufgabe wird die Methode der eLHS nach SCHMIDT et al. [16] unter Verwendung der Software ProSi eingesetzt. Im vorliegenden Fall werden insgesamt 5 eLHS-Sequenzen mit je 30 Proben, also insgesamt 150 Realisierungen erstellt. Abbildung 4.27 zeigt die Abweichung zwischen den Zielkorrelationen der Messdaten und der Korrelation der erzeugten Realisierungen. Die maximale Abweichung Δr beträgt 0.08. Diese Abweichung kann als vernachlässigbar angesehen werden.

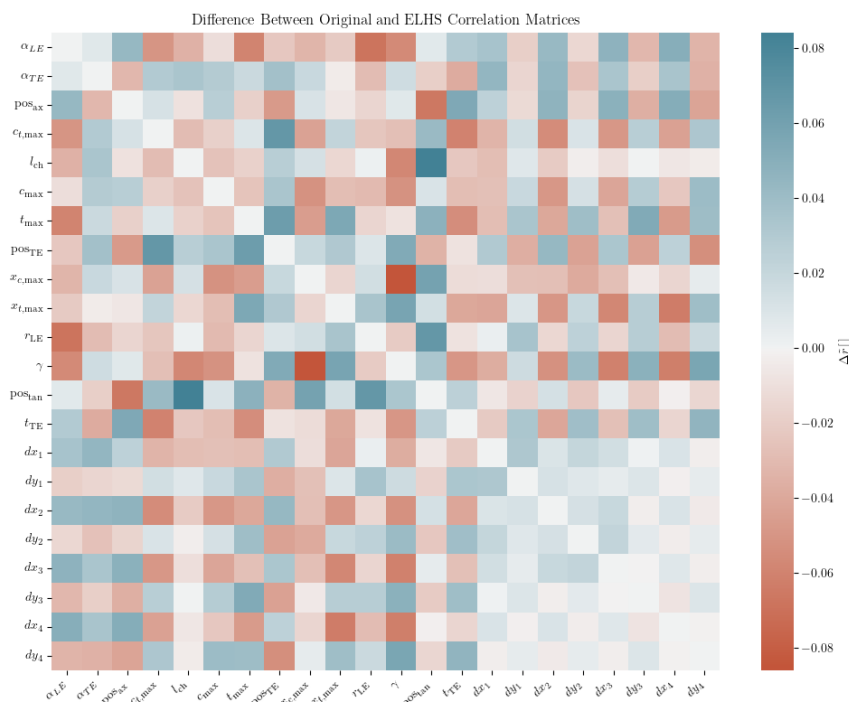


Abbildung 4.27: Abweichung der Korrelationsstruktur zwischen Messdaten und Realisierungen

4.3.5 gpds-fc Prozesskette und Dateneinbindung

Im folgenden Abschnitt wird die bereitgestellte gpds-fc Prozesskette beschrieben, die seitens Rolls-Royce den vollautomatischen Wiederaufbau der Geometrie, inklusive aller Kühlungsfeatures, sowie die Vorbereitung der CFD-, CHT- und FEM-Analyse und deren Berechnung ermöglicht. Leider war es in diesem Projekt aufgrund technischer Limitierungen nicht möglich, die gpds-fc Prozesskette am Institut vollumfänglich zu nutzen. Daher wurden mit Unterstützung von Rolls-Royce Änderungen an der gpds-fc Prozesskette vorgenommen, die ebenfalls in diesem Kapitel aufgezeigt werden.

Geometriewiederaufbau und Simulationsvorbereitung

Der Wiederaufbau der Schaufelaussengeometrie und der Schaufelinnengeometrie erfolgt zunächst getrennt und wird anschliessend innerhalb des Geometriewiederaufbaus zusammengeführt. Für den Wiederaufbau der Schaufelaussen- und Schaufelinnengeometrie werden unterschiedliche Ansätze verfolgt, die beide auf dem parametrischen Wiederaufbau basieren, welcher sich auf eine Mittelwertgeometrie stützt.

Der sektionsbasierte Wiederaufbau der Schaufelblattaussengeometrie erfolgt zunächst ausserhalb der gpds-fc Prozesskette. Grund hierfür ist, dass die am Institut für Turbomaschinen verwendete Parametrisierung nicht mit den üblicherweise von Rolls-Royce verwendeten Parametern übereinstimmt. Da eine Überführung der Parameter nur bedingt möglich ist, stellt die gpds-fc Prozesskette eine Methode zum sektionsbasierten Wiederaufbau der Schaufelblattaussengeometrie anhand von Blade2Parameter generierten Daten zur Verfügung. Blade2Parameter verwendet die Mittelwertgeometrie der Schaufel, auf der die im mittels eLHS generierten Delta-Parameter aufgeprägt werden. Diese Schaufelsektionen werden in eine xml-Datei ausgeschrieben und könne anschliessend in den Datensatz von gpds-fc eingebunden werden. Im Gegensatz dazu erfolgt der parametrische Wiederaufbau der Kühlkanäle direkt innerhalb der gpds-fc-Prozesskette. Die Mittelwertgeometrie und die Delta-Parameter werden in einer XML-Datei definiert, wodurch der Aufbau der Innengeometrie effizient realisiert werden kann.

Da eine vollständige Darstellung aller Schritte den Rahmen dieser Arbeit sprengen würde, werden im Folgenden nur die für die probabilistische Analyse relevanten Aspekte des Geometriewiederaufbaus beschrieben.

Die Aussenkontur der Schaufel wird zunächst in Parablading v23.0 als Vollkörper auf Basis der bereitgestellten Sektionen erstellt. Für die spätere Integration des Winglets wird der obere Bereich der Schaufel durch einen definierten Offset aufgedickt. Die Guttergeometrie wird durch Subtraktion eines nominalen Schnittwerkzeugs erzeugt. Die Plattform und der Schaufelfuss werden als Vollkörperrohling modelliert und anschliessend mit der Aussenkontur verbunden. Ein variabler Radius für das Fillet wird mithilfe des Aerofillettool v10.1.0 hinzugefügt. Weitere geometrische Details wie die Shank- und Wedgeplan-Bereiche werden durch gezielte Subtraktionen basierend auf der Nominalgeometrie gestaltet.

Die Innengeometrie wird mit dem Geometrybuilder v2.4.0 erstellt, welcher speziell für den variablen Aufbau der Kühlkanäle angepasst wurde. Die Mittelwertgeometrie der Kanäle wird durch Splines definiert und basierend auf den Sampling-Parametern für die Schwerpunktverschiebung transformiert. Bereiche ohne Messdaten, wie beispielsweise die Übergänge zwischen den Kernen, werden linear interpoliert. Die Form der definierten Kühlkanalquerschnitte bleibt dabei invariant. Anschliessend erfolgt die Integration der Turbulatoren in die Kühlkanäle. Die Geometrie sowie die Position und Ausrichtung der Turbulatoren bleiben unverändert, da die vorliegenden Messdaten keine zuverlässige Grundlage für eine probabilistische Analyse der Geometrievariationen bieten.

Nach der Subtraktion der Innengeometrie vom Vollkörper werden die Kühlluftlöcher eingefügt. Diese werden relativ zur Aussenkontur in Richtung des Schaufelkerns positioniert, um eine robuste Geometrieerzeugung zu gewährleisten und die realistischen Fertigungsbedingungen durch Laserbohrung besser abzubilden [122].

Abschliessend wird die fertige Turbinenschaufelgeometrie für die weiteren Simulationen vorbereitet. Für die FEM-Analyse wird die kalte Geometrie der Schaufel verwendet. Dabei wird der Schaufelfuss mit einem Turbinenschaufelsegment verschnitten, um definierte Kontaktbedingungen zwischen Schaufelfuss und Scheibe sicherzustellen. Weitere Details hierzu werden in Kapitel 4.3.5 erläutert.

Für die CFD- und CHT-Analyse wird hingegen die heisse Geometrie benötigt. Hierzu wird die Geometrie zunächst auf Basis üblicher Betriebstemperaturen von kalt zu heiss transformiert. Aufgrund des monokristallinen Materials der Schaufel wird ein anisotroper thermischer Ausdehnungskoeffizient verwendet. Zusätzlich wird die Schaufel unter Berücksichtigung der Scheibenausdehnung durch Translation und Rotation an die korrekte Position bei Betriebstemperatur gebracht.

Das Fluidgebiet wird anschliessend basierend auf der heissen Schaufelgeometrie sowie der Gehäuse- und Nabengeometrie abgeleitet. Dabei werden Randbedingungen wie die Mischebene hinter dem ersten Stator, die Austrittsebene des Rotors und die Eintrittsbereiche für Kühlluft und Kavität definiert. Weitere Details zu den Randbedingungen und dem Strömungsgebiet werden in Kapitel 4.3.5 beschrieben.

CFD mit konjugierter CHT Analyse

Im Rahmen der vorliegenden Untersuchung wurde die Vernetzung des Modells in STAR-CCM+ durchgeführt, wobei sich die Vernetzung auf Rotor 1 und die zugehörige Fluiddomäne beschränkte. Die Fluiddomäne wurde mit einem radialen Spalt vernetzt, während die Soliddomäne und die Fluiddomäne gleichzeitig vernetzt wurden. Für die Soliddomäne wurde ein polyedrisches Netz verwendet, während die Fluiddomäne mit einem hybriden Vernetzungsverfahren ausgestattet wurde, das polyedrische Zellen und Prismenschichten im wandnahen Bereich kombiniert. Dieses Verfahren gewährleistet eine hochauflösende Darstellung der Strömungs- und Temperaturprofile in den Wandbereichen.

Die Position der Einlass- und Auslassebenen wurde auf Basis von Voruntersuchungen von Rolls-Royce festgelegt und nicht verändert. Die Vernetzung erfolgte vollautomatisch über ein in Java entwickeltes Template unter Verwendung des Plugins TurbCool V2302.1.1.1 von Rolls-Royce.

Für eine korrekte Vernetzung war es erforderlich, die relevanten Knoten, Flächen und Volumenkörper der Schaufel innerhalb der Prozesskette eindeutig zu kennzeichnen. Anschliessend wurde ein Imprint der Festkörper- und Fluiddomäne aus der CAD-Geometrie abgeleitet, um eine Schnittstelle zwischen den beiden Berechnungsbereichen zu schaffen.

Nach der Definition der Schnittstelle wurde zunächst die Oberflächenvernetzung für beide Domänen erzeugt. Dabei wurde an gemeinsamen Schnittstellen eine identische Vernetzung mit zusammenhängenden Knoten erstellt, wodurch die Konvergenz der Festkörper-Fluid-Wechselwirkung verbessert wurde. Aufbauend auf dieser Oberflächenvernetzung wurde das in Abbildung 4.28 dargestellte Volumennetz erzeugt. Dabei wurden zunächst die wandnahen Prismenschichten vernetzt und anschliessend der wandferne Kernbereich der Geometrie.

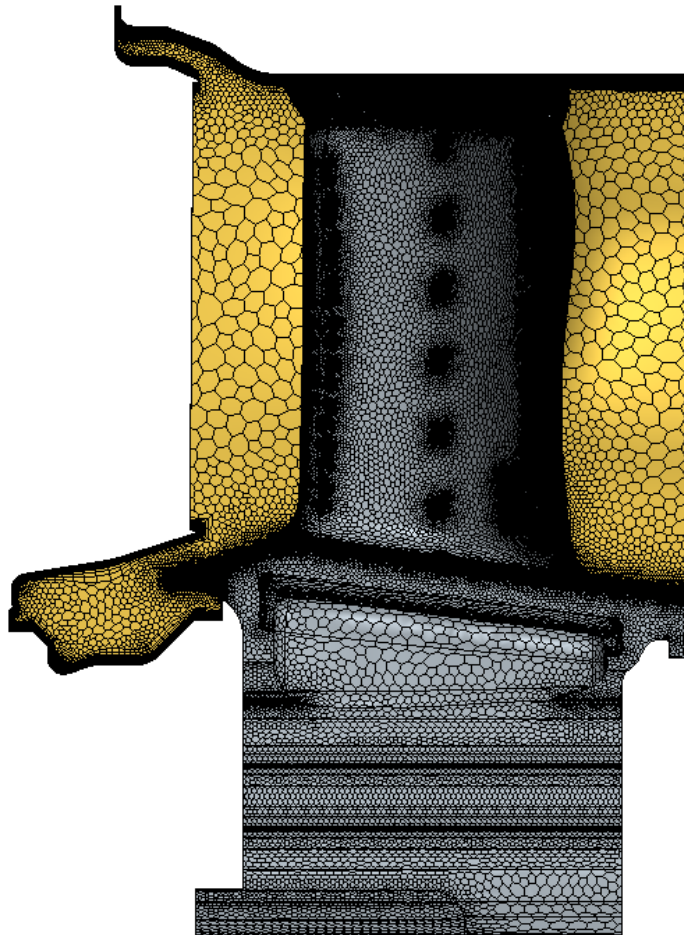


Abbildung 4.28: Vernetzte Fluiddomain mit Rotor

Der Vernetzer in STAR-CCM+ bietet verschiedene Einstellungsmöglichkeiten, um die Qualität der Netzgenerierung zu beeinflussen, darunter:

- Zielzellgrösse,

- Mindestzellengrösse,
- Auflösung der Oberflächenkrümmung,
- Zellwachstumsfaktor,
- Prismenschichtdicke.

Die minimale Zellgrösse und die Zielzellgrösse werden relativ zu einer Basisgrösse angegeben, um die Skalierbarkeit zu verbessern. Da eine einheitliche Zellgrösse im gesamten Rechengebiet nicht sinnvoll ist, wurde eine flächenbezogene Netzverfeinerung durchgeführt, wobei kritische Bereiche wie die Kühlluftbohrungen, der Kühlluftkanal und bestimmte Zonen entlang der Schaufeloberflächen gezielt verfeinert wurden.

Die verwendeten Zellgrössen sind der Tabelle 4.3 zu entnehmen:

Tabelle 4.3: Zellgrössen in verschiedenen Bereichen des CHT-Modells

Bereich	Zielzellengrösse	Mindestzellengrösse
LE, PS, ss	$500 \times 10^{-3} \text{ mm}$	$100 \times 10^{-3} \text{ mm}$
TE	$500 \times 10^{-3} \text{ mm}$	$20.0 \times 10^{-3} \text{ mm}$
Kernbereich	$750 \times 10^{-3} \text{ mm}$	$100 \times 10^{-3} \text{ mm}$
Turbulatoren	$100 \times 10^{-3} \text{ mm}$	$25.0 \times 10^{-3} \text{ mm}$
Kühlluftbohrungen	$250 \times 10^{-3} \text{ mm}$	$500 \times 10^{-3} \text{ mm}$

Um einen dimensionslosen Wandabstand von $y^+ < 5$ zu erreichen, wurde im wandnahen Bereich eine Zellhöhe von $1.00 \mu\text{m}$ verwendet. Die Dicke der Prismenschichten wurde auf $1,25 \mu\text{m}$ festgelegt, um eine feine Auflösung der Wandbereiche zu gewährleisten.

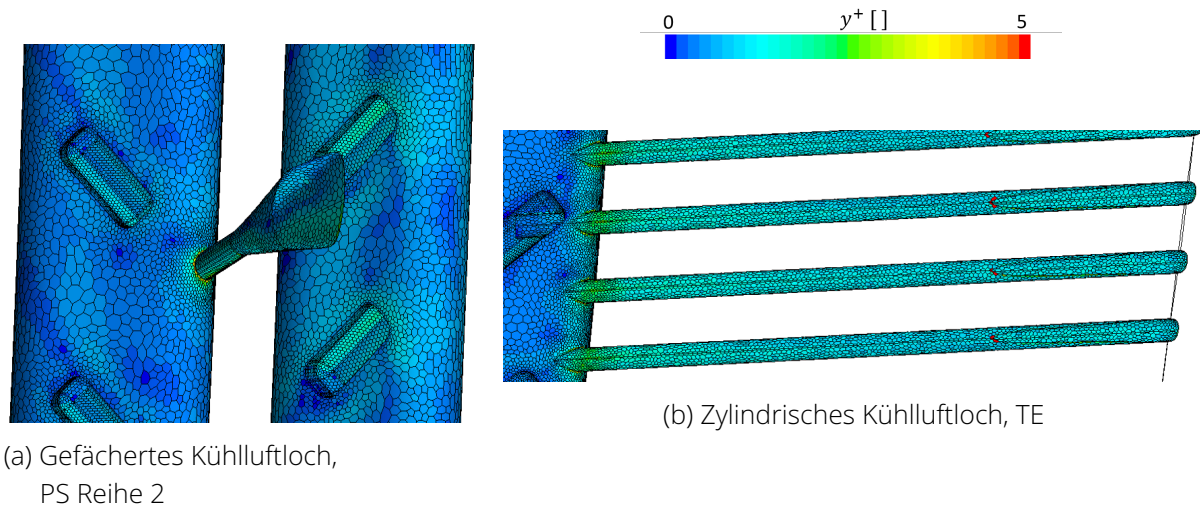
Die Netzqualität wurde anhand der folgenden Kriterien überprüft:

- **y^+ Bereich:** Es wurde ein Wert von $y^+ < 5$ angestrebt, um die korrekte Modellierung der viskosen Unterschicht, wie in Kapitel 4.1.3 beschrieben, zu gewährleisten.
- **Volumenverhältnis:** Das Auftreten starker Unterschiede im Volumen benachbarter Zellen wurde minimiert.
- **Winkelschiefe:** Verzernte Zellen wurden vermieden, um numerische Fehler zu reduzieren.

Trotz umfangreicher Netzverfeinerungen konnte der Wert $y^+ < 5$ in kritischen Bereichen wie Kühlluftlöchern und Turbulatoren nicht immer eingehalten werden, wie in Abbildung 4.29 zu sehen ist. Insbesondere die komplexe Geometrie an den Ein- und Auslässen der Kühlluftöffnungen führte zu erhöhten Unsicherheiten bei der Modellierung der thermischen Randbedingungen.

Für die Simulation werden sowohl die Festkörper- als auch die Fluiddomäne in einem rotierenden System definiert. Die Fluiddomäne wird als ideales Gas mit den Eigenschaften reiner Luft gemäss den Rolls-Royce-Daten angenommen. Die Fluideintritts- und -austrittsbedingungen sind wie folgt definiert:

- **Druck Outlet:** am Outlet des Hauptgaspfades,

Abbildung 4.29: y^+ -Wert und Vernetzung von Kühlluftlöcher

- **Stagnation Inlet:** für den Inlet des Hauptgaspfades und des Kühlluftpfades unter Angabe Totalgrößen und der Tubulenz
- **Massenstrom Inlet:** für den Inlet innerhalb der Cavitäten

Die Randbedingungen wurden von Rolls-Royce aus einer separaten Stator- und Sekundärluftsystemberechnung vorgegeben. Zusätzlich wurden periodische Randbedingungen zwischen den Strömungspfaden der Schaufeln vorgegeben, um den Rechenaufwand zu reduzieren und dennoch das zyklische Verhalten des Systems zu berücksichtigen.

Für die thermische Kopplung zwischen Festkörper und Fluid wird eine Zwei-Wege-Kopplung implementiert, um die Wärmeleitung und Temperaturverteilung präzise abzubilden. Das Material der Festkörperdomäne ist CMSX4 mit temperaturabhängigen Materialeigenschaften und konstanter Dichte.

Die Simulation wurde im stationären Modus durchgeführt, um Rechenzeit zu sparen und gleichzeitig die Genauigkeit für den stationären Betriebszustand der Turbinenschaufel zu gewährleisten. Als Turbulenzmodell wurde das $k - \omega - \text{sst}$ -Modell gewählt, da es die Vorteile des $k - \omega$ -Modell für die Grenzschichtauflösung mit denen des $k - \varepsilon$ -Modell für freie Strömungen verbindet.

Um eine stabile Lösung für die Simulation zu erhalten, wurden insgesamt mehr als 1000 Iterationsschritte durchgeführt. In der Regel konnte jedoch bereits nach ca. 500 Iterationen eine Konvergenz beobachtet werden, die für die meisten Fälle ausreichend war. Um jedoch auch in den Fällen, in denen die Simulation schlecht konvergiert, eine zuverlässige Lösungsgenauigkeit zu gewährleisten, wurde entschieden, die Simulationen bis zum Ablauf aller 1000 Iterationsschritte fortzusetzen. Andere Abbruchbedingungen, die auf einem schnelleren Abbruch basierten, erwiesen sich als zu fehleranfällig, da sie in einigen Fällen keine stabile Lösung lieferten.

Aufgrund der oben genannten Einschränkungen in der Netzqualität ist infolge von numerischen Instabilitäten in den Nachströmbereichen der Kühlluftöffnungen eine periodische Schwankung der Kühllufttemperatur zu beobachten. Daher wurde entschieden, das Temperaturfeld über die

letzten Iterationen zu mitteln. Bei der Betrachtung der Kühllufttemperatur wurde festgestellt, dass die Temperatur innerhalb von ca. 200 Iterationen zwei Perioden durchläuft. Aus diesem Grund wurde das Mittelungskriterium auf 200 Iterationen festgelegt, um ein zuverlässigeres Ergebnis für die weitere Auswertung zu erhalten. In Bezug auf die Materialtemperatur konnte ausgewertet werden, dass die Schwankung der Materialtemperatur geringer ist.

Die Ergebnisse der Berechnungen sind in den Abbildungen 4.30a und 4.30b dargestellt. In Abbildung 4.30a ist die Standardabweichung der Temperatur zu entnehmen. Es ist zu erkennen, dass in weiten Teilen der Schaufel die Standardabweichung unter 5.00 K liegt, was auf eine gute Stabilität der Temperatur in diesen Bereichen hinweist. Auffällig ist jedoch, dass im Bereich der Kühlluftbohrungen die Standardabweichung der Materialtemperatur Werte über 5.00 K erreicht. Diese erhöhte Schwankung deutet auf die komplexen Strömungs- und Wärmeübertragungsvorgänge in diesem Bereich hin, die durch die lokale Netzqualität beeinflusst werden können.

Die Abbildung 4.30b zeigt die mittlere Temperaturverteilung. Diese entspricht der typischen Temperaturverteilung eines Rotors, bei der das Temperaturmaximum an der Vorderkante liegt. Weiterhin ist eine gleichmässige Temperaturverteilung des Materials über weite Teile der Druckseite der Turbinenschaufel zu erkennen. Im Bereich der Filmkühlung hingegen ist die Materialtemperatur etwas niedriger, was auf die effektive Wärmeabfuhr durch den Kühlluftstrom zurückzuführen ist. Auffällig ist, dass entgegen den Erwartungen keine hohen Temperaturen an der Hinterkante auftreten. Dies ist auf die Art der Simulation zurückzuführen, die im stationären Zustand durchgeführt wurde. Die Temperatur an der Hinterkante wird jedoch hauptsächlich durch instationäre Phänomene beeinflusst, die mit der verwendeten numerischen Simulation nicht abgebildet werden können. Die dargestellten Ergebnisse unterstreichen jedoch die Bedeutung der Filmkühlung und verdeutlichen die Grenzen der verwendeten numerischen Methode bei der Abbildung bestimmter instationärer Effekte [45].

Spannungsanalyse

Im Rahmen der Spannungsanalyse wurden zwei wesentliche Ansätze verfolgt, um die strukturelle Integrität der Turbinenschaufeln unter realistischen Betriebsbedingungen zu bewerten. Zunächst wurde eine Finite-Elemente-Simulation mit dem Tool SC03 von Rolls-Royce Deutschland durchgeführt. Anschliessend wurde eine vergleichbare FEM-Berechnung in Star-CCM+ implementiert.

SC03

SC03 ist ein hochentwickeltes, automatisiertes Analysewerkzeug, mit dem komplexe Modelle von Gasturbinenkomponenten strukturell, thermisch und thermomechanisch analysiert werden können. Dabei wird eine bestehende Prozesskette von Rolls-Royce Deutschland verwendet, die eine effiziente Möglichkeit bietet, die Auswirkungen von Temperatur- und Druckänderungen auf die Turbinenschaufel zu bewerten.

Die Prozesskette in SC03 besteht aus mehreren Schritten:

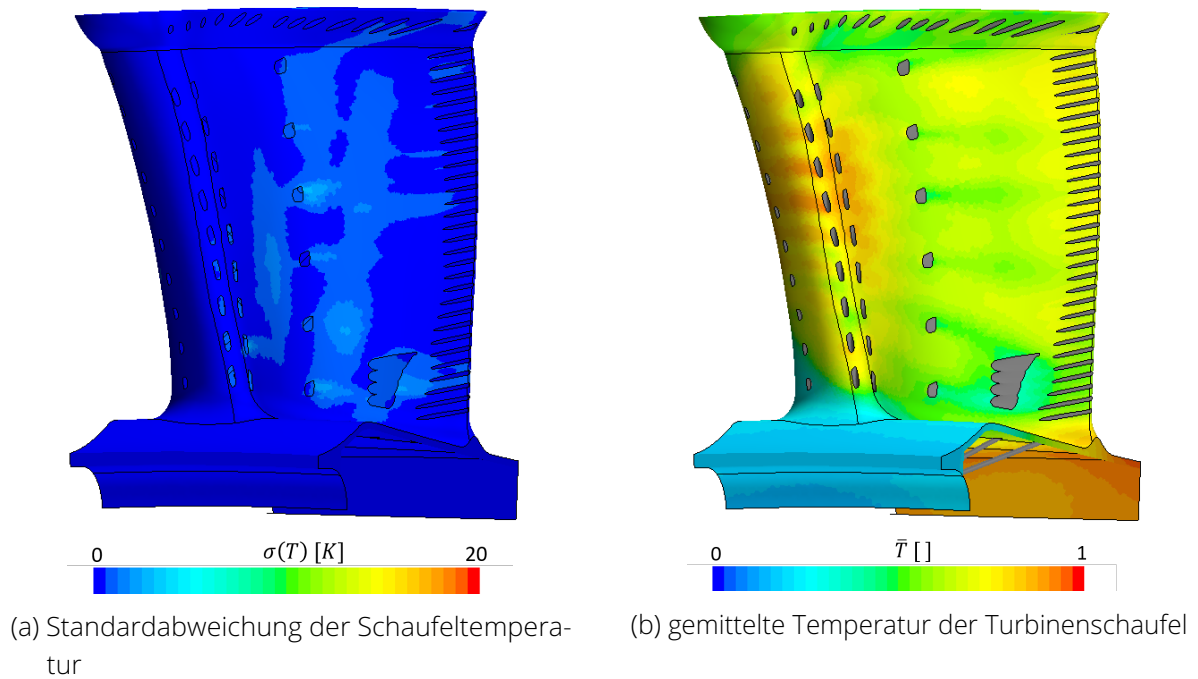


Abbildung 4.30: Temperaturfeld der CHT Berechnung

1. Vernetzung von Turbinenschaufel und Turbinenscheibensegment:

Die Turbinenkomponenten werden zunächst im kalten Zustand vernetzt. Das Netz besteht aus tetraedrischen Elementen mit zentralen Knoten, wie bereits in Abbildung 4.7a gezeigt.

2. Mapping der Temperaturfelder:

Im nächsten Schritt werden die aus Star-CCM+ exportierten Temperaturfelder in Form einer diskreten Punktwolke auf das generierte Netz abgebildet. Dies geschieht mit Hilfe der Nearest-Point-Methode. Dieses Mapping ist von zentraler Bedeutung, da die Temperaturverteilung einen direkten Einfluss auf den Spannungszustand der Bauteile hat.

3. Elastische Geometrieberechnung:

Abschliessend wird eine nichtlineare elastische Geometrieberechnung des Scheibensegments und der Turbinenschaufel durchgeführt, um den Spannungszustand beider Komponenten zu bestimmen. Diese Berechnung liefert wertvolle Informationen über die mechanischen Spannungen, die während des Betriebs auftreten können, und ermöglicht eine fundierte Analyse der strukturellen Integrität.

Im Vergleich zur Geometrie in Star-CCM+ weist das Modell SC03 einige Vereinfachungen auf, die zur Beschleunigung der Berechnungen beitragen. Zum einen enthält die Geometrie keine Turbulatoren und zum anderen sind keine Kühlluftlöcher in der Rechengeometrie enthalten. Trotz dieser Vereinfachungen liefert das Modell eine gute Annäherung an die realen Spannungszustände in wesentlichen Bereichen des Schaufelblattes sowie im Schaufelfussbereich.

Bei der Konvergenzanalyse des Modells stellte sich jedoch heraus, dass das Modell aufgrund der hohen Rechenzeiten nur bedingt für die angestrebte Monte-Carlo-Simulation geeignet ist, was im Folgenden näher erläutert wird.

Zunächst wurde ein Netz für die Spannungsberechnung mit ca. 1.732 Mio. Knoten erstellt, welches plausible Spannungsergebnisse ohne Anomalien und grosse Spannungsgradienten aufwies. Im Zuge der weiteren Konvergenzanalyse sollte ausgehend von einem groben Temperaturexportnetz eine geeignete Auflösung gefunden werden, die eine möglichst genaue Berechnung der mechanischen Spannungen ermöglicht. Diese Exportnetzauflösungen sind in Tabelle 4.4 zusammengefasst.

Exportnetz	Knotenanzahl	Durchschnittliche Rechenzeit [h]
Netz 1	78 000	4.6
Netz 2	100 000	5.8
Netz 3	182 000	9.7
Netz 4	522 000	25.4
Netz 5	1 732 000	80.2

Tabelle 4.4: Berechnungszeiten SC03

Die Auswertungen der einzelnen Rechnungen ergab das die Netzauflösung 1 Temperatursprünge von bis zu 10.0 K aufwies, was zu einer Verspannung innerhalb der Finite-Elemente führt. Dies ist auf die zu geringe Auflösung des Temperaturfeldes zurückzuführen. Wird das Temperaturfeld aus Star-CCM+ direkt in die Rechenknoten von SC03 exportiert, wird der Fehler entsprechend minimiert.

Der Fehler durch thermische Spannungen innerhalb der Elemente ergab durch eine Überschätzung der maximalen von-Mises-Spannung um bis zu 76.0 MPa. Bei entsprechender Rechenetz- und Exportnetzauflösung ergeben sich jedoch Rechenzeiten von etwa 80.0 h pro Realisierung. Dementsprechend ist diese Art der Berechnung für eine Monte-Carlo-Simulation aufgrund der hohen Rechenzeit ungeeignet.

Es wurde daher entschieden, die FEM-Simulation in Star-CCM+ zu implementieren und ausgewählte SC03-Rechnungen zur Validierung des Rechenmodells heranzuziehen.

Star-CCM+

Ein wesentlicher Vorteil bei der Implementierung des FEM-Modells in Star-CCM+ ist die Übernahme wesentlicher Teile des Simulationsmodells, das im Kapitel 4.3.5 beschriebenen CHT-Modell. Die Geometrie der Turbinenscheibe konnte direkt aus dem SC03-Modell in Star-CCM+ importiert werden. Im Gegensatz zum SC03-Modell verwendet die FEM-Berechnung in Star-CCM+ jedoch ein Modell, das Kühlluftbohrungen und Rippen enthält, die aus der CHT-Berechnung stammen.

Die Materialeigenschaften wie Elastizitätsmodul, Wärmeleitfähigkeit, thermischer Ausdehnungskoeffizient, Querkontraktionszahl, Schubmodul und Materialdichte des Turbinenscheibensegments und der Turbinenschaufel konnten aus dem SC03-Modell übernommen werden oder wurden bereits im CHT-Modell verwendet.

Die Vernetzung der Rechengebiete Scheibensegment und Turbinenschaufel erfolgt direkt in Star-CCM+. Da der Solid Stress Solver von Star-CCM+ [153] die Vernetzung mit Polyederele-

menten nicht verarbeiten kann, war eine Neuvernetzung der Bauteile mit Tetraederelementen erforderlich. Ein wesentlicher Vorteil dieser Neuvernetzung besteht darin, dass die Netzauf-
lösung an die spezifischen Anforderungen eines FEM-Netzes angepasst werden kann. Diese
Anpassung unterscheidet sich von der notwendigen Vernetzung der CHT, die auch die Fluid-
domäne berücksichtigen muss.

Dementsprechend konnten folgende Netzanpassungen für die Turbinenscheibe und die Turbi-
nenschaufel vorgenommen werden, wie in Tabelle 4.5 zusammengefasst.

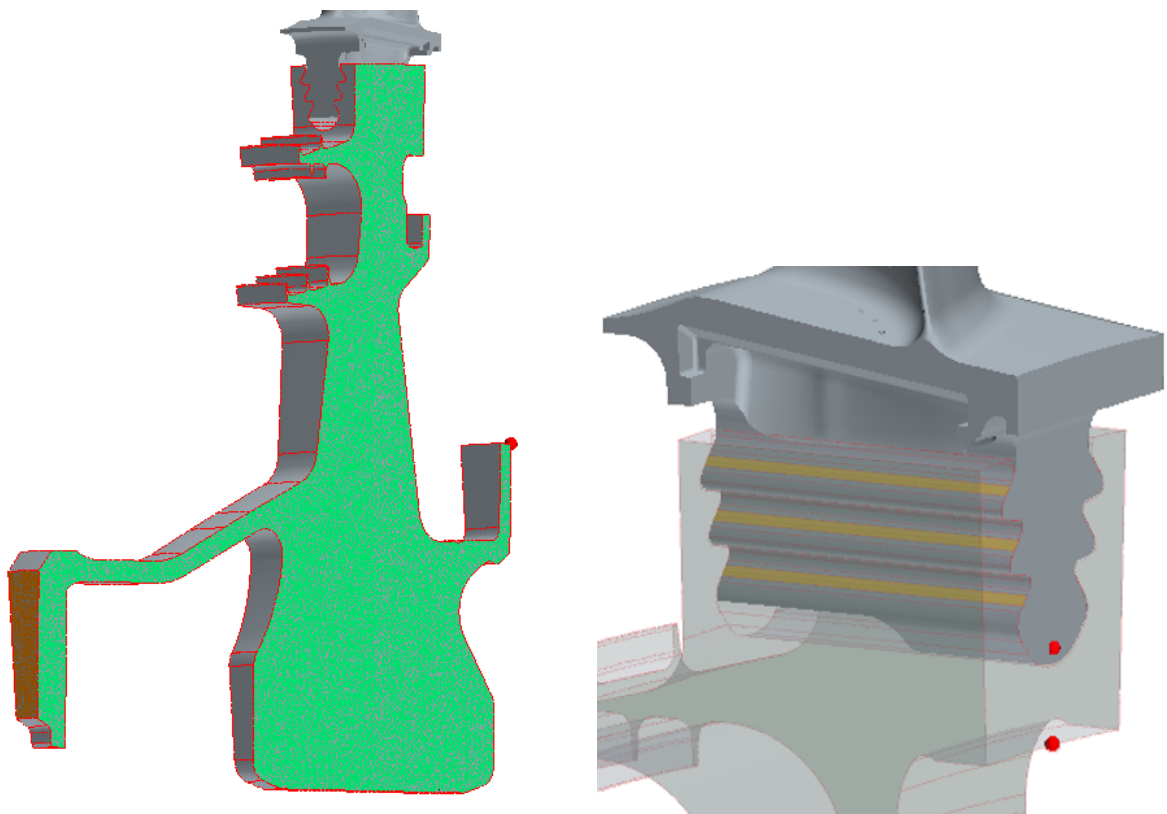
Tabelle 4.5: Zellgrössen in unterschiedlichen Bereichen des FEM Modells

Bereich	Zielzellgrösse	Minimale Zellgrösse
LE	$350 \times 10^{-3} \text{ mm}$	$100 \times 10^{-3} \text{ mm}$
PS, ss	$450 \times 10^{-3} \text{ mm}$	$100 \times 10^{-3} \text{ mm}$
TE	$350 \times 10^{-3} \text{ mm}$	$100 \times 10^{-3} \text{ mm}$
Kühlkanal	$250 \times 10^{-3} \text{ mm}$	$50.0 \times 10^{-3} \text{ mm}$
Turbulatoren	$250 \times 10^{-3} \text{ mm}$	$50.0 \times 10^{-3} \text{ mm}$
Kühlluftbohrungen	$250 \times 10^{-3} \text{ mm}$	$50.0 \times 10^{-3} \text{ mm}$
Schaufelhals	$650 \times 10^{-3} \text{ mm}$	$100 \times 10^{-3} \text{ mm}$
Schaufelfusskontaktfläche	$500 \times 10^{-3} \text{ mm}$	$100 \times 10^{-3} \text{ mm}$
Oberfläche der Scheibe	$650 \times 10^{-3} \text{ mm}$	$100 \times 10^{-3} \text{ mm}$

Da keine explizite Auswertung der Spannungszustände auf dem Scheibensegment erfolgt und weite Bereiche der Scheibe keine grossen Spannungsgradienten aufweisen, konnte hier im Vergleich zur Schaufel eine gröbere Vernetzung gewählt werden. Wie zuvor bei der CHT-Vernetzung erfolgt auch hier zunächst die Vernetzung der Objektoberfläche und anschliessend die Vernetzung des Objektvolumens mit einer Volumen- und Flächenwachstumsrate von 1.5 entsprechend den Empfehlungen von Star-CCM+. Dies ergibt eine Vernetzung von ca. 1 Million Zellen für die Scheibe und ca. 4.7 Millionen Zellen für die Turbinenschaufel.

Im Folgenden wird auf die gewählten Randbedingungen eingegangen. Zum einen wurde das Modell im rotatorischen System definiert. Die Beziehungen zum globalen Koordinatensystem werden ausschliesslich über das Scheibensegment definiert. Wie in Abbildung 4.31a grün dargestellt, wird der Scheibe eine zyklische Randbedingung aufgeprägt. Die axiale Position wird durch die orange Fläche am Scheibenflansch durch eine Loslagerdefinition begrenzt, während die Position in Umfangsrichtung durch radiale Zwangsbedingungen vom Punkt am Flansch zur nächsten Turbinenstufe definiert wird.

Die Randbedingungen für die Turbinenschaufel werden in Abhängigkeit vom Scheibensegment definiert, wie in Abbildung 4.31b gezeigt. Die axiale Randbedingung wird durch eine Knotenzwangsbedingung zwischen Scheibe und Schaufelfuss in axialer Richtung vorgegeben (rote Punkte), um die Implementierung weiterer Bauteile zu vermeiden. Der Kontakt zwischen dem Schaufelfuss wird durch eine reibungsfreie Kontaktbedingung über alle sechs Zähne des Schaufelfusses sichergestellt. Hierfür wurde zunächst ein Inprint des Schaufelfusses in die Scheibe durchgeführt, der ein Mapping der Knotenkräfte von den Schaufeln auf die Schei-



(a) Randbedingung Turbinenscheibensegment

(b) Randbedingungen Turbinenschaufel

Abbildung 4.31: Randbedingungen des FE-Modell

benkontaktf lächen ermöglicht. Im Gegensatz zu den anderen Kontaktbedingungen ist hierfür keine identische Vernetzung der Kontaktflächen auf Seiten der Scheibe und des Schaufelfusses erforderlich, allerdings sind numerische Abweichungen bei der Interpolation der Knotenkräfte zu berücksichtigen [15].

Zusätzlich wird die Schaufel innen und aussen mit Druckkräften aus dem CHT-Modell beaufschlagt. Dies geschieht durch ein Mapping des skalaren Feldes mit der Methode der kleinsten Quadrate, um die Daten auf das neu generierte Netz zu übertragen. Ebenso wird das Temperaturfeld des Festkörpers aus der CHT-Berechnung auf das neue Netz mit der gleichen Methode der kleinsten Quadrate abgebildet, wodurch sichergestellt wird, dass sowohl Druck- als auch Temperaturverteilungen genau berücksichtigt werden.

Für die Berechnung wurde der Sparse Direct Solver von Star-CCM+ verwendet, der eine direkte Lösungsmethode für lineare Gleichungssysteme bietet. Diese Methode eignet sich besonders für die Bearbeitung grosser und komplexer Modelle, da sie eine hohe Genauigkeit und Robustheit bei der Lösung der Gleichungen gewährleistet.

Das resultierende Spannungsbild im Vergleich zum Modell SC03 ist in Abbildung 4.32 dargestellt. Der Vergleich der von Mises-Spannungen aus der Star-CCM+-Modellierung (Abbildung 4.32a) mit den Ergebnissen aus dem SC03-Modell (Abbildung 4.32b) zeigt eine gute Übereinstimmung der Spannungsfelder.

Betrachtet man die Spannungsdifferenzen in Abbildung 4.32c, so erkennt man deutliche Spannungsunterschiede in der Nähe der Kühlluftbohrungen. Diese Unterschiede entstehen durch die Umlenkung des u.a. radialen Kraftflusses um die Kühlluftlöcher. Weiterhin ist zu erkennen, dass insbesondere im Kontaktbereich des Schaufelfusses die grössten Spannungsunterschiede auftreten. Dies lässt sich dadurch erklären, dass die Kontaktbedingungen aus dem Modell SC03 nicht direkt auf die Modellierung in Star-CCM+ übertragen werden konnten. Zudem treten im Kontaktbereich hohe Spannungsgradienten auf, die bereits bei kleinen Vernetzungsunterschieden zu signifikanten Abweichungen in den berechneten Spannungen führen.

Die Kerne der beiden Berechnungen sind in den Abbildungen 4.32d und 4.32e gegenübergestellt. Es ist zu erkennen, dass die Star-CCM+-Berechnung höhere Spannungsgradienten im Bereich der Turbulatoren aufweist. Dies ist darauf zurückzuführen, dass hier eine abrupte Änderung der Wanddicke durch die Turbulatoren auftritt, was zu einer Spannungsüberhöhung am Übergang zu den Rippen führt. Da das Modell SC03 keine Rippen hat, sind diese Effekte hier nicht sichtbar.

Bei der Betrachtung der Spannungsdifferenzen am Kühlluftkanal in Abbildung 4.32f wird weiterhin deutlich, dass durch die Kühlluftbohrungen und die Rippen insbesondere in diesen Bereichen eine Umverteilung des Kraftflusses stattfindet.

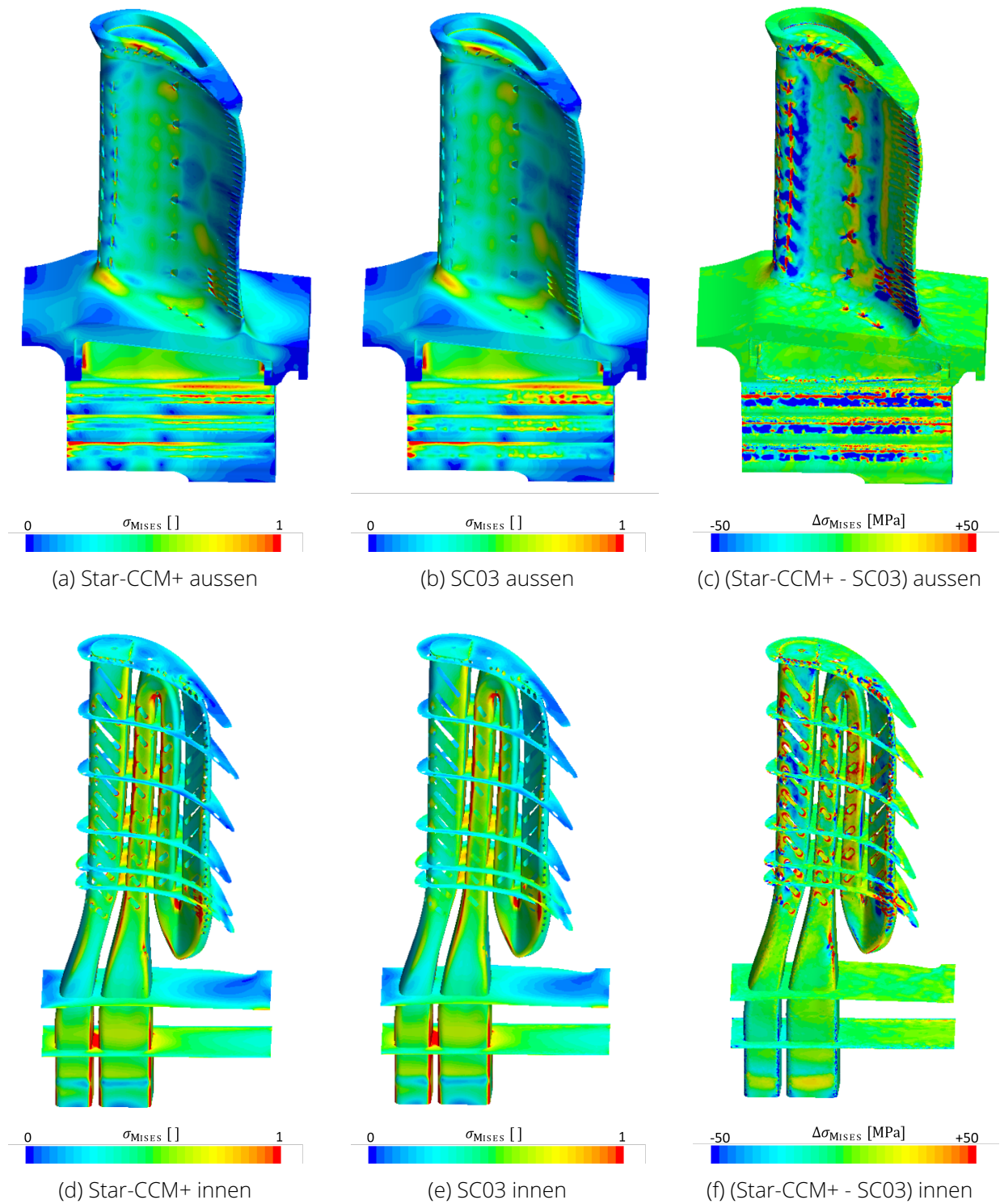


Abbildung 4.32: Vergleich der Von Mises Spannung zwischen Star-CCM+ und SC03

4.4 Zusammenfassung und Ausblick (AP2.1)

4.4.1 Zusammenfassung

Im AP 2.1 wurde die robuste und automatisierte Erzeugung von digitalen Zwillingen von einer Turbinenschaufel umfassend untersucht. Das Hauptziel bestand darin, die während der Produktion erfassten CT- und GOM-Messdaten in die BR000-Prozesskette zu integrieren, um diese für die konjugierte Wärmeübertragung (CHT) und Spannungsanalysen zugänglich zu machen.

Zunächst wurde eine detaillierte Analyse der verfügbaren Messmethoden und deren Genauigkeit durchgeführt. CT-Messungen ermöglichen die zerstörungsfreie Erfassung komplexer innerer Strukturen, während optische Verfahren eine flächenhafte Vermessung der äußeren Geometrie erlauben. Durch die Kombination der Methoden konnte ein umfassendes Bild der Geometrievariationen anhand von drei CT-Schnitten erstellt werden. Die gewonnenen Daten wurden anschliessend statistisch aufbereitet und in probabilistische Modelle überführt, um den Einfluss von Fertigungsstreuungen auf die aerodynamischen Eigenschaften der Schaufeln zu bewerten.

Ein wesentlicher Bestandteil der Arbeit war die parametrische Rekonstruktion der Schaufelgeometrie auf Basis der analysierten Messdaten. Diese Rekonstruktion diente als Grundlage für numerische Simulationen mittels CFD und FEM. In diesem Zusammenhang wurde der Modellierung der Prozesskette besondere Aufmerksamkeit geschenkt. Mit Hilfe dieser Simulationen konnte der Einfluss von Geometrievariationen auf Temperaturverteilungen und Spannungszustände innerhalb der Turbinenschaufeln ermittelt werden.

Die Ergebnisse dieser Untersuchungen zeigen deutlich, dass bereits kleine Abweichungen in der Geometrie signifikante Auswirkungen auf die Spannungsverteilung in den Turbinenschaufeln haben können.

4.4.2 Ausblick

Die vorliegende Arbeit des AP 2.1 legt den Grundstein für weitere Forschungen auf dem Gebiet der digitalen Zwillinge und deren Anwendung in der Triebwerksentwicklung. Um die Genauigkeit weiter zu erhöhen, sind mehrere Schritte erforderlich. Zukünftige Arbeiten sollten darauf abzielen, weitere Datenquellen wie Echtzeit-Messdaten aus dem Triebwerksbetrieb zu integrieren. Dies könnte dazu beitragen, digitale Zwillinge dynamisch anzupassen und ihre Vorhersagekraft zu erhöhen.

Darüber hinaus sollte eine Erweiterung des Turbulatorbereinigungsverfahrens zur automatischen Anpassung an neue Kühlkanalformen in Betracht gezogen werden. Eine genauere Verarbeitung der CFD-Ergebnisse ist ebenfalls von großer Bedeutung. Insbesondere die Verbesserung der CFD-Berechnung durch instationäre Berechnungen könnte dazu beitragen, instationäre Phänomene besser abzubilden und die Simulationsgenauigkeit insgesamt zu erhöhen.

Aufbauend auf den im Rahmen dieses Projektes durchgeführten Voruntersuchungen zur parametrischen Erfassung der Kühlluftöffnungen und Turbulatoren sind die Kenntnisse weiter zu vertiefen, um die Genauigkeit des Rechenmodells zu optimieren. Darüber hinaus ist eine Verbesserung der FEM-Berechnung durch nichtlineare Geometrieberechnungen notwendig, was eine Anpassung der Geometrie im Bereich der Kühlluftöffnungen erfordert.

Schliesslich könnte die Implementierung einer direkten bidirektionalen thermischen Kopplung zwischen Struktur und Fluid in einem Simulationsmodell den Umweg über eine Abbildung der thermischen und Druckdaten auf ein Spannungsmodell vermeiden. Dies würde die Genauigkeit des Modells erheblich verbessern und eine genauere Berechnung des Spannungszustands in der Turbinenschaufel ermöglichen.

5 Teilbericht AP2.3

5.1 Introduction AP2.3

Within the scope of this report, the following work packages (APs) of the joint project LuFo VI: DARWIN are addressed.

- AP 2.3.2: Automated measurement of the roughness of compressor components using optical methods.
- AP 2.3.3: Automated measurement of the surface of compressor components using optical methods.
- AP 2.3.4: Parameterization of the measured roughness fields and creation of digital geometry models.
- AP 2.3.5: Investigation of methods for the simulation-based evaluation of roughness using LES.
- AP 2.3.6: Transfer of the roughness effects into simplified modeling approaches (RANS/URANS).

This project report is structured as follows. In Section 5.2, an automated, efficient and accurate process has been developed for performing surface roughness measurements (AP 2.3.2) of 100 rotor blades, coming from the 10-stage HPC of two in-service RollsRoyce engines. Specifically, in Subsection 5.2.2, the employed measuring system is introduced and the process for obtaining the roughness results is described. Then, a validation of the measuring device is presented in Subsection 5.2.3. Subsection 5.2.4 demonstrates preliminary results from the detailed roughness measurements performed on Rotor 2 blades. The approach for optimizing the measuring point distribution is described in Subsection 5.2.5 and the final roughness measurement results for the entire HPC are summarized in Subsection 5.2.6. It should be noted that the GOM scans (AP 2.3.3) of all rotor blades and stator vanes have been performed (200 in total) and the results were documented internally. The activities relevant to roughness and geometry measurements are illustrated in Fig. 5.1. Section 5.3 describes the selected roughness parameterization (AP 2.3.4), whereas the aerofoil parameterization (generation of digital geometry models) was only partially addressed in this work and is better explained in [147]. In Section 5.4, the CFD-based modelling of the impact of roughness on turbulent flows is attempted, as illustrated in Fig. 5.2. Specifically, the CFD grid generation for performing channel flow simulations is presented in Subsection 5.4.2, followed by a validation of the CFD setup in Subsection 5.4.3. Subsection 5.4.4 demonstrates results for rough channels and a comparison between RANS and wallresolved LES (AP 2.3.5). The database constructed in this work is presented in Subsection 5.4.5 and used for training the proposed roughness function model (AP 2.3.6) in Subsection 5.4.6. Finally, a conclusion of the work is given in Section 5.5.

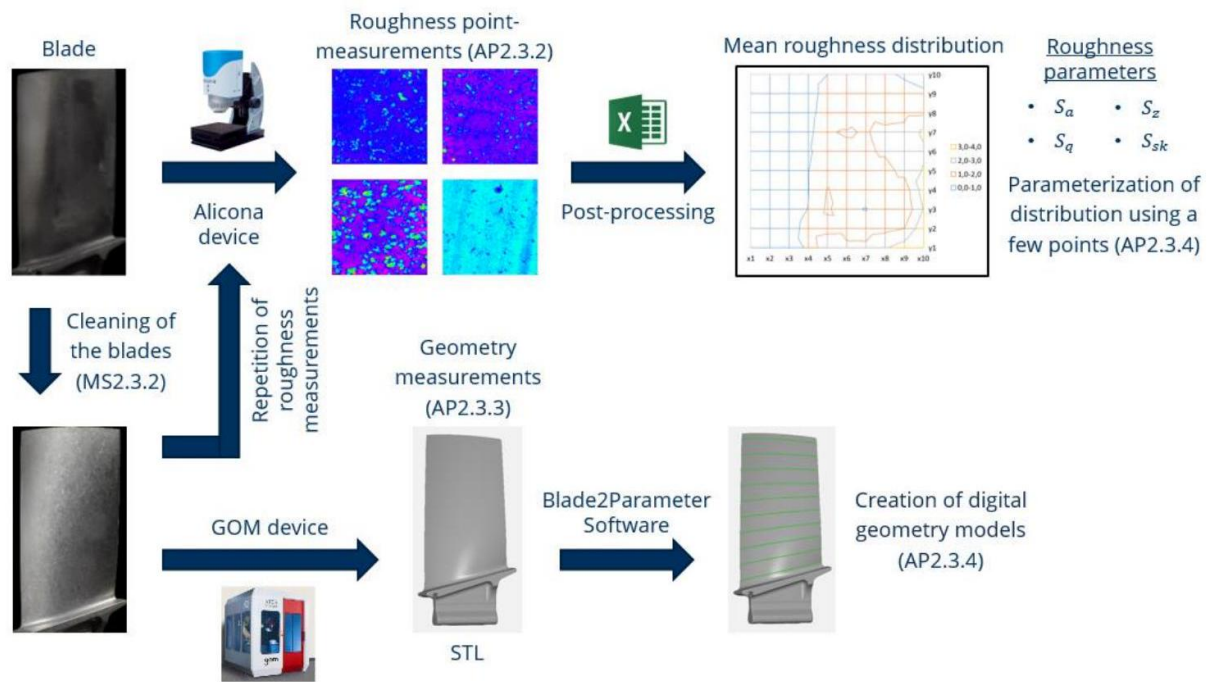


Abbildung 5.1: Processes for roughness and geometry measurements

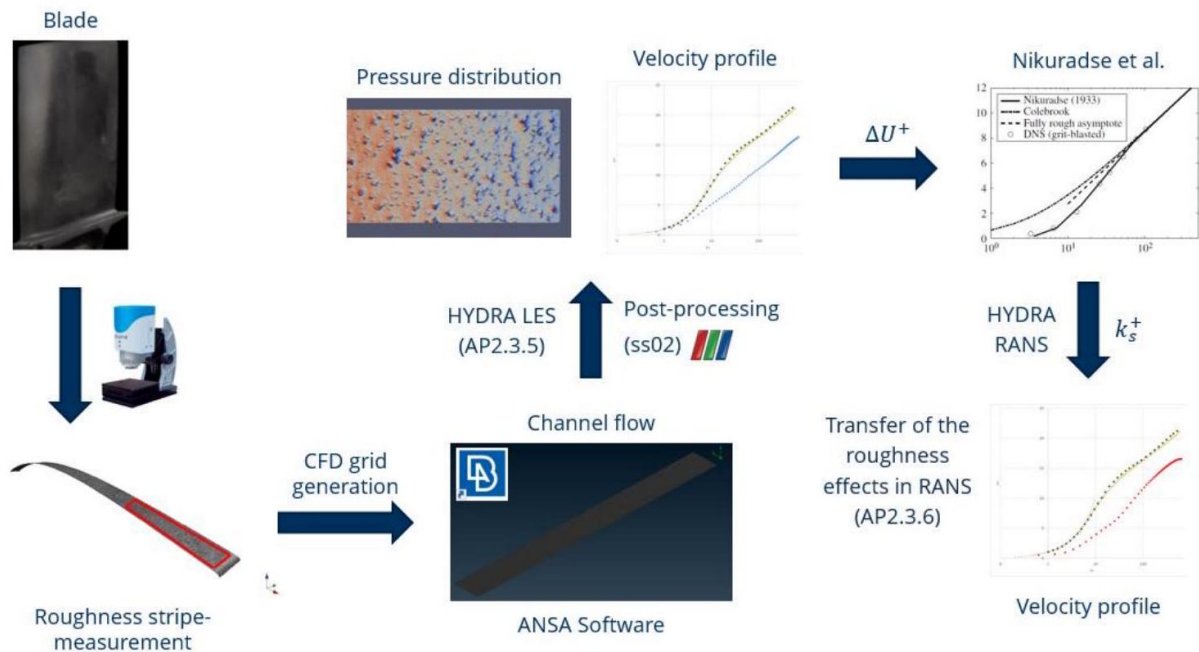


Abbildung 5.2: Simulation-based evaluation of roughness

5.2 Roughness Measurements

5.2.1 Literature Review

Surface roughness investigations on turbomachinery components have been rising in popularity, given the significant impact roughness can have on both compressor and turbine performance [61, 39]. Specifically for compressors, conclusions can be drawn from the experimental investi-

gation conducted by Back et al. [8] to characterize the influence of surface roughness magnitude and location. It was found that compressor performance is more sensitive to roughness on the suction side, especially on the aft 2/3 of the surface, where roughness leads to increased pressure loss. Moreover, for a given roughness, a critical Reynolds number exists at which performance begins to significantly deteriorate.

Obtaining detailed and accurate roughness measurement data is both time-consuming and expensive. Therefore, questions regarding the distribution of roughness along the pressure and suction sides of blades/vanes (or even through the entire engine), as well as its nature (e.g. due to fouling [112], corrosion [164], erosion [64] or impact damage [58]) still remain mostly unanswered in general. Bons et al. [20] shed some light on the unique features of each different roughness type, after performing surface roughness measurements of nearly 100 in-service turbine blades and vanes. Later, Bons [19] also compiled a detailed review of the effects of surface roughness in gas turbines - based on contributions from the relevant research community over the past few decades - to which the interested reader is referred for further information.

Recent advancements in optical surface metrology [34] allow efficient and accurate surface roughness measurements - in contrast to traditional tactile measurements [100], mainly along a line [167] - and make roughness studies more affordable. For instance, Walton et al. [177] present an automated technique to characterize the roughness of exservice compressor blade leading edges, which are found to exhibit a significantly higher surface roughness than other blade regions. In most of the relevant literature [56, 166], roughness is measured at only a few discrete points on the blade and, then, applied to the entire blade surface with the assumption of conformity. However, this simplification can lead to wrong predictions of the impact of roughness and, thus, a detailed investigation of the roughness distribution over the blade surface is necessary to, then, accurately model the effects of real surface roughness in engines. The latter is attempted by Gilge et al. [59], where the surface roughness of in-service rotor blades from different stages of an HPC and a first-stage blisk is measured by a 3D laser scanning microscope. The results from this publication are somewhat consistent with the findings of the present work, but not entirely though.

In this work, an automated, efficient and accurate process has been developed for performing surface roughness measurements of 100 rotor blades, coming from the 10-stage HPC of two in-service Rolls-Royce engines.

5.2.2 Measuring System and Automated Process

The components of the measuring system obtained by Alicona [24] and used to form an automated process for the roughness measurements conducted in this work are shown in Fig. 5.3. On the left, a sample holder is constructed, offering slots (currently 32 in total) for placing all the blades to be measured during the process. The distance between the slots (also called nests) must be the same and only blades of the same type (i.e. rotors from the same stage of the HPC) can be measured without interrupting the process. Given the differences in root geometry amongst the rotor blades, multiple different nests were required during the entire measurement campaign, which were manufactured using 3D printing. In the middle of

Fig. 5.3, a robotic arm (originally from Universal Robots [169]) is employed and trained to pick one blade at a time and bring it under the Alicona sensor, where the measurement is conducted. After the measurement is completed, the robot (also called cobot) returns the blade back to the sample holder and places it into the corresponding nest. The cobot must be trained to perform this pick-and-place process separately for each rotor and - given again the differences in blade root geometry - a few different finger geometries were 3D-printed and employed for an optimal grabbing and holding of the blades. Finally, all the measurement data is transferred to a computer (shown on the right of Fig. 5.3) where the Alicona software is installed and used for post-processing and acquisition of results. This entire automated solution for conducting roughness measurements of the specific hardware at hand was originally designed and tested by Alicona.

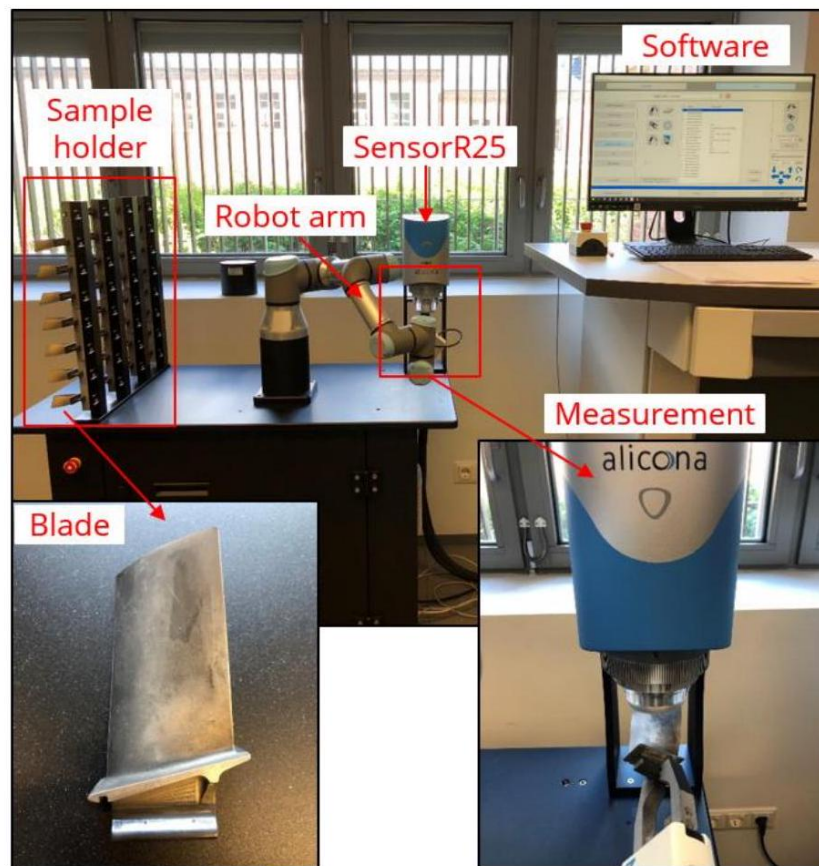


Abbildung 5.3: Roughness measuring system at TU Dresden

Alicona SensorR25 and Universal Robots Cobot

In this work, the highly accurate Alicona InfiniteFocus-SensorR25 [1] was used to perform the surface roughness measurements. Two lenses were provided for this sensor; namely the 10x and 20x lens, the technical specifications of which are given in Table 5.1. The coarser 10x lens is faster and can be used to measure larger square areas of $A = 4 \text{ mm}^2$, with a lower resolution than the 20x lens though, which is able to capture very fine roughness scales within a square area of $A = 1 \text{ mm}^2$. Regarding the lateral and vertical resolution parameters; L_r and V_r

respectively, it should be noted that the values in Table 5.1 do not correspond to the distance between the measuring points on the surface, rather they are related to the filter controlling the sensor resolution. The smaller these values are, the higher the accuracy of the measurement; however, it is advised to avoid values close to the limit, since this can lead to incomplete measurements (i.e. measuring points missing). Hereafter, the 10x lens is employed to explore the roughness distribution over the suction and pressure sides of rotors from the first stages of the HPC, by conducting detailed measurements based on a 100-point grid (Section 2.4), with $L_r = 3\mu\text{m}$ and $V_r = 0.2\mu\text{m}$. The 20x lens is used for the final results instead (i.e. after deriving an optimal distribution for a reduced number of measuring points - Section 2.5), with $L_r = 2\mu\text{m}$ and $V_r = 0.1\mu\text{m}$. In terms of time, a typical measurement (including the roughness computation) takes about 40 sec using the 10x lens and about 75 sec with the 20x lens. Finally, the definition of the roughness parameters Ra and Sa is explained in the following subsection.

Tabelle 5.1: Technical specifications of Alicona SensorR25

Lens	10x	20x
Measuring area A	2 mm $\times$ 2 mm	1 mm $\times$ 1 mm
Finest lateral resolution L_r	$2\mu\text{m}$	$1\mu\text{m}$
Finest vertical resolution V_r	$0.1\mu\text{m}$	$0.05\mu\text{m}$
Measuring speed	$\sim 1.7\text{M}$ measuring points / sec	
Min measurable Ra	$0.3\mu\text{m}$	$0.15\mu\text{m}$
Min measurable Sa	$0.15\mu\text{m}$	$0.075\mu\text{m}$

The automation of the measuring process is primarily attributed to the Universal Robots UR3 e-Series Cobot [170]. Apart from the pick-and-place process, the cobot is responsible for translating the blade under the sensor to perform the measurement at all the predefined/trained locations. In addition, its arm can rotate, which allows to conduct measurements on both sides of the blade in a single uninterrupted process. Finally, the cobot has also been programmed by Alicona to perform measurements of large areas (e.g. stripes from the leading to the trailing edge of the blades), by continuously translating the blade under the sensor and combining multiple smaller square measured areas. This capability has been used for the stripe measurements presented in Section 5.3.

Computing Roughness from a Measurement

The workflow for calculating the roughness parameters from a given measurement is summarized in Fig. 5.4 and the individual steps involved are briefly described in what follows. The entire computation is performed by the Alicona software and only minimal input is required by the user for the selection of the relevant parameters.

First, an accurate measurement has to be performed, for which the selection of the lens together with the corresponding resolution parameters L_r and V_r is critical, as already discussed. Recommendations on how to select these parameters based on the intended application are found in the Alicona manual. Another parameter that should be carefully selected since it

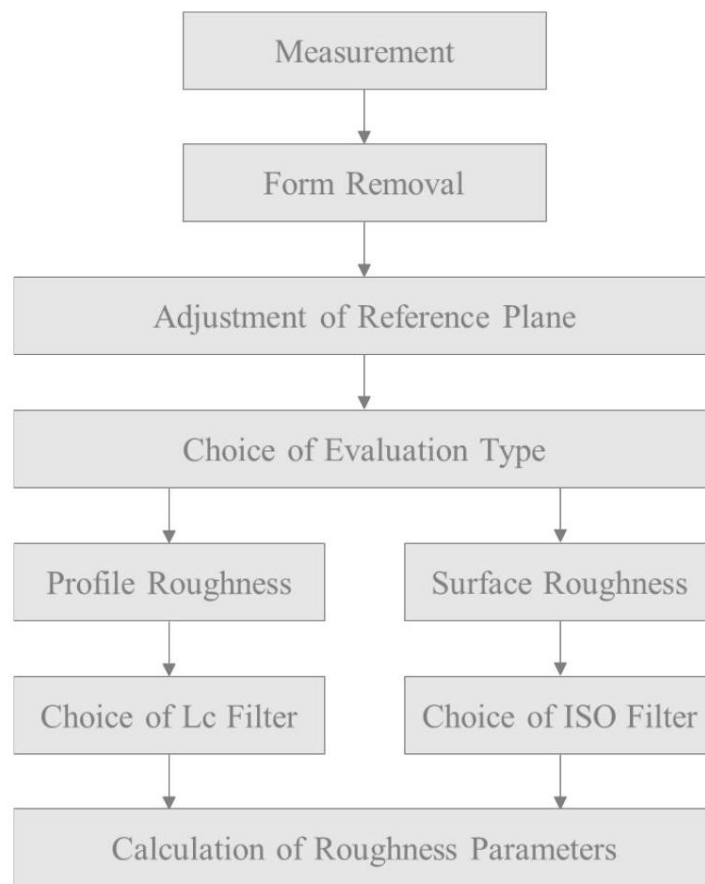


Abbildung 5.4: Workflow from measurement to roughness computation

affects both the validity and the time of the measurement is the z (i.e. vertical axis) range. Its lower bound value should be below the deepest valley of the measured surface and its upper bound above the highest peak. Selecting a larger than required z range leads to unnecessary increase of measurement time. Finally, the ring light settings of the sensor, such as the brightness (exposure time) and contrast parameters, may need to be adjusted accordingly for each case.

In order to obtain correct roughness values, it is important to first remove the true form of the surface (which could be planar, parabolic, cylindrical, etc.) from the measured 3D data-set. Herein, given the geometry of the rotor blades, a polynomial form removal was selected, which was found to perform best using a 3rd-degree polynomial for most cases. After this step, the reference plane has to be adjusted and moved to $z = 0$ before calculating the roughness parameters.

The Alicona software offers two different types of roughness evaluation one can choose from; namely the profile and surface roughness evaluation. The difference between these two types is exemplary illustrated in Fig. 5.5. The profile option evaluates roughness along a defined direction, whereas the surface option uses the entire 3D data-set to evaluate roughness. In terms of filtering, to separate waviness (low frequency components) from roughness, a 1st-order filter is used for profile evaluation, damping the features with a wavelength close to the chosen L_c cutoff value, whereas a wide variety of Gaussian filters (according to ISO) is available for surface evaluation. Guidelines on how to choose a proper L_c value and filtering

method are given in the Alicona manual. For the measurements performed herein, the 1st-order linear planar Gaussian filter was used with $L_c = 800\mu\text{m}$.

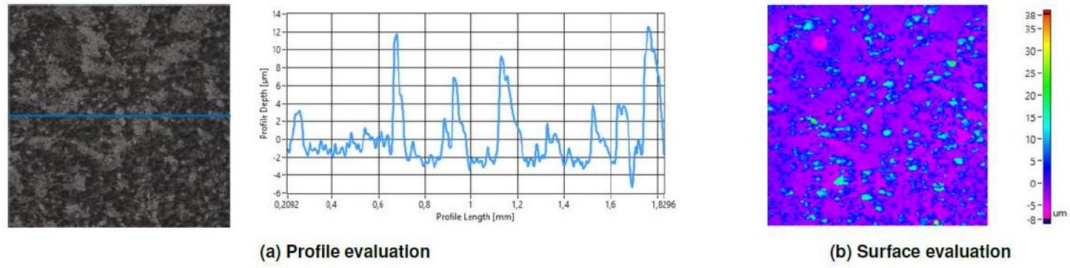


Abbildung 5.5: Example of profile versus surface roughness evaluation

Finally, a long list of roughness parameters can be calculated based on the processed data (such as the mean, RMS, skewness, kurtosis, etc. of the roughness distribution). However, this 1st part of the work focuses only on the mean roughness parameter, which is defined for a profile (of length l) and a surface evaluation as shown in Eqs. (5.1) and (5.2), respectively.

$$Ra = \frac{1}{l} \int_0^l |z| dx \quad (5.1)$$

$$Sa = \frac{1}{A} \int_A |z| dA \quad (5.2)$$

Even though characterising roughness with an Ra value is somewhat outdated, it is still being used in some cases, especially for comparing against data obtained by tactile measurements. In this work, instead, the higher-fidelity Sa parameter is used to evaluate roughness.

5.2.3 Validation of Measuring Device

Prior to conducting the roughness measurements of the in-service HPC rotor blades, several tests of the measuring system were performed, along with a validation of the accuracy of the Alicona sensor which is presented in this section. As a reference for the validation, the highly accurate μ surf custom NanoFocus measuring device [119] from the Chair of Magnetofluidynamics, Measurement and Automation Technology of TU Dresden was employed. The technical specifications of this device are: $A = 1 \text{ mm}^2$, $L_r = 1.56\mu\text{m}$, $V_r = 0.07\mu\text{m}$ (i.e. similar to the Alicona 20 x lens) and its Ra prediction accuracy was found to be about $\pm 25\text{nm}$, based on tests using calibration specimens. Regarding the measured hardware/samples, 5 Rolls-Royce HPC Rotor 3 blades were measured at a marked (randomly selected) location with both devices. To evaluate the repeatability of the results obtained by the Alicona device, each measurement was repeated 4 times. Finally, the Ra (along the diagonal of the measured square area) and Sa values were computed and compared for the two devices.

Figure 5.6 shows the results of the measurements performed for the validation of the Alicona sensor. Overall, the results obtained with the Alicona 20x lens are in a better agreement with the NanoFocus results, than the 10x lens results are (as expected from the lower resolution of

the latter), especially when comparing the Ra values. The error bars in Fig. 5.6 correspond to the min and max values of the repeated measurements and it should be noted that (even if not apparent at the chosen scale) error bars are included for all the points. As expected, the repeatability in terms of Sa values is higher, since a slight deviation of the measurement location has a smaller impact on the area-averaged roughness computation, than when computing the mean along a line, which indicates that Sa is a more robust parameter for characterising roughness. The mean and max differences of the two Alicona lenses (taking the NanoFocus results as reference) are summarized in Table 5.6. From these values it has been concluded that the Alicona sensor using the 20x lens, with a max Sa error of about $0.1\mu m$, is accurate enough to perform the roughness measurements of the HPC rotor blades (where eventually values up to $10\mu m$ were measured).

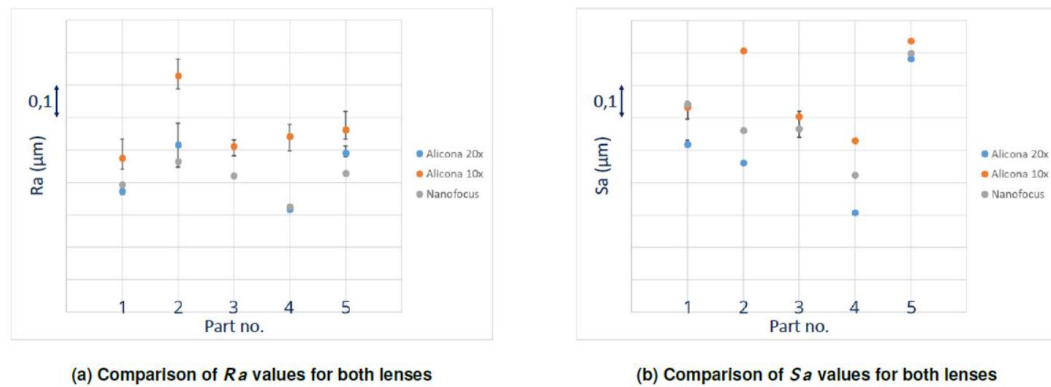


Abbildung 5.6: Validation of Alicona sensor against reference NanoFocus device

Tabelle 5.2: Differences between Alicona lenses and reference NanoFocus device

Alicona lens	10 x	20 x
$\overline{\Delta Ra}$	$0.158\mu m$	$0.035\mu m$
ΔRa_{max}	$0.264\mu m$	$0.062\mu m$
$\overline{\Delta Sa}$	$0.087\mu m$	$0.089\mu m$
ΔSa_{max}	$0.246\mu m$	$0.124\mu m$

5.2.4 Demonstration of Detailed Roughness Measurements for Rotor Blades

In this section, the developed automated process for performing roughness measurements of HPC blades is demonstrated for the detailed measurement of the roughness on the suction side (SS) and pressure side (PS) of the Rotor 2 blades. Specifically, two sets of 5 blades each have been measured, which were obtained from the 10-stage HPC of two in-service Rolls-Royce engines of the same type; namely TW1 and TW2 in what follows. Some information on the two engines are given in Table 5.3. The TW1 engine operated in a region, where the harsher environmental conditions (e.g. sand be present in the atmosphere) are expected to have a higher impact on the roughness of the blades, than for the TW2 engine. However, even though the TW2 engine has a smaller total operating time than the TW1 engine, its more than twice

number of cycles indicates that this engine (by performing shorter flights) has operated more time in the lower parts of the atmosphere, where there is a larger mass concentration of particles. Therefore, it is not apparent which engine's blades are expected to be overall more rough.

Tabelle 5.3: Information on the measured hardware

Engine	TW1	TW2
Operating time	T_1	$0.6T_1$
Number of cycles	N_1	$2.2N_1$

Figure 5.7 shows the 100-point measurement grid used for conducting the detailed roughness measurements on the SS of the Rotor 2 blades, plotted on top of the blade geometry (the dimensions of which have been intentionally distorted to meet the RollsRoyce export control restrictions - same stands for rest of the plots). Moreover, the difference between the two sensor lenses in terms of measuring area is illustrated, from which the 10x lens is used in this case. After all the measurements are completed, a mapping of the S_a values on a non-dimensional Cartesian grid is performed (where x stands for the chord-wise and y for the span-wise direction), to allow comparisons of patterns from the roughness distributions of blades from different stages of the HPC.

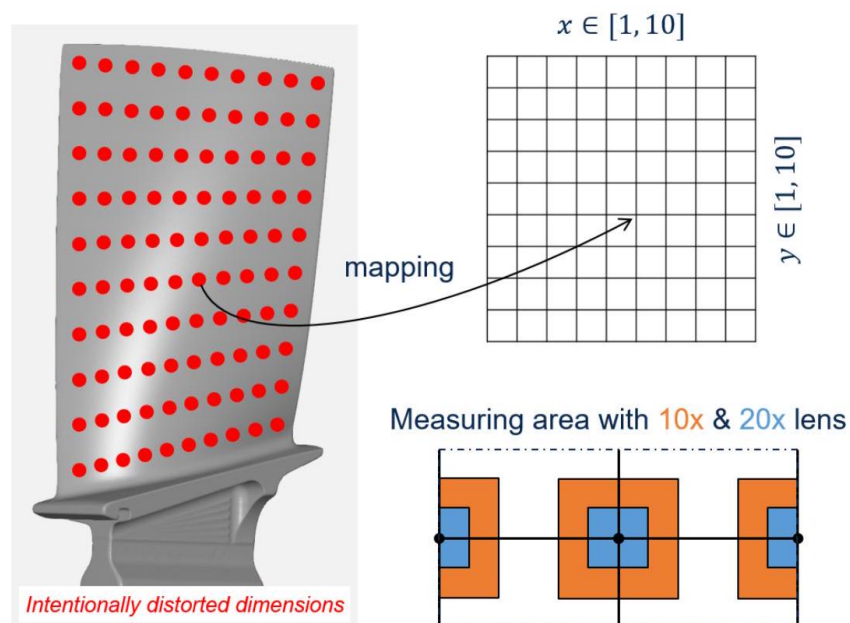


Abbildung 5.7: Measurement grid exemplary shown for Rotor 2 and difference between 10x and 20x lens

Images of the SS and PS of a representative Rotor 2 blade from both engines are included in Figs. 5.8 and 5.9, respectively. These have been taken using a high-resolution Nikon D90 camera. Focusing on the SS, the surface appears to be smooth at the leading edge (LE) region (where the flow is laminar) for both engines; however, distinctive differences are apparent when comparing the coloring of the images for the SS of the two engines. A large dark area can

be identified at the hub, close to the trailing edge (TE) region for the TW1 engine, indicating high roughness values due to dirt concentration possibly caused by the corner flow separation that occurs there. For the TW2 engine, the dark area corresponding to transition (at about 40% chord) is much larger and one would expect relatively high roughness values there. The PS appears to be much smoother for both engines, without any identifiable patterns, apart from the LE region at the tip of the TW2 engine blade that seems to be a bit eroded.

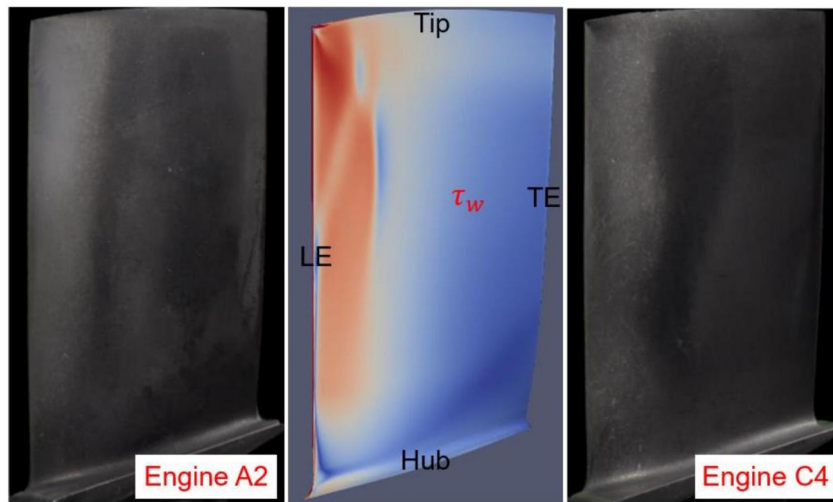


Abbildung 5.8: Rotor 2 suction side images of the two engines and comparison against contour plot of wall shear stress

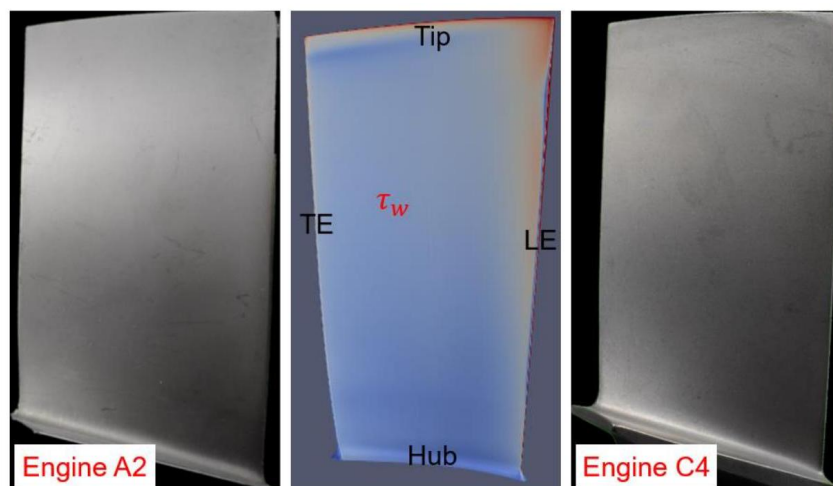


Abbildung 5.9: Rotor 2 pressure side images of the two engines and comparison against contour plot of wall shear stress

In addition, contour plots of the wall shear stress magnitude τ_w are shown in Figs. 5.8 and 5.9. These have been obtained from a steady-state RANS simulation of the full HPC at max takeoff operating conditions, considered to be representative of the conditions at which most of the roughness formation occurs (larger mass concentration of particles close to the ground). The interested reader can find more information regarding the CFD setup and the simulation

results in [147]. Comparing the τ_w plots with the corresponding SS and PS images, it can be seen that they correlate quite well. The idea is that in areas where τ_w approaches zero (blue color), separation occurs and therefore roughness is expected to be high due to fouling, whereas in areas where τ_w is high (red color), the local friction acting on the blade surface is high, thus removing any dirt deposits and keeping the surface smooth. A correlation between the distributions of τ_w and Sa over the SS and PS of the blades has not been sought in this work, however, it may be something worth exploring further in the future.

Finally, the roughness distribution (in terms of Sa values, averaged for the 5 measured blades) over the SS and PS of Rotor 2 for both engines is depicted in Figs. 5.10 and 5.11, respectively. It should be noted that qualitatively similar distributions characterize each individual blade and, thus, no loss of useful information occurs through the averaging. Due to confidentiality restrictions, the exact $\overline{Sa}$ values are hidden, however, the min values are the same for all the plots and $\overline{Sa}_{\max}^{A2,SS} = 2.2\overline{Sa}_{\max}^{C4,SS} = 2.2\overline{Sa}_{\max}^{A2,PS} = 3.3\overline{Sa}_{\max}^{C4,PS}$. Overall, the SS of the TW1 engine blades is more rough than the SS of the TW2 engine blades, with max values appearing at the hub-TE region for the TW1 engine and at about 40% chord of mid-span for the TW2 engine (consistent with the corresponding dark areas in Fig. 5.8). The high roughness on the SS is attributed to dirt deposition, which can be cleaned away. On the PS, the roughness distribution is much more uniform for both engines, with smaller roughness values overall. Highest values appear at the hub-LE region (due to fouling) for the TW1 engine and at both tip-LE and tip-TE regions (due to erosion) for the TW2 engine. Even if not apparent for Rotor 2, the roughness on the PS of most rotor blades is dominated by the presence of smaller or bigger craters, formed by particle impact.

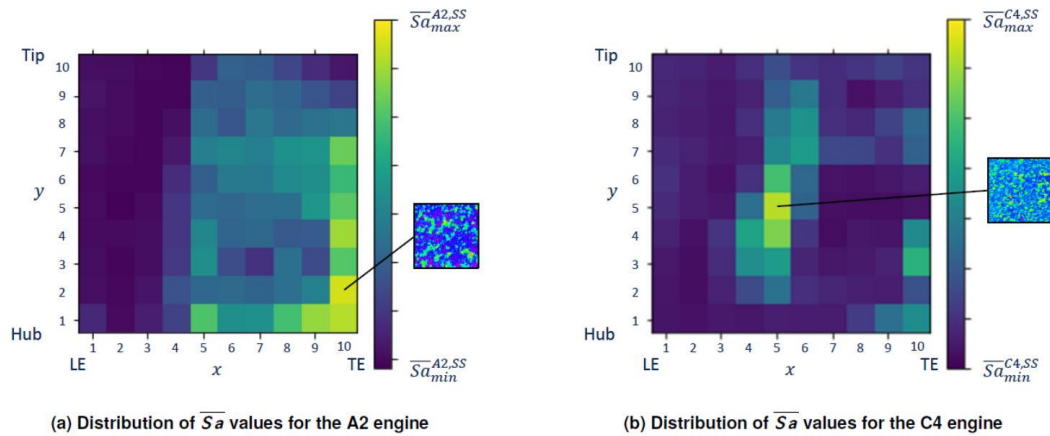


Abbildung 5.10: Detailed roughness distribution over the suction side of Rotor 2 blades

5.2.5 Optimization of Measuring Point Distribution

The detailed measurement process demonstrated in the previous section was employed for measuring the roughness of all the blades from Rotor 1 up to Rotor 4. However, given the fact that this is a fairly time-consuming process (both in terms of training the robot and performing the 100 measurements per blade side), a more efficient alternative was sought,

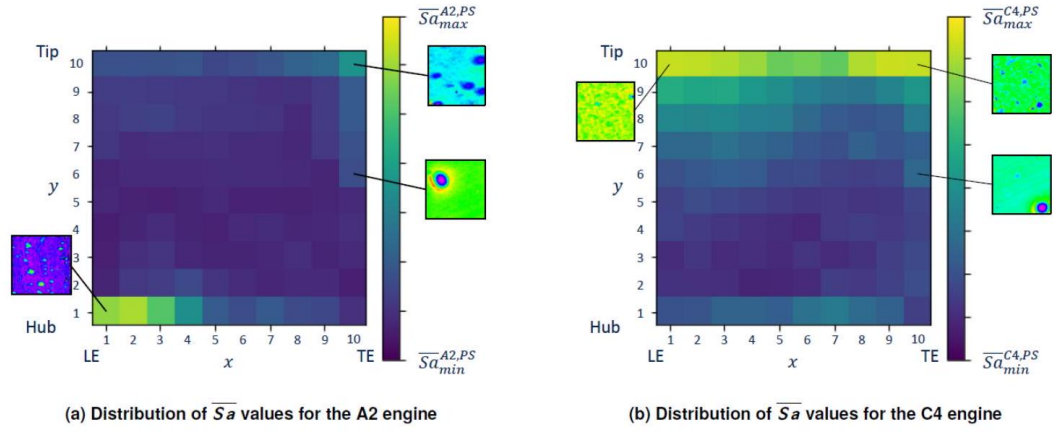


Abbildung 5.11: Detailed roughness distribution over the pressure side of Rotor 2 blades

with an as minimum as possible compromise regarding accuracy. Specifically, it was decided to reduce the number of measuring points per side and use an approximation model for predicting the roughness values in areas where no measurement is to be performed. For this, an optimal placement of the measuring points is required, in combination with a suitable approximation model giving minimal prediction error. The numerical optimization performed herein to determine the location of the measuring points is briefly described in what follows. Assuming the desired amount of measuring points is $N = 10$, the design vector of the optimization problem is written as: $\vec{X} = [X_1, \dots, X_i, \dots, X_N]^T$, where X_i is a unique natural number in $[1, 100]$. Then, a given distribution of the N measuring points over the 100point grid is represented by its corresponding design vector $\vec{X}$. The exact measured roughness values obtained at these locations can be written as: $\vec{Y}(\vec{X}) = [\overline{Sa}_{X_1}, \dots, \overline{Sa}_{X_i}, \dots, \overline{Sa}_{X_N}]^T$, with $\overline{Sa}_{X_i}$ being the mean Sa value (averaged for the 5 measured blades of each engine separately) at the location of point X_i . Based on the selected point distribution and corresponding measurement data, either a regression model via least-squares fitting or an interpolation model can be constructed (i.e. $f(\vec{X}, \vec{Y}) = \widehat{Sa}$) for approximating the $\overline{Sa}$ values over the entire surface. The approximation error (which is also the objective function to be minimized during the optimization) is defined here as

$$Err_{RMS} = \sqrt{\frac{1}{100} \sum_{j=1}^{100} (\overline{Sa}_j - \widehat{Sa}_j)^2} \quad (5.3)$$

This approach was applied using the measurement data from Rotor 2, to derive the point distribution (for both blade sides separately) that minimizes the mean error $\overline{Err}_{RMS} = 0.5 (Err_{RMS}^{A2} + Err_{RMS}^{C4})$. In terms of approximation models, 2nd and 3rd-order polynomial regression, non-uniform rational B-splines (NURBS) fitting as well as radial basis function (RBF) interpolation were tested (implementations from the Python SciPy library [134]), from which the latter was found to give the best results and it was therefore selected. Regarding the optimization approach, given the low cost of the problem's evaluation function, a simple Monte Carlo-based optimization with 50 M shots was performed within 1 h on a local PC. However, even this number of shots is rather small in comparison to the total number of possible point distributions, and it is therefore likely that the optimization result is not the global optimum of

the problem. For this, one could employ (for instance) an evolutionary algorithm being able to deal with discrete design variables as well as the several constraints of the problem (i.e. $\forall i, j$ of $\vec{X}, X_i \neq X_j$). Here, the reasoning behind not investing further into an optimization searching for the global optimum is that just a good enough point distribution was sought, to be then further modified and refined based on some additional criteria (e.g. adequate number of points at the LE and TE, relatively uniform distribution, no large areas without points).

The final selected point distribution, using 14 points for the SS and 12 points for the PS of the blades, along with the corresponding approximate roughness distribution (in terms of $\overline{Sa}$ values) is shown for both engines in Figs. 5.12 and 5.13 (respectively for each side). The red points stand for the locations where exact measurement data is used, whereas the $\overline{Sa}$ values for the remaining locations are obtained by RBF interpolation. A comparison of Figs. 5.12 and 5.13 with the corresponding Figs. 5.10 and 5.11 shows a quite good agreement between the approximate and the exact roughness distribution along both blade sides for both engines. In terms of approximation errors: $Err_{RMS}^{A2,SS} = 0.32\mu m$, $Err_{RMS}^{C4,SS} = 0.21\mu m$, $Err_{RMS}^{A2,PS} = 0.11\mu m$ and $Err_{RMS}^{C4,PS} = 0.07\mu m$, which are considered to be acceptable since they are about an order of magnitude smaller than the corresponding measured range of $\overline{Sa}$ values.

Even though the measuring point distribution was optimized for Rotor 2, it has also been tested for Rotors 1,3 and 4 , giving small approximation errors for all the cases. This can be mainly attributed to the similarities that exist in the roughness distributions in terms of patterns, as well as to the robustness of the chosen point distribution (only for Rotor 1, where the presence of a snubber feature affected the roughness formation at the area around it due to fouling, the point distribution was slightly adjusted). As a result, the derived point distribution was kept fixed during the measurement of all rotor blades performed in this work. It should be noted that the measurements on the selected locations were carried out using the accurate 20x lens.

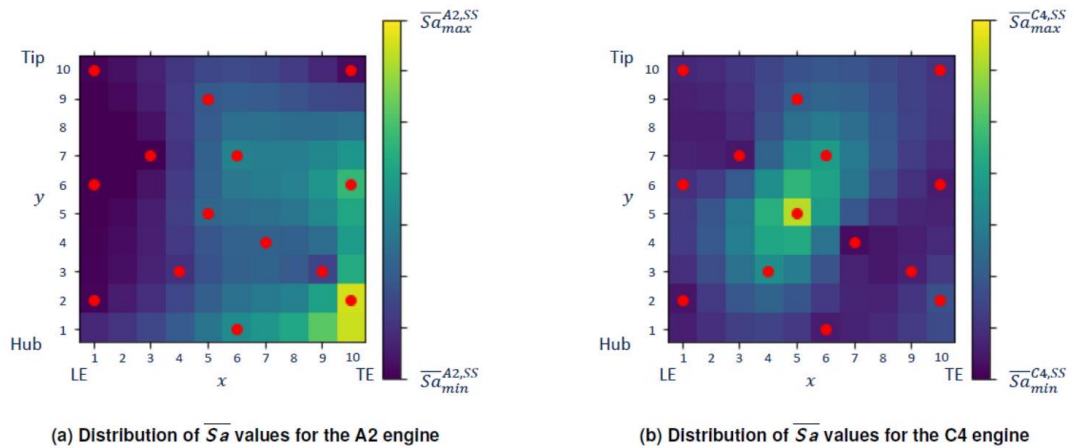


Abbildung 5.12: Approximate roughness distribution over the suction side of Rotor 2 blades. Red points indicate exact measurements used for RBF-based interpolation for the rest

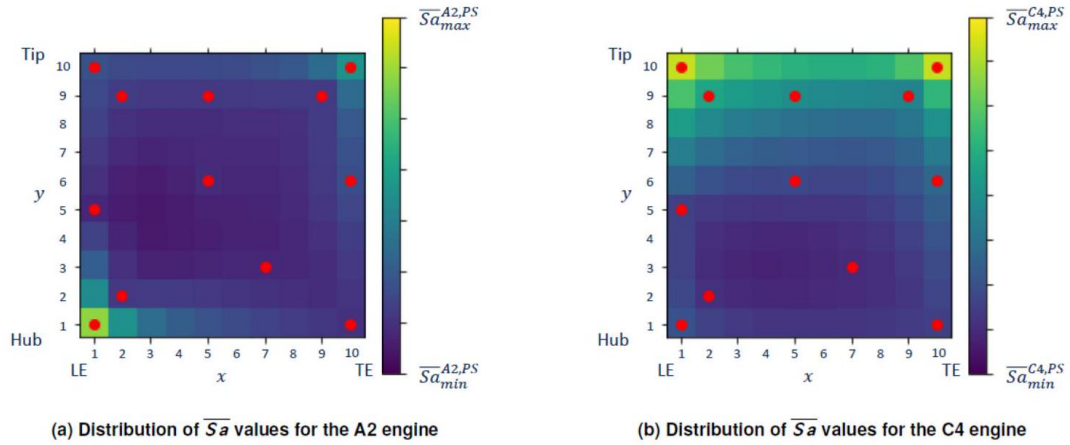


Abbildung 5.13: Approximate roughness distribution over the pressure side of Rotor 2 blades. Red points indicate exact measurements used for RBF-based interpolation for the rest

5.2.6 Summary of Roughness Measurements for Entire HPC

The roughness measurement results for the entire HPC of both engines are summarized in 5.14 and 5.15 (respectively for each side of the rotor blades). Rows 1 and 3 correspond to the measurements performed on the blades from the engines TW1 and TW2 as received from Rolls-Royce, whereas rows 2 and 4 refer to the corresponding measurements performed after cleaning of the blades. Each column contains the results for the specified rotor (starting from Rotor 1 on the left up to Rotor 10 on the right). For visualisation purposes, the surface of the blades was almost uniformly divided into 9

areas and the $\overline{Sa}$ values of the points corresponding to these areas were averaged, giving a single mean value to represent each area. The exact values are of less importance and probably also case-dependent (i.e. for the two specific engines addressed here); thus, they are omitted from the plots, where red indicates high and green low roughness values. The LE, Mid-Chord (MC) and TE areas correspond to 0 – 30% chord, 30 – 70% chord and 70 – 100% chord, respectively, while the Hub, Mid-Span (MS) and Tip areas correspond to 0 – 30% span, 30 – 70% span and 70 – 100% span, respectively.

The cleaning of all the blades was done using an ultrasonic bath device. The detergent Ardrex 6025A (primary cleaner in aero engine overhaul processes) was mixed with water at a 10% concentration and, after performing a few tests, the liquid temperature was adjusted to 80°. After the blades have stayed in the bath for 1h, any dirt is almost completely removed and the blades look bright and shiny (especially the ones from the front rotor stages made out of titanium).

Focusing on the results for the SS (Fig. 5.14) of the uncleaned blades, for both TW1 and TW2 engines, highest values appear at Rotor 1 (below the snubber and at the TE region, due to fouling), followed by Rotor 2 (high values from mid-chord onwards). For the TW1 engine, Rotors 3 and 4 show similarities, with highest values at the hub-TE area. From Rotor 5 up to Rotor 10, only hub sections are rough (due to sand deposits), with Rotor 8 having the highest values. For the TW2 engine, Rotors 3-10 show similarities (in terms of patterns), with highest values

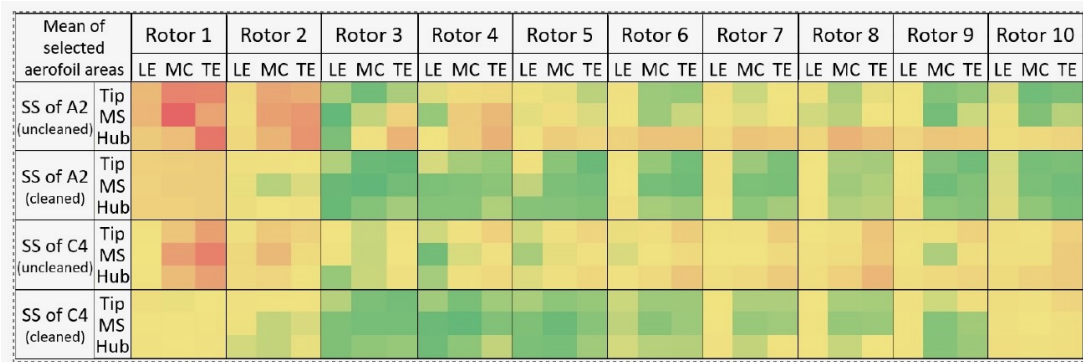


Abbildung 5.14: Mean roughness distribution over the suction side through the entire HPC for both engines (before and after cleaning)

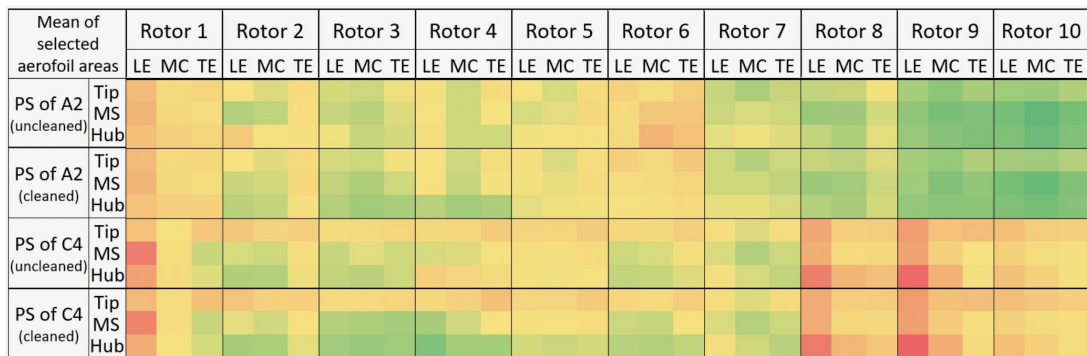


Abbildung 5.15: Mean roughness distribution over the pressure side through the entire HPC for both engines (before and after cleaning)

always at the TE region (especially for the hub and tip sections due to corner flow separations), where corrosion appears from Rotor 6 onwards - again Rotor 8 is the most rough. Overall, the front stages (Rotors 1-4) are more rough for the TW1 engine, whereas the back stages (Rotors 6-10) are more rough for the TW2 engine (more cycles). Finally, cleaning has a very big impact on the roughness of the SS, since it removes the dirt and corrosion which are the dominant components of roughness there. After cleaning, the results look more similar for the two engines. Rotors 1-5 have a lower uniform remaining roughness, with highest values appearing again at Rotor 1 (especially for the TW1 engine), followed by Rotor 2, whereas Rotors 6-10 have higher roughness at the LE region this time. Interestingly, for the TW2 engine, the cleaned Rotor 10 blades are as rough as the cleaned Rotor 1 blades.

Regarding the results for the PS (Fig. 5.15), cleaning has a negligible effect on the results (rows 2 and 4 almost identical to rows 1 and 3), since roughness there is dominated by impact damage and erosion. For the TW1 engine, highest values appear at the LE area of Rotor 1 (due to impact damage), Rotor 2 has highest values at the TE area, while Rotors 3-10 show highly uniform distributions, with the middle stages (Rotors 4-6) having the highest values. For the TW2 engine, highest values appear at the LE region of Rotor 1 as well as of Rotors 8-10 (due to erosion), with Rotor 9 being the most rough, while Rotors 2-7 have similar values with a progressive increase of roughness from hub to tip. Overall, the PS of the TW2 engine is more rough than the PS of the TW1 engine.

A comparison between the SS and PS plots leads to the following generic conclusion: For uncleaned engines, the SS is more rough than the PS at areas where dirt is commonly accumulated (e.g. hub-TE region), whereas for cleaned engines, the PS is overall more rough than the SS.

5.3 Roughness Parameterization

5.3.1 Roughness Stripe Measurements and Definition of Parameters

In this 2nd part of the work (Sections 5.3 and 5.4), the surface roughness distributions/geometries used for the simulations and modelling approach come from stripe measurements - as the one illustrated in Fig. 5.16 - performed on the suction side (since this side was found to be more interesting in terms of roughness) of the in-service HPC blades. The measurements were conducted using the highly accurate Alicona device in combination with the robotic arm responsible for continuously translating the blade under the sensor (from its leading to its trailing edge). A detailed presentation of the measuring system was already given in Section 5.2. The idea here is to use part of the measured stripes to construct the walls of zero-pressure-gradient, fully turbulent channel flow simulations (similar to [178]). For this, initially, 1mm × 10mm stripes were tested; however, it was decided to focus on the most rough part of the stripes (usually the trailing edge area) and form the channel walls by a repetition of such 1mm × 2mm rough patches. Finally, it should be noted that the roughness structures depicted in Fig. 5.16 are due to the accumulation of dirt on the blade, as for all the cases studied hereafter.

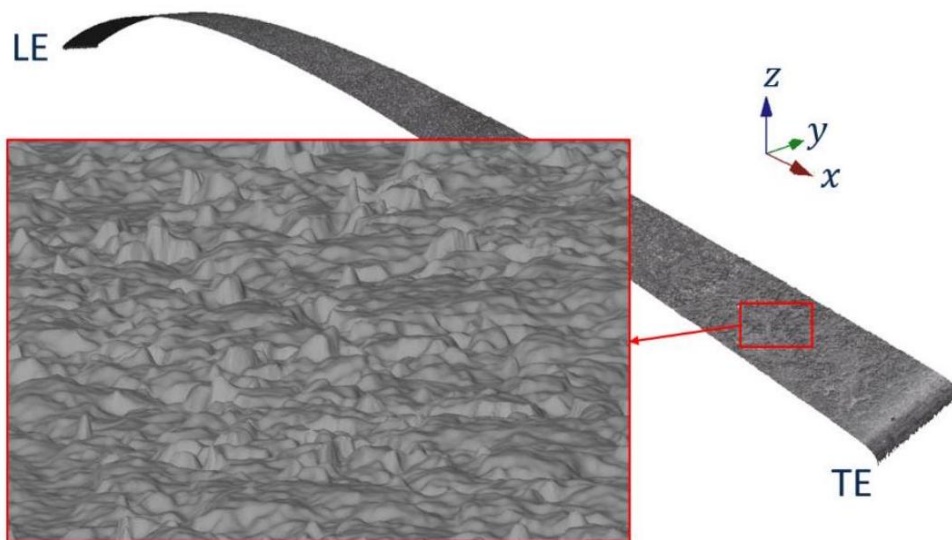


Abbildung 5.16: Roughness stripe measurement exemplary shown for the mid-span of a Rotor 2 blade

Regarding the parameterization of roughness, the following 8 parameters are defined and used hereafter:

$$Sa = \frac{1}{A} \int_A |z(x, y)| dA \quad (5.4)$$

$$Sq = \sqrt{\frac{1}{A} \int_A z(x, y)^2 dA} \quad (5.5)$$

$$Sp = \max(z(x, y)) \quad (5.6)$$

$$Sv = -\min(z(x, y)) \quad (5.7)$$

$$Sk = \frac{1}{ASq^3} \int_A z(x, y)^3 dA \quad (5.8)$$

$$Ku = \frac{1}{ASq^4} \int_A z(x, y)^4 dA \quad (5.9)$$

$$ES = \frac{1}{A} \int_A \left| \frac{dz(x, y)}{dx} \right| dA \quad (5.10)$$

$$AR = \frac{A}{A_p} - 1 \quad (5.11)$$

where Sa is the average roughness height, Sq the RMS height (i.e. the standard deviation of the roughness distribution), Sp and Sv the max peak height and valley depth respectively, Sk and Ku the skewness and kurtosis statistical parameters (i.e. the 3rd and 4th -order moment of the roughness distribution, respectively), ES the effective slope (i.e. the mean surface gradient along the x direction) and AR the total-to-projected area ratio. The z coordinate corresponds to the local roughness height defined by the distance from the zero plane $z = 0$. All these parameters (apart from the ES) are computed by the Alicona software and are readily available after the completion and evaluation of each measurement. The ES parameter is included and computed at a post-processing step, given its importance for accurately predicting the impact of roughness on the flow over a surface (as demonstrated by many researchers - e.g. in [110, 120]).

5.4 Simulation-based Modelling of Roughness

5.4.1 Literature Review

Surface roughness and its impact on fluid flows (regarding boundary layer formation, transition, loss generation, etc.) has been studied since many decades [81]; however, it still is a relatively unexplored field with many open questions remaining. In the detailed review from Bons [19], it is concluded that "accurate models for predicting roughness effects on profile losses and surface heat transfer are still beyond our present capability".

Significant progress has been made with the use of high-fidelity CFD models, which offer an alternative to the predominantly experimental investigations in the field of roughness modelling. There, the first step usually is to model the rough geometry, based on which a high-resolution CFD grid is then generated to perform simulations. Discrete 3D elements, such as rectangular bars [82], sawtooth or scalloped riblets [105], have been used to model regular surface roughness, whereas sinusoidal functions have been proposed for modelling irregular roughness [36]. All these researchers then perform Large Eddy Simulations (LES) to resolve the

flow around the modelled elements. In the present work, this first modelling step is avoided, by accurately measuring the real roughness geometry (recall Section 5.3).

The most suitable parameters to characterize and model roughness are yet to be identified. Usually the equivalent sand-grain roughness parameter k_s introduced by Nikuradse [124] is used, but the appropriate description may well depend on the application at hand, which further complicates this task. Despite the concerns regarding the universality of the k_s parameter, in terms of modelling the effect of roughness on the flow, several attempts have been made throughout the years to derive expressions for predicting k_s from other, easier to compute roughness parameters (such as the mean roughness, indicated by R_a or S_a). An extensive list of recommended expressions can be found in the summary from Bons [19]. More recent empirical and calibrated relations for predicting k_s , after the introduction of the skewness parameter S_k , are proposed by Flack et al. [51] and Forooghi et al. [53], for grit-blasted and numerically generated surfaces, respectively. The importance of the effective slope parameter ES is demonstrated by Meynet et al. [110], as well as in earlier contributions of Goodhand et al. [63], using a different definition though. Finally, Jouybari et al. [83] employed machine learning for training a prediction model for k_s with 17 input roughness parameters, yielding an average prediction error of 8%.

To ultimately model the impact of roughness on the flow over a surface, the roughness function approach is almost exclusively used. The downward shift of the mean velocity profile in the logarithmic region (caused by the roughness) is characterized by the roughness function ΔU^+ . Various different formulas linking ΔU^+ with k_s^+ can be found in the literature, such as the one based on the experiments of Nikuradse [124], the Colebrook formula [31] or the relations obtained by fitting the DNS data from Thakkar et al. [168] and Jouybari et al. [83]. An alternative was proposed by De Marchis et al. [37], where ΔU^+ is directly modelled (for both the transitional and fully rough regimes) based on primary roughness parameters, thus bypassing the need to consider k_s .

The present work is mostly based on the contributions from [83, 37] and nicely complements the two. Specifically, the elegant idea from [37] of modelling ΔU^+ directly is also adopted here, using more roughness parameters though, in the direction set by [83]. Moreover, a corresponding model for predicting k_s is derived for completeness. Contrary to both publications, measured (instead of artificial) roughness geometries are simulated and RANS is found to be a sufficiently accurate alternative to LES and DNS for modelling the impact of roughness.

5.4.2 CFD Grid Generation for Channel Flow Simulations

The body-fitted unstructured CFD grids of the channel flow simulations performed in this work were generated using the ANSA software [5]. Each measured rough geometry (1 mm × 2 mm patch, the form of which has already been removed by the Alicona software - i.e. the middle surface is $z = 0$) is exported from the Alicona software as an STL file and imported in ANSA. It should be noted that, even though the measured data-set is reduced when exported as an STL file, the selected resolution for the triangulation is fine enough to capture the roughness scales (recall Fig. 5.16). The required steps of the grid generation process are discussed in what follows.

First, a rectangular boundary is defined around the triangulation for projecting the boundary nodes, which is done in a smooth manner using the morphing capability available in ANSA. Then, to ensure periodic boundaries in both lateral y and axial/flow x directions, the boundary nodes are aligned as equidistant (in case of a mismatch in the number of nodes, the desired number can be prescribed and the boundaries are reconstructed accordingly). The surface CFD grid is obtained after a number of surface reconstruction steps, aiming at an elimination of the triangular elements (it is desired to have a structured-dominant grid comprised of quads) and an overall improvement of the grid quality. Regarding the quality criteria, the OpenFOAM strict criteria implemented in ANSA were selected (i.e. $\text{skewness}_{\max} < 0.45$, $\text{warping} < 40^\circ$, $\text{angle}_{\max}^{\text{quads}} = 0.45$, $\text{angle}_{\max}^{\text{trias}} = 0.7$). Information on the exact definition of these criteria can be found in the ANSA user guide. The final surface grid - an example of which is shown in Fig. 5.17 - consists of 90-96% quad elements. Even though the aforementioned operations have a non-negligible impact on the underlying geometry, the main roughness features are preserved.

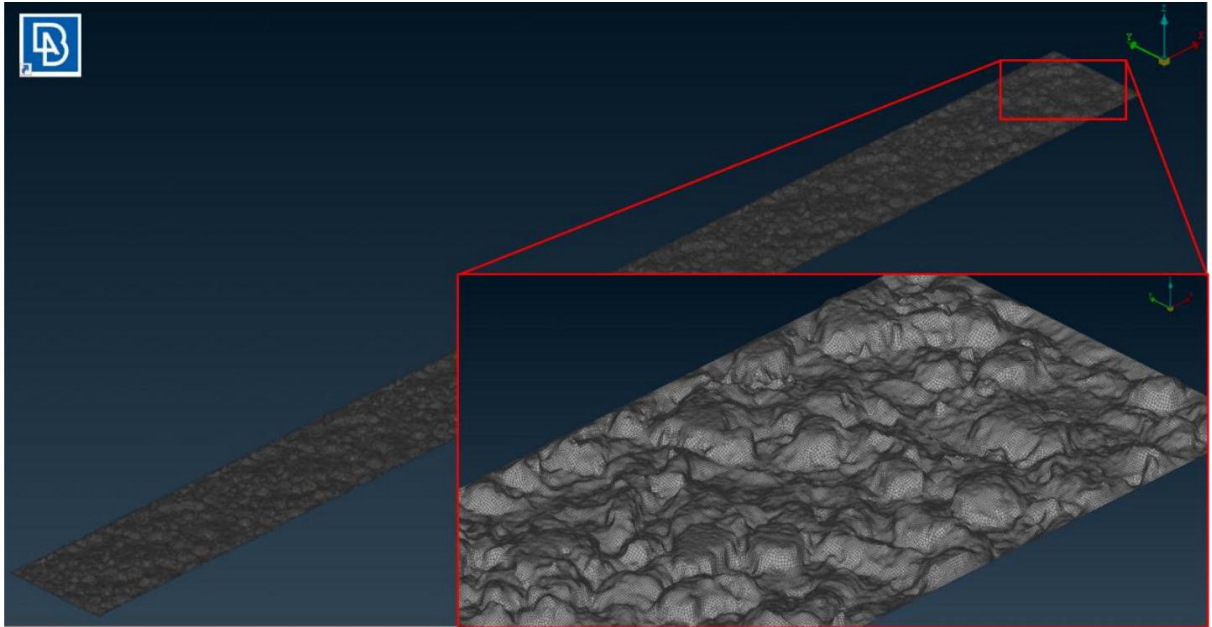


Abbildung 5.17: Example of CFD grid of rough bottom wall generated in ANSA

The volume grid is generated via a two-step extraction of the surface grid in the z direction. For the first step, 38 layers are distributed within a total height of $0.2h$, where h is the half-channel height, with a growth factor of 1.1, whereas for the rest of the half-channel domain, 40 layers are uniformly distributed. The top boundary nodes are aligned with the plane $z = h$, in order to form a symmetry plane which is treated as an inviscid wall during the simulation of the half-channel. To avoid the formation of negative volumes or bad quality hexas/pentas, multiple fix-quality and smoothing steps are performed, based again on the OpenFOAM strict quality criteria for volume cells (i.e. $\text{skewness} < 4$, $\text{warping} > 0.8$, $\text{non-orthogonality} < 60^\circ$).

In the relevant literature, to perform channel flow simulations, most authors generate a rather small CFD domain and impose periodicity also in the x direction, either by adding a body force source term in the Navier-Stokes momentum equations (to compensate the wall friction) [83] or via a Lagrangian recycling method [110]. These approaches have the advantage of keeping

the CFD domain as small as possible, which is particularly important for performing LES or DNS. In this work, however, the CFD solver used does not offer such options and, therefore, long channels had to be constructed to achieve fully-developed turbulent flows. Specifically, the volume domain generated on top of the measured geometry is repeated 5 times in the x direction, to match the desired total length $L = 60h$ of the channel. This leads to CFD domains of 50 – 80M nodes, depending on the case. The final grid density was determined after a simple grid refinement study (doubling the number of grid nodes had only a negligible impact on the quantities of interest). For the smooth channel case (used as a reference and to validate the CFD setup), an even longer channel ($L = 100h$, 80M nodes) had to be constructed to get a fully-developed flow at the outlet. Regarding the rough channels, roughness triggers an earlier transition - causing a faster development of the turbulent flow - and, thus, shorter channels can be used. Finally, it should be noted that the CFD grid of the rough channels was scaled up by a factor of 10, to match the desired channel dimensions and ensure at least a transitionally rough regime [178]. The scaling affects the roughness parameters defined in Eqs. (5.4)-(5.7) (which are re-computed, together with the rest of the parameters, based on the final surface grid); however, it has no impact on the modelling, where non-dimensional parameters (in wall units) are used.

The described CFD grid generation using ANSA unfortunately includes a lot of manual work and it was not possible to automate the process. Especially for cases where roughness is high, in order to generate a high-quality body-fitted grid, many working hours are required, as one has to iterate multiple times between improving the surface and the corresponding extracted volume grid, until a simultaneous good quality of both is reached. It would be much easier to generate Cartesian grids instead (as in [178]), however, the CFD solver used herein cannot run on such grids.

The volume CFD domain of the smooth channel case is exemplary illustrated in Fig. 5.18, where also the type of each boundary surface is annotated. This grid is exported from ANSA in Fluent (.msh) format and used for validating the CFD setup, as presented in the following section.

5.4.3 Validation of CFD Setup for Smooth Channel

In this work, all the CFD simulations were performed using the Rolls-Royce in-house HYDRA [98] suite of codes, which have been extensively validated and applied to various industrial cases [111, 150]. In its core, the non-linear flow solver uses a 2nd-order accurate vertex-centered finite-volume discretization method and either an explicit or an implicit Runge-Kutta pseudo-time-marching scheme, combined with a geometric multigrid technique for convergence acceleration. Details regarding the underlying theory and implemented algorithms are given in [116, 115]. For the Reynolds-Averaged Navier-Stokes (RANS) channel flow simulations, the steady compressible flow equations are solved together with the $k - \omega$ SST turbulence model equations [109]. The coupled system of equations is iteratively solved using the implicit HYDRA solver (without geometric multigrid). To obtain the inlet and outlet boundary conditions for the channel flow simulations, first, a simulation of the long smooth channel ($L = 100h$) was performed (to achieve a fully-developed turbulent flow at the outlet) at a Reynolds number based on the uniform inlet

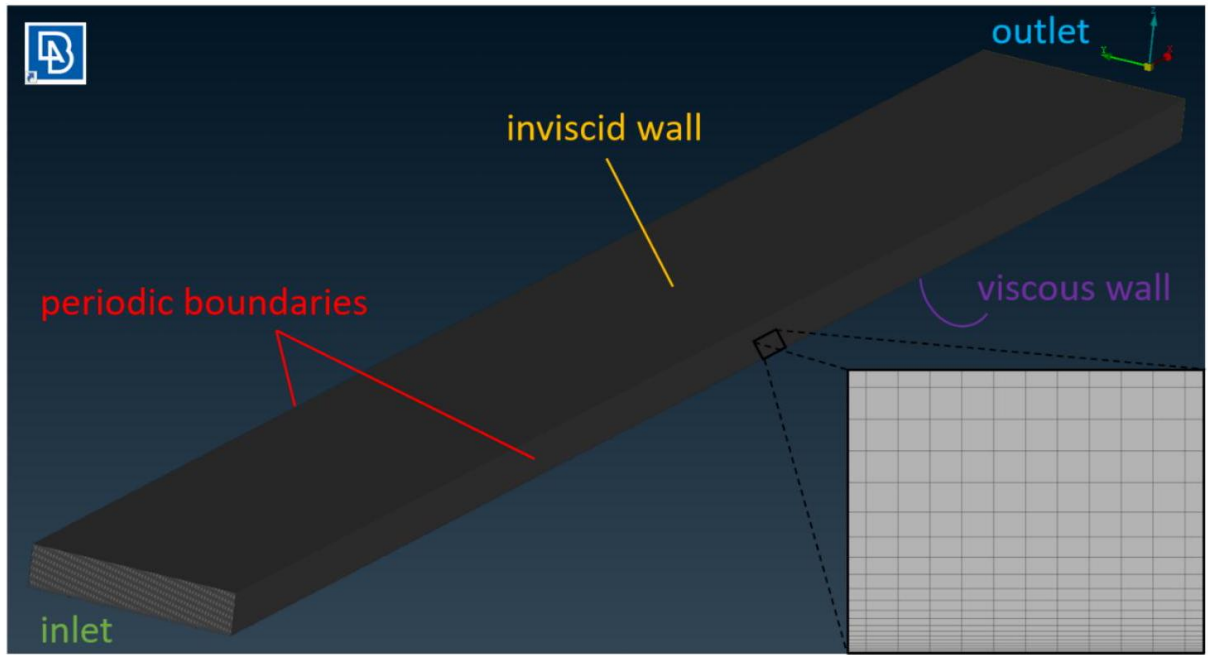


Abbildung 5.18: Volume CFD domain of smooth channel generated in ANSA

velocity U_0 and kinematic viscosity ν_0 of $Re = U_0 h / \nu_0 = 11000$ (similar to [71]). For this simulation, an inlet turbulence intensity of $Tu_i = 6\%$ is prescribed, together with a target mass flow rate (based on U_0) at the outlet. Then, the fully-developed profiles at the channel outlet are extracted and used as boundary conditions for all the subsequent simulations. Specifically, total pressure and temperature profiles as well as the turbulent kinetic energy k and specific dissipation rate ω profiles are prescribed at the inlet, whereas the static pressure profile (corresponding to the desired mass flow rate) is used as an outlet boundary condition.

The RANS simulation of the smooth channel case is performed on the same grid as the wall-resolved Large Eddy Simulation (LES). For the selected Reynolds number, the nondimensional wall distance z^+ of the first series of nodes is below 1 (i.e. there is no need to employ wall functions) and a uniform resolution with $\Delta x^+ = \Delta y^+ = 10$ is chosen for the other directions (typical values for LES, similar to the ones used in [110, 71]). This grid resolution is retained also for the rough channel cases. Regarding the convergence of the RANS simulations, for the long smooth channel case, the RMS residual drops 6 orders of magnitude within 5 hours of running on 1536 CPU cores (Intel® Xeon® E5-2680 v3 at 2.50 GHz), whereas the rough (shorter channel) cases require (on average) about 4 hours on 1152 CPUs for a 5 orders of magnitude residual reduction. The post-processing of the simulations' results is done in SS02; the Rolls-Royce in-house visualisation tool based on ParaView [131]. Of particular interest is the mean velocity profile along the channel height, which is normalized with the friction velocity $u_\tau = \sqrt{\tau_w / \rho}$ (with τ_w being the wall shear stress and ρ the fluid density at the wall). For the smooth channel case, the Reynolds number based on u_τ is $Re_\tau = u_\tau h / \nu = 540$, i.e. very similar to the one selected for the LES by Hinterberger et al. [71] and for the DNS by Moser et al. [117], whose results are used herein as a reference for validation purposes.

For the channel flow explicit LES performed in this work, the spatially-filtered unsteady compressible Navier-Stokes equations are solved, while modelling the sub-grid-scale stresses using

the Sigma model [123]. The Smagorinsky eddy-viscosity model implementation in HYDRA was tested as well; however, it was found to be less accurate than the Sigma model, as also claimed by the authors in [135]. Regarding the time integration scheme, the dual-time-stepping method implemented in HYDRA (same as used for classical URANS problems) is employed. This scheme has no stability limitation (in terms of CFL number) and allows the selection of relatively large physical time steps dt . Here, $dt = 0.01h/U_0$ for the LES, yielding an average CFL number of 6, which is still close to the lower bound of suggested values when using this method. A rather small dt was selected to ensure high accuracy in the boundary layer region. At each physical time step, the convergence of an inner iteration loop is required, for which the 2-3 orders of magnitude drop of the RMS residual (achieved here within 20 iterations) is considered satisfactory.

To initialize the LES, the converged flow solution of the corresponding RANS simulation is used, and the same inlet and outlet boundary conditions are imposed. Moreover, the LES solver allows the introduction of isotropic velocity fluctuations at the inlet plane, aiming at an artificial (also known as synthetic) turbulence generation, as proposed in [35]. For this, along with the intended $Tu_I = 6\%$ and turbulence length scale $l = h$, an average velocity has to be specified (here U_0 is used) for tuning the fluctuations amplitude. The simulation time for the LES was, first, 5 through-flow times $T = L/U_0$ to allow the flow to develop and leave the initial transient state, followed by another 5 through-flow times (adequate to reach statistical convergence), which were used for computing the mean flow. To obtain the mean velocity profile as well as the RMS velocity fluctuations, spatial averaging in the lateral direction y is performed at the outlet plane, where the flow is fully-developed. Finally, regarding the computational effort, one complete LES requires a wall-clock time of about 10 days when running on 3120 CPUs.

In Fig. 5.19, the RANS and LES results from HYDRA for the smooth channel case are validated against the reference results from the relevant literature. A comparison of the non-dimensional mean velocity profile $u^+ = u/u_\tau = f(z^+ = zu_\tau/\nu)$ from each solver is shown in the usual semi-logarithmic plot of Fig. 5.19a. The HYDRA LES results are in very good agreement with the reference data, and the RANS results - even though slightly less accurate (e.g. in the buffer layer) - are still in close agreement. The viscous sublayer relation $u^+ = z^+$ as well as the empirically derived wall-law equation proposed by Spalding [157] (usually employed by wall function models) are also plotted and match the results very well. In Fig. 5.19b, the distributions of the non-dimensional RMS lateral velocity fluctuation $v'^+ = \sqrt{v'v'}/u_\tau$ and shear stress $\overline{u'w''}^+ = \tau_{xz}/(\rho u_\tau^2)$ (derived from the corresponding components of the resolved Reynolds stress tensor) along the normalized channel height z/h are compared for the different unsteady solvers. The HYDRA LES results are again in good agreement with the reference ones. The difference in v'^+ between the LES and DNS near $z = h$ is attributed to the boundary condition imposed there, which for the LES is a nopenetration condition (inviscid top boundary) damping the normal fluctuations and increasing the ones parallel to the plane, whereas there is no such restriction for the DNS, since the entire channel (i.e. up to $z = 2h$) was simulated in [117]. After this validation, the presented CFD setup was considered accurate enough to perform the following rough channel flow simulations.

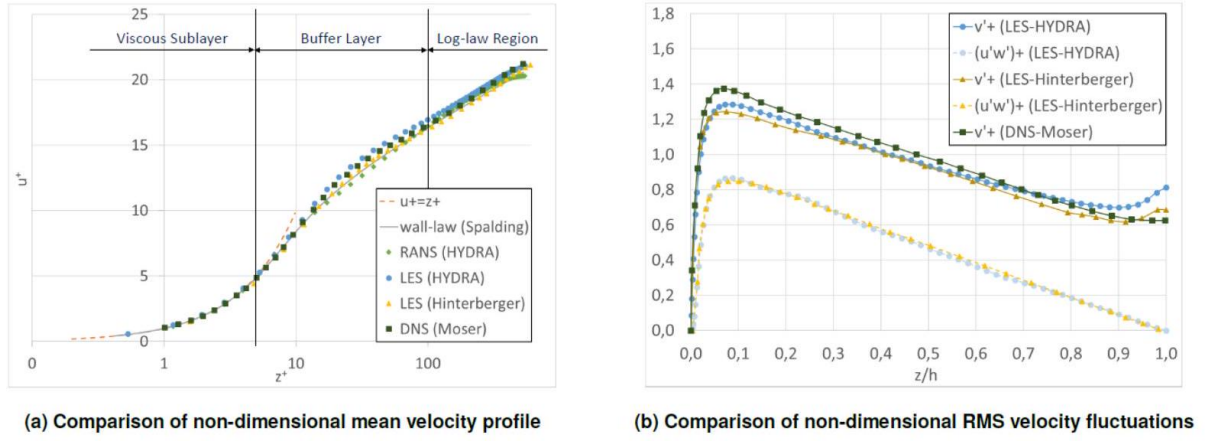


Abbildung 5.19: Validation of HYDRA LES of smooth channel flow ($Re_{\tau} = 540$) against reference LES and DNS data from the literature [46, 48]

5.4.4 Comparison between RANS and Wall-Resolved LES of Rough Channels

The CFD results obtained from the simulation of two rough channel flow test-cases are presented in Fig. 5.20. The impact of roughness is visualized by the downward shift of the non-dimensional mean velocity profile (focusing on the log-law region, where the shift is usually nearly constant) and is commonly characterized by the ΔU^+ value. This is computed as the difference in the u^+ value at the center of the channel (here $z = h$) between the smooth (reference data included in the plots) and the rough channel cases. As long as there is a uniform offset of the log-law region by ΔU^+ , the latter can also be evaluated as the difference at any selected effective height in that region [110]. The reason behind this decrease in u^+ for the rough channel cases is mainly attributed to the increase of the corresponding u_{τ} , the definition of which has to be adjusted for the rough cases - as shown in Eq. (5.12) - in order to include the contributions of the pressure acting on the roughness elements in the total axial stress computation at the wall [36].

$$u_{\tau} = \sqrt{\frac{1}{\rho A_w} \int_{A_w} (\tau_w^x - p n_x) dA_w} \quad (5.12)$$

In Eq. (5.13), the integration is applied to the last 1/6 of the channel bottom wall (corresponding to the last repetition of the measured rough patch with area A_w), where τ_w^x is the axial component of the viscous wall shear stress, p the static pressure, n_x the axial component of the vector normal to dA_w and the average fluid density over A_w is used as ρ . For the LES, the mean flow quantities are employed in Eq. (5.13) for computing u_{τ} . Alternatively, it is also possible to obtain u_{τ} by applying the conservation law of momentum, as done in [110].

The RANS and LES results from HYDRA for the two rough channel cases demonstrated in Fig. 5.20 are similar. Differences exist in the exact non-dimensional mean velocity profile (especially for case-A), however, the computed ΔU^+ values are almost identical. This was observed for a total of 3 cases in this work and, since ΔU^+ is adequate to characterize the effect of roughness, the RANS solver is considered accurate enough and was selected for performing all the rough channel flow simulations required for the modelling purposes hereafter. The ΔU^+ can be used

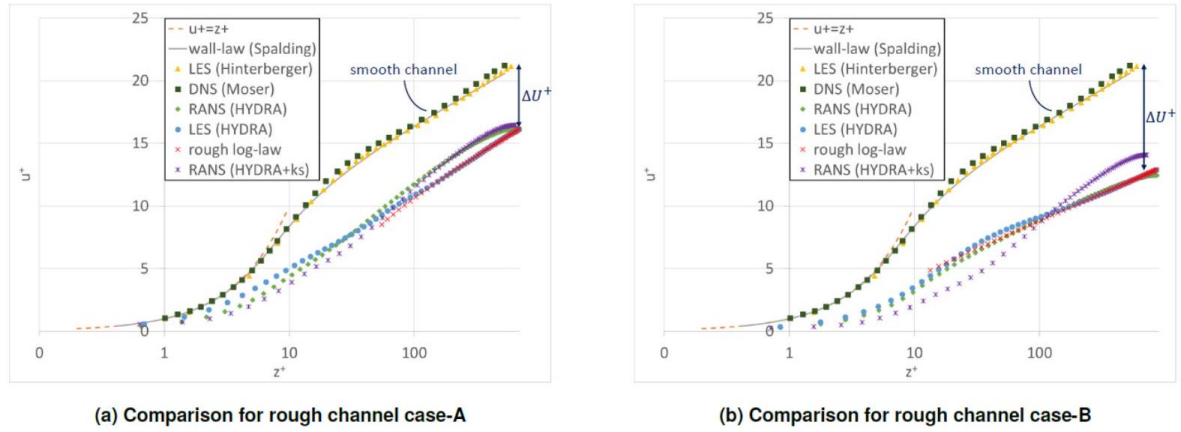


Abbildung 5.20: Non-dimensional mean velocity profiles for HYDRA RANS and LES of two rough channel flow test-cases ($Re_\tau = 540$). Results of RANS with HYDRA roughness function implementation are also shown

to obtain the corresponding k_s^+ value, for which many alternative relations are proposed in the literature, such as the one based on the experiments from Nikuradse [124]. In this work, the latter relation is employed. After the k_s^+ value is known, one can derive a rough log-law (following the work of Jouybari et al. [83]), as shown in Fig. 5.20 for both cases, by fitting Eq. (5.13) on the LES results.

$$u^+ = \frac{1}{\kappa} \ln \left(\frac{z-d}{k_s} \right) + \Gamma \quad (5.13)$$

In Eq. (5.13), d is the zero-plane displacement, computed as explained in [80]. The von Kármán constant κ and offset constant Γ are: $\kappa = 0.38, \Gamma = 8.8$ for case-A and $\kappa = 0.53, \Gamma = 9.5$ for case-B. A successful calibration of these constants would then lead to a generic rough log-law which could be implemented in a CFD solver (as is done in ANSYS CFX [6]) to replace the wall-law equation and simulate the impact of roughness, while running on a smooth grid. The disadvantage of this approach is, though, that one would be forced to use wall functions. An alternative approach is adopted in HYDRA, where a roughness function model implementation is already available, as briefly discussed in subsection 5.4.1 The results from this implementation (RANS solution on the smooth channel using a roughness function with k_s as input) are also plotted in Fig. 5.20. For case-A, the results match very well with the RANS results from the rough channel simulation, whereas for case-B, the comparison is less satisfactory, meaning that there is potentially room for improvement in the current HYDRA implementation.

Roughness Function Model in HYDRA

The current implementation of the roughness function model in HYDRA is based on the work of Aupoix [7], where a detailed description can be found. The idea is to model the impact of roughness via an artificial increase of the turbulent viscosity at the wall. So, for a given input k_s value indicative of the surface roughness to be modelled, the corresponding ΔU^+ value is found, which - in case of the $k - \omega$ turbulence model for example - is used to modify the boundary

condition at the wall and decrease ω . How to find the right k_s is certainly not an easy task and the main aim of this chapter.

Herein, the performed rough channel flow simulations belong to the transitional and fully rough regimes, represented by the following equation from Nikuradse, which is used to obtain k_s^+ from a given ΔU^+ , and vice versa.

$$\Delta U^+ = \frac{1}{\kappa} \ln k_s^+ + C - B \quad \kappa = 0.4 \quad C = 5.5 \quad (5.14)$$

with

$$\begin{aligned} B &= 11.5 - 0.7 \ln k_s^+ \quad , 14 < k_s^+ < 68 \quad (\text{transitionally rough}) \\ B &= 8.48 \quad , 68 < k_s^+ \quad (\text{fully rough}) \end{aligned}$$

5.4.5 Database Construction for Modelling

The database constructed for training the roughness function models presented in this work comprises of the 14 roughness geometries illustrated in Fig. 5.21, where the colouring corresponds to the local height z (separate scaling used for better visualization). These geometries originate from measurements performed on the suction side of the in-service HPC blades (see Section 5.2 for details) and the diversity present at the distributions is attributed to the different type of deposits (e.g. dirt, sand, etc.) on different areas of the blades. As already mentioned, each measured geometry of Fig. 5.21 is repeated 5 times in the flow direction to construct the rough bottom wall of the corresponding channel.

Impact of Reynolds Number

To investigate the impact that Reynolds number has on the rough channel flow simulations, the initially selected $Re_\tau = 540$ was increased to $Re_\tau = 880$ and both setups were tested on a total of 10 rough channel cases. Only RANS simulations were performed for the higher Re_τ , for which the CFD grid had to be refined in the z direction, so that the z^+ of the first layer of nodes is still below 1. Moreover, the inlet and outlet boundary conditions corresponding to a fully-developed turbulent flow for $Re_\tau = 880$ had to be derived, which were obtained (as for the case of $Re_\tau = 540$) after running the prerequisite smooth channel simulation. The convergence of the simulations for the higher Re_τ is slightly deteriorated, with an RMS residual drop of 3 orders of magnitude for the worst case examined, which then enters limit

cycle oscillations (indication of unresolved unsteadiness in the flow). However, the corresponding oscillations in the quantities of interest are still negligible and, therefore, there is no real need of performing an unsteady simulation. Even though no LES is employed to verify the accuracy of the RANS simulations for $Re_\tau = 880$, based on the good agreement of the results for $Re_\tau = 540$, no major discrepancies are expected.

The non-dimensional mean velocity profiles corresponding to the first 6 cases of Fig. 5.21 are plotted for the two different Reynolds numbers in Fig. 5.22. The reference profile (for the smooth channel) is independent of the Re_τ , whereas the profiles for the rough channels

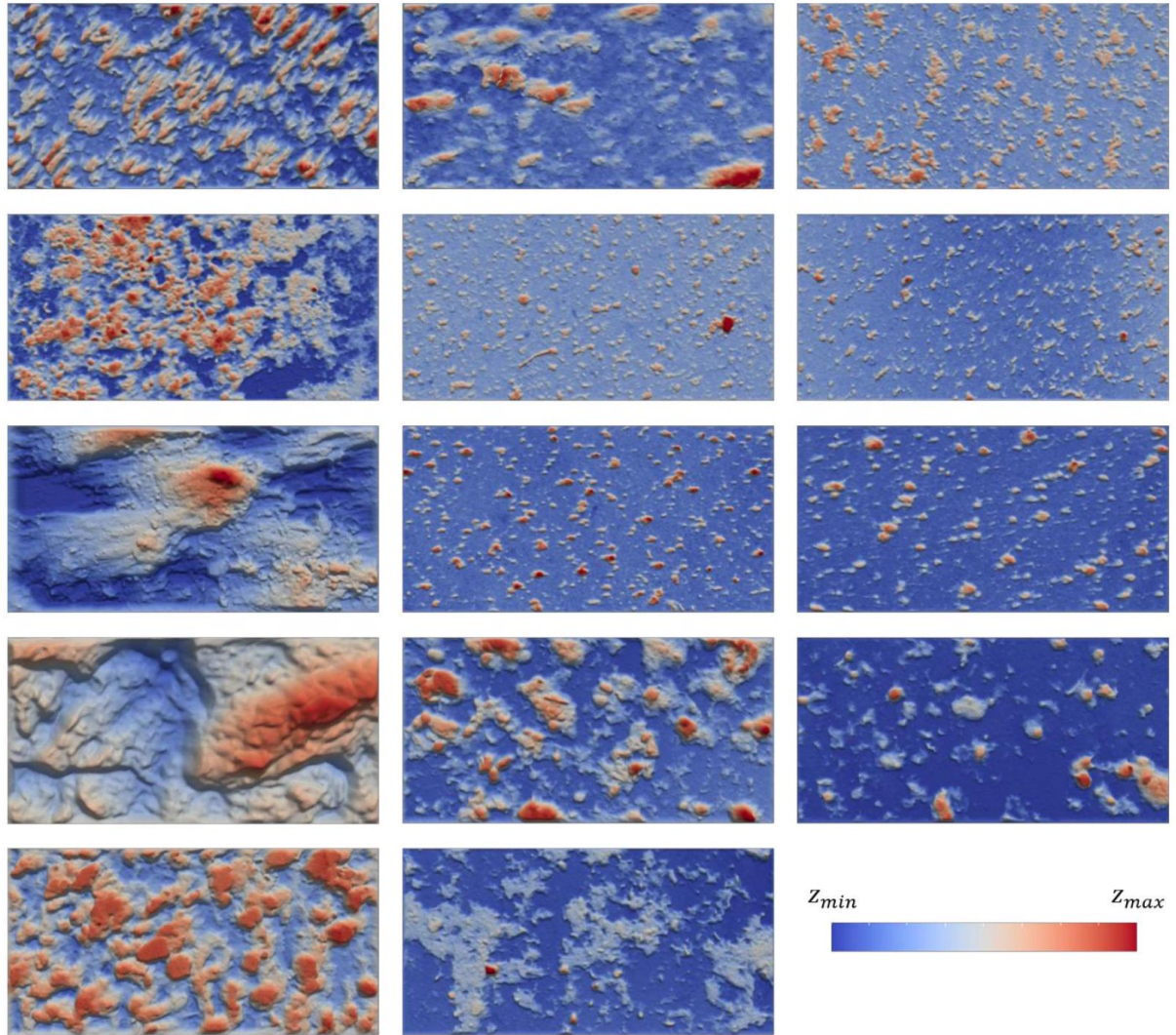


Abbildung 5.21: Database of the 14 measured roughness geometries used for training the roughness function model

move in the expected direction (as also shown in [37]) when increasing the Reynolds number. Finally, the question to be answered is which Re_τ to use for performing the simulations upon which the roughness function model will be based, since the choice is likely to have an impact on the final result. It was decided to select the higher Reynolds number, as more cases would then belong to the fully rough regime, which is more suitable for modelling purposes [83].

5.4.6 New Roughness Function Model

The results of the 14 rough channel flow simulations at $Re_\tau = 880$ are summarized in Fig. 5.23, where each roughness geometry is represented by the 8 roughness parameters defined in Eqs. (5.4)-(5.11) and its impact on the flow is quantified by the ΔU^+ . The idea here is to derive a roughness function model that predicts ΔU^+ from the roughness parameters (correlations of the former with the latter are shown in the last row of Fig. 5.23). To do so, given the relatively limited database at hand, the most important roughness parameters need to be identified. Moreover,

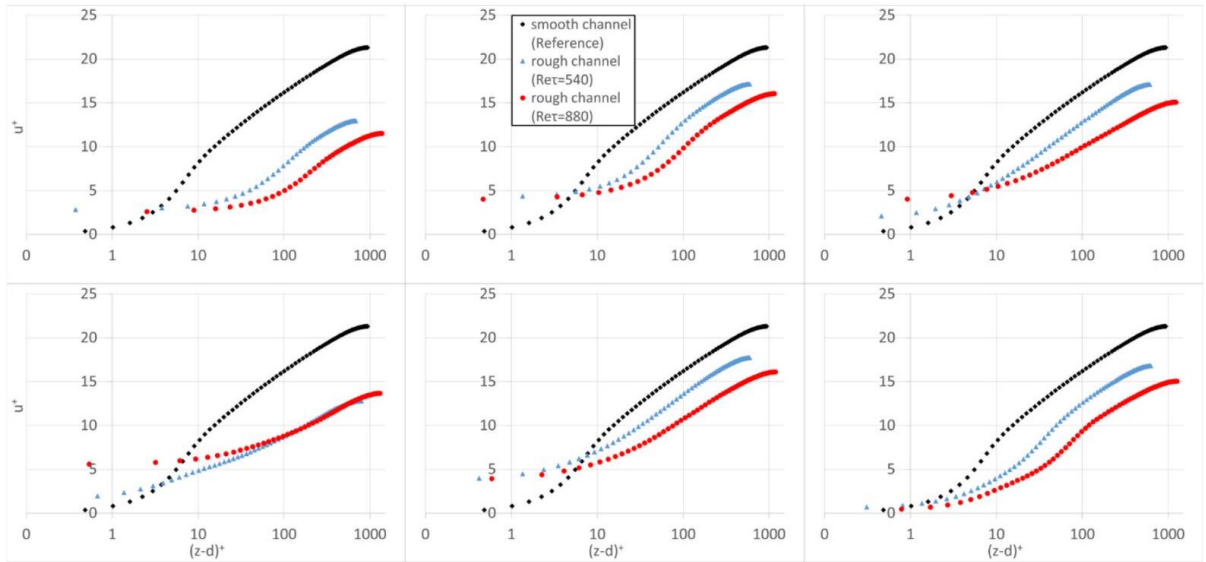


Abbildung 5.22: Non-dimensional mean velocity profiles corresponding to the first 6 cases of Fig. 4.21 for two different Reynolds numbers

since some of the roughness parameters are strongly correlated to others (e.g. Sa with Sq - Fig. 5.23), it is not recommended to use all of them for building a prediction model. In what follows, first, only two parameters; namely the Sq^+ and ES are selected as input parameters for the model (following the work of De Marchis et al. [37]), whereas eventually a new higher fidelity model based on 5 of the roughness parameters (Sq^+ , S_k , K_u , ES and AR) is derived.

Model based on Two Parameters

As proposed by De Marchis et al. [37], the product of the Sq^+ and ES parameters exhibits a very strong correlation with ΔU^+ , which was also observed for the data generated in this work, as shown in Fig. 5.24. By fitting a linear curve on the point-cloud of the plot (using a semi-logarithmic scale), the following roughness function model is obtained:

$$\Delta U^+ = \frac{1}{K} \ln (ES \cdot Sq^+) + 2.94 \quad (5.15)$$

which is almost identical to the one proposed in [37], with a slightly different offset constant (2.94 instead of 3.5). This surprisingly close agreement increases the confidence in our results (obtained by RANS) and suggests that the model of Eq. (5.15) - despite its simplicity - is a relatively robust model, which is also independent of the Reynolds number [37].

One can go a step further and also derive the corresponding relation for predicting k_s in the fully rough regime. This is done by comparing Eq. (5.15) with Eq. (5.14) (for the fully rough regime), which leads to:

$$k_s = 10.7 ES \cdot Sq \quad (5.16)$$

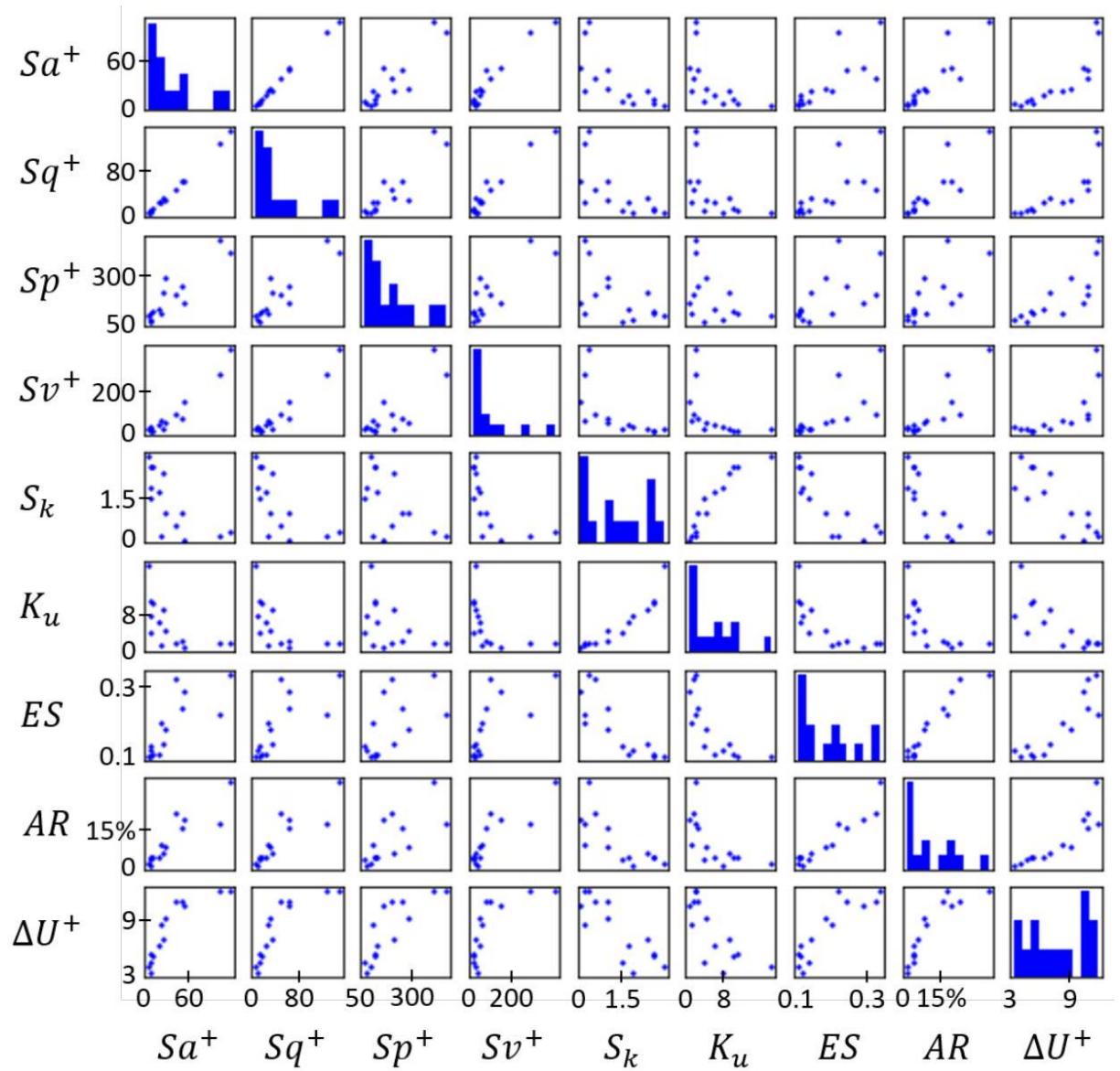


Abbildung 5.23: Scatter plots of roughness parameters (input) and ΔU^+ (output) corresponding to the database of Fig. 4.21

Even though k_s is only indirectly modelled in this manner (via a direct modelling of ΔU^+), the above relation should be more accurate than most of the commonly used empirical and/or calibrated expressions found in the literature [19].

Model based on Five Parameters

Herein, an improvement to the model presented in the previous subsection was attempted, aiming at a higher fidelity roughness function model with more roughness parameters as input. Since it is desired to simultaneously model both the transitional and fully rough regimes, first, an approximate relation between ΔU^+ and Nikuradse's k_s^+ (based on the data generated in this work) is derived, covering both regimes and obtained from Fig. 5.25:

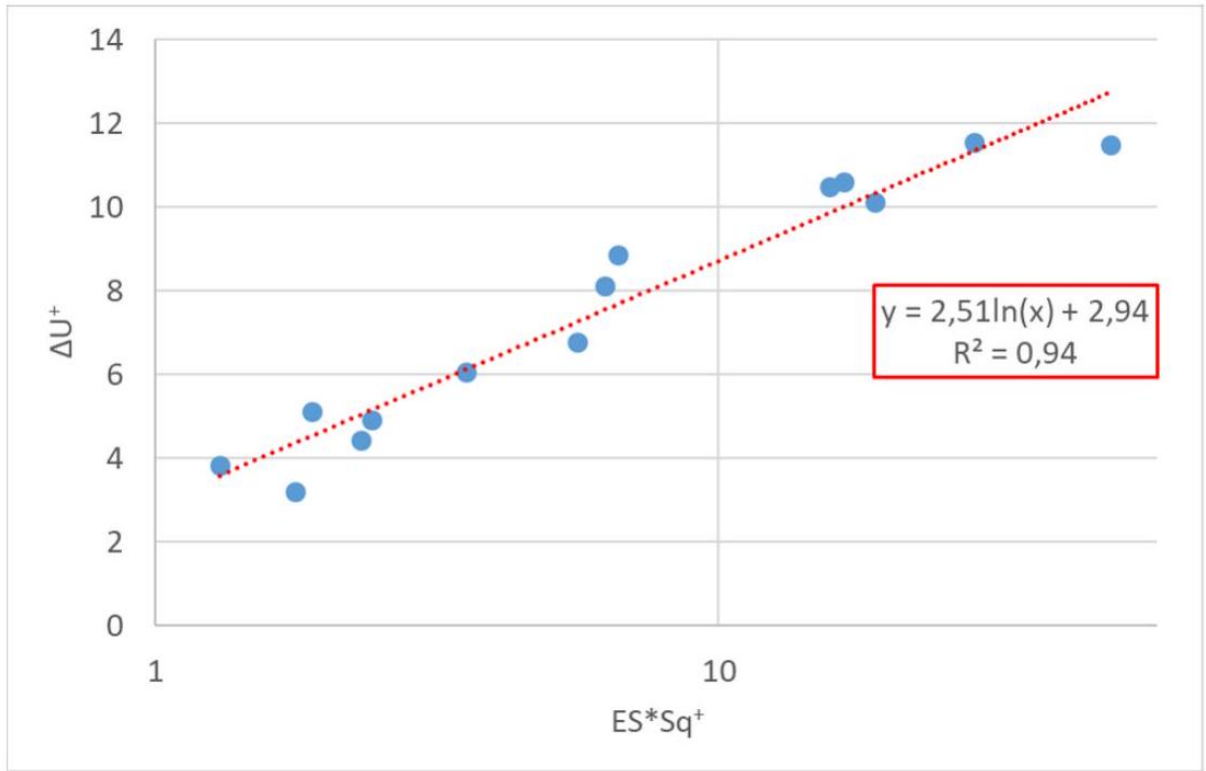


Abbildung 5.24: Prediction of ΔU^+ using a model based on two roughness parameters

$$\Delta U^+ = \frac{1}{\kappa^*} \ln k_s^+ - 4.71 \quad (5.17)$$

with $\kappa^* = 0.35$. Equation (5.17) is very similar to the one derived by Thakkar et al. [168] for describing the fully rough regime. Then, the following roughness function model is assumed:

$$\Delta U^+ - \frac{1}{\kappa^*} \ln Sq^+ = c_0 + \frac{c_1}{\kappa^*} \ln ES + \frac{c_2}{\kappa^*} \ln AR + \frac{c_3}{\kappa^*} \ln S_k + \frac{c_4}{\kappa^*} \ln K_u$$

the coefficients of which are obtained by means of a least squares linear regression to the data, giving: $c_0 = 2.77, c_1 = 0.8, c_2 = -0.12, c_3 = 0.06, c_4 = -0.13$. As a result, the proposed model is written:

$$\Delta U^+ = \frac{1}{\kappa^*} \ln \left(\frac{Sq^+ ES^{0.8} S_k^{0.06}}{AR^{0.12} K_u^{0.13}} \right) + 2.77 = \frac{1}{\kappa^*} \ln k_{vas}^+ - 2.77 \quad (5.18)$$

The prediction accuracy of this new model is depicted in Fig. 5.26, from which, after comparing it with Fig. 5.24, it can be seen that the coefficient of determination R^2 remains the same, regardless of the model used. This indicates that the inclusion of more roughness parameters (five instead of two) in the modelling (at least for the approach and database tested here) did not offer an additional improvement, and it can be concluded that the new model of Eq. (5.18) and the bi-parametric model of Eq. (5.15) are equivalent in terms of prediction quality. Finally, the new proposed relation for computing k_s is derived by comparing Eqs. (5.17) and (5.18), which leads to:

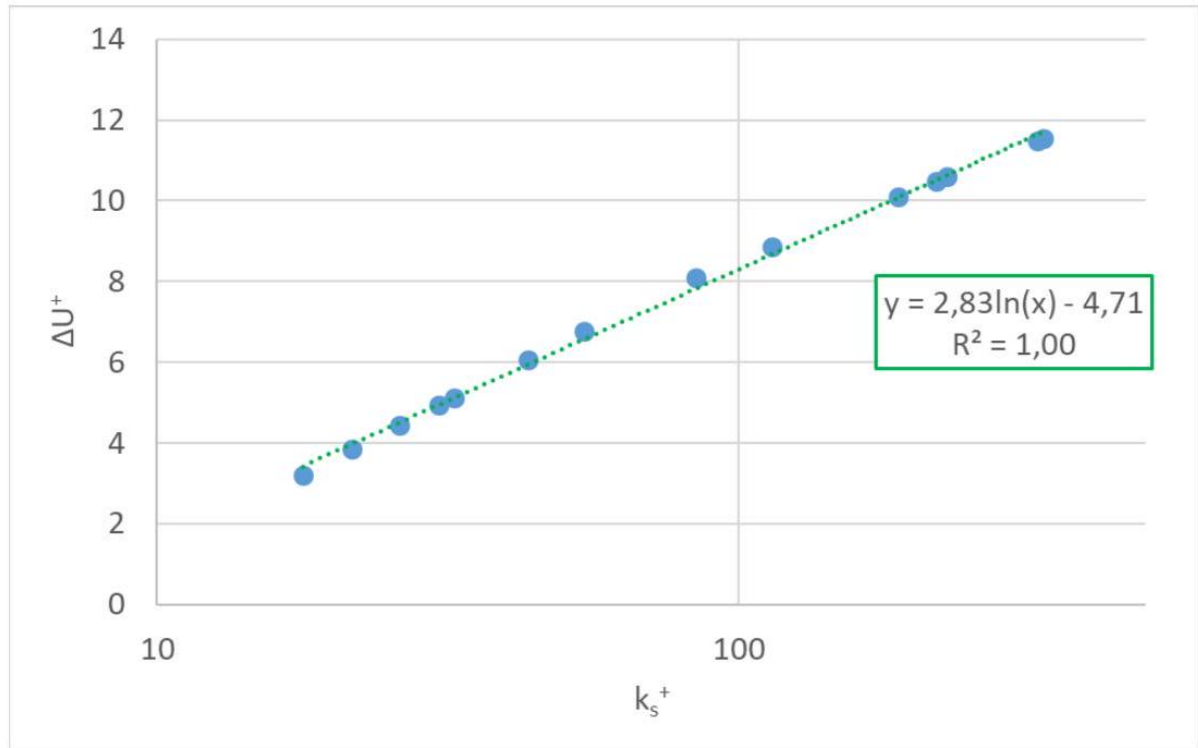


Abbildung 5.25: Approximate relation between ΔU^+ and Nikuradse's k_s^+ covering both transitional and fully rough regimes

$$k_s = 14 \frac{SqES^{0.8}S_k^{0.06}}{AR^{0.12}K_u^{0.13}} \quad (5.19)$$

Equation (5.19), where AR is in %, is theoretically valid for both transitional and fully rough regimes, however, it should be applied only to roughness geometries characterized by a positive skewness ($S_k > 0$), since this was the case for all the data used to train the model, which is not guaranteed to perform well also for cases with $S_k < 0$. The quality of this approximation model for k_s can be quantified by its mean prediction error, computed here as the difference between the predicted values from Eq. (5.19) and the actual Nikuradse's k_s values from Eq. (5.14), which is equal to 21% for the training data-set. This might seem as a relatively large error, however, when compared with error results coming from expressions proposed in the literature for computing k_s [29,30], it can be seen that this is not the case. For example, even the 30 degrees-of-freedom polynomial model derived in [83] gave a mean prediction error of 12% (indicating that an accurate prediction of k_s is a very challenging task). As a result, the prediction quality of the relatively simple k_s model derived in this work (via a direct modelling of ΔU^+) is deemed acceptable, also given the beneficial fact that the roughness parameters used in Eq. (5.19) can be directly obtained by performing a typical surface roughness measurement.

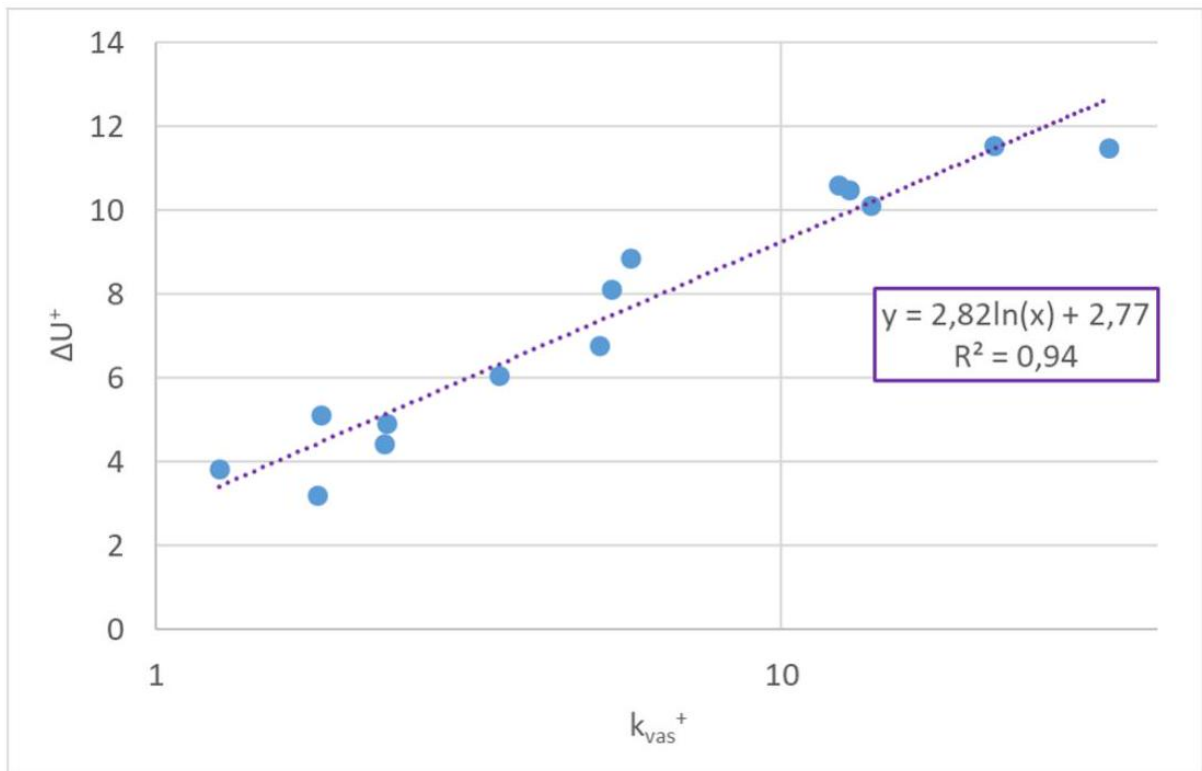


Abbildung 5.26: Prediction of ΔU^+ using a new model based on five roughness parameters

5.5 Summary and Conclusions (AP2.3)

In the 1st part of this work (presented in Section 5.2), an automated, efficient and accurate process has been developed for performing surface roughness measurements of HPC blades. The process was applied for the measurement of 100 rotor blades in total, coming from the 10-stage HPC of two in-service Rolls-Royce engines. The obtained roughness distributions over the SS and PS of the blades, through the entire HPC, offer valuable information regarding the formation as well as the nature of roughness, which can be used as a stepping stone for arriving to some generic conclusions. Finally, the impact of cleaning the blades was also investigated. The corresponding work was published in [174].

From the roughness measurement campaign completed in Section 5.2, the following potentially generic conclusions can be drawn:

- The roughness on the SS of the blades is attributed to fouling, mainly at the areas where corner flow separation occurs (i.e. hub and tip TE regions), especially for the front rotor stages (as expected). Corrosion may also form at these same areas of the blades for the back rotor stages, which are made of Nickel-based alloys.
- The roughness on the PS of the blades is attributed to a combination of erosion and impact damage, mainly expected at the LE region of the blades for the front stages, as well as for the back stages sometimes.

- The cleaning of the blades has a significant impact on the SS roughness and a negligible one on the PS roughness (expected, given the respective roughness type). The remaining/permanent roughness on the SS is overall lower than on the PS.

Even though the development of the measuring process was a time-consuming task (particularly due to the required training of the robot), conducting further roughness measurements of HPC blades from engines of the same type is now very straightforward and requires only minimal user interaction. Moreover, a second process has been developed for the measurement of the HPC stator vanes (details not included in this report).

In the 2nd part of this work (presented in Sections 5.3 and 5.4), the CFD-based modelling of the impact of roughness on turbulent flows was addressed. For this, real roughness geometries obtained from measurements conducted on in-service compressor blades are used as a basis to construct turbulent rough channel flow simulations. High-quality body-fitted CFD grids which are fine enough to capture the roughness scales are generated on top of the measured geometries and wall-resolved LES as well as RANS

simulations are performed. After validating the CFD setup, it was found that the RANS solver with the $k - \omega$ SST turbulence model provides adequate accuracy for modelling the effect of roughness, and was therefore selected (given its much smaller computational cost compared to the LES) for performing the simulations. Finally, two roughness function modelling attempts were made, based on a relatively small database comprised of 14 rough channels. The first model uses the RMS roughness and effective slope as input parameters and is in good agreement with the model proposed in [37]. The second model is based on 5 roughness parameters and offers a new expression for computing k_s , with promising accuracy. The corresponding work was published in [172].

In terms of future work, it is planned to implement the proposed roughness function model into the HYDRA RANS solver and compare its performance against the one of the current model. Moreover, it is of interest to investigate the impact of roughness on flows characterized by negative pressure-gradients (e.g. using divergent channels), which are more representative of the flow in a compressor, and for which only limited research is currently available.

6 Teilbericht AP3.1

6.1 Einführung

In AP3.1 wurden die Anforderungen an ein gemeinsames Daten- und Metadaten-Repository für die Performanceanalyse von Hydra-Simulationen definiert und umgesetzt.

- *AP3.1.1* Anforderungsdefinition für das gemeinsame Daten- und Metadaten-Repository für alle datengetriebenen Arbeitspakete
- *AP3.1.2* Offline- und Online Sammlung und Auswertung von Performance-Daten und Metadaten von Hydra-Simulationen
- *AP3.1.3* Aufsetzen eines gemeinsamen Daten- und Metadaten-Repositorys und geeignete Zugriffs-Schnittstellen

Dieses Kapitel stellt zunächst die zugrunde liegenden Konzepte (RDF, SPARQL) vor (Abschnitt 6.2), leitet daraus die Anforderungen ab (Abschnitt 6.3), beschreibt die realisierte Infrastruktur einschließlich Backends und scorep-db (Abschnitt 6.4), und schließt mit einer Zusammenfassung und einem Ausblick (Abschnitt 6.5).

6.2 Ausgewählte Grundlagen

6.2.1 RDF Grundlagen

RDF (Resource Description Framework) [32] ist ein vom W3C standardisiertes Modell zur Beschreibung von Informationen in strukturierter, maschinenlesbarer Form. Es bildet die Grundlage für viele semantische Technologien und wird eingesetzt, um Daten und deren Beziehungen zueinander formal abzubilden.

Das zentrale Konzept von RDF ist die Darstellung von Wissen in Form von Tripeln, bestehend aus Subjekt, Prädikat und Objekt. Jedes Tripel stellt eine Aussage dar, z.B.: „*JW*“ ist eine Person und heißt mit vollem Namen James Wales. Ein Beispiel zeigt Listing 6.1. Diese Darstellung ermöglicht es, Daten flexibel zu modellieren, zu erweitern und zu verknüpfen – unabhängig von einer festen Schema- oder Tabellenstruktur wie in klassischen relationalen Datenbanken und kann als Graphstruktur (Triplestore) interpretiert werden.

RDF-Daten können in verschiedenen Serialisierungen gespeichert werden, darunter Turtle und JSON-LD, letzteres erlaubt eine Integration in JSON-basierte Systeme.

Im Kontext dieser Arbeit wurde RDF verwendet, um Metadaten zu Konfigurations-, Laufzeit- und Analyseinformationen formal zu erfassen. Die semantische Struktur erlaubt dabei:

- eine eindeutige Identifikation und Beschreibung von Ressourcen (z.B. Simulationen, Messpunkten, Konfigurationsparametern)
- eine automatische Verknüpfung verwandter Informationen über IRI-basierte Referenzen,
- sowie eine anschließende Auswertung mittels SPARQL.

RDF bildet damit die semantische Grundlage für interoperable Analysen, datengetriebene Automatisierung und eine nachhaltige Wiederverwendbarkeit der erhobenen Messdaten in verteilten und heterogenen HPC-Umgebungen.

Die erzeugten Datensätze sind als Linked Data zu verstehen. Der Begriff ‚Open‘ wird hier nicht im Sinne öffentlich zugänglicher Daten verwendet; der Zugriff ist kuratiert, die FAIR-Prinzipien [179] werden dennoch adressiert.

```

1 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
2 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
3 @prefix foaf: <http://xmlns.com/foaf/0.1/> .
4
5 <#JW> a          foaf:Person ;
6     foaf:name     "James Wales" ;
7     foaf:interest <http://www.wikimedia.org> .
8
9 <http://www.wikimedia.org> rdfs:label "Wikimedia" .

```

Quelltext 6.1: Beispielhafte Darstellung von RDF unter Nutzung der Turtle Syntax und der „FOAF“ (friend of a friend) Ontologie

6.2.2 SPARQL und Triplestores

SPARQL (SPARQL Protocol and RDF Query Language) [133] ist die standardisierte Abfragesprache für RDF und stellt ein zentrales Werkzeug zur Analyse und Weiterverarbeitung semantischer Daten dar. Mit SPARQL lassen sich strukturierte Anfragen über RDF-Datenbestände formulieren, die typischerweise als Tripel (Subjekt – Prädikat – Objekt) organisiert sind.

Die Sprache ermöglicht nicht nur einfache Musterabfragen, sondern unterstützt auch komplexe Konstrukte wie Filter, Aggregationen, Unterabfragen, Joins und Pfadsuchen. Dadurch können sowohl spezifische Details (z.B. *Welche Experimente wurden mit Compiler X durchgeführt?*) als auch übergreifende Strukturen (z.B. *Wie verändern sich Laufzeitverhalten und Speicherverbrauch in Abhängigkeit der Problemgröße?*) extrahiert werden.

SPARQL eignet sich hervorragend für die Analyse von Metadaten, wie sie im Rahmen dieser Arbeit in RDF/JSON-LD Form erhoben wurden. Die gezielte Abfrage ermöglicht beispielsweise:

- das Extrahieren spezifischer Laufzeitinformationen
- das Auffinden von Konfigurationsunterschieden zwischen Experimenten
- die semantische Verknüpfung mit externem Wissen (z.B. Ontologien oder Benchmarkdaten)
- oder das Generieren von Reports und Dashboards zur Laufzeit.

SPARQL-Anfragen können gegen lokale RDF-Datenbanken oder sogenannte SPARQL-Endpunkte ausgeführt werden, die über das Netz erreichbar sind. Dies erlaubt eine flexible Integration in bestehende Analysepipelines sowie eine Anbindung an kollaborative Dateninfrastrukturen. Listing 6.2 zeigt eine Anfrage zum RDF-Beispiel aus Listing 6.1.

```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
4
5 SELECT ?name
6 WHERE
7 {
8     ?person a          foaf:Person ;
9             foaf:name    ?name ;
10            foaf:interest <http://www.wikimedia.org> .
11 }

```

Quelltext 6.2: Beispielhafte Nutzung von SPARQL zur Abfrage des zuvor gezeigten RDF-Beispiels

Durch die Verwendung von SPARQL wird eine maschinenlesbare, formale Analysegrundlage geschaffen, die sowohl für wiederholbare Performanceanalysen als auch für automatisierte Diagnose- und Optimierungsschritte genutzt werden kann.

6.3 Anforderungen durch Daten und Metadaten

Metadaten werden von Score-P und von dem Strömungslöser Hydra erzeugt. Score-P erzeugt JSON-LD-Dateien, welche nach dem Schema <https://scorep-fair.github.io/schema/v0.1/ontology.ttl> definiert sind.

Weiterhin sind die Metadaten des Strömungslösers wichtig, um die Performance-Daten in einen Kontext zu setzen. Hydra verfügt über ein kontrolliertes Vokabular, welches eine gültige Löserkonfiguration beschreibt. Dieses erlaubt die Überprüfung der Konfiguration auf Gültigkeit, definiert allerdings auch den gültigen Konfigurationsraum.

Weitere relevante Parameter sind das Berechnungsnetz und die Randbedingungen (falls dateibasiert beschrieben).

Im Verlauf des Projektes haben sich die Performance (Meta-)Daten als hauptsächliche Daten herausgestellt. Theoretisch lassen sich allerdings auch andere Daten speichern - besonders bieten sich die Simulationsergebnisse an, denn diese werden ebenso durch die Konfiguration beschrieben.

Der Fokus bleibt dennoch auf dem (Meta-)Daten Management mit dem Ziel der Anwendung im Performance Engineering, obgleich sich sämtliche RDF-Daten einpflegen und speichern lassen.

- Es sollen in etwa 1.00×10^3 Datensätze gespeichert werden
- 10.0MB bis 10.0GB je Datensatz
- Metadaten-Schema soll von Nutzerseite erweiterbar sein
- Es soll eine grafische Benutzeroberfläche verfügbar sein
- Es soll eine Skript-Schnittstelle verfügbar sein

6.4 Repositorium und Zugriffsschnittstellen

6.4.1 Infrastruktur — Dataverse & ToMeDa

Zu Beginn des Projektes war Dataverse [88] favorisiert, letztendlich aber verworfen und durch RDF ersetzt (Näheres im nächsten Abschnitt). Bei Dataverse handelt es sich um eine Open-Source-Webanwendung, um wissenschaftliche Daten zu teilen, zu bewahren, zu zitieren, zu explorieren und zu analysieren.

Durch die Nutzung haben sich allerdings folgende Punkte als Schwächen herausgestellt:

- Komplexer Verbund aus Webserver, Suchmaschine und Datenbank sowie deren Interaktion
- Komplexer Ablauf zum Erweitern der kontrollierten Vokabulare, welche einen Administrator samt Systemneustart erforderlich machen
- Unübliche Datenformate (TSV) mit ineinander geschachtelten Tabellen, welche den hohen angestrebten Automatisierungsgrad deutlich einschränken.

Dataverse ist eine sehr mächtige Software, die allerdings für den hier angedachten Einsatz zu komplex und aufwändig zu betreiben ist.

Bevor diese Schwachstellen erkannt wurden, wurden erste Skripte erstellt, um eine automatisierte Interaktion aufzubauen. Dieses Tool hieß ToMeDa (Tool für Metadaten) und verfolgte den Ansatz einer aktiven Nutzung eines Metadatenschemas. Das Schema wurde mithilfe von Pydantic, einer Python-Parserbibliothek, implementiert. Diese wird dazu genutzt, sicherzustellen, dass eine Datenstruktur nach Verarbeitung korrekt vorliegt und sämtliche Anforderungen an Typ und Existenz erfüllt sind.

Angedachtes Schema Das Schema basierte auf EngMeta [146], welches von Björn Schembera im Rahmen seiner Promotion zum Zwecke der Datenbeschreibung auf HPC Systemen entwickelt wurde. Dieses umfasst einen Grundstock von beschreibenden Metadaten. Dieses wurde händisch in eine Pydantic Datenstruktur überführt. Diese ist zu Beginn leer und wurde als NestedText [96] Datei geschrieben. Diese sollte dann von Nutzer angepasst werden: Jedem Feld kann eine Operation zugewiesen werden - zu diesem Zweck wurde eine eigene kleine Skriptsprache entwickelt, die mehrere Aktionen unterstützt. Hier wurde beispielsweise angegeben, wie bestimmte Daten ermittelt werden (z.B. Softwareversionen).

In einem nächsten Schritt wurde diese Datei eingelesen und die definierten Operationen ausgeführt. Das so gefüllte Pydantic-Objekt wurde geparsed und die Metadaten lagen als JSON vor.

Neben der JSON konnte auch ein JSON-Schema exportiert werden, aus welchem die für Dataverse benötigten Datenbeschreibungen abgeleitet werden konnten.

Kritische Betrachtung Die Nutzung von ToMeDa ist aus mehreren Gründen gescheitert.

- Die Entwicklung einer eigenen Skriptsprache ist zu komplex. So müssen neben der Syntax beispielsweise auch die Möglichkeit für das Ausführen von Bash-Befehlen implementiert,

was sicherheitskritisch und wartungsintensiv ist (Injection-Risiken, Plattformabhängigkeiten).

- Die Annahme war, dass ToMeDa die Daten aus den verschiedenen Quellen extrahiert und dann entsprechend des Schemas zusammenführt. Dies ist allerdings praktisch nicht möglich, denn bei einer Änderung auf Quellseite müsste auch das Extraktionsskript auf ToMeDa-Seite abgeändert werden, was in Zukunft viele Inkompatibilitäten nach sich zieht.
- Für die Implementierung ist auf interne Pydantic-Datenstrukturen zurückgegriffen worden. Diese standen bei der Nutzung von Pydantic 2.0 nicht mehr zur Verfügung, was große Änderungen der ToMeDa Infrastruktur bedeutet hätte.
- Hinzu kommen die Komplikationen bei der Nutzung eines neuen Schemas, dessen Nutzung ein Administrator einschalten muss, was diesen Ansatz starr macht.

Aus den obigen Gründen wurde der ToMeDa Ansatz verworfen und ein zweiter Ansatz verfolgt. Im nächsten Ansatz ist davon ausgegangen worden, dass jede mögliche Datenquelle ihre Daten in einem kompatiblen Format selber zur Verfügung stellt.

6.4.2 Infrastruktur — RDF & scorep-db

Als zweiten Ansatz wurde die RDF-basierte Lösung übernommen. In der folgenden Tabelle 6.1 werden der verworfene ToMeDa Ansatz mit dem scorep-db Ansatz verglichen.

Herausforderung ToMeDa	Lösung in scorep-db
Eigene Skriptsprache zum Befüllen der Metadatenfelder	Programme liefern Metadaten direkt als JSON/JSON-LD
Ein gemeinsames Schema für alle Datenquellen	Jede Quelle bringt eine eigene Ontologie mit
Abhängigkeit von internen Pydantic-Strukturen	Generische rdflib/JSON-LD, orchestriert von scorep-db
Schema-Erweiterung erfordert Admin/Neustart	Nutzerdefinierte Ontologien sofort nutzbar

Tabelle 6.1: Vom Dataverse/ToMeDa-Ansatz zur RDF-Lösung mit scorep-db.

6.4.3 scorep-db

Das im Rahmen des Projektes entwickelte Programm scorep-db [143] ermöglicht die einfache Nutzung der RDF-Datenbank mit den aufgenommen Score-P Traces. Zur weiteren Nutzung müssen noch einige weitere Informationen gesetzt werden. Diese werden vorzugsweise über Umgebungsvariablen gesetzt, da diese üblicherweise konstant bleiben. Diese umfassen Adressen von Datenbanken und deren Anmeldedaten und der zu verwendende Modus für Metadaten und Performance-Daten. Diese Einstellungen können in einer .env-Datei

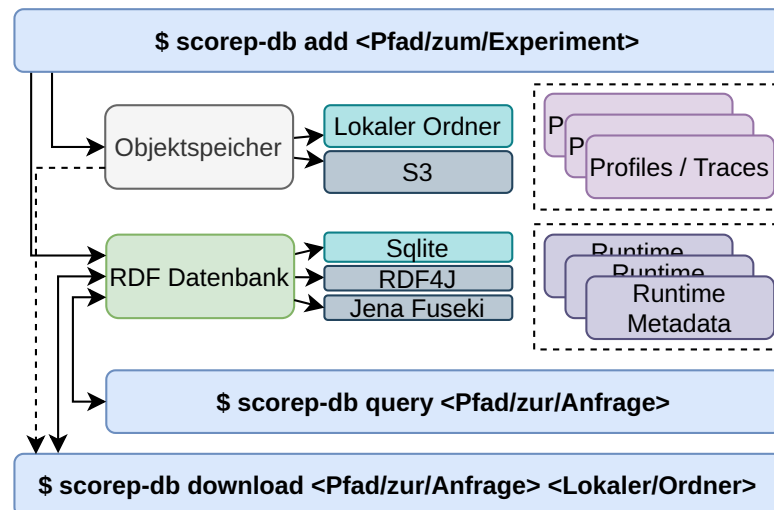


Abbildung 6.1: Vereinfachter Überblick über `scorep-db`. Weitere `scorep-db` Befehle sind in Tabelle 6.2 aufgelistet.

gespeichert werden, deren Pfad über die Umgebungsvariable `SCOREP_DB_CONFIG_FILE` gesetzt wird.

6.4.4 Verfügbare RDF-Triplestore-Backends

Die Lösung muss Offline- und Online-Modi bieten, um Daten von Clustern ohne oder mit nur beschränktem Internetzugang zu sammeln. Somit sind zwei Modi implementiert, jeweils mit einem Speicherort für die RDF-Metadaten und einem für die tatsächlichen Performance-Daten (Profile und Trace Event Daten). Tabelle 6.3 gibt eine Übersicht über die verfügbaren Backends, deren Nutzung und Modus.

Diese können auch gemischt werden: Beispielsweise eine optimierte Online RDF-Datenbank mit einem lokalen (HPC-)Verzeichnis für große Datenmengen.

6.4.5 Beispielnutzung von `scorep-db`

Zum Schutz sensibler Betriebs- und Geschäftsgeheimnisse sind die hier dargestellten Informationen mit dem öffentlich zugänglichen CFD Benchmark Code NAS-NPB [121] erstellt worden. Ein Ausschnitt eines mit `scorep-db` geparsten Datensatzes in Quelltext 6.3 in Turtle Syntax dargestellt. Zu sehen sind Entitäten des `ExperimentRun`, `Program` und `Executable` (vgl. Abb. 3.3). Dieser Datensatz kann nach Hinzufügen zur Datenbank via `scorep-db add` durch SPARQL abgefragt werden, wie schematisch in Abb. 6.1 dargestellt. Das Ergebnis von SPARQL-Abfrage 6.4 ist in 6.5 zu sehen. Abgefragt werden alle `ExperimentRun` mitsamt dem Ausführungszeitpunkt, dem Wert des `timestamp` Prädikats.

Auch komplexere Abfragen sind möglich, wie aus den Beispielen von `scorep-db` zu entnehmen ist, denn SPARQL unterstützt Aggregation und Filterfunktionen, um beispielsweise nur unterdurchschnittliche Ausführungszeiten auszugeben.

```

1 <http://internal-uid.org/77b09c0c1d4a8709b4dc32152446055c7db5828de972a4aa88e73966eb0f1a6b> a
   scorep:ExperimentRun ;
2   scorep:timestamp "2025-02-14T11:26:15"^^xsd:dateTime ;
3   scorep:usesProgram <http://internal-uid.org/
   c4f157b8ed5de309b04465e9122260920a8c386375fb52a74e502781a4acf911> .
4
5 <http://internal-uid.org/c4f157b8ed5de309b04465e9122260920a8c386375fb52a74e502781a4acf911> a
   scorep:Program ;
6   scorep:hasEnvironmentContext <http://internal-uid.org/5
   c24928efe30e3a0a45e9aa322b90c9fe617f2a5a0367dc653405b1ca9f0587a> ;
7   scorep:programArguments "" ;
8   scorep:programName "sp-mz.W.x" ;
9   scorep:usesExecutable <http://internal-uid.org/59
   edf371fd11eb12910a9590f68133eea39eae38cc6cdc109297500e581453df> ;
10  scorep:usesSubsystems <http://internal-uid.org/
   b14028f07d93765a9597e7790b78a8a246eae49e783e4735e28fabe454e6cb1d> .
11
12 <http://internal-uid.org/59edf371fd11eb12910a9590f68133eea39eae38cc6cdc109297500e581453df> a
   scorep:Executable ;
13   scorep:hasEnvironmentContext <http://internal-uid.org/
   e0554891647e4fb7b8d40aa276d2f27eb13f058bcc03c0493dd21f1694f5e1c1> ;
14   scorep:isConstructedFrom [shortened for brevity] ;
15   scorep:isLinkedBy <http://internal-uid.org/33
   fc7f72594273bfdf45d1835f541f8e3ebc5b6d9e4a9930468fb8e6352bb9be> ;
16   scorep:name "sp-mz.W.x" ;
17   scorep:scorepRevision <http://internal-uid.org/7c7d33d10-dirty> ;
18   scorep:scorepVersion "Score-P 10.0-dev"^^xsd:string ;
19   scorep:timestamp "2025-02-14T11:26:15"^^xsd:dateTime ;
20   scorep:usesFlags <http://internal-uid.org/74
   d356b4808eab6ca9b5501764f02cd2a6d6dc83b88f444b31bbe3c839e9086a> .

```

Quelltext 6.3: Ausschnitt eines Score-P RDF-Datensatzes erstellt mit scorep-db parse <experiment-dir> basierend auf einer NAS-NPB Ausführung.

Befehl	Argumente	Kommentar
scorep-db add	[<code>-append-files</code> <code><solver.ttl></code>] <code><scorep-dir></code>	Fügt zur Datenbank hinzu
scorep-db clear		Löscht die Datenbanken
scorep-db merge		Kopiert von <i>lokaler</i> zur <i>online</i> Datenbank
scorep-db health_check		Überprüft, ob die spezifizier- te Datenbank zur Verfügung steht.
scorep-db get-id	<code><scorep-dir></code>	Gibt die ExperimentRun-ID zu- rück
scorep-db parse	<code><scorep-dir></code>	Gibt das geparsete JSON-LD im Turtle Format zurück
scorep-db query	<code><query.sparql></code>	Fragt die Datenbank ab
scorep-db download	<code><query.sparql></code> <code><download-dir></code>	Lädt die Ergebnisse herunter

Tabelle 6.2: Übersicht der scorep-db Befehle

Datenbank	Daten	Betriebsmodus
SQLite	RDF-Metadaten	offline
RDF4J	RDF-Metadaten	online
Apache Jena Fuseki	RDF-Metadaten	online
Lokales Verzeichnis	Performance-Daten	offline
S3 Objektspeicher	Performance-Daten	online

Tabelle 6.3: Übersicht der verfügbaren scorep-db Daten-Backends

6.5 Zusammenfassung und Ausblick

6.5.1 Zusammenfassung

In AP3.1 wurden die Anforderungen an die Performance-Datensammlung definiert. Es wurden zwei Softwareprototypen entwickelt: ToMeDa mit dem Ziel Dataverse als Backend zu benutzen, und scorep-db mit einem Triplestore- und Objektspeicher-Backend. Der erste Prototyp (ToMeDa) hat gezeigt, dass die ursprünglich getroffenen Designentscheidungen falsch waren und im zweiten Prototyp (scorep-db) korrigiert wurden. Mit scorep-db wurden letztendlich auch die Performance-Daten aufgezeichnet und sowohl offline als auch online gespeichert.

```

1 PREFIX scorep: <http://scorep-fair.github.io/schema/v0.1/ontology.ttl#>
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3
4 SELECT ?experiment ?experiment_date
5 WHERE {
6   ?experiment rdf:type          scorep:ExperimentRun ;
7               scorep:timestamp ?experiment_date .
8 }

```

Quelltext 6.4: SPARQL-Abfrage zur Ermittlung aller durchgeführten Experimente.

```

1 Result 1:
2   experiment: http://internal-uid.org/77
3               b09c0c1d4a8709b4dc32152446055c7db5828de972a4aa88e73966eb0f1a6b
4   experiment_date: 2025-02-14T11:26:15
5 -----

```

Quelltext 6.5: Ergebnis einer SPARQL-Abfrage zur Ermittlung aller durchgeführten Experimente.

6.5.2 Ausblick

Hervorzuheben ist, dass der RDF-basierte Ansatz eine allgemeine Lösung für *Linked Data* darstellt und so auch auf Gebiete außerhalb von Performance- und Applikationsdaten erweitert werden kann. So ist es denkbar, dass auch Job-Monitoring-Systeme wie PIKA [43] Daten in RDF-Form bereitstellen, um das Laufzeitverhalten im Kontext aller laufenden Applikationen zu setzen. Auch weitere Performance Tools könnten um eine RDF-kompatible Datenbeschreibung erweitert werden, um diese direkt nutzen zu können.

Aber auch die Nutzung von maschinellem Lernen auf RDF-Datenbanken ist denkbar, um komplexe Zusammenhänge auch automatisch abzuleiten. In diesem Zusammenhang sind auch Agenten ein mögliches weiteres Forschungsthema, welche selbständig über den Graph iterieren und Transformationen ausführen, welche den Graphen mit weiteren Informationen anreichern. Ein so strukturierter Arbeitsablauf könnte möglicherweise die Komplexität der Interoperabilität verschiedenster Aufgaben senken.

7 Teilbericht AP. 3.4.8 - 3.4.11

Author: Sebastian Strönisch

This chapter presents the results of the workpackages 3.4.8 - 3.4.11. The workpackage titles are

AP 3.4.8 *Application of ML methods for the prediction of academic flow fields*

AP 3.4.9 *Development of ML methods for the prediction of stationary flow fields on meshes from AP2.3*

AP 3.4.10 *ML-based prediction of stationary and instationary three-dimensional turbine flows*

AP 3.4.11 *Integration of a trained ML model in the Hydra CFD solver framework*

whereas the results have either already been published in scientific papers [161, 162, 163] or will be part of the dissertation of Sebastian Strönisch, titled „Distributed Graph Neural Networks for Turbomachinery Flow Field Prediction“, which is scheduled to be submitted in 2025.

The following sections of this chapter are complete or condensed excerpts from the aforementioned dissertation.

In the following, the related work to graph based machine learning models is presented, the selected model architecture outlined and the developed training strategies explained. Finally, the results of [163] are presented. An integration of the surrogate model into the Hydra solver was not possible during the project runtime due to technical challenges with differences in the multigrid definition of the solver and the surrogate model.

7.1 Related Work - Graph Neural Networks for Flow Field Predictions

Machine learning refers to the process of adjusting the parameters of an adaptive model using a set of samples, called the *training set*, with the objective of making accurate predictions for data not encountered during training [17]. The ability to generalize to previously unseen examples is a key performance measure of such models. To evaluate this generalization, a separate collection of samples, the *test set*, is employed. Both the training and test sets consist

of paired examples of input and target values, derived from observations of the event or system under study. A single example, referred to here as a data sample, is defined by N input features $\{x_1, \dots, x_N\}$ and M target values $\{y_1, \dots, y_M\}$. Typically, an input vector $\mathbf{x} \in \mathbb{R}^n$ is used to predict an output $y \in \mathbb{R}$, where the dimensionalities of $\mathbf{x}$ and y depend on the specific learning task.

7.1.1 Fundamental Concepts of Graph Neural Networks

A graph is a linked data structure that represents relationships between entities and is widely used in computer science. Formally, a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of nodes $\mathcal{V}$, each with a node state $\mathbf{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_{|\mathcal{V}|}\}$, and a set of edges $\mathcal{E} \subseteq \{\{x_i, x_j\} \mid x_i, x_j \in \mathcal{V}, x_i \neq x_j\}$ connecting them. Each node i is represented by a latent feature vector $\mathbf{v}_i \in \mathbb{R}^d$, where d denotes the number of latent features.

Message-Passing Graph Neural Networks

As noted by Gilmer u. a. [60], most existing GNN layers, whether spatial or spectral, can be expressed as two learnable functions repeatedly applied to the graph's edges and nodes. At layer ℓ , a message-passing neural network (MPNN) layer performs:

$$\mathbf{m}_{ij}^{(\ell)} = \varphi_e^{(\ell)}(\mathbf{v}_i^{(\ell)}, \mathbf{v}_j^{(\ell)}, e_{ij}) \quad (\text{Message}), \quad (7.1)$$

$$\tilde{\mathbf{m}}_i^{(\ell)} = \square_{j \in \mathcal{V}(i)} \mathbf{m}_{ij}^{(\ell)} \quad (\text{Aggregation}), \quad (7.2)$$

$$\mathbf{v}_i^{(\ell+1)} = \varphi_v^{(\ell)}(\mathbf{v}_i^{(\ell)}, \tilde{\mathbf{m}}_i^{(\ell)}) \quad (\text{Update}), \quad (7.3)$$

where $\square$ is a permutation-invariant operator such as sum, mean, or max. The functions φ_e and φ_v are often implemented as MLPs with shared parameters $\mathcal{W}_{\{e,v\}}^{(\ell)}$ for all nodes and edges, enabling them to generalize to graphs with varying sizes and structures.

By stacking K such layers, information propagates across the K -hop neighbourhood of each node, analogous to how deeper CNNs enlarge the receptive field in image processing [62]. Scalable training methods, such as in *GraphSAGE* [65], sample a fixed subset $\mathcal{S}_i \subseteq \mathcal{V}_i$ of each node's neighbours. This inductive sampling with learnable aggregators enables representation learning for unseen graphs while reducing memory requirements. Edges are undirected, and self-loops are excluded; e_{ij} denotes the edge between nodes i and j . For introductory details on graph theory, see for example [152].

Early GNN formulations, such as that by Scarselli u. a. [145], applied a single learnable transformation iteratively to all node states until convergence, inspired by recurrent neural networks (RNNs). While theoretically sound, this method suffered from long convergence times and large memory costs. A major shift came with Bruna u. a. [25], who extended CNNs to graphs in the spectral domain using the Laplacian eigenbasis, enabling true weight sharing but at cubic complexity in the number of nodes. Defferrard u. a. [38] reduced this cost by representing spectral filters as truncated Chebyshev polynomials, achieving strictly K -hop localised filters at $\mathcal{O}(K \mid \mathcal{E} \mid)$ complexity. Building on this, Kipf u. a. [89] proposed the Graph Convolutional Network (GCN) layer, simplifying the Chebyshev expansion to $K = 1$ and using a renormalized adjacency matrix to yield an efficient first-order approximation.

Physics-Informed Graph Neural Networks

The *GraphSAGE* framework [65] showed that a shared neighbourhood aggregator can produce embeddings for previously unseen graphs. Battaglia u. a. [13] generalised this concept by introducing a graph network (GN) block that updates edges, nodes, and a global state vector, extending the MPNN formulation (7.1)–(7.3) with a global update φ_u . Let $E = \{\mathbf{e}_k\}$ denote edge states, $V = \{\mathbf{v}_i\}$ node states, and $\mathbf{u}$ a global state vector. A GN block then computes:

$$\mathbf{e}'_k = \varphi^e(\mathbf{e}_k, \mathbf{v}_{s(k)}, \mathbf{v}_{r(k)}, \mathbf{u}), \quad (7.4)$$

$$\bar{\mathbf{e}}_i = \rho^{e \rightarrow v}(\{\mathbf{e}'_k \mid r(k) = i\}), \quad (7.5)$$

$$\mathbf{v}'_i = \varphi^v(\mathbf{v}_i, \bar{\mathbf{e}}_i, \mathbf{u}), \quad (7.6)$$

where $\varphi^{\{e,v,u\}}$ are learnable MLPs and $\rho^{(\cdot)}$ are permutation-invariant reductions (e.g. sum). This structure embeds a relational inductive bias: physical laws act locally on node pairs (edges), aggregate at nodes (control volumes), and influence a global conserved quantity.

The GN block is typically placed within an Encode–Process–Decode architecture [13], allowing raw inputs to be embedded into a latent space, processed, and decoded to outputs. Sanchez-Gonzalez u. a. [141] first demonstrated that a GN block could replace explicit numerical integrators and generalise to new system parameters. Sanchez-Gonzalez u. a. [140] extended this to continuum fluid simulations by representing each particle as a node and learning particle–particle interactions via multiple GN blocks, achieving long-horizon stability and near-solver accuracy.

7.1.2 Graph Neural Networks for Flow Field Predictions

MeshGraphNets (MGN) [129] extended the GN framework to mesh-based CFD. Each vertex in the unstructured mesh is treated as a graph node, edges are taken from mesh connectivity, and edge attributes include geometric distances. Messages approximate local differential operators, and boundary conditions are imposed as fixed target values. While successful on 2D benchmarks (airfoils, cylinders), MGN was tested only on cases with a few thousand nodes.

Scaling to industrial meshes with millions of nodes introduces two challenges: (i) GPU memory limitations and (ii) *over-smoothing* [181, 30], where deep or densely connected message-passing layers cause node embeddings to drift toward the graph-wide mean, erasing high-frequency flow features [126]. Over-smoothing also appears in sequential GCN stacks [103, 102, 18] and can only be partly alleviated by coupling with low-fidelity solvers [14].

Our initial work [161] used GCN layers to replace convolutions for steady 2D flow prediction. We later developed MGN-NoComm and MGN-Halos [162], training on subgraphs from domain partitioning. MGN-NoComm discards cross-partition edges, while MGN-Halos retains them as *halo* connections and exchanges messages across GPUs using neighbour-AllToAll. Although conceptually enabling full-domain learning, the initial MGN-Halos implementation underperformed, likely due to a latent technical issue. Strönisch u. a. extended this to Distributed BSMGN, integrating bi-stride pooling to enable steady and unsteady turbomachinery predictions on meshes with over $2 \cdot 10^6$ nodes.

Other works explored related scaling strategies: Hines u. a. ensured only nodes with intact neighbourhoods contributed to the loss in subgraph learning, avoiding cross-partition communication; Barwey u. a. implemented a consistent cross-partition message-passing mechanism, scaling to 10^9 nodes in synthetic tests but without surrogate modelling. Multigrid-inspired designs such as Bi-Stride MeshGraphNets (BSMGN) [30] use fixed coarse-fine mappings in a U-Net-like structure to reduce cost and partially mitigate over-smoothing by preserving multi-scale features. This architecture is also employed in the present work.

To investigate the performance of multiscale GNNs for large-scale 3D turbine flows, we developed a distributed variant of the Bi-Stride MeshGraphNet (BSMGN) [30], building on our earlier work [162] and initially omitting halo communication. In [163], this distributed BSMGN was evaluated on both steady and unsteady turbine cases, producing predictions for surface quantities as well as for the complete volumetric flow field. The corresponding results are presented in Abschnitt 7.3.

In the present work, this concept is extended by integrating the halo-exchange mechanism first proposed in [162], resulting in a synchronous distributed BSMGN capable of message passing across partition boundaries. The details of this implementation are described in Abschnitt 7.2.3.

7.2 Distributed Multiscale Graph Machine Learning Model

This chapter presents the graph neural network (GNN) architecture forming the basis of the surrogate models developed in this work, along with the distributed training strategies explored during its development. The discussion begins with a detailed definition of the learning tasks in Abschnitt 7.2.1. Next, Abschnitt 7.2.2 describes the selected GNN model and the modifications required to adapt existing approaches for large-scale fluid flow prediction, where high resource demands make scaling to large meshes a key challenge. Subsequently, Abschnitt 7.2.3 outlines three strategies for applying GNNs to meshes whose size exceeds the memory capacity of a single GPU. Finally, Abschnitt 7.2.4 summarises the main findings of this chapter.

7.2.1 Description of Learning Tasks and Evaluation Modes

The definition of the surrogate's target task must be established before model selection, and the evaluation procedure must remain consistent throughout the study.

In related works such as [129] and [30], the primary Learning Task (LT) is the prediction of unsteady two-dimensional flow evolution around cylinders and airfoils. In this work, additional LTs target turbomachinery design workflows, where surrogates approximate physical quantities Q_i to reduce computational cost.

Learning Task

The problem formulation follows [163] and extends it to four LTs.

Consider a parametric design setting in which each geometry is defined by a parameter vector $\mathbf{p}_i \in \mathbb{R}^p$. For each geometry, a mesh $\mathcal{M}_i$ is generated. In the turbomachinery cases here, an

unstructured grid generator produces meshes with varying node counts n_i , depending on geometry.

A steady-state 3D CFD simulation then solves the RANS equations with a turbulence model, producing the solution

$$\mathbf{q}_i^R = f(\mathbf{p}_i, \mathcal{M}_i), \quad (7.7)$$

where $\mathbf{q}_i^R$ contains pressure p , density ρ , velocities u, v, w , and turbulence variable(s).

Steady RANS is often sufficient for design-oriented tasks [55, 26]. However, certain applications, such as rotor surface temperature or blade forcing prediction, require unsteady simulations. Here, URANS is employed to resolve the dominant unsteady phenomena at reduced cost compared to LES [176, 48].

For periodic turbomachinery flows, the unsteady solution over n_T timesteps is expressed as

$$\mathbf{q}_i^t = g(t, \mathbf{p}_i, \mathcal{M}_i), \quad \text{with} \quad (7.8)$$

$$Q_i = [\mathbf{q}_i^1, \mathbf{q}_i^2, \dots, \mathbf{q}_i^{n_T}], \quad (7.9)$$

and the mean field $\overline{Q}_i$ often more closely reflects reality than $\mathbf{q}_i^R$ [48].

The mesh $\mathcal{M}_i$ is treated as a graph $\mathcal{G}_i = (\mathcal{N}_i, \mathcal{E}_i)$, with node coordinates $C_i \in \mathbb{R}^{n,3}$, node types $N_i \in \mathbb{R}^n$, and adjacency matrix

$$A_i = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_e], \quad (7.10)$$

where each edge j connects nodes k and l : $\mathbf{a}_j = [k, l]$.

Node types include *normal*, *inflow*, *outflow*, *blade*, *hub*, and *shroud*. In some LTs, $\mathbf{p}_i$ is duplicated across all nodes and concatenated to the node features.

The four LTs are:

LT 1: Predict the time-averaged flow field either from mesh and parameters only:

$$\hat{\mathbf{q}}_i = \hat{g}(\mathbf{p}_i, \mathcal{M}_i), \quad (7.11)$$

or additionally using $\mathbf{q}_i^R$:

$$\hat{\mathbf{q}}_i = \hat{g}(\mathbf{p}_i, \mathcal{M}_i, \mathbf{q}_i^R). \quad (7.12)$$

LT 2: Predict the unsteady field at $t + 1$ from timestep t :

$$\hat{\mathbf{q}}_i^{t+1} = \hat{g}(\mathcal{M}_i, \mathbf{q}_i^t), \quad (7.13)$$

where $\mathbf{q}_i^t$ includes ρ, u, v, w , and p but excludes turbulence variables.

LT 3: Predict the time-averaged surface pressure (cf. 7.11) restricted to the surface mesh $\mathcal{M}_{S,i}$, including surface normals $S_i \in \mathbb{R}^{n,3}$.

LT 4: Predict unsteady surface pressure analogous to 7.13, again restricted to $\mathcal{M}_{S,i}$ and including S_i .

Evaluation Modes

Following [129, 30, 163], two evaluation modes are used:

- **Nextstep:** The model takes $\mathbf{q}_i^t$ and predicts $\mathbf{q}_i^{t+1}$ (eq. 7.13).
- **Rollout:** Iteratively applies eq. 7.13 using predicted states as input:

$$\hat{\mathbf{q}}_i^{t+1} = \hat{g}(\mathcal{M}_i, \hat{\mathbf{q}}_i^t) . \quad (7.14)$$

The main accuracy metric is RMSE:

$$e_i = \sqrt{\frac{1}{n} \sum_{j=1}^n (\hat{\mathbf{q}}_i^{t+1} - \mathbf{q}_i^{t+1})^2} , \quad (7.15)$$

with n the number of predictions. For single physical quantities, the relative error is

$$e_{\text{rel},i} = \frac{e_{\text{abs}}(q_i) - \min(q_i)}{\max(q_i) - \min(q_i)} , \quad (7.16)$$

where e_{abs} is the absolute error. Rollout errors over k final frames are averaged as

$$e_{\text{R},k} = \frac{1}{k} \sum_{i=0}^k e_i . \quad (7.17)$$

Two-Dimensional Cylinderflow Testcase

The Cylinderflow dataset, introduced in [129] and later used in [30], simulates incompressible 2D flow around a cylinder, transitioning from the initial state to either steady or oscillatory unsteady regimes. The Navier–Stokes equations are solved on irregular triangular meshes (COMSOL¹), capturing phenomena such as vortex shedding and wake formation. Figure 7.1 shows a representative frame.

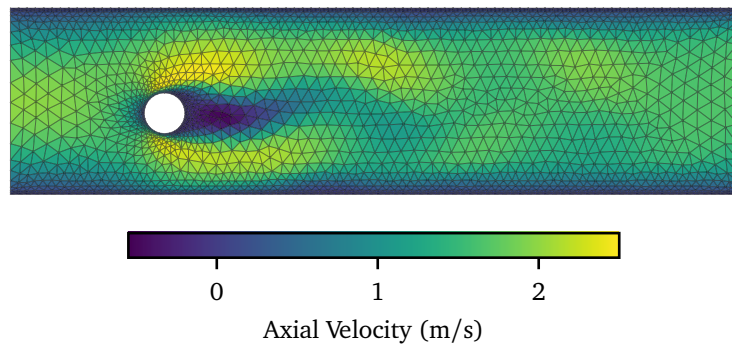


Abbildung 7.1: Single Frame of a Cylinderflow trajectory

Each trajectory spans 600 timesteps of $\Delta t = 0.01$ s. Meshes contain about 1885 nodes, with diversity introduced by varying cylinder size and position.

¹<https://www.comsol.com/cfd-module>

No Reynolds or Strouhal numbers are provided. Using assumptions from [54] (e.g., $\nu = 1 \times 10^{-3} \text{ m}^2/\text{s}$) and geometry inference from mesh density, the training set covers Re from 14 to 2183 (mean ≈ 291). Some trajectories reach steady states; others remain unsteady².

This dataset is used here to benchmark distributed training strategies, since the second dataset in Abschnitt 7.3 exceeds single-GPU capacity, preventing a one-device BSMGN reference.

7.2.2 Model Selection

All surrogate models developed in this work are based on the U-Net-shaped Bi-Stride Multiscale Graph Network (BSMGN) [30], illustrated in Fig. 7.2. BSMGN was the first publicly available multiscale MGN and demonstrated higher accuracy than the original MGN architecture. For related literature on multiscale GNNs for fluid flow prediction, see Abschnitt 7.1.

In previous work on turbomachinery flow prediction [162], the MGN architecture was used without a multigrid hierarchy; thus, no inter-level transitions were required. The distributed MGN in [162] was implemented according to the concepts later formalised in Abschnitt 7.2.3 and Abschnitt 7.2.3. Here, the same concepts are extended to the BSMGN framework.

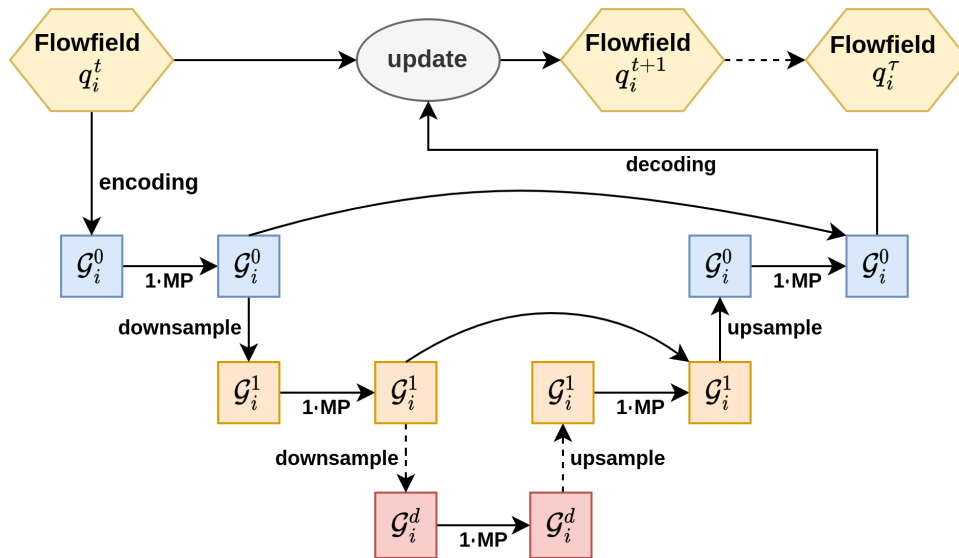


Abbildung 7.2: BSMGN architecture, adapted from [30] as presented in [163].

BSMGN follows the Encode–Process–Decode structure [141], where the processor operates on a sequence of increasingly coarse meshes $\mathcal{G}^0 \rightarrow \mathcal{G}^d$. These coarse levels are generated via bi-stride pooling (Abschnitt 7.2.2), starting from the finest mesh $\mathcal{G}^0$ and retaining every second breadth-first-search (BFS) frontier. Because each discarded vertex shares an edge with at least one retained neighbour, the coarse mesh remains connected, and the discarded–retained edges provide interpolation weights for later unpooling.

In [30], a single message-passing (MP) block is applied per level. Although more MPs per level are possible, they are not explored here. The contracting path reduces the latent state down to $\mathcal{G}^d$, enabling long-range interactions with only d MPs, compared to many more in a flat MGN. The latent state is then prolonged back to $\mathcal{G}^0$ with an MP at each level.

²<https://sites.google.com/view/meshgraphnets#h.6gyr94evchy0>

Downsampling is algebraic and parameter-free. After pooling, the adjacency matrix is row-normalised, and a contribution matrix $\mathbf{W}_{e,ij}$ is computed. Node features and a running weight $\mathbf{W}_n$ are aggregated through $\mathbf{W}_e$, analogous to multigrid restriction. Upsampling reverses this process using $\mathbf{W}_e^T$, returning features to discarded vertices in a manner similar to a transposed convolution. This interpolation requires no additional connectivity or parameters due to the guaranteed adjacency between pooled vertices and their neighbours. The full details are given in Abschnitt 7.2.2.

BSMGN supports both unsteady flow prediction (LT 2 and LT 4) and direct prediction of time-averaged fields (LT 1 and LT 3, Abschnitt 7.3). For the latter, the model outputs the target variables directly, without updating the input. A special case, denoted Vol-R2AV, updates the steady RANS solution $\mathbf{q}_i^R$ to predict the time-averaged URANS field $\hat{\mathbf{q}}_i$.

While effective, the architecture has certain limitations. The mesh hierarchy is fixed from a single BFS seed, and although heuristics like Close-Center or MinAve [30] generally work well, a poorly chosen seed can produce unbalanced coarse graphs, reducing accuracy and efficiency. Restricting the processor to one MP per level reduces memory use and runtime but may limit the representation of complex dynamics. Finally, the non-parametric coarse-fine transitions save parameters and memory but cannot learn task-specific prolongation kernels, which may be advantageous in some cases.

These aspects are not investigated further here. Instead, the unmodified BSMGN is adopted and extended with synchronous neighbourhood communication for distributed training on partitioned graphs.

Communication inside graph neural networks

Before developing the synchronous distributed training approach, we identify where the BSMGN architecture exchanges information across node neighbourhoods. We focus on the Eulerian CFD code path; details for Lagrangian simulations are given in [30].

BSMGN follows the *encode-process-decode* paradigm [13, 141]. On the finest level $\mathcal{G}^0$, the *encode* module maps nodal inputs to a latent representation and the *decode* module maps latent features to predictions. All coarser levels operate purely in latent space and consist of a *process* block with exactly one message passing (MP) step per level, following [30]. The *encode* and *decode* modules are two-layer MLPs; therefore, only the *processor* (MP, downsampling, upsampling) requires analysis with respect to neighbourhood communication.

Message passing At level ℓ , geometric edge attributes are computed on-the-fly from the node coordinates $\mathbf{c}^\ell$ (rather than precomputed at batch assembly). A preprocessing operator ρ^ℓ forms the relative displacement and its norm for each edge (i,j) :

$$\mathbf{x}_p^\ell \leftarrow \rho^\ell \left(\mathbf{x}_{ij}^\ell, \|\mathbf{x}_{ij}^\ell\| \right). \quad (7.18)$$

These edge attributes are concatenated with sender/receiver node features and passed through an edge MLP $\varphi^{e,\ell}$. Edge messages are then aggregated at receivers, and a node MLP $\varphi^{v,\ell}$

produces the residual update:

$$\mathbf{e}_{ij}^\ell \leftarrow \varphi^{e,\ell}(\mathbf{x}_p^\ell, \mathbf{v}_i^\ell, \mathbf{v}_j^\ell), \quad (7.19)$$

$$\hat{\mathbf{v}}_i^\ell \leftarrow \mathbf{v}_i^\ell + \varphi^{v,\ell}(\mathbf{v}_i^\ell, \sum_j \mathbf{e}_{ij}^\ell). \quad (7.20)$$

Aggregation is implemented via `torch_scatter.scatter()` with `reduction="sum"`.³ Two data dependencies imply halo exchanges on partitioned graphs: (i) accessing neighbour coordinates and features $(\mathbf{c}_j^\ell, \mathbf{v}_j^\ell)$ to build edge inputs for $\varphi^{e,\ell}$; and (ii) evaluating the receiver-side sum $\sum_j \mathbf{e}_{ij}^\ell$.

Downsampling (restriction) Inter-level transfer is non-parametric and mirrors multigrid restriction. A running weight field $w_{n,i}$ is initialised to one on $\mathcal{G}^0$ and accumulated across levels. The downsampling from level ℓ to $\ell+1$ proceeds in three steps:

(1) *Row normalisation and single-weight convolution.* Let A_{ij} be the (boolean) adjacency. First normalise rows,

$$\hat{A}_{ij} = \frac{A_{ij}}{\sum_j A_{ij}}, \quad (7.21)$$

then propagate running weights,

$$w_{n,ij} = w_{n,i} \hat{A}_{ij}. \quad (7.22)$$

This step is local to the sender i .

(2) *Edge contribution (normalised sender weights).* Form the contribution table

$$w_{e,ij} = \frac{w_{n,ij}}{\sum_i w_{n,ij}}, \quad (7.23)$$

which requires the receiver-side sum $\sum_i w_{n,ij}$; consequently, a halo exchange is needed across partition boundaries.

(3) *Latent aggregation onto the coarse nodes.* Restrict latent node features by weighted aggregation

$$\mathbf{v}_j = \sum_i \mathbf{v}_i w_{e,ij}, \quad (7.24)$$

which again requires halo access to sender features $\mathbf{v}_i$ for receivers j near the partition boundary.

Upsampling (prolongation) Returning from level $\ell+1$ to ℓ uses the same contribution table. First, coarse features are *unpooled* to the fine index set (placing features at retained vertices and zeros elsewhere). Then a neighbourhood update distributes coarse information to the discarded fine vertices via

$$\mathbf{v}_i = \sum_j \mathbf{v}_j w_{e,ij}, \quad (7.25)$$

which is algebraically equivalent to applying the transpose mapping (akin to a transposed convolution). This step requires halo access to coarse features $\mathbf{v}_j$ when edges (i,j) cross partitions.

³See <https://pytorch-scatter.readthedocs.io/en/latest/functions/scatter.html>.

End-to-end data flow across one level One pass from level ℓ to $\ell+1$ and back comprises: (i) restriction of $(\mathbf{v}^\ell, \mathbf{c}^\ell)$ via (7.21)–(7.24) to obtain $(\mathbf{v}^{\ell+1}, \mathbf{c}^{\ell+1})$ and the contribution table $w_{e,ij}$; (ii) one message-passing step at level $\ell+1$ using $\mathbf{c}^{\ell+1}$ to form edge attributes and update $\mathbf{v}^{\ell+1}$; and (iii) prolongation to level ℓ via (7.25). In distributed training, halo exchanges occur in four places: edge preprocessing (coordinates of neighbours), MP aggregation, restriction steps (2) and (3), and the prolongation update.

Creating distributed multigrids

For distributed MGNN training, the computational graph derived from the CFD simulation⁴ must be both partitioned and converted into a multigrid hierarchy.

Following [30, 29], this work uses a bi-stride pooling strategy to construct coarser graphs from the fine numerical mesh. The application of distributed BSMGN models to other multigrid generation schemes is deferred to future work.

Before creating the multigrid levels, the fine graph is partitioned into n_B subdomains using Metis [86], assigning each node a block identifier stored in B . The bi-stride pooling process requires a set of seed nodes, and [29] describes two seeding heuristics that were both used in [30] and are therefore adopted here with minor modifications:

1. *MinAve*, which selects in each cluster the node with the minimum average geodesic distance to all other nodes in that cluster; and
2. *NearCenter* (termed *CloseCenter* in [29]), which selects the node closest to the geometric centroid of the cluster.

The modification introduced in this work extends both algorithms to also return the corresponding block IDs of the chosen seed nodes. The *MinAve* approach performs a BFS from each node to compute all-pairs distances, resulting in $\mathcal{O}(N^2)$ time complexity; it is therefore only feasible for smaller cases such as the CylinderFlow dataset. For the larger turbomachinery datasets in Abschnitt 7.3, *NearCenter* is preferred because it requires only a single centroid distance computation per cluster, resulting in linear complexity.

Two distinct strategies for creating partitioned multigrids are used. For the synchronous training strategy described in Abschnitt 7.2.3, no halo information is required and the bi-stride pooling is applied independently to each partition. In contrast, for the approach that maintains halo information across partitions, the pooled seeds and their block IDs are computed globally before coarse graphs are generated.

The bi-stride pooling procedure proceeds in a cluster-aware fashion to ensure that nodes from disconnected components remain separate across levels. Clusters are identified using a BFS-based DetermineClusters procedure, which collects all nodes reachable from an arbitrary seed and repeats until all nodes are assigned. This ensures that multiple rigid bodies or separate domains remain independent in the multigrid hierarchy.

Once seeds L_s and their block IDs L_b are available, a SplitEdges procedure partitions the original edge list into intra-block and inter-block subsets. This involves: 1. Building a local node index map N_b for each block, remapping both node IDs and intra-block edges $\mathcal{E}_{\text{intra},b}$ to contiguous

⁴See Abschnitt 7.2.1 for the mapping between graph and CFD notation.

local IDs; 2. Identifying sender and receiver nodes in edges that connect different blocks to form the inter-block edge list $\mathcal{E}_{\text{inter},b,i}$; and 3. Mapping global pooled node IDs to local IDs, producing the list $L_{b,s}$ of locally indexed pooled seeds.

This top-down approach to multi-level partitioning inherits coarse partitions directly from the fine mesh without rebalancing, which can lead to computational load imbalances on coarser levels. More advanced schemes could incorporate multi-level partitioning or repartitioning to equalise workload while minimising inter-partition communication. Such improvements are left for future work; the reader is referred to [10] for a survey of graph partitioning techniques that address these issues.

7.2.3 Strategies for Distributed Training

Training on partitioned graphs requires a strategy to handle information flow across subdomain boundaries. In CFD, this is commonly addressed through halo nodes, which are copies of boundary nodes from neighbouring partitions that are refreshed each timestep to ensure stencil completeness. This concept inspires similar approaches for distributed GNN training. In this work, three data-parallel variants of BSMGN are implemented and evaluated, each differing in how they handle (or omit) halo exchanges. The original single-device implementation from [30] serves as a reference.

Isolated but Distributed

IsoT applies standard data parallelism without any inter-partition communication. Each device processes its own graph partition, using a separate multigrid hierarchy generated for that partition. The model and training loop are wrapped in PyTorch DDP, ensuring that no sample is processed twice within an epoch. This approach was used in [163] for both steady and unsteady turbomachinery flow predictions. An analysis of its performance scaling is given in the forthcoming dissertation of Sebastian Strönisch.

Synchronous Distributed

SyncT extends IsoT by adding synchronous halo exchanges at all points in the BSMGN pipeline where neighbour information is needed (Abschnitt 7.2.2). Only node features are exchanged; edge attributes are rebuilt in each message-passing step. Halo nodes store copies of boundary nodes from adjacent partitions, ensuring that distributed training preserves global connectivity. Implementation details and evaluation results are provided in the forthcoming dissertation of Sebastian Strönisch.

Asynchronous Distributed

AsyncT seeks to remove the constraint that the number of devices must match the number of partitions. It does this by performing an initial forward pass without gradient tracking to collect halo information, followed by a second pass that uses the recorded data for training. While conceptually promising, the current implementation is incomplete and suffers from limitations

in the halo multigrid generation process, which restricts feasible configurations. Consequently, AsyncT is not further evaluated in this work, and a more robust multigrid generation process is left for future research.

7.2.4 Conclusion

This section introduced the multiscale GNN architecture selected for this work, along with the two evaluation modes: *Nextstep*, which measures single-step prediction accuracy from ground-truth inputs, and *Rollout*, which assesses long-term prediction quality by iteratively feeding back model outputs. Reproducibility was ensured through fixed random seeding, and challenges in distributed data loading were addressed.

To enable scaling from the single-device reference implementation (RefT) to large-scale flow field data, three distributed training strategies were outlined. IsoT applies standard data parallelism without any inter-partition communication. SyncT adds synchronous halo exchange at all levels of the multiscale model, restoring cross-partition connectivity. AsyncT proposes an asynchronous halo exchange to remove the one-device-per-partition constraint, but remains incomplete and was not further evaluated.

Preliminary investigations (cf. forthcoming dissertation of Sebastian Strönisch) indicate that SyncT can match RefT in accuracy, confirming that halo exchange can recover most of the original information flow lost through partitioning. However, this comes at substantially higher computational and runtime cost. Given the practical aim of applying the methods to large, real-world datasets, all subsequent results in this work are obtained using the more efficient IsoT strategy. Optimising SyncT, for example by tuning PyTorch or NCCL communication parameters, remains a subject for future work.

7.3 Prediction of Turbomachinery Flow Using GNNs

Although numerous GNN methodologies have been demonstrated on academic-scale numerical meshes, only a limited number of studies address the scaling challenges that arise from the resource-intensive nature of these models. Even fewer attempts have been made to apply a distributed GNN to an industry-scale dataset.

This chapter presents, with minor editorial adjustments, results previously published in our peer-reviewed article [163].

The surrogate model Dist-BSMGN is applied on a real-world dataset and deployed to predict physical quantities of either the full flow domain or the airfoil surface. Additionally, the model is either used for the prediction of time-averaged or time-resolved quantities. The different model versions are listed in Tab. 7.1. The following sections will first present results for the full domain time-averaged models Vol-AV and Vol-R2AV. Afterwards, the time-resolved full domain predictions of Vol-U are shown. The following two sections describe and present predictions made by the surface models Surf-AV and Surf-U, of which the first is trained on time-averaged and the latter on unsteady surface values. Finally, the experiments on the real-world dataset using IsoT are concluded in section 7.3.7.

Tabelle 7.1: Features of Dist-BSMGN models

LT	Model	Input	Target
1	Vol-AV	$\mathcal{M}_i^d, N_i, \mathbf{p}_i$	$\bar{Q}_i$
1	Vol-R2AV	$\mathcal{M}_i^d, N_i, \mathbf{p}_i, Q_{R,i}$	$(\bar{Q}_i - Q_{R,i})$
2	Vol-U	$\mathcal{M}_i^d, N_i, Q_i^t$	Q_i^{t+1}
3	Surf-AV	$\mathbf{M}_{S,i}^d, N_i, S_i, \mathbf{p}_i$	$\bar{Q}_{S,i}$
4	Surf-U	$\mathbf{M}_{S,i}^d, N_i, S_i, \mathbf{p}_i, Q_{S,i}^t$	$Q_{S,i}^{t+1}$

7.3.1 Turbine Test Case

The dataset that is used in this work to asses the capabilities of the GNN is based on an optimization task of a representative high-pressure turbine stage of an aircraft engine. The optimization

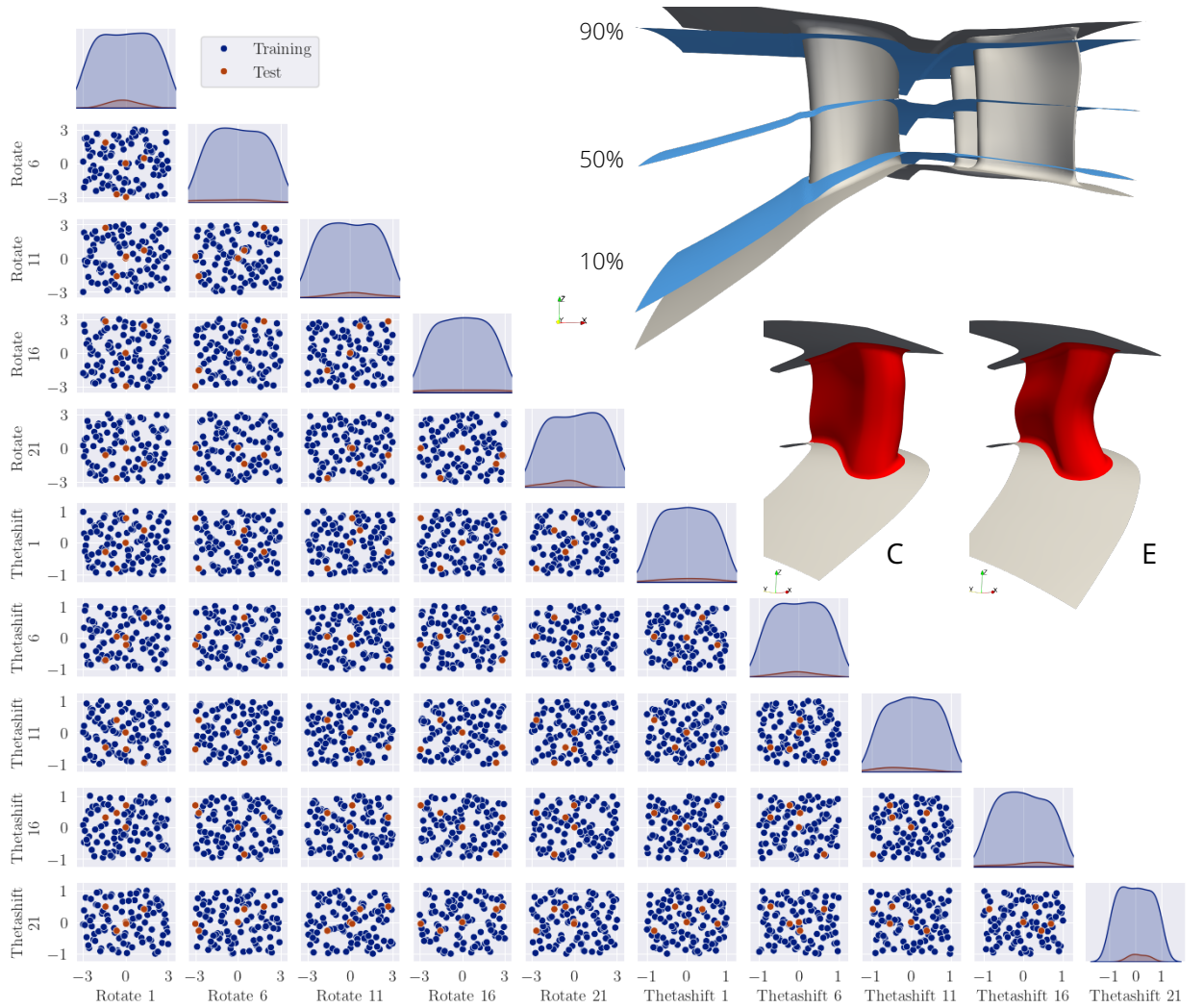


Abbildung 7.3: Pairplot of sampling strategy (left), computational setup with visualization planes (top) and two stator samples (right), from [163]

task was to find a stator geometry for an optimal force distribution on the downstream rotor blade. The corresponding numerical setup of the sector model is comprised of the stator to be optimized and two rotor blades of the subsequent blade row.

The baseline geometry is depicted in Fig. 7.3 on the top right. The full dataset consists of 101 URANS simulations of the turbine stage with different stator geometries which were generated using an optimal Latin-Hypercube Sampling (oLHS) DoE for ten design parameters describing section shifts in axial and circumferential direction over radial height, see [3]. The split into 96 training and 5 test geometries is described further below. Some examples of the generated stator geometries from the test set are depicted in Fig. 7.3 on the right. The rotor blade was not varied in this setup.

The blade counts of the turbine stage have been selected to enable unsteady simulations using the sector model with a mesh for the stator domain of sizes between 750×10^3 and 970×10^3 nodes for the different geometries that were generated. The surface of the stator is discretized with roughly 13.0×10^3 points. The rotor domain with two blades has a mesh size of 2.50×10^6 points, and both airfoils have a surface mesh with 30.0×10^3 points each.

The Rolls-Royce CFD solver Hydra [99] using the Spalart-Allmaras one equation turbulence model [156] was used to solve the unsteady flow. In particular the sector model period is resolved with 70 timesteps. Six periods were simulated to reach the periodic solution which consists of 70 snapshots and is assembled in the matrix Q_i .

Fig. 7.3 visualizes on the bottom the optimal Latin Hypercube designs of the 10 geometry parameters that were used to create the dataset and on the top right the cuts on three different radial heights that are used for visual inspection of the predicted flow quantities in the following. Fig. 7.4 gives a summary of the normalized variation of turbine stage efficiency η versus reduced

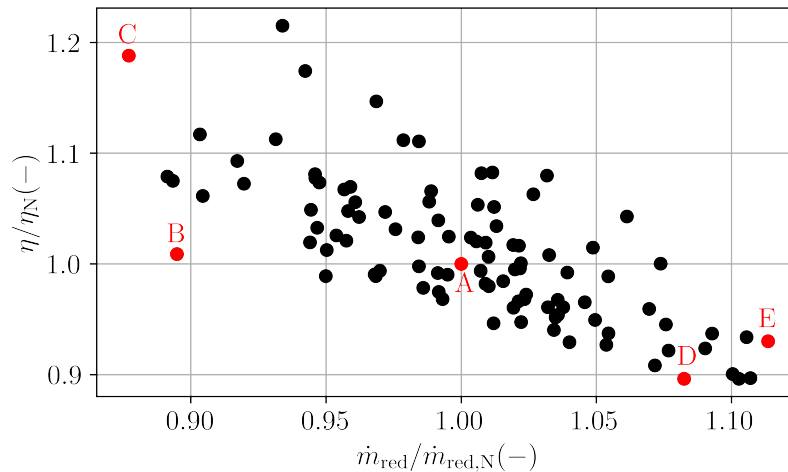


Abbildung 7.4: Turbine stage efficiency over reduced massflow of train (black) and test (red) samples normalized to nominal case (sample A)

massflow $\dot{m}_{\text{red}}$. Based on this scatterplot, the nominal flow solution (sample A) and four other test samples are selected in the outer regions of the result space (marked in red in Fig. 7.4) to also assess the surrogate model prediction quality for outliers.

Tab. 7.2 summarizes key information on the compute clusters that were used for this work. The RANS and URANS were performed on the CPUs of *TUD ZIH Romeo*. The training and inferencing of the ML models is done on GPUs on *TUD ZIH Alpha Centauri*. The RANS simulation required approx. 2 h on 32 CPUs and the URANS approx. 16 h on the same amount of CPUs. Only accounting for the computing time, one sample requires approx. 576 CPUh.

Tabelle 7.2: Hardware specifications from [185]

Node Specification / System	TUD ZIH - Romeo	TUD ZIH - Alpha Centauri
Processor	2× AMD EPYC 7702 @ 2.0 GHz	2× AMD EPYC 7352 @ 2.3 GHz
Accelerator	-	8× NVIDIA A100-SXM4
Sockets×Cores	2 × 64	2 × 24
Memory	512 GB	1 TB
Interconnect	Infiniband HDR 100 GBit/s	Infiniband HDR 200 GBit/s
Max. Power [†]	≈ 650 W	≈ 5200 W

[†] Maximum sustained power from ZIH-internal measurements.

In order to train a GNN on these large graphs, each simulation mesh $\mathcal{M}_i$ is partitioned and an isolated multigrid is build for every subgraph. The fine level $\mathcal{G}_i^0$ in Fig. 7.2 therefore represents only a single partition for the graphs that span the full domain (LT 1 and LT 2). Graphs that only represent blade surfaces (LT 3 and LT 4) are small enough for a single device memory and not partitioned, so that $\mathcal{G}_i^0$ represents the full surface mesh. A possible alternative is discussed in section 7.3.4. The stator domain (S) is divided into 20 partitions and the rotor domain (R) into 50 partitions. Separate models are trained for each domain without any information exchange. Training one model for the full turbine stage is subject to future work.

7.3.2 LT1: Time-Averaged Volume Quantities

This section presents the models Vol-AV and Vol-R2AV trained to predict the time-averaged flow field of stator and rotor domain with a separate model each. As presented in Tab. 7.1, the models were trained on either just the multigrid $\mathcal{M}_i^d$ and the stator geometry parameters $\mathbf{p}_i$ or additionally on the flow results from the RANS simulation $Q_{R,i}$. In the latter case the target were not the absolute flow quantities but rather the difference between RANS and time-averaged URANS flow field. All models were trained on 2 GPUs (cf. Tab. 7.2) with a batchsize of one per device. The models were stopped at epoch e , when the test loss did not decrease significantly in a certain time interval (early stopping). Both models have a latent dimension of 128 and 6 multigrid levels, which result in 11 MPs in total and a model with 1.79×10^6 trainable parameters. Tab. 7.3 presents the results of both models for the two domains respectively. The superiority of Vol-R2AV is clearly apparent for both domains in terms of accuracy. The time for an inference of a new sample is longer for Vol-R2AV as a RANS has to be performed first. Besides that Vol-R2AV

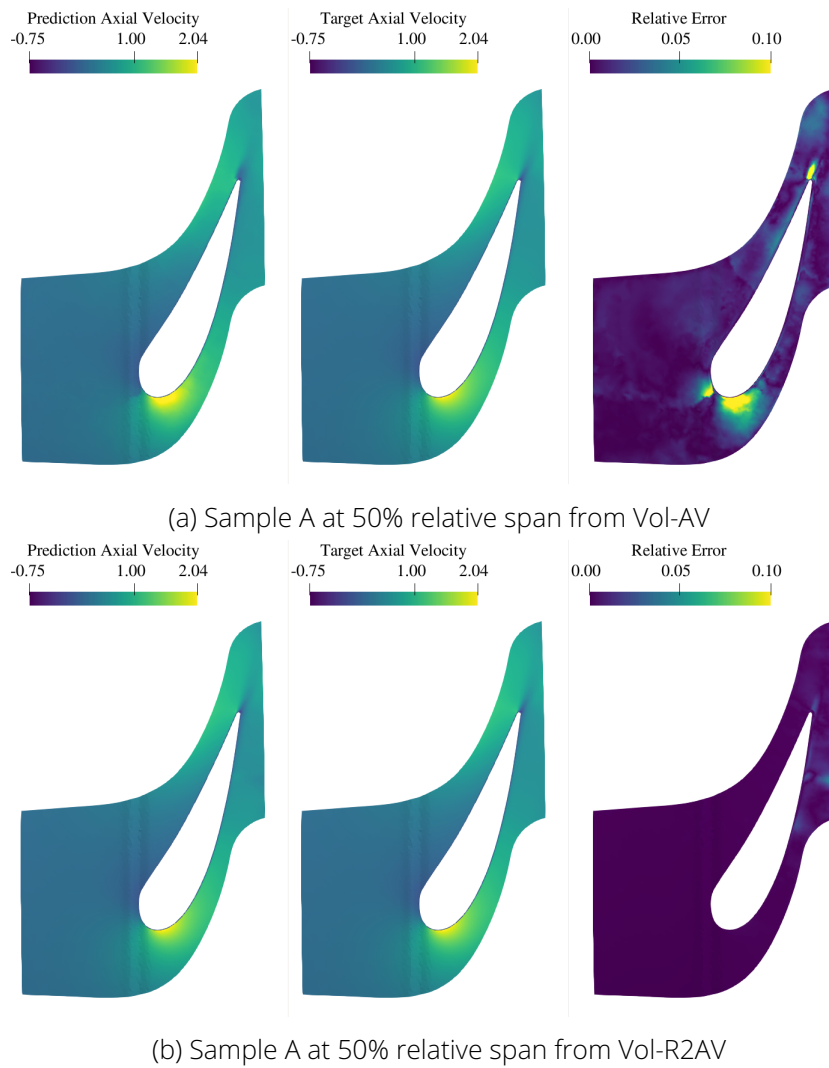
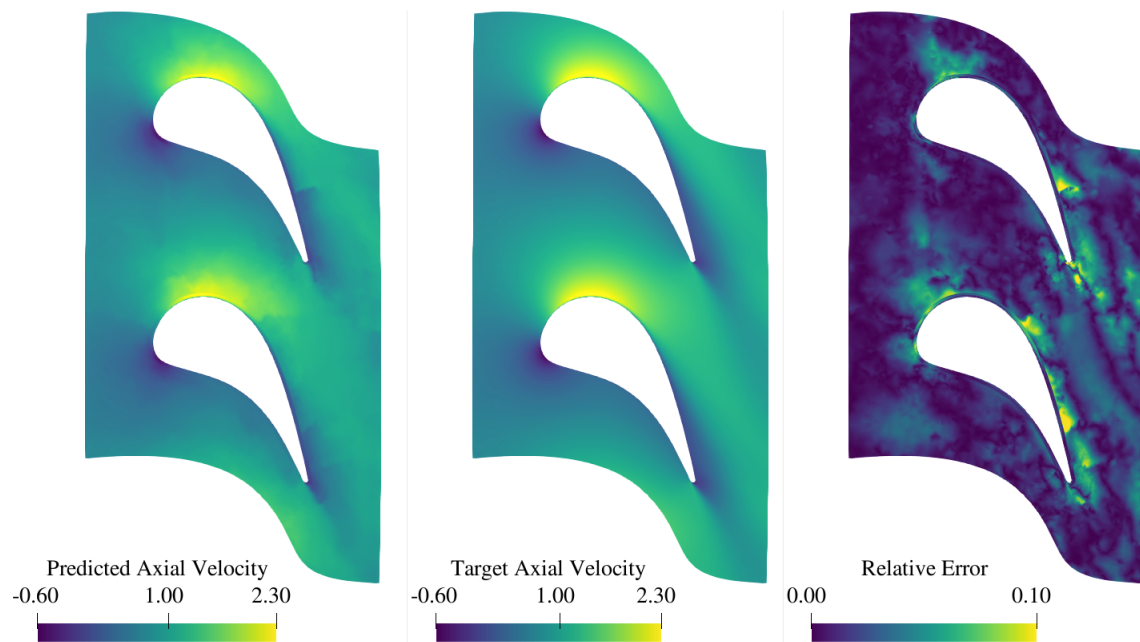


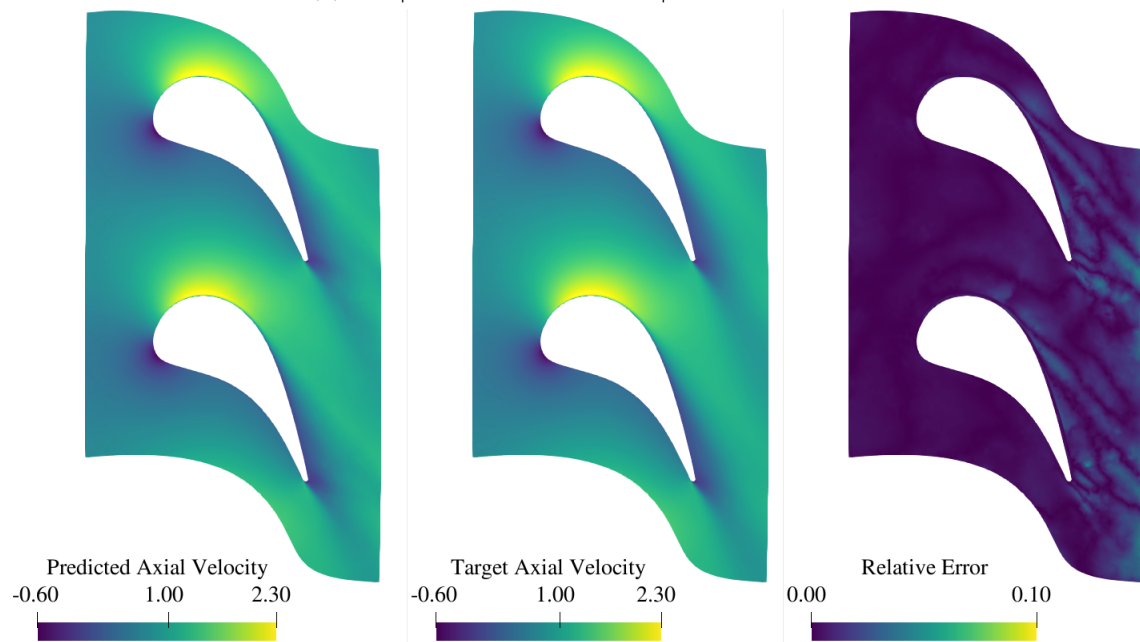
Abbildung 7.5: Comparison of radial cuts through Time-Averaged Stator Flowfield Predictions

produces more accurate predictions with less training time and computational resources. Fig. 7.5 depicts a 50% radial cut through predictions of the nominal stator domain from both models. It presents a direct comparison of a predicted quantity, here axial velocity u , the target and the relative error. It becomes clear that the prediction of Vol-AV in Fig. 7.5a deviates more from the target at leading and trailing edge, whereas the deviations of Vol-R2AV are not apparent in the same intensity. The same type of comparison is made for predictions of the nominal rotor in Fig. 7.6, which shows that model Vol-R2AV yields significantly lower deviations from the target. Fig. 7.7 supports these findings, as it shows significant lower relative error throughout the whole test set for the stator and the rotor for model Vol-R2AV compared to Vol-AV. That Vol-R2AV also predicts the other four flow quantities sufficiently well is presented in the appendix of [163] showcasing the pressure field.

So far, the predictions were only presented on mid-span. The predictions however are made on the complete flow field so in [163] in the appendix, the axial velocity predictions on 10% and 90% relative span are featured, which are similarly well predicted.



(a) Sample A at 50% relative span from Vol-AV



(b) Sample A at 50% relative span from Vol-R2AV

Abbildung 7.6: Comparison of radial cuts through Time-Averaged Rotor Flowfield Predictions

Tabelle 7.3: Mean RMSE and standard error for the turbine test set and training time for full domain time-averaged predictions (LT1).

Model	Row	t_{train} [h]	e	RMSE $\times 10^{-3}$
Vol-AV	S	25.05	100	166.47 ± 15.27
Vol-R2AV	S	4.68	30	68.45 ± 13.60
Vol-AV	R	100.5	200	123.84 ± 8.91
Vol-R2AV	R	16.28	35	48.32 ± 5.34

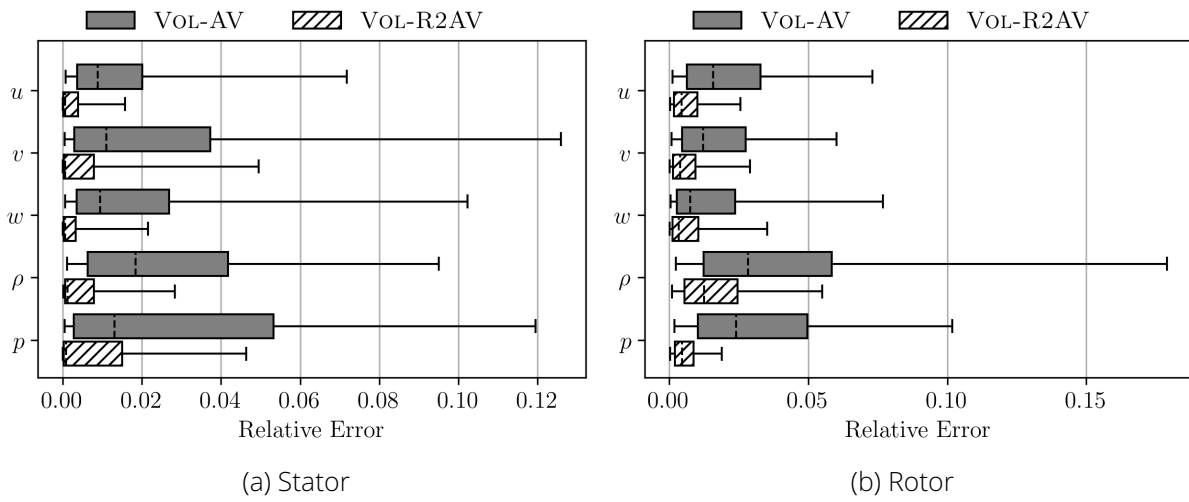


Abbildung 7.7: Comparison of relative test set error for time-averaged full domain predictions (LT1) with median (dashed) and 5 and 95% quantiles (whiskers).

7.3.3 LT2: Unsteady Volume Quantities

This section presents the model Vol-U trained to predict the unsteady evolution of three dimensional turbine flow. The Vol-U models were trained on a full compute node with eight GPUs (cf. Tab. 7.2), a latent size of 64, 6 multigrid levels and a batch size of one per device. Tab. 7.4 presents the comparison of test set errors for the stator and rotor domain. For the prediction of the flow field in the stator domain, an MGN version without multigrid [162] is used as comparison to show the benefits of hierarchical GNNs. Model Vol-U yields a lower test set error for rollouts as well as nextstep prediction after only 15 epochs (early stopping) compared to the previous versions of MGN [162] applied to the stator of this dataset. Even though the number of model parameters is higher (cf. Tab. 7.4) Vol-U needs only 1.2 days to train on the stator flow and 4.1 days for the rotor. The MGN version without multigrid required 23 days for training on the stator domain alone [162]. Training on the rotor domain was not considered due to the unjustifiably high computational resources required (approx. 70 days).

Fig. 7.8 presents a cut at 50% relative span through the rollout predictions of the rotor domain of sample A from model Vol-U. The error of the first step prediction is quite small but the error clearly built up after 8 rollout steps, pictured in Fig. 7.8b. Here, only 16 of the 70 timesteps are

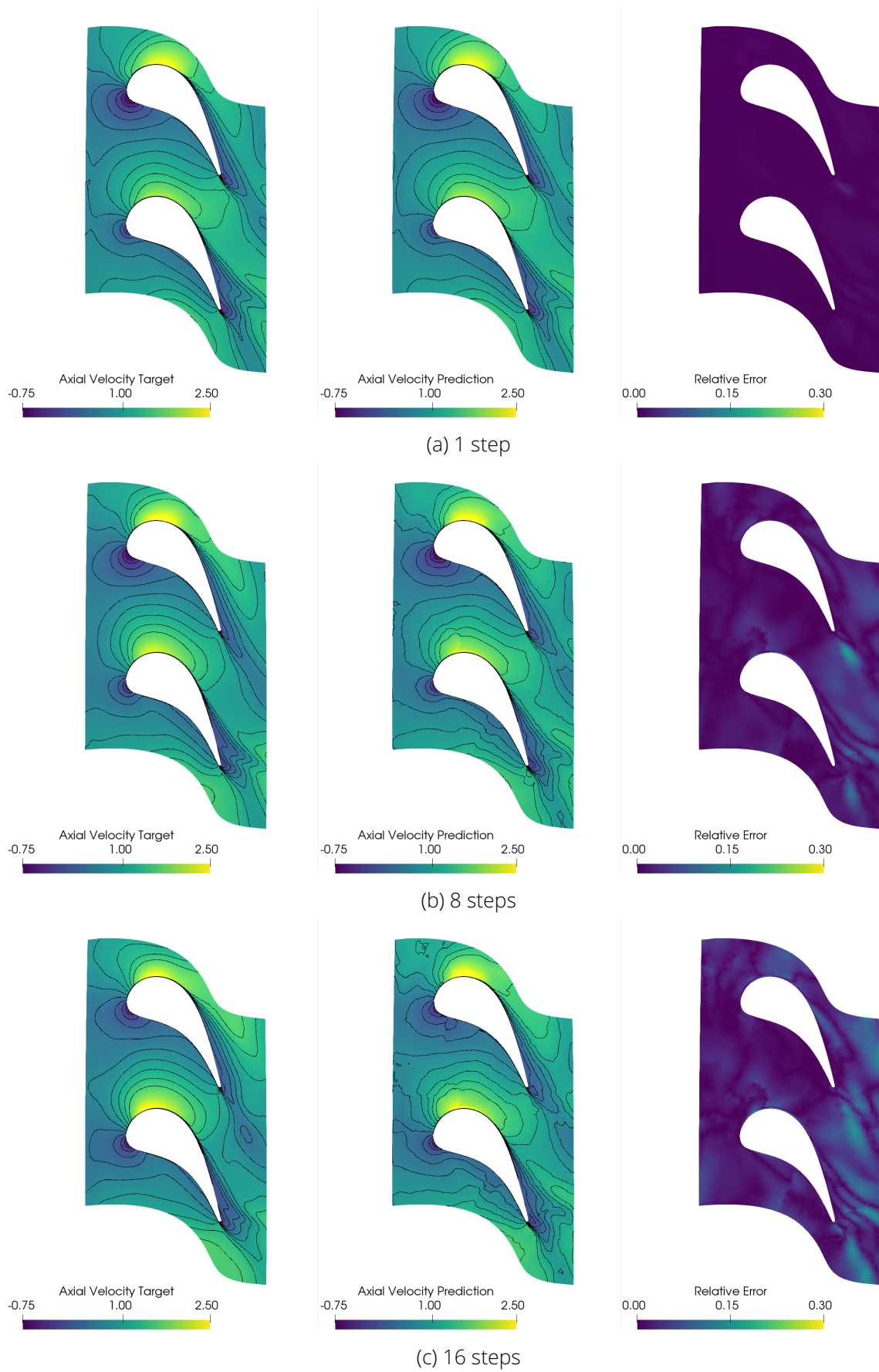


Abbildung 7.8: 50% cuts through rollout of rotor domain predictions from model Vol-U for sample A

Tabelle 7.4: Comparison of training time, model parameters and mean and standard error of RMSE over the turbine test set for LT2 models. Models trained with m parameters using d multigrid levels, k message passing steps

Model	Row	t_{train} [h]	e	d	k	m $\times 10^{-6}$	RMSE 1-step $\times 10^{-3}$	RMSE rollout-all $\times 10^{-3}$	RMSE nextstep $\times 10^{-3}$
MGN [162]	S	560	500×10^3	0	10	0.18	96.76 ± 4.28	99.51 ± 3.90	94.10 ± 0.53
Vol-U	S	29	15	6	11	0.45	7.39 ± 1.76	63.58 ± 10.44	3.67 ± 0.86
Vol-U	R	99	20	6	11	0.45	20.13 ± 1.89	202.29 ± 9.94	10.40 ± 1.07

shown, as the prediction Fig. 7.8c already lost a lot of the physical flow pattern, e.g. the sharp edges in the suction side contours. How this erroneous prediction influences a probe value in the flow field, is shown in section 7.3.5.

A limitation of this work is the indirect bias of the surrogate on the timestep width used in the reference simulation. The variation of the timestep width in the URANS simulation and its affect on the surrogates generalizability performance is subject to future work.

7.3.4 LT3: Time-Averaged Surface Quantities

For the predictions of surface quantities on numerical meshes with a fixed number of points, established methods like PODI can be employed as a comparison to the ML models, analogue to the comparisons of GNN and PODI models in [70]. The PODI model used here employs a gaussian process regressor for the interpolation part. For details on PODI for aerodynamic surrogate models the reader is referred to [136, 70].

Tabelle 7.5: Mean and standard error of RMSE and training time over the turbine test set for time-averaged surface predictions.

Model	Row	t_{train} [h]	e	RMSE $\times 10^{-3}$
Surf-AV	S	1.1	1500	59.82 ± 9.78
Surf-AV	R	6.32	2300	27.20 ± 3.18
PODI	R	0.0028	–	37.84 ± 4.84

Tab. 7.5 presents the prediction error for the two ML models and PODI. In addition, Fig. 7.9 presents for all three model the node-wise distribution of the relative error, which is the absolute error scaled to the value range. For the rotor surface the comparison between PODI and the BSMGN Surf-AV model shows the dominance of the ML model. However, the ML model needs multiple orders of magnitudes longer for the training with more resources, as both Surf-AV

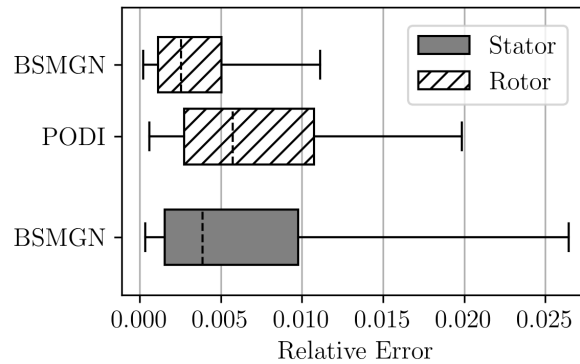


Abbildung 7.9: Nodewise relative Error for Surface Pressure Prediction of Test set with whiskers at 5th and 95th percentiles and dashed median.

models are trained on a single GPU (cf. Tab. 7.2) on the full training set⁵. The relative error distribution of the second BSMGN model that was trained on the stator surface pressure scatters more than for the rotor surface pressure. Here, 4 multigrid levels worked better than the 6 used for the rotor surface pressure. Both model are trained with a latent vector size of 32 and a batch size of two.

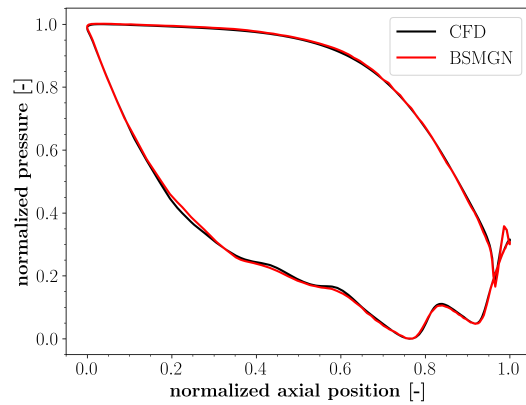
Fig. 7.10 presents a comparison of the surface pressure distribution from the prediction of Surf-AV with the CFD target at 50% relative span for the nominal sample A and one of the outliers (sample C). It becomes apparent that Surf-AV performs slightly better than PODI. However, considering the very short fitting time of PODI of 10s and an inference time of around 0.024s on *ZIH Romeo* (cf. Tab. 7.2), both models can be considered valid options for the task at hand. The inference timings presented in section 7.3.6 in Tab. 7.6 show that Surf-AV in comparison needs around 0.0107s in average on one GPU for the prediction of both rotor surfaces and approx. 0.0121s in average for the prediction of the stator surface pressure. In the appendix of [163] a three dimensional view on suction and pressure side of the two rotor blades of sample A is presented additionally.

The capability of GNNs to work on variable sized meshes comes into play when applied on the stator dataset, so that Fig. 7.10 only features a comparison of the prediction of Surf-AV with the CFD pressure distribution. It becomes clear that the model excels for the prediction of the nominal case as pictured in Fig. 7.11. But it also shows good performance for the other test samples. Figure 7.12 features a three-dimensional view on the surface pressure prediction of sample C where most of the deviations from the target are present on the suction side. On the pressure side no large deviations are visible.

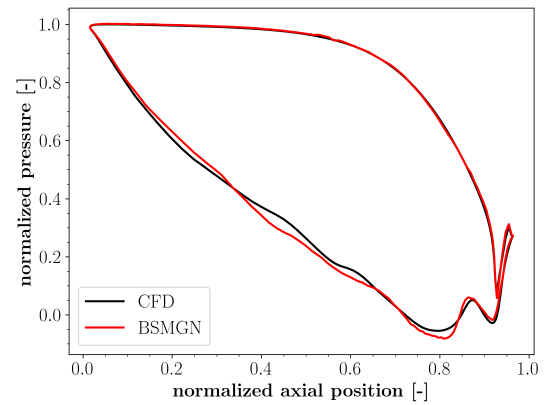
7.3.5 LT4: Unsteady Surface Quantities

In this section, model Surf-U is trained on the unsteady surface pressure values of the rotor domain. The latent size was chosen to be 32 and only a single GPU was used with batchsize 2. After 8.43h and 50 epochs the model training was stopped. As for the volume model Vol-U, the rollout error increases quite fast from $(24.47 \pm 3.54) \times 10^{-3}$ after the first step to

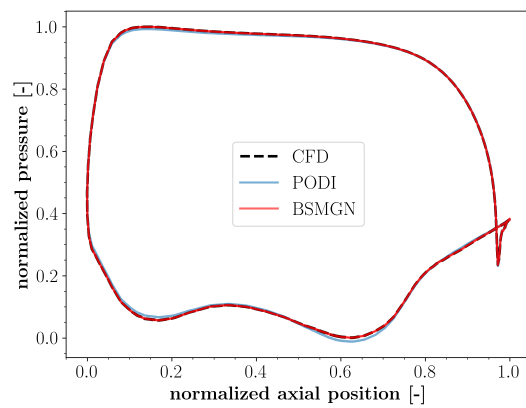
⁵A study with less training samples showed higher generalization errors. Whether these models are sufficiently accurate can only be evaluated with a downstream task, which is why this study is not presented here.



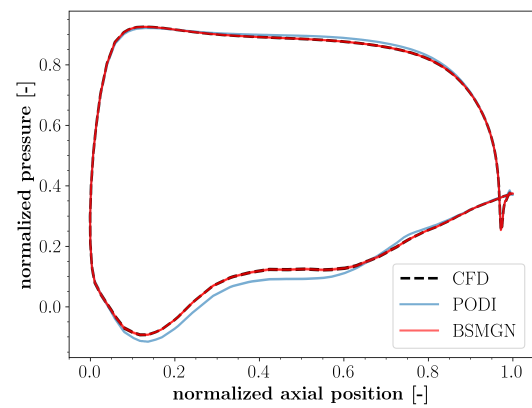
(a) Stator A



(b) Stator C



(c) Rotor A



(d) Rotor C

Abbildung 7.10: Surface Pressure Distribution at 50% radial span for samples A and C.

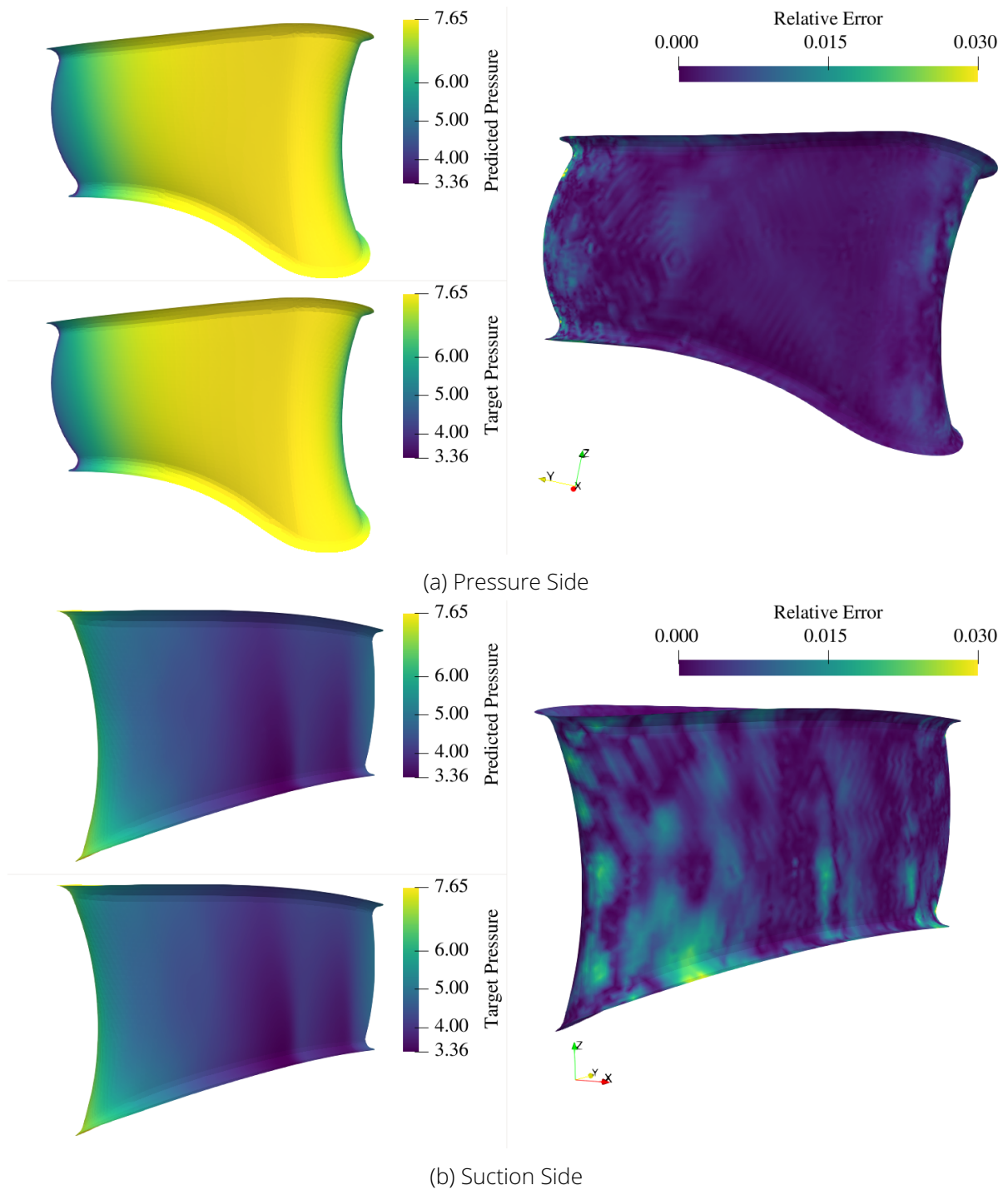


Abbildung 7.11: Stator Surface Pressure prediction of sample A from model Surf-AV. Note different color scale for relative error.

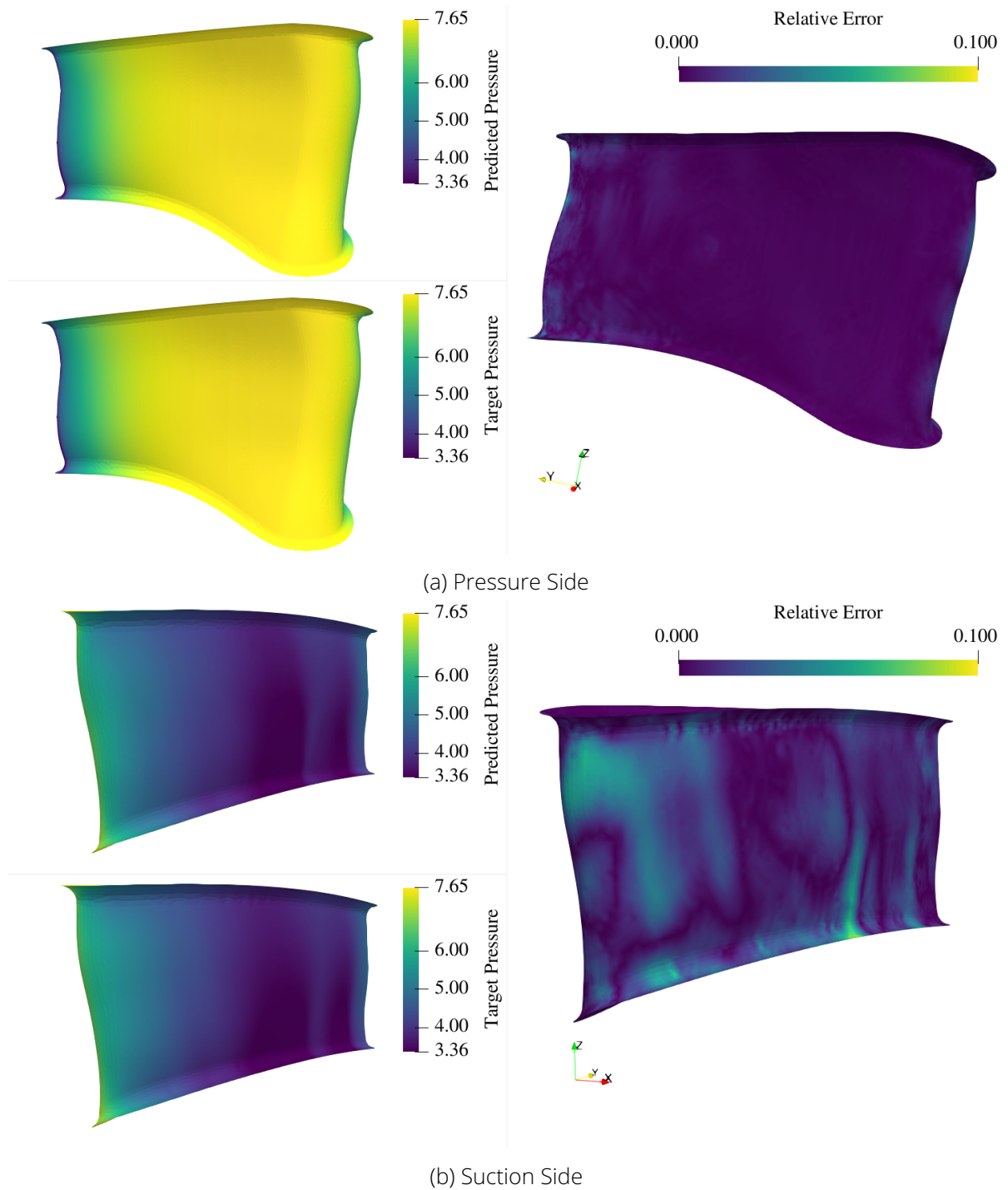


Abbildung 7.12: Stator Surface Pressure prediction of sample C from model Surf-AV. Note different color scale for relative error.

$(214.19 \pm 17.38) \times 10^{-3}$ after all steps. However, analogue to Vol-U, the nextstep error stays low at $(12.95 \pm 1.81) \times 10^{-3}$.

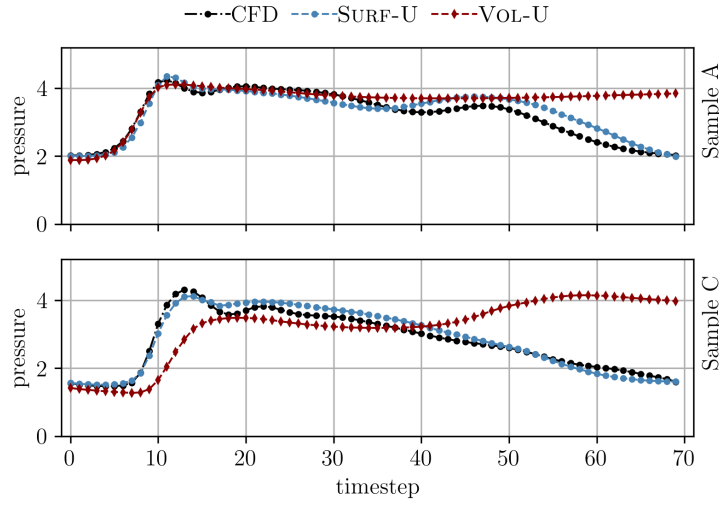


Abbildung 7.13: Comparison of point with highest amplitude over time from sample A and C.

One of the use-cases of this LT is to assess whether this approach would be feasible for unsteady probes on the surface. Therefore, the point with the largest value amplitude over time in the nominal CFD simulation is selected. Fig. 7.13 presents a comparison of the target CFD and the predictions of Surf-U and Vol-U for samples A and C. As mentioned previously, the rollout of Vol-U is not able to predict the evolution of surface values. Thus a second point that was moved slightly above the wall is used for a comparison to the CFD results instead. Fig. 7.13 shows in contrast to the rollout of Vol-U, the rollouts of model Surf-U stay very close to the target and deviate only for the outlier sample C in the last 20 steps more significantly.

7.3.6 Training and Inference Costs

This section derives a rule of thumb to compare the resources required for the various surrogates with the cost of a simulation. Training and inference costs need to be put into perspective with the resources required to produce the reference simulation in order to assess the computational efficiency of the approach. Here, the fluid solver Hydra [99] was used to create the dataset. The same flow solver was used in [118] in comparison to a GPU-implementation of it. From this work a performance (not power) equivalent translation of 1 NVIDIA V100 GPUh $\approx$ 150 CPUh can be derived. A rough performance-wise translation of the NVIDIA V100 GPU (used in [118]) computing time to the NVIDIA A100 GPU (used in this chapter) would be to double the spent GPUh, which corresponds to measurements from other authors [41]. With this rule of thumb (i.e. 1 GPUh $\approx$ 300 CPUh) the training times of the models presented in this work can be put into perspective of the resources required for one URANS simulation. The costs are denoted c and are presented as CPUh of the equivalent URANS in Tab. 7.6 calculated with the above rule of thumb. Note that this does not account for the 55.0×10^3 CPUh it took to produce the 96 training samples, which also includes the RANS simulations.

Additionally, the average inference timings of a new sample calculated on one GPU of *Alpha Centauri* (cf. Tab. 7.2) are presented. Using one node with 128 cores, the reference URANS

setup finishes in roughly 4 hours (using an average from three separate runs), i.e. 525 CPUh. This average runtime for a URANS is used to calculate the equivalent number of simulations $\tilde{n}_{\text{URANS}}$ one could've run instead of training a surrogate model. The values are rounded to the next integer. So for the CPUh spend on training e.g. a Vol-U model for the rotor domain alone, about 453 URANS simulation could've been performed.

To put the inference times in Tab. 7.6 into perspective, an inference step for a prediction of a steady rotor surface from model Surf-AV would be 1.38 mio. times faster than the corresponding URANS. A full rollout of model Vol-U for the rotor domain would take 70 predictions 0.0108s, which is 19.5 thousand times faster than the reference URANS.

Tabelle 7.6: Training Costs and Average Inference Timings

LT	Model	C_{Training} [CPUh]		$t_{\text{Inference}}$ [s]		$\tilde{n}_{\text{URANS}}$ [-]	
		Rotor	Stator	Rotor	Stator	Rotor	Stator
1	Vol-AV	60 300	15 030	0.0101	0.0107	239	29
1	Vol-R2AV	9 768	2 808	0.0106	0.0106	19	5
2	Vol-U	237 600	69 600	0.0108	0.0108	453	133
3	Surf-AV	1 896	330	0.0107	0.0121	4	1
4	Surf-U	2 646	-	0.0105	-	5	-

7.3.7 Conclusion

This section revisited the results from [163], demonstrating the application of the isolated but distributed training strategy IsoT to an industry-scale turbomachinery dataset with varying stator geometries. The model was used to predict both time-averaged and unsteady flow quantities in the volume and on blade surfaces.

For the time-averaged prediction tasks (LT 1 and LT 3), BSMGN delivered promising performance. In LT 1 (volume quantities), a mixed approach combining RANS CFD with BSMGN achieved lower relative error and shorter training time, but required a costly RANS simulation for inference. In LT 3 (surface pressure), BSMGN slightly outperformed a PODI surrogate, while offering the advantage of handling variable-size meshes without remeshing.

For the unsteady flow prediction tasks, BSMGN achieved low next-step errors in the full-volume rollout task (LT 2) but lost stability after only a few timesteps. Still, compared to earlier approaches, training time and resource usage were reduced while accuracy improved. In LT 4 (unsteady surface pressure), the model produced stable rollouts and low next-step errors.

Two key limitations of IsoT were also identified. First, partitioning is mandatory to fit the full mesh into GPU memory, but isolated multigrid creation restricts information flow and introduces systematic error. A potential remedy, full-batch training with halo exchange (SyncT, cf. Abschnitt 7.2), proved too resource- and time-intensive, with training times increasing by at least an order of magnitude in preliminary tests. Second, the inherently high computational cost of

GNN training (cf. 7.3.6) challenges the approach's viability in a production toolchain. This study showed that carefully restricting surrogate targets (e.g. to surface quantities) can substantially reduce training costs, but the ultimate value of such surrogates can only be assessed after integration into an optimization or uncertainty-quantification workflow, which remains future work.

7.4 Summary and Outlook

This work set out to develop a surrogate model for predicting turbomachinery fluid flow. An extensive review of existing methods identified graph neural networks (GNNs) as the most promising approach for the problem at hand. Due to the considerable computational demands of GNNs, the study focused on distributed training as a means to enable scaling to industry-sized datasets. Three distributed training strategies were proposed, with the IsoT approach implemented and evaluated on a real-world turbine dataset.

The results demonstrate that IsoT can be applied successfully to large turbomachinery meshes, although its partitioning strategy introduces limitations in information exchange. These findings highlight the trade-off between scalability and predictive fidelity in distributed GNN training. Future research should aim to refine distributed training strategies to improve both efficiency and accuracy. Promising directions include adaptive resource allocation during training, integrating advanced subgraph learning with optimised halo communication, and reducing memory overhead through more efficient multigrid generation. Further investigations into transferability across different but related flow scenarios, as well as the incorporation of uncertainty quantification, could enhance the robustness and applicability of GNN-based surrogates in engineering design and optimisation pipelines.

8 Zusammenfassung und Ausblick

Die Ergebnisse dieses Forschungsprojektes zeigen eindrucksvoll, wie die gewonnenen Forschungsergebnisse einen entscheidenden Beitrag zur Weiterentwicklung zukünftiger Flugtriebwerke leisten können. Die steigenden Anforderungen an Effizienz, Leistungsfähigkeit, Umweltverträglichkeit und Betriebssicherheit erfordern innovative Ansätze, die über konventionelle Entwicklungsprozesse hinausgehen. Die hier vorgestellten Methoden und Analysen bilden die Grundlage, um den wachsenden Herausforderungen in der Luftfahrtindustrie effizient zu begegnen und technologische Grenzen nachhaltig zu verschieben.

Die präzise Modellierung von Triebwerkskomponenten sowie die Berücksichtigung von Unsicherheiten durch probabilistische Methoden sind zentrale Bausteine für die Entwicklung neuer Triebwerksgenerationen. Durch die umfassende Analyse von Fertigungsstreuungen und deren Einfluss auf die Aerodynamik und Struktur von Turbinenschaufeln können bereits in frühen Entwicklungsphasen robuste Prognosen für zukünftige Designs erstellt werden. Dies führt zu einer signifikanten Reduktion von Entwicklungsrisiken, da potenzielle Schwachstellen frühzeitig identifiziert und adressiert werden können.

Ein besonderer Fortschritt liegt in der systematischen Erfassung und Auswertung von Geometriedaten mittels hochmoderner Messverfahren. Die im Rahmen dieser Arbeit eingesetzten optischen und CT-basierten Messmethoden, wie sie beispielsweise bei Rolls-Royce Deutschland standardmäßig zur Qualitätssicherung verwendet werden, ermöglichen eine präzise Analyse der Innen- und Außengeometrien von Turbinenschaufeln und stellen die Grundlage für die Arbeiten im AP 2.1 dar. Die im Arbeitspaket AP 2.1 durchgeführten Monte-Carlo-Simulationen in Kombination mit Extended Latin Hypercube Sampling und der parametrischen Rekonstruktion von Schaufelgeometrien eröffnen völlig neue Möglichkeiten für die präzise Vorhersage von Temperaturverteilungen und Spannungsverteilungen in Turbinenschaufeln. Diese Simulationen beruhen auf einer aerothermale Kopplungsanalysen, die entscheidend zur Verbesserung der thermischen Effizienz und der mechanischen Belastbarkeit von Triebwerkskomponenten beiträgt. Durch die Integration dieser Methoden in den Entwicklungsprozess können Bauteile gezielt für höhere Belastungen ausgelegt werden, was die Lebensdauer der Komponenten verlängert und die Betriebssicherheit erhöht.

Im Arbeitspaket AP 2.3 lag der Fokus auf der Untersuchung der Oberflächenrauheit von Hochdruckkompressorschaufeln (HPC) unter realen Betriebsbedingungen. Hierbei wurde ein au-

tomatisiertes Verfahren zur detaillierten Rauheitsmessung entwickelt, das den Einsatz eines hochpräzisen Alicona-Sensors in Kombination mit einem Pick-and-Place-Roboterarm umfasst. Dieses System ermöglicht die Durchführung von Rauheitsmessungen mit hoher Wiederholgenauigkeit an einer Vielzahl von Rotorblättern. Durch die Validierung der Messergebnisse mit einem NanoFocus-Gerät sowie die Optimierung der Messpunktverteilung konnten nicht nur die Messgenauigkeit gesteigert, sondern auch die Effizienz des Messprozesses erheblich verbessert werden.

Die erfassten Rauheitsdaten wurden anschließend verwendet, um CFD-Simulationen mit wand-aufgelösten LES- und RANS-Methoden durchzuführen. Diese Simulationen lieferten detaillierte Einblicke in die Auswirkungen von Oberflächenrauheiten auf die turbulenten Strömungsverhältnisse innerhalb der Triebwerkskomponenten. Durch die systematische Analyse des Einflusses von Reynolds-Zahlen und die Entwicklung eines neuen Rauheitsfunktionsmodells konnte gezeigt werden, wie sich unterschiedliche Rauheitsparameter auf die Strömungscharakteristika auswirken. Dies ermöglicht es, den Einfluss von Fertigungs- und Betriebsbedingungen bereits in der Konstruktionsphase zu berücksichtigen und so die aerodynamische Effizienz von Hochdruckkompressoren gezielt zu optimieren.

AP 1.4 erweiterte die Messinfrastruktur Score-P um eine FAIR-konforme Metadatenerfassung und wendet sie am gekoppelten Strömungslöser Hydra an. Durch eine JSON-LD/RDF-basierte Beschreibung wurden Messdaten mit eindeutig identifizierbaren Artefakten (Code-Version, Build-Flags, Umgebung, Input) verknüpft und projektübergreifend auswertbar. Am Anwendungsfall Hydra zeigte sich, dass die Score-P Infrastruktur reale Kopplungs- und MPI-Probleme sichtbar macht und zielgerichtete Optimierungen ermöglicht. So konnten Engpässe (z. B. I/O-Muster) identifiziert, Abhilfe implementiert und messbare Laufzeitgewinne erzielt werden. Upstream-Beiträge zu Score-P und CubeLib sichern Nachhaltigkeit und ermöglichen der Community die unmittelbare Nachnutzung. In Summe schafft AP 1.4 die Basis für reproduzierbare Performance-Analysen gekoppelter HPC-Anwendungen und erleichtert datengetriebene Entscheidungen über Plattformen und Konfigurationen hinweg.

AP3.1 schaffte die Grundlage für ein gemeinsames Daten- und Metadaten-Repository zur Performanceanalyse von Hydra-Simulationen. RDF und SPARQL ermöglichen es, Messdaten und Kontextwissen flexibel zu verknüpfen und reproduzierbar auszuwerten. Damit wurden FAIR-Prinzipien praktisch umgesetzt: Daten sind auffindbar, zugänglich (kuratiert), interoperabel und nachnutzbar. Ein Triplestore plus Objektspeicher erlaubte die zentrale Speicherung von Ergebnisdaten. Langfristig schafft AP3.1 die Basis, um weitere Datenquellen (z. B. Job-Monitoring, Simulationsergebnisse) einzubinden und datengetriebenes Performance-Engineering zu skalieren.

AP3.4 hat erfolgreich einen graph-basierten Modellierungsansatz aus dem Stand der Technik zur Vorhersage von Turbinen-Strömungsfeldern angewandt. Während Graph-Neuronale Netzwerke auf akademischen Datensätzen bereits sehr gute Vorhersageergebnisse liefern konnten, zeigte sich bei der Übertragung auf industrielle Netze mit hoher Auflösung ein erheblicher Ressourcenbedarf, insbesondere im Hinblick auf die Skalierung der Modelle. Zur Adressierung dieser Herausforderung wurden in diesem AP zwei verteilte Lernansätze entwickelt: einer auf Basis isolierter Partitionen ohne Kommunikation und ein zweiter unter Berücksichtigung eines expliziten Halo-Austauschs zur Konsistenz zwischen den Partitionen. Der Ansatz mit

isolierten Partitionen konnte erfolgreich auf ein reales Turbomaschinen-Strömungsfeld angewendet werden und diente damit als erster Proof-of-Concept für den praktischen Einsatz von GNNs in diesem Kontext. Eine Einschränkung der Vorhersage auf Oberflächenwerte ließ auch den Ressourcenbedarf praxistauglicher werden. Die entwickelten Konzepte bilden eine solide Grundlage für zukünftige Arbeiten, in denen sowohl die Anwendung datengetriebener Ersatzmodelle im Design-Workflow untersucht als auch deren Integration in bestehende CFD-Prozesse weiterentwickelt werden kann.

Die gesammelten Erkenntnisse verdeutlichen, dass die Kombination aus präziser Messtechnik, fortschrittlicher numerischer Simulation und datenbasierter Optimierung der Schlüssel für die Triebwerksentwicklung von morgen ist. Insbesondere die Integration von probabilistischen Methoden in den Entwicklungsprozess ermöglicht es, Unsicherheiten systematisch zu bewerten und die Robustheit von Triebwerksdesigns nachhaltig zu verbessern. Dies führt nicht nur zu einer höheren Effizienz und Betriebssicherheit, sondern auch zu einer Reduktion von Entwicklungskosten und -zeiten.

Darüber hinaus leisten die Forschungsergebnisse einen wichtigen Beitrag zur Erreichung der Klimaziele der Luftfahrtindustrie. Durch die Optimierung der aerodynamischen Eigenschaften und die Reduzierung von Fertigungstoleranzen können der Kraftstoffverbrauch und die CO₂-Emissionen signifikant gesenkt werden. Dies ist ein entscheidender Schritt hin zu nachhaltigeren Antriebskonzepten, die den Anforderungen zukünftiger Generationen von Flugzeugen gerecht werden.

Insgesamt zeigt diese Arbeit, dass die hier entwickelten Methoden und Modelle nicht nur einen unmittelbaren Nutzen für die Optimierung bestehender Triebwerke bieten, sondern auch das Potenzial haben, die Entwicklung von innovativen Technologien für die Luftfahrtindustrie der Zukunft maßgeblich zu beeinflussen.

9 Liste der Veröffentlichungen

Im Rahmen der Bearbeitung wurden folgende Artikel veröffentlicht:

- [173] I. Vasilopoulos u. a. „Roughness Investigations on In-Service High-Pressure Compressor Blades – Part I: An Automated Process for High-fidelity Roughness Measurements“. In: *ASME Turbo Expo 2023: Turbine Technical Conference and Exposition*. Boston, MA, USA: American Society of Mechanical Engineers, 2023
- [171] I. Vasilopoulos u. a. „Roughness Investigations on In-Service High-Pressure Compressor Blades – Part II: Roughness Parameterization and CFD-based Modelling of Its Impact on Turbulent Flows“. In: *ASME Turbo Expo 2023: Turbine Technical Conference and Exposition*. Boston, MA, USA: American Society of Mechanical Engineers, 2023
- [142] M. Sander u. a. „Towards Large-Scale Top-Down Microarchitecture Analysis Using the Score-P Framework“. In: *Euro-Par 2024: Parallel Processing Workshops*. Hrsg. von S. Caino-Lores u. a. Cham: Springer Nature Switzerland, 2025, S. 189–200. isbn: 978-3-031-90200-0
- [143] M. Sander u. a. „Working Towards FAIRness in Performance Data“. In: *Proceedings of the 35th Parallel CFD International Conference 2024*. Bd. 69. Schriften des Forschungszentrums Jülich IAS Series. 35th Parallel CFD International Conference 2024, Bonn (Germany), 2 Sep 2024 - 4 Sep 2024. Jülich: Forschungszentrum Jülich GmbH Zentralbibliothek, Verlag, 2. Sep. 2024, S. 192–196
- [161] S. Strönisch u. a. „Flow field prediction on large variable sized 2D point clouds with graph convolution“. In: *Proceedings of the Platform for Advanced Scientific Computing Conference*. PASC '22. New York, NY, USA: Association for Computing Machinery, 12. Juli 2022, S. 1–10. isbn: 978-1-4503-9410-9
- [162] S. Strönisch u. a. „Multi-GPU Approach for Training of Graph ML Models on large CFD Meshes“. In: *AIAA SCITECH 2023 Forum*. American Institute of Aeronautics und Astronautics, 19. Jan. 2023
- [163] S. Strönisch u. a. „Prediction of steady and unsteady flow quantities using Multiscale Graph Neural Networks“. In: *Journal of Turbomachinery* (14. Nov. 2024)

Literaturverzeichnis

- [1] Alicona InfiniteFocus-SensorR25. Accessed 28 December 2022.
- [2] D. Amirante, V. Ganine, N. J. Hills und P. Adami. „A coupling framework for multi-domain modelling and multi-physics simulations“. In: *Entropy* 23 (2021).
- [3] P. Amtsfeld. „Methoden für die beschleunigte aerodynamische Optimierung von Turbinenschaufeln“. Diss. Brandenburgischen Technischen Universität Cottbus-Senftenberg, 2018.
- [4] J. D. Anderson. *A History of Aerodynamics and Its Impact on Flying Machines*. Cambridge Aerospace Series 8. Cambridge: Cambridge university press, 1997. isbn: 978-0-521-45435-3.
- [5] ANSA. ANSA. <https://www.beta-cae.com/ansa.htm>. Accessed: 28 December 2022.
- [6] ANSYS CFX. Zugriff am 28. Dezember 2022.
- [7] B. Aupoix. „A General Strategy to Extend Turbulence Models to Rough Surfaces: Application to Smith’s k-L Model“. In: *Journal of Fluids Engineering* 129.10 (2007), S. 1245–1254.
- [8] S. Back, G. Hobson, S. Song und K. Millsaps. „Effects of Reynolds Number and Surface Roughness Magnitude and Location on Compressor Cascade Performance“. In: *Journal of Turbomachinery* 134.5 (2012), S. 051013.
- [9] T. Backhaus. „Uncertainty Propagation of Real Geometry Effects on Jet Engine Compressor Blisks“. Doktorarbeit. Dresden: Technische Universität Dresden, 2020.
- [10] D. A. Bader, H. Meyerhenke, P. Sanders und D. Wagner. „Graph Partitioning and Graph Clustering“. In: *Contemporary Mathematics*. 10th DIMACS Implementation Challenge Workshop (2012). issn: 1098-3627.
- [11] B. Baldwin und H. Lomax. „Thin-Layer Approximation and Algebraic Model for Separated Turbulent Flows“. In: *16th Aerospace Sciences Meeting*. 1978.

- [12] S. Barwey, R. Balin, B. Lusch, S. Patel, R. Balakrishnan, P. Pal, R. Maulik und V. Vishwanath. „Scalable and Consistent Graph Neural Networks for Distributed Mesh-based Data-driven Modeling“. In: *SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*. SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis. Nov. 2024, S. 1058–1070.
- [13] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. Ballard, J. Gilmer, G. Dahl, A. Vaswani, K. Allen, C. Nash, V. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li und R. Pascanu. *Relational inductive biases, deep learning, and graph networks*. 17. Okt. 2018.
- [14] F. D. A. Belbute-Peres, T. Economon und Z. Kolter. „Combining Differentiable PDE Solvers and Graph Neural Networks for Fluid Flow Prediction“. In: *Proceedings of the 37th International Conference on Machine Learning*. International Conference on Machine Learning. ISSN: 2640-3498. PMLR, 21. Nov. 2020, S. 2402–2411.
- [15] J. Bernander. *Contact-Mode Interfaces in Simcenter STAR-CCM+*. volume, 12. Apr. 2024.
- [16] A. Beschorner, M. Voigt und K. Vogeler. „Monte Carlo Cross Validation for Response Surface Benchmark“. In: *Proceedings of the 12th International Probabilistic Workshop*. 2014.
- [17] C. Bishop. *Pattern Recognition and Machine Learning*. 1. Aufl. New York, NY, USA: Springer, 2006. isbn: 978-0-387-31073-2.
- [18] D. Blechschmidt und D. Mimic. „A Machine Learning Approach for the Prediction of Time-Averaged Unsteady Flows in Turbomachinery“. In: *ASME Turbo Expo 2023*. Turbomachinery Technical Conference and Exposition. Boston, 2023.
- [19] J. Bons. „A Review of Surface Roughness Effects in Gas Turbines“. In: *Journal of Turbomachinery* 132.2 (2010), S. 021004.
- [20] J. Bons, R. Taylor, S. McClain und R. Rivir. „The Many Faces of Turbine Surface Roughness“. In: *Journal of Turbomachinery* 123.4 (2001), S. 739–748.
- [21] P. Bourke. *Polygons, Meshes*. Polygons und meshes. 5. Okt. 2021. url: [http : / / paulbourke . net / geometry / polygonmesh /](http://paulbourke.net/geometry/polygonmesh/) (besucht am 05. 10. 2021).
- [22] T. Bray. *The JavaScript Object Notation (JSON) Data Interchange Format*. RFC 8259. Dez. 2017.
- [23] J. Broenink, D. Brugali, T. Kroeger und B. MacDonald, Hrsg. *Simulation, Modeling, and Programming for Autonomous Robots: 4th International Conference, SIMPAR 2014, Bergamo, Italy, October 20-23, 2014. Proceedings*. 1st ed. 2014. Lecture Notes in Artificial Intelligence 8810. Cham: Springer International Publishing : Imprint: Springer, 2014. 1 S. isbn: 978-3-319-11900-7.
- [24] *Bruker Alicona*. Accessed 28 December 2022.
- [25] J. Bruna, W. Zaremba, A. Szlam und Y. LeCun. „Spectral Networks and Locally Connected Networks on Graphs“. In: *International Conference on Learning Representations (ICLR2014)*. 21. Mai 2014.

- [26] G. Bruni, S. Maleki und S. Krishnababu. „C(NN)FD - Deep Learning Modelling of Multi-Stage Axial Compressor Aerodynamics“. In: ASME Turbo Expo. Memphis, 17. Juni 2025.
- [27] W. J. G. Bräunling. *Flugzeugtriebwerke: Grundlagen, Aero-Thermodynamik, Ideale Und Reale Kreisprozesse, Thermische Turbomaschinen, Komponenten, Emissionen Und Systeme*. 3. Aufl. VDI-Buch. Berlin Heidelberg: Springer-Verlag, 2009. isbn: 978-3-540-76368-0.
- [28] A. Calotoiu, T. Hoefler, M. Poke und F. Wolf. „Using automated performance modeling to find scalability bugs in complex codes“. In: *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. 2013, S. 1–12.
- [29] Y. Cao. „Advancing Physics-Based Simulations: Integrating Conventional and Machine-Learning Approaches for Enhanced Computational Efficiency“. Dissertation. Los Angeles, California: University of California (UCLA), 2024.
- [30] Y. Cao, M. Chai, M. Li und C. Jiang. „Efficient Learning of Mesh-Based Physical Simulation with Bi-Stride Multi-Scale Graph Neural Network“. In: *Proceedings of the 40th International Conference on Machine Learning*. International Conference on Machine Learning. ISSN: 2640-3498. PMLR, 3. Juli 2023, S. 3541–3558.
- [31] C. Colebrook. „Turbulent Flow in Pipes, With Particular Reference to the Transition Region Between the Smooth and Rough Pipe Laws“. In: *J. Inst. Civil Eng.* 11.4 (1939), S. 133–156.
- [32] R. Cyganiak, D. Wood und M. Lanthaler. *RDF 1.1 Concepts and Abstract Syntax*. <https://www.w3.org/TR/rdf11-concepts/>. W3C Recommendation. 2014.
- [33] R. A. Dandekar, M. Cohen und N. Kirkendall. „Sensitive Micro Data Protection Using Latin Hypercube Sampling Technique“. In: *Inference Control in Statistical Databases*. Springer, 2002, S. 117–125.
- [34] R. Danzl, F. Helmli und S. Scherer. „Focus Variation – a Robust Technology for High Resolution Optical 3D Surface Metrology“. In: *Journal of Mechanical Engineering* 57.3 (2011), S. 245–256.
- [35] L. Davidson. „Using Isotropic Synthetic Fluctuations as Inlet Boundary Conditions for Unsteady Simulations“. In: *Advances in Applied Fluid Mechanics* 1.1 (2007), S. 1–35.
- [36] M. De Marchis, B. Milici und E. Napoli. „Large Eddy Simulations on the Effect of the Irregular Roughness Shape on Turbulent Channel Flows“. In: *International Journal of Heat and Fluid Flow* 80 (2019), S. 108494.
- [37] M. De Marchis, D. Saccone, B. Milici und E. Napoli. „Large Eddy Simulations of Rough Turbulent Channel Flows Bounded by Irregular Roughness: Advances Toward a Universal Roughness Correlation“. In: *Flow Turbul. Combust.* 105 (2020), S. 627–648.
- [38] M. Defferrard, X. Bresson und P. Vandergheynst. „Convolutional neural networks on graphs with fast localized spectral filtering“. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS’16. Red Hook, NY, USA: Curran Associates Inc., 5. Dez. 2016, S. 3844–3852. isbn: 978-1-5108-3881-9.

- [39] G. Delvaux, R. Toracchio, E. Boufidi, E. Croner und F. Fontaneto. „The Modelling of Roughness Effect on the Performance of a Controlled Diffusion Airfoil“. In: *ASME Turbo Expo*. GT2022-79522. 2022.
- [40] G. D. C. Dhondt. *The Finite Element Method for Three-Dimensional Thermomechanical Applications*. Hoboken, NJ: Wiley, 2004. 340 S. isbn: 978-0-470-85752-6.
- [41] D. Di Giusto und J. Castagna. „A scalable algorithm for many-body dissipative particle dynamics using multiple general purpose graphic processing units“. In: *Computer Physics Communications* 280 (1. Nov. 2022), S. 108472. issn: 0010-4655.
- [42] F. Diermeier und M. Voigt. *ProSi MANUAL*. 2020.
- [43] R. Dietrich, F. Winkler, A. Knüpfer und W. Nagel. „Pika: Center-wide and job-aware cluster monitoring“. In: *2020 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE. 2020, S. 424–432.
- [44] Eckart Laurien und Herbert Oertel jr. *Numerische Strömungsmechanik: Grundgleichungen Und Modelle - Lösungsmethoden - Qualität Und Genauigkeit*. 5. Aufl. Springer Vieweg, 2013. isbn: 978-3-658-03144-2.
- [45] M. Effendy, Y. Yao, J. Yao und D. Marchant. „DES Study of Blade Trailing Edge Cutback Cooling Performance with Various Lip Thicknesses“. In: *Applied Thermal Engineering* 99 (Apr. 2016), S. 434–445. issn: 13594311.
- [46] D. Eschweiler, M. Wagner, M. Geimer, A. Knüpfer, W. E. Nagel und F. Wolf. „Open Trace Format 2: The Next Generation of Scalable Trace Formats and Support Libraries“. In: *Applications, Tools and Techniques on the Road to Exascale Computing*. 2012.
- [47] M. Färber, D. Lamprecht und Y. Susanti. „AutoRDF2GML: Facilitating RDF Integration in Graph Machine Learning“. In: *The Semantic Web - ISWC 2024 - 23rd International Semantic Web Conference, Baltimore, MD, USA, November 11-15, 2024, Proceedings, Part III*. Hrsg. von G. Demartini, K. Hose, M. Acosta, M. Palmonari, G. Cheng, H. Skaf-Molli, N. Ferranti, D. Hernández und A. Hogan. Bd. 15233. Lecture Notes in Computer Science. Springer, 2024, S. 115–133.
- [48] J. H. Ferziger und M. Perić. *Numerische Strömungsmechanik*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. isbn: 978-3-540-67586-0 978-3-540-68228-8.
- [49] J. H. Ferziger, M. Perić und R. L. Street. *Numerische Strömungsmechanik*. 2., aktualisierte Auflage. Berlin [Heidelberg]: Springer Vieweg, 2020. 663 S. isbn: 978-3-662-46543-1.
- [50] R. A. Fisher. „Statistical Methods for Research Workers“. In: *Breakthroughs in Statistics*. New York: Springer, 1992, S. 66–70.
- [51] K. Flack, M. Schultz, J. Barros und Y. Kim. „Skin-Friction Behavior in the Transitionally-Rough Regime“. In: *International Journal of Heat and Fluid Flow* 61 (2016), S. 21–30.
- [52] A. Florian. „An Efficient Sampling Scheme: Updated Latin Hypercube Sampling“. In: *Probabilistic Engineering Mechanics* 7.2 (Jan. 1992), S. 123–130. issn: 02668920.
- [53] P. Forooghi, A. Stroh, F. Magagnato, S. Jakirlić und B. Frohnappfel. „Toward a Universal Roughness Correlation“. In: *Journal of Fluids Engineering* 139.12 (2017), S. 121201.

- [54] M. Fortunato, T. Pfaff, P. Wirnsberger, A. Pritzel und P. Battaglia. „MultiScale MeshGraph-Nets“. In: ICML 2022 2nd AI for Science Workshop. 21. Juli 2022.
- [55] C. Frey, G. Ashcroft, M. Müller und J. Wellner. „Analysis of Turbomachinery Averaging Techniques“. In: *Journal of Turbomachinery* 145.51006 (25. Nov. 2022). issn: 0889-504X.
- [56] S. Gbadebo, T. Hynes und N. Cumpsty. „Influence of Surface Roughness on Three-Dimensional Separation in Axial Compressors“. In: *J. Turbomach.* 126.4 (2004), S. 455–463.
- [57] M. Geimer, F. Wolf, B. J. Wylie, E. Ábrahám, D. Becker und B. Mohr. „The Scalasca performance toolset architecture“. In: *Concurrency and computation: Practice and experience* 22.6 (2010), S. 702–719.
- [58] A. Ghenaiet, R. Elder und S. Tan. „Particles Trajectories Through an Axial Fan and Performance Degradation due to Sand Ingestion“. In: *ASME Turbo Expo 2001*. 2001, 2001–GT–0497.
- [59] P. Gilge, A. Kellersmann, J. Friedrichs und J. Seume. „Surface Roughness of Real Operationally Used Compressor Blade and Blisk“. In: *Proc. Inst. Mech. Eng. G: J. Aerosp. Eng.* 233.14 (2019), S. 5321–5330.
- [60] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals und G. E. Dahl. „Neural message passing for Quantum chemistry“. In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML’17. Sydney, NSW, Australia: JMLR.org, 6. Aug. 2017, S. 1263–1272.
- [61] J. Goeing, H. Seehausen, V. Pak, S. Lueck, J. Seume und J. Friedrichs. „Influence of Combined Compressor and Turbine Deterioration on the Overall Performance of a Jet Engine Using RANS Simulation and Pseudo Bond Graph Approach“. In: *J. Glob. Power Propuls. Soc.* 4 (2020), S. 296–308.
- [62] I. Goodfellow, Y. Bengio und A. Courville. *Deep Learning*. 1. Aufl. The MIT Press, 18. Nov. 2016. isbn: 978-0-262-03561-3.
- [63] M. Goodhand, K. Walton, L. Blunt, H. Lung, R. Miller und R. Marsden. „The Limitations of Using ‘Ra’ to Describe Surface Roughness“. In: *J. Turbomach.* 138.10 (2016), S. 101003.
- [64] A. Hamed, W. Tabakoff, R. Rivir, K. Das und P. Arora. „Turbine Blade Surface Deterioration by Erosion“. In: *Journal of Turbomachinery* 127.3 (2005), S. 445–452.
- [65] W. L. Hamilton, R. Ying und J. Leskovec. „Inductive representation learning on large graphs“. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS’17. Red Hook, NY, USA: Curran Associates Inc., 4. Dez. 2017, S. 1025–1035. isbn: 978-1-5108-6096-4.
- [66] K. Heinze. *HolisTurb: Realgeometrieeffekte von Hochdruckturbinenschaufeln*. Abschlussbericht. Dresden: TU Dresden, 2013.
- [67] K. Heinze. *Realgeometrieeffekte von Hochdruckturbinenschaufeln*. Abschlussbericht. TU Dresden, 2013, S. 121.

- [68] K. Heinze, M. Meyer, J. Scharfenstein, M. Voigt und K. Vogeler. „A Parametric Model for Probabilistic Analysis of Turbine Blades Considering Real Geometric Effects“. In: *CEAS Aeronautical Journal* 1.5 (5 2014), S. 41–51.
- [69] H. Herwig und A. Moschallski. *Wärmeübertragung: Physikalische Grundlagen - Illustrierende Beispiele - Übungsaufgaben mit Musterlösungen*. Wiesbaden: Springer Fachmedien Wiesbaden, 2014. isbn: 978-3-658-06207-1 978-3-658-06208-8.
- [70] D. Hines und P. Bekemeyer. „Graph neural networks for the prediction of aircraft surface pressure distributions“. In: *Aerospace Science and Technology* 137 (Juni 2023), S. 108268. issn: 12709638.
- [71] C. Hinterberger, J. Fröhlich und W. Rodi. „2D and 3D Turbulent Fluctuations in Open Channel Flow with $R_\tau = 590$ Studied by Large Eddy Simulation“. In: *Flow Turbulence and Combustion* 80 (2008), S. 225–253.
- [72] S. Holzhausen. „Erfassungsplanung Nach Dem Optimierungsprinzip Am Beispiel Des Streifenprojektionsverfahrens“. Doktorarbeit. Dresden: Technische Universität Dresden, 2015.
- [73] J. Hsieh. *Computed Tomography: Principles, Design, Artifacts, and Recent Advances*. 2nd ed. [SPIE PM 188]. Bellingham, Wash: SPIE Press, 2009. 556 S. isbn: 978-0-470-56353-3 978-0-8194-7533-6.
- [74] L. Högner. *Abschlussbericht LuFo: PERFEKT - Ganzheitliche Probabilistische Analyse von Hochdruckturbinen-Rotorschaukeln Unter Verwendung von CT-Daten*. Abschlussbericht. Dresden, 17. Aug. 2017.
- [75] L. Högner. *Abschlussbericht LuFo V: Prestige - Methoden Zur Integration Der Variabilität Komplexer Freiformgeometrie in Multidisziplinäre Simulationen*. Dresden, 13. Jan. 2022.
- [76] L. Högner, S. Knebel, M. Voigt, R. Mailach und M. Meyer. „Quantification of X-Ray Measurement Uncertainty Based on Optical Measurement Data of Turbine Blades“. In: *Proceedings of ASME Turbo Expo 2017*. GT2017-63704. Juni 2017.
- [77] L. Högner, M. Voigt, R. Mailach, M. Meyer und U. Gerstberger. „Probabilistic Finite Element Analysis of Cooled High-Pressure Turbine Blades—Part A: Holistic Description of Manufacturing Variability“. In: *Journal of Turbomachinery* 142.10 (10 Okt. 2020). issn: 0889-504X.
- [78] T. Ilsche, R. Schöne, M. Bielert, A. Gocht und D. Hackenberg. „Io2s — Multi-core System and Application Performance Analysis for Linux“. In: *2017 IEEE International Conference on Cluster Computing (CLUSTER)*. 2017.
- [79] *ISO 25178-606:2015: Geometrical Product Specification (GPS) – Surface Texture: Areal – Part 606: Nominal Characteristics of Non-Contact (Focus Variation) Instruments*. DIN, Dez. 2016.
- [80] P. Jackson. „On the Displacement Height in the Logarithmic Velocity Profile“. In: *Journal of Fluid Mechanics* 111 (1981), S. 15–25.
- [81] J. Jiménez. „Turbulent Flows Over Rough Walls“. In: *Annual Review of Fluid Mechanics* 36 (2004), S. 173–196.

- [82] J. Joo, G. Medic und O. Sharma. „Large-Eddy Simulation Investigation of Impact of Roughness on Flow in a Low-Pressure Turbine“. In: *ASME Turbo Expo 2016*. 2016.
- [83] M. Jouybari, J. Yuan, G. Brereton und M. Murillo. „Data-Driven Prediction of the Equivalent Sand-Grain Height in Rough-Wall Turbulent Flows“. In: *J. Fluid Mech.* 912 (2021).
- [84] B. Kader. „Temperature and Concentration Profiles in Fully Turbulent Boundary Layers“. In: *International Journal of Heat and Mass Transfer* 24.9 (Sep. 1981), S. 1541–1544. issn: 00179310.
- [85] H. Kahmen. *Vermessungskunde*. Reprint 2019. Berlin: De Gruyter, 2020. isbn: 978-3-11-087406-8.
- [86] G. Karypis und V. Kumar. „METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices“. In: *Computer Science & Engineering (CS&E) Technical Reports* (1997).
- [87] A. Kemper und A. Eickler. *Datenbanksysteme - Eine Einführung*. De Gruyter Oldenbourg, 2015. isbn: 9783110443752.
- [88] G. King. *An introduction to the dataverse network as an infrastructure for data sharing*. 2007.
- [89] T. N. Kipf und M. Welling. „Semi-Supervised Classification with Graph Convolutional Networks“. In: *International Conference on Learning Representations (ICLR17)*. 6. Feb. 2017.
- [90] B. Klein. *FEM: Grundlagen und Anwendungen der Finite-Element-Methode im Maschinen- und Fahrzeugbau*. Springer-Verlag, Feb. 2012. isbn: 978-3-8348-2134-8.
- [91] A. Knüpfer, H. Brunst, J. Doleschal, M. Jurenz, M. Lieber, H. Mickler, M. S. Müller und W. E. Nagel. „The vampir performance analysis tool-set“. In: *Tools for High Performance Computing: Proceedings of the 2nd International Workshop on Parallel Tools for High Performance Computing, July 2008, HLRS, Stuttgart*. 2008.
- [92] A. Knüpfer, C. Rössel, D. an Mey, S. Biersdorff, K. Diethelm, D. Eschweiler, M. Geimer, M. Gerndt, D. Lorenz, A. Malony, W. E. Nagel, Y. Oleynik, P. Philippen, P. Saviankou, D. Schmidl, S. Shende, R. Tschüter, M. Wagner, B. Wesarg und F. Wolf. „Score-P: A Joint Performance Measurement Run-Time Infrastructure for Periscope, Scalasca, TAU, and Vampir“. In: *Tools for High Performance Computing 2011*. 2012.
- [93] O. Kolditz. *Computational Methods in Environmental Fluid Mechanics*. Berlin: Springer, 2011. isbn: 978-3-642-07683-1.
- [94] D. P. Kroese, T. Taimre und Z. I. Botev. *Handbook of Monte Carlo Methods*. Bd. 706. John Wiley & Sons, 2013.
- [95] P. Krysl. *Finite Element Modeling with Abaqus and Python for Thermal and Stress Analysis*. San Diego: Pressure Cooker Press, 2017.
- [96] K. S. Kundert. *NestedText*. <https://github.com/KenKundert/nestedtext>. Accessed: 2025-07-14. 2020.
- [97] A. Lange. „Probabilistische Performanceuntersuchung Eines Hochdruckverdichters Unter Berücksichtigung Geometrischer Variabilität“. Diss. Dresden: TU Dresden, 2014.

- [98] B. Lapworth. „HYDRA-CFD: A Framework for Collaborative CFD Development“. In: *International Conference on Scientific and Engineering Computation (IC-SEC)*. 2004.
- [99] L. Lapworth. „Hydra-CFD: a framework for collaborative CFD development“. In: *International conference on scientific and engineering computation (IC-SEC)*. Bd. 30. 2004.
- [100] R. Leach. *Fundamental Principles of Engineering Nanometrology*. William Andrew, 2010.
- [101] Lee Wen Yao. „Manufacturing Variability in High Pressure Turbine Rotor Blades“. University of Cambridge, Nov. 2019.
- [102] J. Li, Y. Li, T. Liu, D. Zhang und Y. Xie. „Multi-fidelity graph neural network for flow field data fusion of turbomachinery“. In: *Energy* 285 (15. Dez. 2023), S. 129405. issn: 0360-5442.
- [103] J. Li, T. Liu, Y. Wang und Y. Xie. „Integrated graph deep learning framework for flow field reconstruction and performance prediction of turbomachinery“. In: *Energy* 254 (1. Sep. 2022), S. 124440. issn: 0360-5442.
- [104] D. Ma. „Novel Casting Processes for Single-Crystal Turbine Blades of Superalloys“. In: *Frontiers of Mechanical Engineering* 13.1 (1 März 2018), S. 3–16. issn: 2095-0233, 2095-0241.
- [105] A. Malathi, M. Nardini, A. Vaid, N. Vadlamani und R. Sandberg. „Profile Loss Reduction of High-Lift Turbine Blades With Rough and Ribbed Surfaces“. In: *Journal of Turbomachinery* 145.2 (2023), S. 021001.
- [106] M. Marxer, C. Bach und C. P. Keferstein. *Fertigungsmesstechnik: Alles Zu Messunsicherheit, Konventioneller Messtechnik Und Multisensorik*. Wiesbaden: Springer Fachmedien Wiesbaden, 2021. isbn: 978-3-658-34168-8.
- [107] M. D. McKay, R. J. Beckman und W. J. Conover. „A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code“. In: *Technometrics* 21.2 (2 1979), S. 239–245. issn: 0040-1706.
- [108] F. R. Menter. „Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications“. In: *AIAA Journal* 32.8 (1994), S. 1598–1605. issn: 0001-1452, 1533-385X.
- [109] F. Menter. „Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications“. In: *AIAA Journal* 32.8 (1994), S. 1598–1605.
- [110] S. Meynet, A. Barge, V. Moureau, G. Balarac, G. Lartigue und A. Hadjadj. „Roughness-Resolved LES of Additive Manufacturing-Like Channel Flows“. In: *ASME Turbo Expo 2022*. 2022.
- [111] J. Mihalyovics, B. Christian, D. Peitsch, I. Vasilopoulos und M. Meyer. „Aerodynamic and Experimental Investigations on Optimized 3D Compressor Airfoils“. In: *ASME Turbo Expo*. GT2018-76826. 2018.
- [112] K. Millsaps, J. Baker und J. Patterson. „Detection and Localization of Fouling in a Gas Turbine Compressor From Aerothermodynamic Measurements“. In: *ASME Turbo Expo 2004*. 2004, GT2004–54173.
- [113] Minitab. *Erste Schritte Mit Der Minitab Statistical Software*. Minitab, 2024.
- [114] Minitab. *Methoden Für Identifikation Der Verteilung*. 2024.

- [115] C. Misev. „Development and Optimization of an Implicit CFD Solver in Hydra“. Diss. UK: University of Surrey, 2017.
- [116] P. Moinier, J. Müller und M. Giles. „Edge-Based Multigrid and Preconditioning for Hybrid Grids“. In: *AIAA Journal* 40.10 (2002), S. 1954–1960.
- [117] R. Moser, J. Kim und N. Mansour. „Direct Numerical Simulation of Turbulent Channel Flow up to $R_\tau = 590$ “. In: *Physics of Fluids* 11.4 (1999), S. 943–945.
- [118] G. R. Mudalige, I. Z. Reguly, A. Prabhakar, D. Amirante, L. Lapworth und S. A. Jarvis. „Towards Virtual Certification of Gas Turbine Engines With Performance-Portable Simulations“. In: *2022 IEEE International Conference on Cluster Computing (CLUSTER)*. 2022 IEEE International Conference on Cluster Computing (CLUSTER). ISSN: 2168-9253. Sep. 2022, S. 206–217.
- [119] *NanoFocus ((μ))surf custom*. Accessed 28 December 2022.
- [120] E. Napoli, V. Armenio und M. De Marchis. „The Effect of the Slope of Irregularly Distributed Roughness Elements on Turbulent Wall-Bounded Flow“. In: *J. Fluid Mech.* 613 (2008), S. 385–394.
- [121] NASA Advanced Supercomputing Division. *NAS Parallel Benchmarks, Multi-Zone Versions*. <https://www.nas.nasa.gov/software/npb.html>. 2003.
- [122] A. Nath. „Laser Drilling of Metallic and Nonmetallic Substrates“. In: *Comprehensive Materials Processing*. Elsevier, 2014, S. 115–175. isbn: 978-0-08-096533-8.
- [123] F. Nicoud, H. Toda, O. Cabrit, S. Bose und J. Lee. „Using Singular Values to Build a Subgrid-Scale Model for Large Eddy Simulations“. In: *Physics of Fluids* 23 (2011), S. 085106.
- [124] J. Nikuradse. *Strömungsgesetze in Rauhen Rohren*. Techn. Ber. 361. VDI-Forschungsheft, 1933.
- [125] J. O’Connor, J.-P. Aumasson, S. Neves und Z. Wilcox-O’Hearn. *BLAKE3: One Algorithm, One Hash Function, Many Applications*. <https://github.com/BLAKE3-team/BLAKE3>. Accessed: 2025-02-18. 2020.
- [126] K. Oono und T. Suzuki. „Graph Neural Networks Exponentially Lose Expressive Power for Node Classification“. In: *International Conference on Learning Representations*. 23. Sep. 2019.
- [127] S. Patankar. *Numerical Heat Transfer and Fluid Flow*. Boca Raton, FL: CRC Press, 2018. isbn: 978-1-4822-3421-3.
- [128] K. Pearson. „Notes on Regression and Inheritance in the Case of Two Parents“. In: *Proceedings of the Royal Society of London* 58 (1895), S. 240–242.
- [129] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez und P. Battaglia. „Learning Mesh-Based Simulation with Graph Networks“. In: *International Conference on Learning Representations*. International Conference on Learning Representations. 12. Jan. 2021.
- [130] J. Pleming und R. Manteufel. „Replicated Latin Hypercube Sampling“. In: *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. Austin, Texas: American Institute of Aeronautics and Astronautics, 18. Apr. 2005. isbn: 978-1-62410-065-9.

- [131] A. Prabhakar, C. Goddard, D. Amirante, I. Reguly, A. Gerstenberger, J. Suhrmann, S. Jarvis, L. Lapworth und G. Mudalige. „Virtual Certification of Gas Turbine Engines – Visualizing the DLR Rig250 Compressor“. Preprint submitted to Parallel Computing. 2023.
- [132] A. Prots, L. Schlüter, M. Voigt, R. Mailach und M. Meyer. „Sensitivity Analysis of Performance Parameters of a Compressor Blade With Correlated Profile Parameters“. In: *ASME Turbo Expo 2023: Turbomachinery Technical Conference and Exposition*. Boston, MA: American Society of Mechanical Engineers, 2023.
- [133] E. PRUDHOMMEAUX. „SPARQL query language for RDF“. In: <http://www.w3.org/TR/rdf-sparql-query/> (2008).
- [134] *Python SciPy RBF*. Accessed 28 December 2022.
- [135] M. Rieth, F. Proch, O. Stein, M. Pettit und A. Kempf. „Comparison of the Sigma and Smagorinsky LES Models for Grid Generated Turbulence and a Channel Flow“. In: *Computers and Fluids* 99.22 (2014), S. 172–181.
- [136] M. Ripepi, M. J. Verveld, N. W. Karcher, T. Franz, M. Abu-Zurayk, S. Görtz und T. M. Kier. „Reduced-order models for aerodynamic applications, loads and MDO“. In: *CEAS Aeronautical Journal* 9.1 (1. März 2018), S. 171–193. issn: 1869-5590.
- [137] P. Russo. *Handbook of X-ray Imaging: Physics and Technology*. Hrsg. von P. Russo. 1. Aufl. CRC Press, 14. Dez. 2017. isbn: 978-1-351-22825-1.
- [138] E. Saliby. „Descriptive Sampling: A Better Approach to Monte Carlo Simulation“. In: *Journal of the Operational Research Society* 41.12 (1990), S. 1133–1142.
- [139] C. Sallaberry, J. Helton und S. Hora. „Extension of Latin Hypercube Samples with Correlated Variables“. In: *Reliability Engineering & System Safety* 93.7 (7 Juli 2008), S. 1047–1059. issn: 09518320.
- [140] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec und P. W. Battaglia. „Learning to simulate complex physics with graph networks“. In: *Proceedings of the 37th International Conference on Machine Learning*. Bd. 119. ICML’20. JMLR.org, 13. Juli 2020, S. 8459–8468.
- [141] A. Sanchez-Gonzalez, N. Heess, J. T. Springenberg, J. Merel, M. Riedmiller, R. Hadsell und P. Battaglia. „Graph Networks as Learnable Physics Engines for Inference and Control“. In: *Proceedings of the 35th International Conference on Machine Learning*. International Conference on Machine Learning. ISSN: 2640-3498. PMLR, 3. Juli 2018, S. 4470–4479.
- [142] M. Sander, H. Tröpgen, C. von Elm und R. Schöne. „Towards Large-Scale Top-Down Microarchitecture Analysis Using the Score-P Framework“. In: *Euro-Par 2024: Parallel Processing Workshops*. Hrsg. von S. Caino-Lores, D. Zeinalipour, T. D. Doudali, D. E. Singh, G. E. M. Garzón, L. Sousa, D. Andrade, T. Cucinotta, D. D’Ambrosio, P. Diehl, M. F. Dolz, A. Jukan, R. Montella, M. Nardelli, M. Garcia-Gasulla und S. Neuwirth. Cham: Springer Nature Switzerland, 2025, S. 189–200. isbn: 978-3-031-90200-0.

- [143] M. Sander, W. R. Williams und B. Wesarg. „Working Towards FAIRness in Performance Data“. In: *Proceedings of the 35th Parallel CFD International Conference 2024*. Bd. 69. Schriften des Forschungszentrums Jülich IAS Series. 35th Parallel CFD International Conference 2024, Bonn (Germany), 2 Sep 2024 - 4 Sep 2024. Jülich: Forschungszentrum Jülich GmbH Zentralbibliothek, Verlag, 2. Sep. 2024, S. 192–196.
- [144] P. Saviankou, M. Knobloch, A. Visser und B. Mohr. „Cube v4: From performance report explorer to performance analysis tool“. In: *Procedia Computer Science* 51 (2015).
- [145] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner und G. Monfardini. „The Graph Neural Network Model“. In: *IEEE Transactions on Neural Networks* 20.1 (Jan. 2009), S. 61–80. issn: 1941-0093.
- [146] B. Schembera und D. Iglezakis. „The genesis of EngMeta-a metadata model for research data in computational engineering“. In: *Research Conference on Metadata and Semantics Research*. Springer. 2018, S. 127–132.
- [147] L. Schlüter, M. Voigt, R. Mailach, R. Schmidt und B. Becker. „Probabilistic CFD Analysis of a High-Pressure Compressor Under Consideration of Manufacturing and In-Service Variability“. In: *Draft submitted to ASME Turbo Expo* (2023). GT2023-100528.
- [148] R. Schmidt. *Extension of Latin Hypercube Samples While Maintaining the Correlation Structure*. 1. Aufl. Proceedings of the 12th International Probabilistic Workshop. DE: VDG Weimar, 25. Nov. 2014.
- [149] M. Schuth und W. Buerakov. *Handbuch Optische Messtechnik: Praktische Anwendungen Für Entwicklung, Versuch, Fertigung Und Qualitätssicherung*. München: Carl Hanser Verlag GmbH & Co. KG, 2017. isbn: 978-3-446-43661-9.
- [150] S. Shahpar und S. Caloni. „Aerodynamic Optimization of High-Pressure Turbines for Lean-Burn Combustion System“. In: *J. Eng. Gas Turbines Power* 135.5 (2013), S. 055001.
- [151] D. Singh, A. Hamed und W. Tabakoff. „Simulation of Performance Deterioration in Eroded Compressors“. In: *ASME Turbo Expo 1996: International Gas Turbine and Aeroengine Congress and Exhibition*. Birmingham, UK: American Society of Mechanical Engineers, 1996, V001T01A112.
- [152] S. S. Skiena. *The Algorithm Design Manual*. Texts in Computer Science. Cham: Springer International Publishing, 2020. isbn: 978-3-030-54255-9 978-3-030-54256-6.
- [153] S. D. I. Software. *Simcenter STAR-CCM+ User Guide, Version 2302*. Siemens, 2023.
- [154] M. Sosnowski, J. Krzywanski, K. Grabowska und R. Gnatowska. „Polyhedral Meshing in Numerical Analysis of Conjugate Heat Transfer“. In: *EPJ Web of Conferences* 180 (2018). Hrsg. von P. Dančová, S. 02096. issn: 2100-014X.
- [155] P. Spalart und S. Allmaras. „A One-Equation Turbulence Model for Aerodynamic Flows“. In: *30th Aerospace Sciences Meeting and Exhibit*. 30th Aerospace Sciences Meeting and Exhibit. Reno,NV,U.S.A.: American Institute of Aeronautics and Astronautics, 6. Jan. 1992.
- [156] P. R. Spalart und S. R. Allmaras. „A one-equation turbulence model for aerodynamic flows“. In: *Recherche Aerospatiale* 1 (1994), S. 5–21.

- [157] D. Spalding. „A Single Formula for the 'Law of the Wall'". In: *Journal of Applied Mechanics* 28.3 (1961), S. 455–458.
- [158] C. Spearman. „The Proof and Measurement of Association between Two Things". In: *The American Journal of Psychology* 100.3/4 (3/4 1987), S. 441–471. issn: 0002-9556.
- [159] M. Sporny, D. Longley, G. Kellogg, M. Lanthaler und N. Lindström. „JSON-LD 1.0". In: *W3C recommendation* 16 (2014), S. 41.
- [160] P. Steinke. *Finite-Elemente-Methode: Rechnergestützte Einführung*. Springer-Verlag, Juli 2012. isbn: 978-3-642-29505-8.
- [161] S. Strönisch, M. Meyer und C. Lehmann. „Flow field prediction on large variable sized 2D point clouds with graph convolution". In: *Proceedings of the Platform for Advanced Scientific Computing Conference*. PASC '22. New York, NY, USA: Association for Computing Machinery, 12. Juli 2022, S. 1–10. isbn: 978-1-4503-9410-9.
- [162] S. Strönisch, M. Sander, M. Meyer und A. Knüpfer. „Multi-GPU Approach for Training of Graph ML Models on large CFD Meshes". In: *AIAA SCITECH 2023 Forum*. American Institute of Aeronautics und Astronautics, 19. Jan. 2023.
- [163] S. Strönisch, M. Sander, M. Meyer und A. Knüpfer. „Prediction of steady and unsteady flow quantities using Multiscale Graph Neural Networks". In: *Journal of Turbomachinery* (14. Nov. 2024).
- [164] H. Sun, J. Ma, Z. Wang und L. Cao. „The Research on Compressor Performance Degradation Caused by Surface Roughness Enlargement due to Corrosion in Marine Environments". In: *ASME Turbo Expo 2019*. 2019, GT2019–90784.
- [165] Symscape. *Polyhedral, Tetrahedral, and Hexahedral Mesh Comparison*. 25. Feb. 2013.
- [166] E. Syverud und L. Bakken. „The Impact of Surface Roughness on Axial Compressor Performance Deterioration". In: *ASME Turbo Expo 2006*. 2006.
- [167] R. Taylor. „Surface Roughness Measurements on Gas Turbine Blades". In: *Journal of Turbomachinery* 112.2 (1990), S. 175–180.
- [168] M. Thakkar, A. Busse und N. Sandham. „Direct Numerical Simulation of Turbulent Channel Flow Over a Surrogate for Nikuradse-Type Roughness". In: *J. Fluid Mech.* 837 (2018).
- [169] *Universal Robots*. Accessed 28 December 2022.
- [170] *Universal Robots UR3 e-Series Cobot*. Accessed 28 December 2022.
- [171] I. Vasilopoulos, P. Adami, M. Voigt, M. Meyer und R. Mailach. „Roughness Investigations on In-Service High-Pressure Compressor Blades – Part II: Roughness Parameterization and CFD-based Modelling of Its Impact on Turbulent Flows". In: *ASME Turbo Expo 2023: Turbine Technical Conference and Exposition*. Boston, MA, USA: American Society of Mechanical Engineers, 2023.
- [172] I. Vasilopoulos, P. Adami, M. Voigt, M. Meyer und R. Mailach. *Roughness Investigations on in-Service High-Pressure Compressor Blades–Part II: Roughness Parameterization and CFD-based Modelling of its Impact on Turbulent Flows*. ASME Turbo Expo 2023, GT2023-101157. 2023.

- [173] I. Vasilopoulos, M. Rostamian, M. Voigt, M. Meyer und R. Mailach. „Roughness Investigations on In-Service High-Pressure Compressor Blades – Part I: An Automated Process for High-fidelity Roughness Measurements“. In: *ASME Turbo Expo 2023: Turbine Technical Conference and Exposition*. Boston,MA,USA: American Society of Mechanical Engineers, 2023.
- [174] I. Vasilopoulos, M. Rostamian, M. Voigt, M. Meyer und R. Mailach. *Roughness Investigations on in-Service High-Pressure Compressor Blades–Part I: An Automated Process for High-Fidelity Roughness Measurements*. ASME Turbo Expo 2023, GT2023-100809. 2023.
- [175] VDI e. V., Hrsg. *VDI-Wärmeatlas*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. isbn: 978-3-642-19980-6 978-3-642-19981-3.
- [176] H. K. Versteeg und W. Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Pearson Education, 2007. 530 S. isbn: 978-0-13-127498-3.
- [177] K. Walton, L. Blunt, L. Fleming, M. Goodhand und H. Lung. „Areal Parametric Characterisation of Ex-Service Compressor Blade Leading Edges“. In: *Wear* 321 (2014), S. 79–86.
- [178] Z. Wang, X. Chi, T. Shih und J. Bons. „Direct Simulation of Surface Roughness Effects with a RANS and DES Approach on Viscous Adaptive Cartesian Grids“. In: *34th AIAA Fluid Dynamics Conference and Exhibit*. AIAA 2004-2420. 2004.
- [179] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne u. a. „The FAIR Guiding Principles for scientific data management and stewardship“. In: *Scientific data* 3.1 (2016), S. 1–9.
- [180] K. Zangl, R. Danzl und F. Helml. „Vertical Focus Probing and the Use in Dimensional Metrology“. In: (2019).
- [181] L. Zhao und L. Akoglu. „PairNorm: Tackling Oversmoothing in GNNs“. In: *International Conference on Learning Representations*. 2020.
- [182] O. C. Zienkiewicz und R. L. Taylor. *The Finite Element Method*. 5th ed. Oxford ; Boston: Butterworth-Heinemann, 2000. 3 S. isbn: 978-0-7506-5049-6 978-0-7506-5055-7 978-0-7506-5050-2.
- [183] O. C. Zienkiewicz und R. L. Taylor. *The Finite Element Method*. 5th ed. Oxford Auckland Boston [etc.]: Butterworth-Heinemann, 2000. isbn: 978-0-7506-5049-6 978-0-7506-5055-7 978-0-7506-5050-2.
- [184] O. C. Zienkiewicz und R. L. Taylor. *The Finite Element Method*. 5th ed. Oxford Auckland Boston [etc.]: Butterworth-Heinemann, 2000. isbn: 978-0-7506-5049-6 978-0-7506-5055-7 978-0-7506-5050-2.
- [185] ZIH, TU Dresden. *HPC Ressources - ZIH HPC Compendium*. 2025.