



# Practical issues in the optimization of CFD based engineering problems

## Dissertation

zur Erlangung des akademischen Grades

**Doktoringenieur**  
**(Dr.-Ing.)**

von M.Sc. László Daróczy

geb. am 29. Januar 1988 in Miskolc

genehmigt durch die Fakultät für Verfahrens- und Systemtechnik  
der Otto-von-Guericke-Universität Magdeburg

Promotionskommission: Prof. Dr.-Ing. Dominique Thévenin (Vorsitz)  
Priv.-Doz. Dr.-Ing. Gábor Janiga (Gutachter)  
Prof. Dr. Sebastian Sager (Gutachter)  
Prof. Dr. Jari Härmäläinen (Gutachter)

eingereicht am: 01. Juli 2016

Promotionskolloquium am: 04. November 2016

# Contents

<b>Abstract</b>	<b>6</b>
<b>Kurzzusammenfassung</b>	<b>7</b>
<b>Acknowledgement</b>	<b>8</b>
<b>Nomenclature</b>	<b>9</b>
<b>Introduction</b>	<b>14</b>
<b>1 Introduction to optimization</b>	<b>16</b>
1.1 Mathematical optimization . . . . .	16
1.1.1 Optima . . . . .	17
1.1.2 Classification of optimization problems . . . . .	17
1.1.3 Classification of optimization methods . . . . .	19
1.2 Optimization in engineering practice . . . . .	22
1.2.1 Optimization of CFD-based problems . . . . .	22
1.2.2 Governing equation in CFD . . . . .	23
1.2.3 Chosen optimization algorithms for the present study . . . . .	23
1.3 Optimization Algorithm Library++ (OPAL++) . . . . .	24
1.3.1 Feasibility and dominance in OPAL++ . . . . .	24
1.3.2 Automatization of workflow . . . . .	26
1.3.3 Initialization of the first generation . . . . .	28
1.3.4 Optimization algorithms . . . . .	31
1.3.5 Advanced post-processing methods . . . . .	33
<b>2 Optimization of a heat-exchanger - Analyzing the Pareto front</b>	<b>35</b>
2.1 Aim of the analysis . . . . .	35
2.1.1 Optimization of heat exchangers in the literature . . . . .	36
2.2 Setup of CFD model . . . . .	37
2.2.1 Tube bank heat exchanger configuration . . . . .	37
2.2.2 Validation of the CFD model . . . . .	38
2.3 Setup of the optimization . . . . .	40
2.3.1 Design variables . . . . .	40
2.3.2 Constraints . . . . .	41
2.3.3 Objective functions . . . . .	42
2.3.4 Automation . . . . .	43

2.3.5	Optimization settings . . . . .	43
2.4	Results . . . . .	44
2.4.1	Comparing Pareto fronts . . . . .	44
2.4.2	Cyclic vs. normal version . . . . .	44
2.4.3	<i>Assym</i> vs. <i>Sym<sub>1</sub></i> vs. <i>Sym<sub>2</sub></i> . . . . .	46
2.4.4	Analysis of the Pareto front . . . . .	47
2.5	Conclusions and summary . . . . .	50
<b>3</b>	<b>Extending the heat exchanger example - What is the impact of parameterization?</b>	<b>51</b>
3.1	Setup of CFD model . . . . .	51
3.1.1	Tube bank heat exchanger configuration . . . . .	51
3.1.2	Physical modeling . . . . .	52
3.1.3	Spatial discretization . . . . .	53
3.1.4	Mesh independency . . . . .	53
3.2	Setup of the optimization . . . . .	54
3.2.1	Design variables . . . . .	54
3.2.2	Constraints . . . . .	55
3.2.3	Objective functions . . . . .	56
3.2.4	Automation . . . . .	56
3.2.5	Optimization settings . . . . .	57
3.3	Results . . . . .	57
3.3.1	Efficiency of parameterizations . . . . .	57
3.3.2	Effect of domain size . . . . .	57
3.4	Summary and conclusions . . . . .	59
<b>4</b>	<b>VAWT blade optimization</b>	<b>60</b>
4.1	Introduction to wind energy . . . . .	61
4.1.1	Importance of wind energy . . . . .	61
4.1.2	Description of wind . . . . .	61
4.1.3	Classification of wind turbines . . . . .	63
4.1.4	Darrieus wind turbines . . . . .	64
4.1.5	Physics of the H-Darrieus wind turbine . . . . .	65
4.2	Numerical validation of the model . . . . .	70
4.2.1	Performance evaluation of Darrieus wind turbines . . . . .	70
4.2.2	CFD evaluation of Darrieus wind turbines in the scientific literature	70
4.2.3	Limitations in the comparison of numerical and experimental investigations . . . . .	71
4.2.4	Computational method . . . . .	72
4.2.5	CFD setup . . . . .	77
4.2.6	Systematic numerical analysis . . . . .	80
4.2.7	Experimental validation process . . . . .	88
4.2.8	Conclusions and summary . . . . .	93
4.3	Optimization of a H-Darrieus rotor blade . . . . .	95
4.3.1	Introduction to the current analysis . . . . .	95
4.3.2	Setup of CFD Model . . . . .	96

4.3.3	Setup of the optimization . . . . .	98
4.3.4	Results . . . . .	101
4.3.5	Possible improvements . . . . .	109
4.3.6	Conclusions and summary . . . . .	109
<b>5</b>	<b>Optimization of VAWT blades revisited - Extension to 3D</b>	<b>111</b>
5.1	Blade tips of Darrieus rotors . . . . .	112
5.2	Setup of CFD Model . . . . .	112
5.2.1	Physical model . . . . .	113
5.2.2	Boundary conditions . . . . .	113
5.2.3	Spatial discretization . . . . .	113
5.2.4	Multi-fidelity approach . . . . .	113
5.2.5	Validation of the model . . . . .	115
5.3	Setup of the optimization . . . . .	118
5.3.1	Parameterization . . . . .	118
5.3.2	Automated workflow . . . . .	120
5.4	Results . . . . .	120
5.4.1	Performance coefficient far away from the wingtip . . . . .	121
5.4.2	Performance coefficient of the complete rotor . . . . .	122
5.4.3	Effect of the wingtip on the distribution of the energy generation . . . . .	122
5.4.4	Size of wingtip vortex . . . . .	123
5.5	Conclusions and summary . . . . .	124
<b>6</b>	<b>Question of robustness - Uncertainty Quantification</b>	<b>125</b>
6.1	Aim of the analysis . . . . .	125
6.2	Uncertainties and Polynomial Chaos Expansion . . . . .	126
6.2.1	Sources of uncertainty . . . . .	126
6.2.2	Quantification of uncertainty . . . . .	127
6.2.3	Polynomial Chaos Expansion . . . . .	127
6.2.4	Evaluation of Polynomial Chaos Expansion (PCE) . . . . .	129
6.2.5	Polynomial Chaos Expansion in CFD . . . . .	131
6.2.6	Workflow of non-intrusive Polynomial Chaos Expansion in CFD . . . . .	131
6.3	Validation of the implementation . . . . .	132
6.3.1	Validation of the implemented numerical integration method . . . . .	132
6.3.2	Validation of the implemented generalized PCE method . . . . .	133
6.4	Setup of CFD Model . . . . .	136
6.4.1	Selected physical models . . . . .	136
6.4.2	Selected rotor geometry . . . . .	137
6.5	Setup of Uncertainty Quantification . . . . .	137
6.5.1	Uncertain parameters . . . . .	137
6.5.2	Applied PCE expansion . . . . .	138
6.6	Results . . . . .	138
6.6.1	Performance coefficient . . . . .	138
6.6.2	Torque for the quasi-periodic revolution . . . . .	139
6.6.3	Convergence . . . . .	141
6.7	Conclusions and summary . . . . .	142

<b>7</b>	<b>Optimization of medical problems - Expert driven optimizations</b>	<b>144</b>
7.1	Aneurysms and their consequences . . . . .	145
7.2	Treatment of aneurysms . . . . .	145
7.3	Towards patient-specific treatment . . . . .	146
7.4	Expert-driven Computer Aided Stent Evaluation (ECASE) . . . . .	147
7.4.1	Reconstruction and preparation of the geometry . . . . .	148
7.4.2	Virtual stent deployment . . . . .	148
7.4.3	Mesh and CFD setup . . . . .	149
7.4.4	Post-processing . . . . .	150
7.5	Exemplary optimization . . . . .	150
7.5.1	Analyzed clinical case . . . . .	151
7.5.2	Vascular reconstruction and preprocessing . . . . .	151
7.5.3	Stent geometry . . . . .	151
7.5.4	Stent deployment . . . . .	152
7.5.5	Meshing . . . . .	152
7.5.6	Parameterization & optimization setup . . . . .	152
7.5.7	Setup of CFD Model . . . . .	152
7.5.8	Results . . . . .	154
7.6	Future improvements . . . . .	157
7.7	Second proof-of-concept . . . . .	157
7.7.1	Alternative stent deployment . . . . .	157
7.7.2	Setup of the optimization . . . . .	158
7.7.3	Results . . . . .	159
7.8	Conclusions and summary . . . . .	160
<b>8</b>	<b>HYENA - Practical engineering optimization algorithms</b>	<b>161</b>
8.1	Introduction . . . . .	161
8.1.1	Iterative methods . . . . .	162
8.2	Analysis of optimization process from engineering point of view . . . . .	163
8.2.1	Accuracy . . . . .	164
8.2.2	Resource-efficient optimization . . . . .	165
8.2.3	Adaptive Operator Selection (AOS) . . . . .	166
8.2.4	Preventing loss of information . . . . .	166
8.3	HYENA . . . . .	167
8.3.1	Step 0: Initialization . . . . .	167
8.3.2	Step 1: Generating new individuals . . . . .	167
8.3.3	Step 2: Evaluation and updating archive . . . . .	170
8.3.4	Step 3: Efficiency of the operators . . . . .	171
8.3.5	Step 4: Updating weights . . . . .	171
8.3.6	Summary . . . . .	171
8.4	Results . . . . .	172
8.4.1	SCH1 . . . . .	173
8.4.2	KUR . . . . .	174
8.4.3	ZDT4 (n=4) . . . . .	176
8.4.4	FON . . . . .	176
8.4.5	POL . . . . .	177

8.4.6	XFOIL-based optimization . . . . .	178
8.5	Conclusions and summary . . . . .	181
	<b>Summary</b>	<b>182</b>
	<b>Bibliography</b>	<b>184</b>
	<b>List of relevant publications</b>	<b>203</b>
	<b>List of all publications</b>	<b>206</b>

# Abstract

Optimization embraces everything humans do: from common everyday tasks to engineering development, optimization is present everywhere. In early history, technology was advancing very slowly, as improvements were achieved with a primitive trial-and-error approach. However, with the appearance of the different mathematical optimization techniques and with their integration into the engineering practice, improvements are achieved at an accelerating rate. Optimization is becoming more wide-spread, starting from the easier analytical cases even to computationally demanding fields.

The ultimate goal of the present thesis is to visit the different topics of Computational Fluid Dynamics-based optimization (or shortly, CFD-O), to provide an overview of the difficulties frequently encountered by engineers on different fields and to propose solutions for selected areas and issues. Thus, focus will be given to the distinctive aspects of engineering optimization. Most importantly, an optimization software is developed as part of this thesis to support the implementation of selected analyses and a new algorithm will be proposed to take into account all practical aspects encountered in the various chapters.

This will be achieved through a series of carefully chosen CFD problems (e.g., heat exchangers, medical problems, wind turbines). A successful automation and optimization process will be implemented for many problems, which were previously not – or at least not at this level of complexity – considered in the corresponding scientific literature. Among others, an efficient way for the optimization of arrangement problems will be identified, optimization of airfoil geometries for wind turbine applications based on 2D CFD computations will be considered, a 3D parameterization method for tube-like geometries (wings, winglets, channels, etc.) will be proposed. Though many engineers perceive optimization as an automation to replace humans not only in manufacturing, but in the research and development phase as well, experts cannot be eliminated from the process. An optimization process delivers a meaningful answer only to a well-formulated question. To provide an example for such an expert-driven problem, a patient-specific treatment will be considered for intracranial aneurysms. Additionally, robustness will be considered and the order needed for Uncertainty Quantification will be analyzed for a wind turbine to show the need for advanced methods, as applied in the present work.

## **Keywords**

Computational Fluid Dynamics (CFD), optimization, Evolutionary Algorithms (EA), wind energy, heat exchanger, medical optimization, mesh generation

# Kurzzusammenfassung

Optimierung umfasst alles, was Menschen tun: von alltäglichen Aufgaben bis zu industriellen Entwicklungen. Optimierung ist überall präsent. In der früheren Geschichte der Menschheit kam die Entwicklung der Technik nur sehr langsam voran, da Verbesserungen mit einer primitiven Versuch-und-Irrtum-Methode erzielt wurden. Doch mit dem Auftreten der verschiedenen mathematischen Optimierungstechniken und mit ihrer Integration in die technische Praxis werden Verbesserungen mit einer beschleunigenden Geschwindigkeit erreicht. Optimierung wird immer häufiger angewendet, von den einfachsten analytischen Fällen bis zu rechenintensiven Bereichen.

Das Ziel der vorliegenden Arbeit ist es, die verschiedenen Themen von Computational Fluid Dynamics basierter Optimierung (oder kurz, CFD-O) zu betrachten, einen Überblick von den unterschiedlichsten, bei Ingenieuren häufig getroffenen Schwierigkeiten zu geben und Lösungen für ausgewählte Bereiche und Fragen zu bieten. Der Fokus liegt auf den wichtigen Aspekten der praxisrelevanten, technischen Optimierung. Im Rahmen dieser Arbeit ist zur Unterstützung der ausgewählten Analysen eine Optimierungssoftware entwickelt worden. Außerdem wird ein neuer Algorithmus vorgeschlagen, um alle praktischen Aspekte der unterschiedlichen Kapitel zu berücksichtigen.

Dies wird durch eine Reihe von sorgfältig ausgewählten CFD-Problemen (z.B. Wärmeübertrager, medizinische Probleme, Windkraftanlagen) erreicht. Eine erfolgreiche Automatisierung und Optimierung wird für viele Probleme umgesetzt, die bisher in der einschlägigen Literatur nicht oder nur mit weniger Komplexität betrachtet wurden. Unter anderem werden ein effizienter Weg zur Optimierung eines Anordnungsproblems beschrieben, die Optimierung des Schaufelprofils für Windkraftanlagen auf Basis von 2D-CFD-Berechnungen in Betracht gezogen und ein 3D-Parametrisierungsverfahren für wellenförmige Geometrie (Flügel, Winglets, Kanäle usw.) vorgeschlagen. Obwohl viele Ingenieure Optimierung als Ersetzung der Menschen betrachten (nicht nur in der Produktion, sondern auch in der Forschungs- und Entwicklungsphase), können Experten aus dem Verfahren nicht ausgeschlossen werden. Ein Optimierungsprozess liefert eine sinnvolle Antwort nur auf eine gut formulierte Frage. Um ein Beispiel für ein solches expertengeführtes Problem zu nennen, wird eine patientenspezifische Behandlung bei einem intrakraniellen Aneurysma betrachtet. Ebenso wird am Beispiel einer Windkraftanlage Robustheit berücksichtigt, um die notwendige polynomische Ordnung einer Quantifizierung von Unsicherheiten zu bestimmen. Damit kann die Notwendigkeit der hochentwickelten Methoden präsentiert werden.

## **Schlüsselwörter**

Computational Fluid Dynamics (CFD), Optimierung, Evolutionärer Algorithmus (EA), Windenergie, Wärmeübertrager, medizinische Optimierung, Gittergenerierung

# Acknowledgement

I would like to express my special appreciation and thanks to my advisor Dr. Gábor Janiga, for his continued support and for encouraging my research. I would especially like to thank Professor Dominique Thévenin for the long discussions and his support provided for my work; and to my colleagues for providing a welcoming and motivating working environment.

I also have to thank the members of my PhD committee, Professor Dominique Thévenin, Professor Sebastian Sager, Professor Jari Hämäläinen and Dr. Gábor Janiga for investing time and providing interesting and valuable feedback.

I gratefully acknowledge the financial support by the BMBF (German Federal Ministry of Education and Research) through the AiF-ZIM project KF2473103WZ3, for finishing the validation process (Chapter 4) for wind turbines.

And at last but not least, I would like to thank my family and especially my parents, for all their love and encouragement.

# Nomenclature

## Abbreviations

ABL	Atmospheric Boundary Layer
ACGD	Approximated Continuous Generational Distance
AOA	angle-of-attack
AOS	Adaptive Operator Selection
BEM	Blade Element Momentum theory
CAD	Computer Aided Design
CFD	Computational Fluid Dynamics
CFD-O	Computational Fluid Dynamics based Optimization
CFL	Courant-Friedrichs-Lewy number
CPU	Central Processing Unit
CSV	Comma Separated Values
CT	Computer Tomography
DE	Differential Evolution
DES	Detached Eddy Simulation
DNS	Direct Numerical Simulation
DOE	Design-Of-Experiment
EA	Evolutionary Algorithm
ECASE	Expert-driven Computer Assisted Stent Evaluation
EP	Evolutionary Programming
ES	Evolution Strategy
EU27	First 27 members of European Union
FIFO	first in, first out
FSI	Fluid Structure Interaction
GA	Genetic Algorithm
GD	Generational Distance
HAWT	Horizontal Axis Wind Turbine
HYENA	Hybrid ENgineering optimization ALgorithm
LDV	Laser Doppler Velocimetry
LES	Large Eddy Simulation
LHS	Latin Hypercube Sample
LOS	Language for OPAL++ Scripting
LSQ	Least Squares method
MLSQ	Moving Least Squares method
MOGA	Multi-Objective Genetic Algorithm
MPI	Message Passing Interface

MRI	Magnetic Resonance Imaging
NIPCE	Non-Intrusive Polynomial Chaos Expansion
NSGA-II	Non-dominated Sorting Genetic Algorithm-II
NURBS	Non-uniform rational B-spline
ODL	OPAL++ Dynamic Library
OMOPSO	Optimized Multi-objective PSO
OPAL++	OPTimization Algorithm Library++
PC	Point Collocation
PCE	Polynomial Chaos Expansion
PISO	Pressure Implicit with Splitting of Operator
PIV	Particle Image Velocimetry
PSO	Particle Swarm Optimization
RBDO	Reliability Based Design Optimization
RBF	Radial Basis Function
RBO	Robust Design Optimization
RSM	Response Surface Method
SAS	Scale Adaptive Simulation
SBX	Simulated Binary cross-over
SFTP	Secure File Transfer Protocol
SIMPLE	Semi-Implicit Method for Pressure-Linked Equations
SPEA2	Strength Pareto Evolutionary Algorithm 2
SSH	Secure Shell Protocol
STL	STereoLithography
TP	Tensor Product quadrature method
TSR	Tip-Speed-Ratio
TVR	Turbulent Viscosity Ratio
UDF	User Defined Function
UQ	Uncertainty Quantification
VAWT	Vertical Axis Wind Turbine
VV	Variable Volume
WSS	Wall Shear Stress

## Greek Symbols

Symbol	Description	Units
$\alpha$	angle-of-attack	rad
$\alpha$	mode strength/empirical constant for wind power law	–
$\beta$	preset pitch angle	rad
$\gamma$	convergence metric/intermittency/step size	–
$\gamma$	winglet angle	rad
$\delta$	distance	m
$\delta_{ij}$	Kroenecker-delta	–
$\Delta$	difference/spread	–
$\Delta t$	time step	s
$\epsilon$	tolerance used for $\epsilon$ -dominance/infinitesimal number	–
$\epsilon$	turbulent dissipation rate	$\text{m}^2\text{s}^{-3}$

$\eta$	distribution index/efficiency	–
$\Theta$	rotation of camberline	rad
$\lambda$	tip speed ratio (TSR)	–
$\mu$	dynamic viscosity	kg m s <sup>-1</sup>
$\mu$	mean value	–
$\nu$	kinematic viscosity	m <sup>2</sup> s <sup>-1</sup>
$\xi$	friction factor	–
$\xi$	uncertain variable	–
$\rho$	density	kg m <sup>-3</sup>
$\sigma$	solidity/standard deviation/ $\sigma$ -algebra	–
$\phi$	azimuth angle	rad
$\phi$	radial basis function	–
$\Phi$	van der Corput sequence	–
$\psi$	non-dimensional parameter	–
$\omega$	angular velocity	rad s <sup>-1</sup>
$\omega$	specific rate of dissipation	s <sup>-1</sup>
$\Omega$	boundary/sample space	–

## General Symbols

Symbol	Description	Units
$A$	archive	–
$A$	area/projected area	m <sup>2</sup>
$AR$	Aspect Ratio	–
$c$	camberline function/calibration coefficient	–
$c$	chord	m
$c$	speed of sound	m s <sup>-1</sup>
$c_p$	specific heat capacity	J kg <sup>-1</sup> K <sup>-1</sup>
$C$	coverage of set/constraint violation	–
$C_{L/D/N/T}$	lift/drag/normal/thrust coefficient	–
$C_p$	power coefficient	–
$C_{Q/M/T}$	torque coefficient	–
$C_{x,y}$	global force coefficients	–
$d$	maximal distance between elements of the Pareto front	–
$D$	diameter	m
$E$	energy	J
$E$	expected value	–
$f$	objective function	–
$F$	force	N
$g$	equality constraint	–
$h$	inequality constraint	–
$H$	height	m
$I$	moment of inertia	kg m <sup>2</sup>
$I$	turbulent intensity	–
$k$	number of equality constraints/number of tournaments	–
$k$	thermal conductivity	W m <sup>-1</sup> K <sup>-1</sup>

$k$	turbulent kinetic energy	$\text{m}^2\text{s}^{-2}$
$k_l$	laminar kinetic energy	$\text{m}^2\text{s}^{-2}$
$l$	number of inequality constraints	—
$L$	length/integral length scale	m
$M$	Mach number	—
$MR$	maximal rank	—
$n$	number of variables	—
$N$	number of individuals/blades/revolutions	—
$\mathcal{N}$	standard normal distribution	—
$N_{crit}$	turbulence level (in XFOIL)	—
$o$	objectives	—
$\mathcal{O}$	Bachmann-Landau symbol	—
$p$	non-dimensional position along the camberline/number of computers	—
$p$	pressure	Pa
$p$	probability	—
$P$	Pareto front/probability measure	—
$P$	performance	W
$\dot{Q}$	volumetric flow rate	$\text{m}^3\text{s}^{-1}$
$\dot{Q}$	heat transfer	$\text{J s}^{-1}$
$r/R$	radius	m
$R$	growth rate	—
$R$	response	—
Re	Reynolds-number	—
$s$	first derivative	—
$S$	mesh size	m
$t$	thickness	m
$t$	thickness function	—
$t$	time	s
$T$	period time/total time	s
$T$	temperature	K
$T$	torque	N m
$u$	velocity	$\text{m s}^{-1}$
$\mathcal{U}$	uniform distribution	—
$v$	velocity	$\text{m s}^{-1}$
$V$	validity	—
$V$	volume	$\text{m}^3$
$V_a$	induction velocity	$\text{m s}^{-1}$
$w$	weight/Gaussian weights/weight function	—
$W$	local relative velocity	$\text{m s}^{-1}$
$x$	first spatial coordinate	m
$\mathbb{X}$	set of feasible designs	—
$y$	second spatial coordinate	m
$y^+$	non-dimensional wall distance	—
$z$	third spatial coordinate	m
$z_0$	roughness length	m

## Subscripts and Superscripts

<i>act</i>	actual value
<i>c</i>	camber/cross-over
<i>crowd</i>	crowding
<i>D</i>	drag
<i>eq</i>	equivalent
<i>inc</i>	incremental
<i>L</i>	lift/leading edge
<i>m</i>	mutation
<i>max</i>	maximal
<i>mean</i>	mean value
<i>mech</i>	mechanical
<i>meta</i>	metamodel
<i>min</i>	minimal
<i>N</i>	normal
<i>norm</i>	normalized
<i>obj</i>	objective
<i>opt</i>	optimal
<i>ref</i>	reference value
<i>sym</i>	symmetry
<i>T</i>	thrust
<i>T</i>	trailing edge
<i>t</i>	thickness
<i>th.</i>	theoretical
<i>tot/total</i>	total
<i>upd</i>	update
<i>var</i>	variable
<i>x, y, z</i>	Cartesian coordinate system directions

## Vectors and Matrices

Symbol	Description	Units
$\Psi$	random variable vector	—
$\mathbf{A}$	surface vector	$\text{m}^2$
$\mathbf{e}$	unit vector	—
$\mathbf{F}$	force vector	N
$\mathbf{p}$	position vector	m
$\mathbf{u}$	wind velocity vector	$\text{m s}^{-1}$
$\mathbf{v}$	velocity vector	$\text{m s}^{-1}$
$\mathbf{x}$	design variable vector	—
$\mathbf{x}$	position vector	m
$\mathbf{y}$	objective function vector	—

# Introduction

“Excellent firms don’t believe in excellence  
- only in constant improvement and  
constant change.”

---

*Tom Peters*

The complete human history is the result of our wish to strive for excellence. Using the definition of Cambridge Dictionary, optimization is “the act of making something as good as possible”. Optimization practically embraces everything we do, from everyday tasks (schedule daily task, buy groceries) to engineering processes (more, better, faster, cheaper); optimization is present everywhere.

The speed of engineering design processes are accelerating as well, mainly due to the quickly changing customer demands, competitive markets and stricter regulations. The design of products in such an environment is not possible anymore with the usual, manual trial-and-error engineering approach. Thus, systematic automated processes have to be introduced not only in the manufacturing, but in the research & development phases as well.

For industrial sectors heavily relying on Fluid Dynamics (e.g., automotive industry, energy sector, aeroplane applications, etc.), these processes depend on Computational Fluid Dynamics (or CFD), thus CFD-based optimization (or shortly CFD-O [212]) is required. However, **CFD-O** is a multidisciplinary field requiring highly trained experts excelling on all relevant fields simultaneously. More specifically, such experts must have an excellent knowledge (see Fig. 1) of

- (a) *Mathematical optimization* to be able to choose an adequate parameterization, optimization method and settings,
- (b) *Geometry and mesh generation* processes to implement it in an automated way into the process,
- (c) *Computational Fluid Dynamics* to choose the appropriate models for the simulations,

- (d) *Information technologies* to be able to implement the process in an automated way using the available hardware and software environment,
- (e) *Practical engineering aspects* in order to guide the optimization into the direction of designs and configurations of practical significance and to avoid designs which cannot be manufactured,
- (f) *Expert knowledge* required for the specific field being considered.

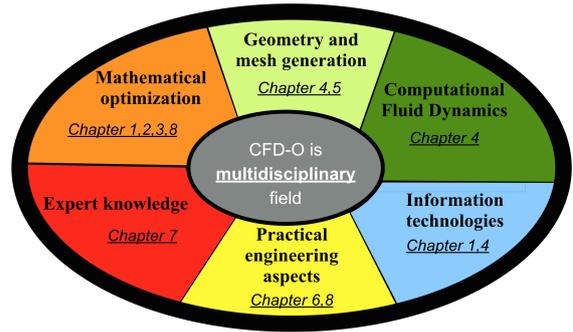


Figure 1: Multidisciplinary aspects of CFD-O

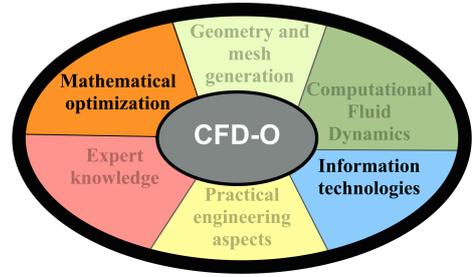
In order to present all these different aspects, selected examples will be visited in the present thesis 1) to answer the most important questions engineers face in CFD-O problems and 2) to provide a good starting point for everybody interested specifically in CFD-O. Thus, Chapter 1 will provide an introduction to *Mathematical Optimization*, followed by Chapter 2, which will emphasize that the mathematical aspects of even a relatively simple engineering problem can be challenging, by relying on the example of a two-dimensional laminar heat-exchanger. Chapter 3 will further extend the example of Chapter 2 to demonstrate the importance of parameterization methods and parameter space.

Afterwards, based on wind turbines, an example will be presented in Chapter 4 where *Computational Fluid Dynamics*, i.e., validation of the appropriate model is more demanding than the optimization itself (which is quite surprising, as the optimization took months with more than one thousand of processor cores). Additionally, this chapter will also use a very complex *Geometry and mesh generation* method, which will be extended to 3D in Chapter 5.

Afterwards, Chapter 6 will present a very important *Practical engineering aspect*, i.e., robustness, which can only be achieved with efficient Uncertainty Quantification (UQ) methods. For this, UQ will be presented for the wind turbine already used in Chapter 4.

Chapter 7 will present a special field of optimization based on a patient-specific medical treatment process, where *Expert knowledge* is of key importance, i.e., it has to be included not only in the planning, but in the actual optimization process as well. Finally, based on all the experiences gained in the present thesis, a new optimization algorithm is proposed in Chapter 8, to accommodate the most important *Practical engineering aspects* into a *Mathematical optimization* algorithm.

Of course, various aspects of *Information technologies* will be also discussed in the different chapters. Most of all, Chapter 1 will focus on the developed optimization software for the present thesis and Chapter 4 will present a very demanding automation and optimization process. Using these selected examples based on original research and publications, all relevant aspects of CFD-O will be presented.



# Chapter 1

## Introduction to optimization

“Good, better, best: never let them rest until the good is better and the better is best.”

---

*St. Jerome*

*The present chapter will especially focus on questions related to the theory of Mathematical Optimization, and actually putting the theory into practice by using Information Technology.*

### 1.1 Mathematical optimization

Let us assume a problem with  $n$  independent variables:

$$\mathbf{x} = (x_1, x_2, \dots, x_n)^T, \quad \mathbf{x} \in \mathbb{X}, \quad (1.1)$$

where  $\mathbf{x}$  is the decision variable vector (or design variable) and  $\mathbb{X}$  the feasible domain. The feasible domain is defined by  $k$  equality constraints,  $l$  inequality constraints and the bounds for each variable:

$$\mathbf{x} \in \mathbb{X} \iff \begin{cases} g_i(\mathbf{x}) = 0 & (i = 1 \dots k) \\ h_j(\mathbf{x}) \leq 0 & (j = 1 \dots l) \\ x_{k,min} \leq x_k \leq x_{k,max} & (k = 1 \dots n) \end{cases} \quad (1.2)$$

Besides the variables  $m$  objective functions are defined:

$$\mathbf{y}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))^T. \quad (1.3)$$

The  $y_i(\mathbf{x})$  function may be defined explicitly (analytical problems) or implicitly (numerical problems). Without any loss of generality the optimization can be defined as

$$\textcircled{O} : \begin{cases} \mathbf{y}(\mathbf{x}) \xrightarrow{\mathbf{x}} \min \\ \text{so that } \mathbf{x} \in \mathbb{X} \end{cases} . \quad (1.4)$$

If maximization is required, the function simply has to be transformed by  $f_i^*(\mathbf{x}) = -f_i(\mathbf{x})$ .

In the followings, the terms ‘design variable’, ‘decision variable vector’ and ‘configuration’ will be used as synonyms.

### 1.1.1 Optima

The ultimate goal of any mathematical optimization process is to identify the optimum (or optima). Without loss of generality, the different types of optima will be presented for  $m = 1$ . The design variable  $\mathbf{x}_0$  is a global optimum (see Fig. 1.1), if and only if:

$$f(\mathbf{x}_0) \leq f(\mathbf{x}) \quad \forall \mathbf{x} \in \mathbb{X} \quad (1.5)$$

i.e., it is at least as good, as any other design variable. It is not trivial to find this design variable without having an extensive knowledge of the objective function (for implicit problems, without computing the objective function for each design variable, one cannot ensure that no better configuration exists). When using local search methods (e.g., many gradient-based methods, see later), the optimization method might get stuck into a local optimum (see Fig. 1.2), without finding the real global optimum:

$$f(\mathbf{x}_0) \leq f(\mathbf{x}_0 + \epsilon), \quad \forall \epsilon : \mathbf{x} + \epsilon \in \mathbb{X}, \quad |\epsilon| \ll 1 \quad (1.6)$$

As a result, global optimization methods are recommended to avoid this problem. These methods are however time-consuming.

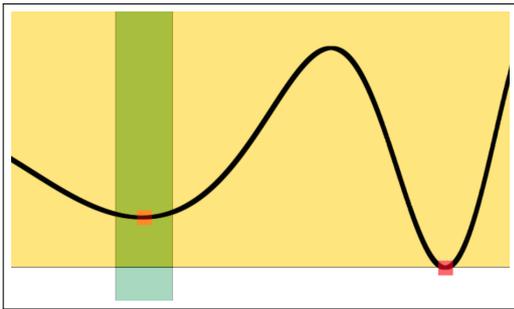


Figure 1.1: Example of local (left) and global optimum (right)

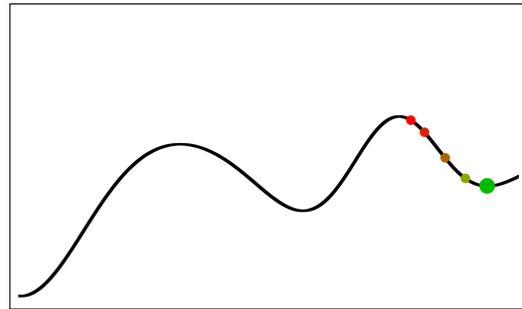


Figure 1.2: Example of gradient based method getting stuck in local optimum (moving from red to green point, based on the gradient)

### 1.1.2 Classification of optimization problems

Optimization tasks may be divided into different groups, based on the definition. Table 1.1 summarizes the most important groups of optimization problems. As for most numerical simulation based problems it cannot be decided, if the problem is convex, convexity will not be discussed here.

#### 1.1.2.1 Number of objectives

**Single-objective** problems are the easiest problems, with only one objective function defined ( $m = 1$ ).

Table 1.1: Classification of optimization problems

$k = l = 0$	unconstrained
$k \neq 0 \cap l \neq 0$	constrained
$m = 1$	single-objective
$m \geq 2$	multi-objective
$\mathbf{y}(\mathbf{X}_1^{\text{opt}}) = \mathbf{y}(\mathbf{X}_2^{\text{opt}}) \implies \mathbf{X}_1^{\text{opt}} = \mathbf{X}_2^{\text{opt}}$	uni-modal
$\mathbf{y}(\mathbf{X}_1^{\text{opt}}) = \mathbf{y}(\mathbf{X}_2^{\text{opt}}) \not\implies \mathbf{X}_1^{\text{opt}} = \mathbf{X}_2^{\text{opt}}$	multi-modal

**Multi-objective** problems are defined with more than one objective function ( $m > 1$ ). These problems are actually very common, as most objectives in real life are concurring as well (e.g., minimize the price and simultaneously maximize the quality of a product). As a result, deciding which design variable is better, may be difficult. When comparing two design variables, one may use the Pareto dominance, as defined by:

$$\mathbf{x}_1 \prec \mathbf{x}_2 \iff \begin{cases} \forall i \in [1\dots m] : f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2) \\ \exists j \in [1\dots m] : f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2) \end{cases} \quad (1.7)$$

i.e.,  $\mathbf{x}_1$  design variable dominates  $\mathbf{x}_2$  design variable if and only if  $\mathbf{y}(\mathbf{x}_1)$  is not worse in any objective value as  $\mathbf{y}(\mathbf{x}_2)$  and is better at least in one objective value. During a multi-objective optimization process our goal is now to determine the true Pareto front (i.e., the set of non-dominated designs) with the best possible resolution. The Pareto-front is defined as:

$$P(\mathbb{X}) := \{\mathbf{x} \in \mathbb{X} \mid \nexists \hat{\mathbf{x}} \in \mathbb{X} : \hat{\mathbf{x}} \prec \mathbf{x}\}. \quad (1.8)$$

This set contains all designs, which are not dominated by any member of the feasible domain. The exact set is usually impossible to reach (except for some simple analytical cases). Thus, it will be approximated by an iterative algorithm. Multi-objective problems may be transformed into single-objective problems by using different scalarization techniques. The most well-known method is the simple linear weight function [65]:

$$f = \sum_{i=1}^m w_i f_i \quad . \quad (1.9)$$

Through the transformation, only a single point of the Pareto front can be retained. Hence, to apply such a simplified, fixed-weight method, a priori information is needed for the *decision making* process. “Without such knowledge or due to faulty assumptions, a lumped analysis can rapidly lead to suboptimal results” [56].

**Many-objective** problems have unfortunately no exact definition, as there is no agreement for the definition yet. Some define them with 3, 4 or more objectives ( $m > 3$  or  $m > 4$ ) [141]. Mathematically, there is no difference between multi- and many-objective problems, but very different algorithms are required to solve such problems, as due to the *curse of dimensionality*, the size of the Pareto-front increases exponentially with  $m$ .

### 1.1.2.2 Modality

For single-modal problems only one optimum exists, while for multimodal problems more optima exist. Thus, the same objective values can correspond to different design variable vectors, and a bijective correspondence between  $\mathbf{x}$  and  $\mathbf{y}$  does not exist.

Such multimodal solutions have large engineering significance, as one of those design variables will be usually superior for practical purposes due to a priori unknown constraints (for instance, it might be easier or cheaper to manufacture). Mathematically formulated:

$$\mathbf{y}(\mathbf{X}_1^{\text{opt}}) = \mathbf{y}(\mathbf{X}_2^{\text{opt}}) \not\Rightarrow \mathbf{X}_1^{\text{opt}} = \mathbf{X}_2^{\text{opt}}. \quad (1.10)$$

Depending on the goal of the optimization study (find only one better solution *or* find all optima), different methods are required. Usually, finding a single optimum for multimodal problems might be easier than for single-modal (if there are more optima, the probability of finding one may be higher), getting all solutions of multimodal problems is much harder (without knowing the number of optima, it may be difficult to determine, when to stop the optimization algorithm).

### 1.1.3 Classification of optimization methods

The methods proposed for solving the problems are as different, as the methods themselves. Methods can be classified according to many different aspects; two possible classifications will be presented below.

#### 1.1.3.1 Black box versus non-black box methods

Optimization algorithms may take advantage of knowledge of the problem directly, or handle the evaluation of the objective function(s) as a black-box (see Fig. 1.3). Black box methods have absolutely no knowledge of the problem itself. Although this seems to be a disadvantage, this is not necessarily true. In industrial practice, many commercial software are applied, where access to the source code is not possible due to license restrictions.

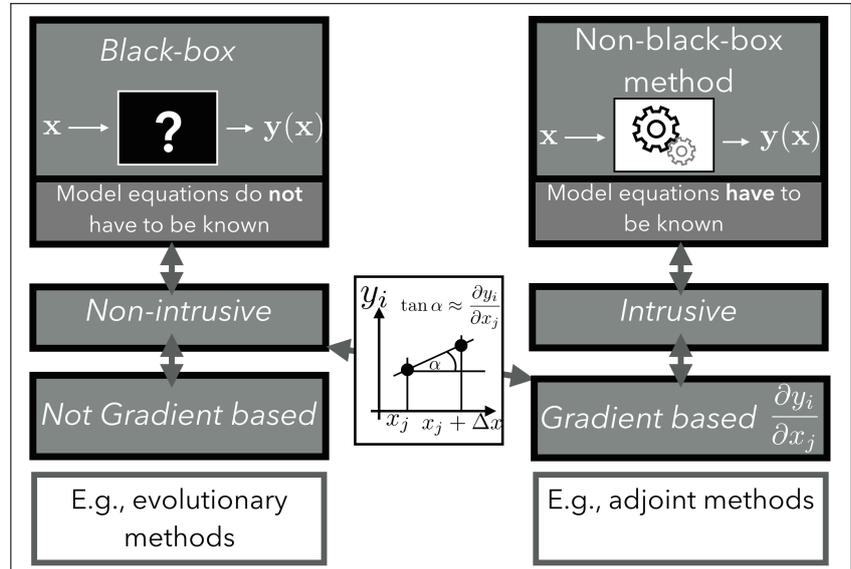


Figure 1.3: Classification of optimization methods

In such cases, only black box methods can be used, i.e., black box methods can be applied to a broader range of problems and are more flexible. A disadvantage of black-box

methods is their significantly higher computational cost.

In contrast, one can also take advantage of knowing, how the objective functions look like. In CFD, such methods are called adjoint methods, where the flow solver has to be modified (thus, the optimization algorithm has to get access to the problem in an intrusive manner). This increases the implementation time significantly, but provides the derivatives ( $\frac{\partial y_i}{\partial x_j}$ ) of the objectives with regard to the variables [175]. Based on the derivatives, one can simply advance using a hillclimbing method into the direction of the optimum. Although being fast, these methods can get trapped into a local minimum (i.e., they are local optimization methods, not global optimization methods).

### 1.1.3.2 Gradient-based and gradient-free methods

As presented earlier, with non-black box methods, gradients can be computed directly. If gradients are known, the optimization becomes a trivial task. If  $\gamma_n$  designates a (small) step size, one may advance into the direction of a local optimum by simply moving from a reference design variable  $\mathbf{x}_0$  step-by-step. In case of single-objective optimization, e.g.,:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \gamma_n \nabla f(\mathbf{x}_n) . \quad (1.11)$$

Gradient-based methods are not necessarily adjoint methods, as gradients may be reconstructed using  $n$  additional function evaluations for an  $n$  dimensional case. As a result, gradient-based black-box methods suffer from the “curse of dimensionality” and become unaffordable very rapidly. Gradient-based methods are local search methods, as they rely only on local information.

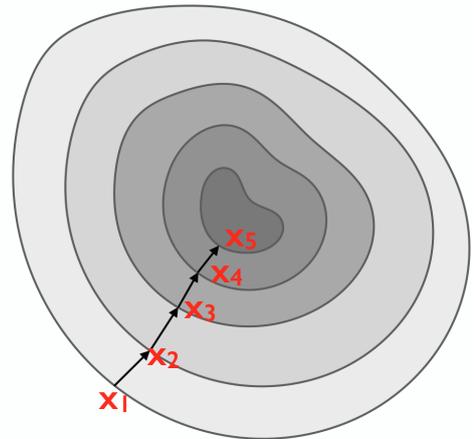


Figure 1.4: Gradient-based optimization

Gradient-free methods handle the function evaluations as black-box. These are iterative methods using different heuristics or mimicking different natural phenomena (e.g., swarm-based optimization, evolution, simulated annealing, etc.). The most popular methods, the Evolutionary Algorithms are usually population-based (especially suited for multi-objective or multi-modal problems), although some exceptions exist with a single candidate solution at each time (e.g., hill climbing) [197]. To speed up convergence, one may implement hybrid methods as well, combining local and global search [105].

### 1.1.3.3 Other classifications

As every year several dozens of new methods are published [93], algorithms could be classified based on many different criteria (e.g., single/multi-modal; local/global optimization; single/multi-objective; etc.), filling many books on this subject. This is out of the scope of the present thesis.

### 1.1.3.4 Iterative optimization methods

In the following, only black-box, gradient-free methods will be considered due to the commercial software used in the present thesis. These methods are usually iterative in nature. Consider a modified form of an iterative search procedure as defined by Laumanns et al. [136]:

```

0: optimize( $\mathbf{y}(\mathbf{x})$ ):
1:  $c := 0$ 
2:  $A^{(0)} := \emptyset$ 
3: while  $terminate(A^{(c)}, i) = false$  do
4:    $c := c + 1$ 
5:    $F^{(c)} := generate(A^{(c-1)})$ 
6:   compute  $\mathbf{y}(\mathbf{x}) \quad \forall \mathbf{x} \in F^{(c)}$ 
7:    $A^{(c)} := update(A^{(c-1)}, F^{(c)})$ 
8: end while
9: return  $A^{(c)}$ 

```

where  $c$  is the iteration cycle counter,  $A^{(c)}$  is the archive in step  $c$  and  $F^{(c)}$  is the list of new individuals generated by the different operators. Compared to [136] it is emphasized that in Step 5 the new design variables are generated based on the old archive and Step 6 is added to emphasize that generating the new design variables and evaluating them is logically different (in CFD, this is especially true). Finally, the Pareto front is approximated with

$$P(\mathbb{X}) \approx P(A^{(C)}) = \{\mathbf{x} \in A^{(C)} \mid \nexists \hat{\mathbf{x}} \in A^{(C)} : \hat{\mathbf{x}} \prec \mathbf{x}\}, \quad (1.12)$$

if the total number of generations is  $C = \max(c)$ . The quality of this approximation can be evaluated using different metrics, e.g., set coverage ratio, spacing, generational distance, inverted generational distance, generalized spread [65], hyper volume indicator [234], etc. In single-objective cases, the Pareto front degenerates into a single point.

For Step 5 different operators can be applied. If a genetic operator is used, the approach is a genetic optimization method; if vector operations are used, it is Differential Evolution (DE); if the individuals are based on an interpolation method, it is a metamodel-based optimization technique.

### 1.1.3.5 Which method to use?

“An inherent difficulty of optimization is that one has to choose for each problem the appropriate optimization method. An algorithm, which is efficient for an unimodal problem with a single optimum might fail for noisy or multimodal problems. There is no algorithm that is efficient for all problems” [58]. This is the result of the so called “*no free lunch theory*”, which states, that “...all algorithms for optimization will give the same average performance when averaged over all possible functions, which means that the universally best method does not exist for all optimization problems” [228, 230].

Thus, it is usually very difficult to make a decision a priori, for which reason optimization software still heavily rely on expert knowledge.

## 1.2 Optimization in engineering practice

The following section is based on:

DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Workshop on Turbomachine Optimization based on Computational Fluid Dynamics. In *Conference on Modelling Fluid Flow (the 16th International Conference on Fluid Flow Technologies)* (Budapest, Hungary, 2015), J. Vad, Ed [58].

Optimization in the engineering practice is somewhat different, compared to the mathematical optimization. E.g., while in mathematical optimization one is interested in approximating the exact solution with as much accuracy as possible, this is completely irrelevant for practical purposes: one has to take into account manufacturing precision (i.e., it makes no sense to optimize the dimensions of a car with nanometer precision).

Furthermore, except the most unique cases, one is not interested to ensure that the real optimum is found. Instead, one has to find a configuration providing *an improvement of economical interest*. I.e., if one can build a new product, which is better than the old one and it is economical to enter the market with it, the task is completed.

The large differences between the theoretical and practical approach lead often to misunderstandings between the two communities. Engineers are not interested in optimization methods, which require “only” 25 000 function evaluations, while mathematicians may find methods based on heuristic approach irrelevant, if proof of convergence cannot be provided. From the mathematical point of view, the convergence of each Genetic Algorithm (GA) to the optima is not guaranteed (however, for some algorithms, there is a guarantee [136]). Nevertheless, they proved to be very efficient for practical applications. One should not forget, that many other engineering methods are also based on heuristics.

### 1.2.1 Optimization of CFD-based problems

When using CFD-based optimization (or CFD-O [212]), additional difficulties are faced during the optimization process, summarized as follows:

- Objective function values are not explicitly known and have to be computed based on numerical simulations. Thus, gradients are usually not available.
- Function evaluations are very costly, requiring from a couple of minutes of computational time (see Chapter 2) up to several days (see Chapter 4 and 5).
- Due to numerical noise and model uncertainties, the objective functions are usually noisy.
- During the optimization, differences of the variables below manufacturing tolerance are irrelevant (e.g., optimization of a car geometry with nm precision).
- The geometry and mesh have to be created/morphed for each design variable in an automated and robust way.
- Different software (including proprietary commercial software) have to be coupled to cooperate for the optimization.

As a result, speed and efficiency is of key importance in CFD-O [58].

## 1.2.2 Governing equation in CFD

In the present thesis, flow simulations will be restricted to the viscous flow of Newtonian, incompressible fluids. Simulations in Chapter 2 and 7 will be in the laminar flow regime; all other Chapters will include only turbulent simulations, using different turbulence models. Additionally, Chapters 2, 3 and 7 will rely on steady-state simulations; the other Chapters will present transient simulations. In a Cartesian coordinate system the flow of such fluids can be described with the mass conservation (i.e., continuity):

$$\nabla \cdot \mathbf{u} = 0 \quad (1.13)$$

and the Navier-Stokes (or momentum conservation) equation:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = \mathbf{g} - \frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} \quad (1.14)$$

where  $\rho$  is the density,  $\mathbf{u}$  velocity vector,  $p$  is the pressure,  $\mathbf{g}$  is the gravitational acceleration vector and  $\nu$  is the kinematic viscosity.

Unfortunately, the Navier-Stokes equation does not have an analytic solution (except in the simplest cases). Thus, different numerical techniques have to be applied to approximate the solution. For this, one has to generate a grid and transform the partial differential equation using Finite Difference, Finite Element or Finite Volume method into a system of algebraic equations. In the present study, only Finite Volume discretizations are considered, using the commercial software packages ANSYS Fluent [6] or CD-Adapco StarCCM+ [37].

## 1.2.3 Chosen optimization algorithms for the present study

For CFD-O problems, probably the most popular methods are the different Evolutionary Algorithms (EA). Especially interesting are Genetic Algorithms (GAs), although there are other methods as well, e.g., Evolution Strategy (ES) [184,194] and Evolutionary Programming (EP). Unfortunately, Evolutionary Algorithms do not have a clear definition. According to Dan Simon “Terminology is imprecise and context-dependent ... [EA is] an algorithm that evolves a problem solution over many iterations” [197]. There is still a serious debate, if swarm optimization methods belong to EAs.

Most of these methods are more resistant to numerical noise than gradient-based methods; they are robust and flexible. Thus, this semi-stochastic, soft-computing [197] approach is especially well-suited for engineering and CFD-based optimizations, as presented in many studies [117].

Genetic Algorithms (GA) are based on an analogy with natural selection. Just like in nature, characteristics of each individual (configuration or design) is described with a collection of genes. One may use many different coding schemes for the genes, based on binary or floating-point representation. In each generation, the objective function of the single individuals is evaluated. Based on the fitness of the different individuals a new set of designs (offsprings) is generated. Just like in evolution, only the best individuals can reproduce (*selection*). These parents create offsprings using a set of genetic operators (cross-over and mutation). Only the best individuals can survive (survival of the fittest), i.e., inferior individuals will be eliminated during the process.

As the best individuals will be favored, the population will drift toward better solutions, converging in each generation closer to the true Pareto front (or single optimum). Throughout the iterations, an appropriate balance must be kept to ensure simultaneously diversity of the solutions and convergence toward the Pareto front; i.e., an appropriate emphasis must be put onto the choice of the parameters of the GA in order to obtain a balance between exploration and exploitation. In the present thesis, only GA methods will be used.

## 1.3 Optimization Algorithm Library++ (OPAL++)

In order to analyze the most important aspects of CFD-O, a software (OPTimization Algorithm Library++ or simply OPAL++) was developed to serve as a framework for the optimization. Although OPAL++ builds on top of our considerable experience with OPAL (see for instance [110]), it is based on a completely new structure. OPAL++ is an object-oriented multi-objective optimization and parameterization framework developed in C++. OPAL++ is highly portable, supporting Microsoft Windows, most common Linux distributions (e.g., CentOS, Fedora, Scientific Linux, Redhat, etc.) and Mac OS operating system as well [52]. The complete source code of OPAL++ is around 92 000 lines.

The software has already been successfully applied to many different problems [54, 55, 56, 57, 60, 62, 119, 153, 154] and is focused especially on CFD-related problems.

### 1.3.1 Feasibility and dominance in OPAL++

As many different Evolutionary Algorithms (EA) are available in OPAL++, a unified approach had to be chosen for dominance (which design variable is better) and constraint handling. Only such methods were chosen for implementation, where instead of using a cost or fitness function, comparison of individuals is sufficient (i.e., no roulette wheel selection can be applied).

In OPAL++, all individuals store the values of the design variables

$$\mathbf{x} = (x_1, x_2, \dots, x_n)^T. \quad (1.15)$$

OPAL++ supports only inequality constraints, however, this is not really a restriction, as equality constraints can be reformulated into inequality constraints:

$$h_j(\mathbf{x}) = 0 \iff |h_j(\mathbf{x})| \leq 0. \quad (1.16)$$

For constraint handling, the constraint violation approach as implemented in NSGA-II was chosen [68], although many different methods (e.g., penalty function) could be applied [44]. To additionally improve the efficiency, constraints are classified into two groups, i.e., design constraints (only depending on design variables) and functional constraints (depending on objective function values as well):

$$h_j(\mathbf{x}) \leq 0 \ (j = 1 \dots l) \rightarrow \begin{cases} h_j^d(\mathbf{x}) \leq 0 \ (j = 1 \dots l_d) \\ h_j^f(\mathbf{x}, \mathbf{y}) \leq 0 \ (j = 1 \dots l_f) \end{cases} \quad (1.17)$$

For analytic problems this approach may seem redundant, as one may transform functional constraints into design constraints by:

$$h_j^f(\mathbf{x}, \mathbf{y}) \equiv h_j^f(\mathbf{x}, \mathbf{y}(\mathbf{x})). \quad (1.18)$$

However, in CFD, this is not as simple. Based upon this definition, the evaluation process can already be given:

- (1) First, design constraints are evaluated. Constraint violation is defined as:

$$C = \left( \sum_{j=1}^{l_d} \max \left( \frac{h_j^d(\mathbf{x})}{w_i^d}, 0 \right) \right) \cdot [1 + C_{diff} \cdot \text{sgn}(l_f)], \quad (1.19)$$

where  $C_{diff}$  is an appropriately large constant and  $w_i^d$  are normalization factors. This equation ensures, that  $C_{diff}$  is only added to the constraint violation, if there are functional constraints and a constraint violation is present. For  $C \neq 0$ , no additional steps are performed.

- (2) If  $C = 0$  holds, objective function(s) will be evaluated:

$$\mathbf{y}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))^T. \quad (1.20)$$

- (3) Functional constraints are evaluated:

$$C = \sum_{j=1}^{l_f} \max \left( \frac{h_j^f(\mathbf{x}, \mathbf{y})}{w_i^f}, 0 \right) \quad (1.21)$$

As one can see, due to the presence of  $C_{diff}$ , design variables, which violate only functional constraints (i.e., a successful function evaluation can be performed) are superior compared to design variables, which violate design constraints.

Additionally, a validity bit is defined ( $V$ ) to provide support for realistic applications. During CFD-O, many unspecified problems can be encountered, where defining a constraint or constraint violation in an explicit form is impossible. Causes for such situations are for instance, failed mesh operations, software crash, divergence, etc. Configurations, where the flow solver experiences problems should still be avoided. If such an error is encountered, the individual can be marked as **invalid** ( $V = 0$ ).

When comparing two ( $A$  and  $B$ ) individuals (note, that as previously explained, OPAL++ is based only on methods, for which comparison of the individuals is sufficient), the following approach is adapted:

#### I. Comparison based on validity

- (a) If  $V_A = V_B = 0$ , both individuals are equivalent.
- (b) If  $V_A = 1, V_B = 0$ ,  $A$  dominates  $B$ .
- (c) If  $V_A = V_B = 1$ , proceed to comparison based on constraint violation.

#### II. Comparison based on constraint violation

- (a) If  $C_A = C_B > 0$ , both individuals are equivalent.
- (b) If  $C_A < C_B$ ,  $A$  dominates  $B$ .
- (c) If  $C_A = C_B = 0$ , proceed to comparison based on objective functions.

### III. Comparison based on objective functions

- (a) Comparison is based on dominance used by the algorithm, i.e., usually based on Pareto-dominance:

$$\mathbf{x}_1 \prec \mathbf{x}_2 \iff \begin{cases} \forall i \in [1\dots m] : f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2) \\ \exists j \in [1\dots m] : f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2) \end{cases} \quad (1.22)$$

However, in some cases, e.g., by OMOPSO,  $\epsilon$ -dominance might be applied.

The above presented method of OPAL++ provides a very robust approach and was successfully applied in many studies, as will be presented in the followings.

### 1.3.2 Automatization of workflow

In order to create an optimization setup, OPAL++ requires two script files, the master and simulation script. The master script contains the definition of the optimization, the list of necessary input and output files and the settings for Evolutionary Algorithms.

The second file, the simulation script contains the workflow for evaluating a single design variable. OPAL++ provides many different commands to ease the coupling of different software. Both script files rely on “Language for OPAL++ Scripting” (LOS), a specific script language developed for the present application. Thanks to its very simple syntax, LOS has a very steep learning curve and eliminates the need to setup many different scripts in different languages.

```
//...
//Substitute the values of variables in a file
OPAL2S.TOOLS.SUBSTITUTE(STR/*example.txt*/);
//Wait 10 seconds
OPAL2S.TOOLS.WAIT(10.0);
//Check, a specific condition
OPAL2S.CONTROL.IF(STR/*variable1>0*/);
    //Do something
OPAL2S.CONTROL.END;
//...
```

Before an optimization can be started, OPAL++ will check the syntax for errors, eliminating common mistakes and providing a robust execution.

When an optimization is started, OPAL++ will evaluate the different individuals in parallel, relying on Message Passing Interface (MPI). A single node will serve as master (controller) node, responsible for performing all operations of the EAs, administrative, disk operations and for managing the workload of the different nodes (see Fig. 1.5). OPAL++ uses an internal job management system; the different configurations are evaluated in a FIFO (first in, first out) manner. The other nodes will serve as workers, performing the same steps defined in the simulation script again and again, for different design variables.

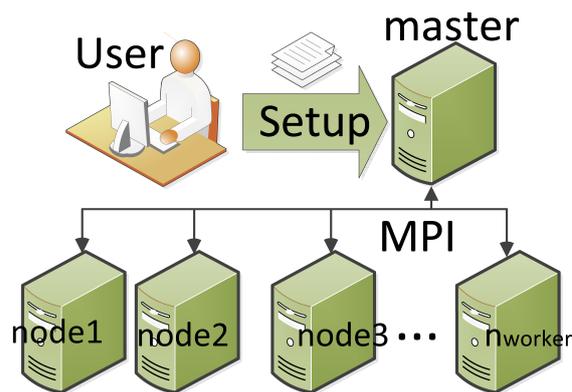


Figure 1.5: Parallel execution in OPAL++

As the CFD solver can be started using many CPU cores (or even computers), this enables a two-level parallelization with drastic speed-ups. 100% parallelization efficiency cannot be achieved, as discussed later in Chapter 2 and 4.

The optimization process can be stopped at any time and OPAL++ will be able to continue from the last completely finished generation (it saves the whole memory image after each generation).

### 1.3.2.1 Support for heterogeneous systems

A common problem in practical optimization setups is that some software components may support only a single operating system (e.g., PTC Creo supports only Microsoft Windows). This breaks the parallelization scheme and results in significant problems. In order to by-pass this issue, OPAL++ provides support directly for SSH and SFTP operations (though in Linux one could simply use the available *sftp* and *ssh* commands, Windows provides no alternative), based on the LibSSH2 library. The user can access all SFTP and SSH related commands in the simulation script file using LOS syntax.

Furthermore, OPAL++ is able to maintain an SSH channel, i.e., not only single commands can be executed but a connection can be kept open.

### 1.3.2.2 Mutex for synchronization

Although the largest speed-up can be achieved, if the same number of nodes is used in parallel as the number of individuals in a single generation, this is not always possible. Not only the available computational resources may be limited, but if commercial applications are applied in the workflow, the number of licenses might be restricted as well. To eliminate this problem, in OPAL++ so-called mutexes can be defined. If a specific section of the automated workflow is protected by a mutex, only a defined number of nodes can enter this section simultaneously. This ensures that no evaluation of an individual can fail due to exhausted licenses.

### 1.3.2.3 Other features

Besides the features listed above, OPAL++ provides support for many other features (e.g., user-defined equations, dynamic C++ libraries loaded directly into OPAL++, etc.), which will not be detailed here for the sake of length. For a full list of options, please refer to the OPAL++ manual [52].

## 1.3.3 Initialization of the first generation

For each population-based optimization algorithm the initialization of the first generation is a very important question. Although the most easiest and widely used method is a simple random population, it is not the most efficient method. Random populations increase the effect of genetic drift and do not guarantee, that the whole design space is equally covered, resulting in some cases in lack of information for several domains.

To address this issue, instead of using a completely random initialization, one may use different Design-Of-Experiment methods to cover the parameter space. Especially for high-dimensional problems, where the curse-of-dimensionality applies, using an appropriate initialization can enhance convergence speed significantly. There are hundreds of different methods, just to mention a few, pseudorandom or quasirandom series, Latin Hypercube Sampling, full or fractional factorial design, Plackett-Burman design, Box-Behnken design, D-optimality, V-optimality, I-optimality [162], etc.

The detailed explanation of all these methods is out of the scope of the present thesis, as one can fill several books with it [162]. A couple of methods were chosen for OPAL++; in the followings a short description of the implemented methods is given.

### 1.3.3.1 Pseudorandom initialization

Although C++ provides a random generator (*int rand(void)*), it is not well suited for the present case, as it does not provide uniformly distributed random numbers, but lower numbers are favored. More advanced methods, e.g., the Mersenne-Twister random generator is supported since C++11, but for compatibility reasons (to support older systems as well), OPAL++ was developed in C++98.

To address this issue, a 623-dimensional Mersenne-Twister pseudorandom generator was implemented in OPAL++ [150], which has an enormous  $2^{19937} - 1$  cycle. OPAL++ uses a single global pseudorandom generator. If necessary, one can reconstruct the optimization methods by forcing OPAL++ to use the same seed as in a previous optimization: thus, exactly the same operations will be performed.

### 1.3.3.2 Quasirandom initialization

Although pseudorandom generators follow the rule of big numbers, i.e., they guarantee a uniform distribution for thousands and millions of points, this is not the case for a couple of points ( $< 200$ ), as the probability of an equally distributed series is small, resulting in large “gaps” in the parameter space. Instead, one may use so-called low-discrepancy sequences. These sequences mimic the properties of random numbers (i.e., uniform distribution for large number of points), but additionally avoid clustering and

gaps in order to provide uniform coverage of the domain. In reality, these sequences are totally deterministic.

**Halton sequence** Let us take an arbitrary number  $n$  and prime number  $b$ . In this case,  $n$  can be represented in the  $b$  number system as:

$$n = \sum_{j=0}^m d_j(n)b^j, \quad (1.23)$$

with  $d_j(n)$  the digits of the number. In this case, the  $n$ -th element of the van der Corput sequence is defined as:

$$\Phi_b(n) = \sum_{j=0}^m d_j(n)b^{-j-1}. \quad (1.24)$$

Halton sequence is a simple  $k$ -dimensional low-discrepancy sequence, which uses for each dimension a van der Corput sequence of different prime number base. Let us denote in what follows the prime numbers as  $P(i)$ , with  $P(1) = 2$ ,  $P(2) = 3$ ,  $P(3) = 5$ , ... Then, the  $n$ -th point of the  $k$ -dimensional Halton sequence can be computed as

$$\mathbf{Halton}(n) = (\Phi_{P(1)}(n), \Phi_{P(2)}(n), \dots, \Phi_{P(k)}(n)) \quad (1.25)$$

Due to the correlation between the dimensions, it is not recommended to use a Halton-sequence over a dimension of 7 [70].

**Hammersley sequence** Hammersley sequence is based on a simple modification of the Halton-sequence, i.e., the last dimension is replaced with  $n/N$ , where  $N$  is the number of points.

$$\mathbf{Hammersley}(n) = (\Phi_{P(1)}(n), \Phi_{P(2)}(n), \dots, \Phi_{P(k-1)}(n), n/N) \quad (1.26)$$

As a result, Hammersley sequence still suffers from degradation in multi-dimensional cases, but provides a better coverage for 2...5 dimensions. One disadvantage of this method is that one has to know in advance the number of points  $N$ .

**Sobol sequence** Sobol' sequence is one the most well-known and well-spread method, based on primitive polynomials [201]. The biggest advantage of the method is that it does not suffer from degradation, even for 260 dimensions [91]. For practical applications, the first point  $(0, 0, \dots, 0)$  has to be discarded [122].

### 1.3.3.3 Latin Hypercube Sample (LHS)

Let us create a Latin Hypercube Sample (LHS) for a  $D$ -dimensional problem and  $N$  points (i.e., samples). In this case, for each dimension  $d$   $N$  different levels have to be defined  $(x_d(i), i = 1 \dots N)$ . An LHS contains  $N$  design variables

$$\mathbf{LHS}(j) = (x_1(i_{1,j}), x_2(i_{2,j}), \dots, x_D(i_{D,j}),) \quad j = 1 \dots N, \quad (1.27)$$

so, that each level for each dimension can appear only once. I.e., for the indices it holds, that

$$i_{d,j_1} = i_{d,j_2} \iff j_1 = j_2. \quad (1.28)$$

The levels can be defined in different ways. The easiest is to use equally distributed points in the parameter space. If the parameter space is defined as  $[x_{min,d}; x_{max,d}]$  (called marginal LHS design in OPAL++), the points will be computed as:

$$x_d(i) = x_{min,d} + (x_{max,d} - x_{min,d}) \cdot \frac{i - 1}{N - 1}. \quad (1.29)$$

If one does not want to include the boundary of the domain, alternatively,  $(x_{min,d}; x_{max,d})$  (called centralized LHS in OPAL++) could be used as well. Additionally, instead of using deterministic levels, one could generate the levels based on a uniform random distribution, e.g., for the marginal random LHS (as called in OPAL++):

$$x_d(i) = x_{min,d} + \frac{x_{max,d} - x_{min,d}}{N} \cdot [i - 1 + \mathcal{U}(0, 1)] \quad (1.30)$$

Generating an LHS is not difficult, one has to simply use random permutations of the  $1, 2, \dots, N$  series for the indices of the different dimensions. This also means, that there are a vast number of different LHS designs for each  $D, N$  combinations.

OPAL++ supports additionally a so-called near-orthogonal LHS approach. This means, that for the two-dimensional case 200 000 random LHS samples are created and OPAL++ chooses the one, where the correlation between the first and second dimension is the smallest. Afterwards, for creating a near-orthogonal LHS with  $d$  dimensions ( $d > 2$ ), OPAL++ takes the near-orthogonal LHS of  $d - 1$  dimensions, adds 200 000 different permutations for the last dimension, and chooses the design variable which minimizes the maximal element of the correlation matrix. This is continued, until  $D$  is reached. Although this method does not guarantee finding an orthogonal LHS (orthogonal design does not exist for each  $N, D$  design variable [142]), it provides very good results while staying practical.

#### 1.3.3.4 Constrained initialization

When performing constrained optimization, large areas of the parameter space maybe infeasible. As a result, the first generation may contain only a couple of feasible individuals, resulting in inferior convergence speed and coverage. To eliminate this error, OPAL++ supports a so-called constrained initialization. If the first generation has to contain  $N$  individuals, OPAL++ searches for the smallest  $N_{mod}$  number, which satisfies, that out of the  $N_{mod}$  design variables  $N$  are feasible (with regard to the design constraints). Afterwards, the optimization will be started using only the  $N$  feasible design variables. Constrained initialization can be applied in combination with all pseudo- or quasi-random sequences.

#### 1.3.3.5 Listfile

Additionally, in some cases it might be necessary to add specific design variables to the first generation. By adding a close-to-optimum configuration convergence speed can increase or one can use OPAL++ instead of an optimization simply for a parameter study, i.e., to test specific design variables (e.g, for a mesh-independency study). In this case, simply a list of design variables can be given to OPAL++.

### 1.3.3.6 Comparison of methods

The different methods are compared in Fig. 1.6. Fig. 1.6 (a) shows a completely random initialization for the  $[0;1] \times [0;1]$  domain, using 128 samples. Similarly, Fig. 1.6 (b) presents a Halton-sequence, Fig. 1.6 (c) a Hammersley sequence, Fig. 1.6 (d) a Sobol sequence, Fig. 1.6 (e) a near-orthogonal LHS while Fig. 1.6 (f) shows a constrained Hammersley-sequence.

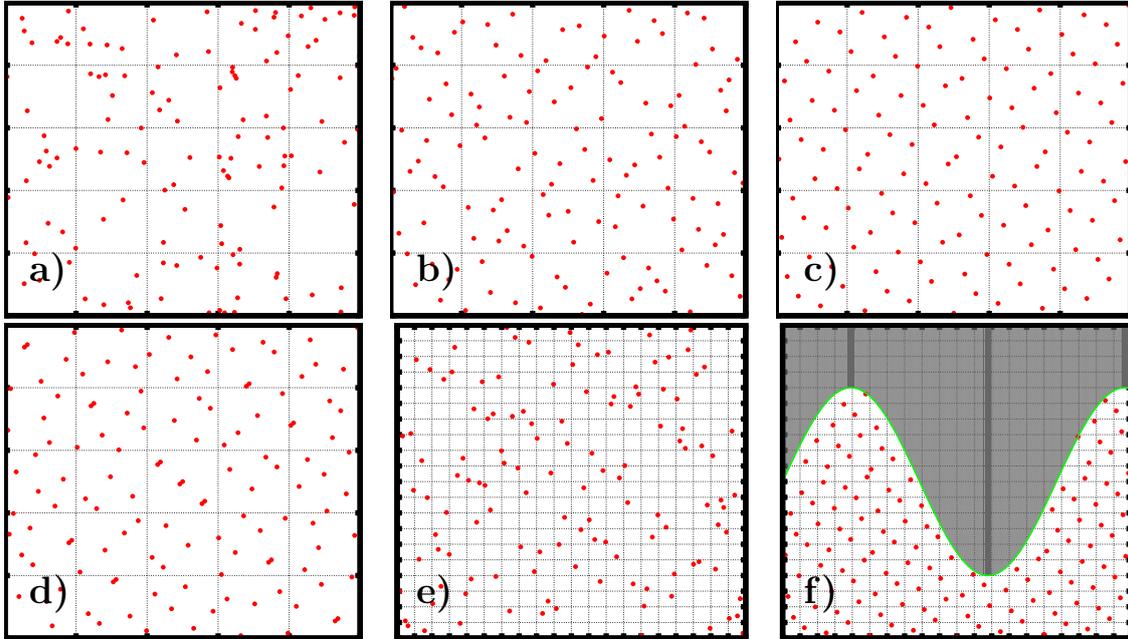


Figure 1.6: Comparison of different initialization methods for the first generation in OPAL++ (a) Random (b) Halton (c) Hammersley (d) SOBOL (e) Near-orthogonal LHS (f) Constrained Hammersley

### 1.3.4 Optimization algorithms

OPAL++ supports many different optimization algorithms, mainly focusing on multi-objective methods. It is out of the scope of the present thesis to provide a detailed description for each method, thus, appropriate reference(s) will be provided.

#### 1.3.4.1 Single-objective algorithms

For single-objective optimization, the following three methods have been implemented. These methods cannot be used for multi-objective problems. Additionally, all methods follow a completely different principle.

- **Firefly**: Firefly is a swarm optimization method, mimicking the behavior of fireflies [230]. It is extremely well suited for multi-modal problems, as fireflies can self-organize around local and global optima. The main advantage of firefly method compared to GAs is the exchange of information between the individuals.

- **Differential Evolution (DE)**: DE is a fast and robust global optimization method. Instead of mimicking swarm behavior or evolution, it is based on vector operations. For each iteration, new vectors are generated by mixing randomly chosen vectors (mutation) and mixing them with target vector [206]. OPAL++ provides support for *de/best/1/exp*, *de/rand/1/exp*, *de/best/1/bin* and *de/rand/1/bin* schemes.
- **GENETIC1**: a simple, single-objective genetic algorithm implemented in OPAL++:
  - All variables have real representation.
  - Each generation contains  $N = \text{const.}$  individuals.
  - Tournament with 2 cycles is used to select parents (can be changed if necessary).
  - SBX is used for cross-over.
  - Polynomial mutation operator of NSGA-II is applied [67, 68].
  - Selection method depends on the settings (by default, the life of the individuals can last only for 1 generation; however, one can also activate elitist selection).

Although the method is very simple, it proved to be very efficient, as it will be discussed in Chapter 4.

#### 1.3.4.2 Multi-objective algorithms

The following multi-objective methods are implemented in OPAL++:

- **NONE**: NONE denotes, that no optimization has to be performed; only the design variables contained in the Design-Of-Experiment (thus, first generation) have to be evaluated.
- **HYENA**: Hybrid adaptive multi-objective optimization method proposed by the current author based on the experiences gained in the present thesis. The method will be discussed in Chapter 8 in a detailed manner.
- **NSGA-II**: Non-dominated Sorting Genetic Algorithm-II developed by Kalyanmoy Deb et al. [68]. NSGA-II is one of the most popular methods in multi-objective optimization, with already several thousands of applications. As a result, it expresses a higher level of maturity compared to novel algorithms. This method will be used in Chapter 2 and 3. NSGA-II is not only fast due to the use of non-dominated sorting algorithm, but preserves diversity as well by using crowding distance. It is an elitist method without an external archive. However, as there is no proof of convergence for NSGA-II, partial deterioration can occur. Sometimes, designs of the true Pareto front can even get lost during the iterations [136]. To overcome this problem, HYENA uses an external archive.
- **Omni-optimizer**: an improved version of NSGA-II, developed by Kalyanmoy Deb [69]. The method provides support for the optimization of multi- or single-modal and multi- or single-objective problems as well. Despite the improvements, it cannot compete with the popularity of NSGA-II. The idea of modified crowding distance implemented for the support of multi-modal problems has been used in HYENA as well.

- **OMOPSO**: Optimized Multi-Objective Particle Swarm Optimization (OMO-PSO). PSO methods are very different from Genetic Algorithms; instead of genes they mimic the behavior of swarms. OMOPSO represents an improvement over the normal single-objective PSO, as it provides support for multi-objective problems and uses additionally techniques such as crowding, mutation and  $\epsilon$ -dominance [196].
- **SPEA2**: Strength Pareto Evolutionary Algorithm 2 developed by ETH Zürich [233]. SPEA2 provides a similar performance to NSGA-II.
- **FastGPA**: FastGPA provided an improvement compared to other methods (e.g., NSGA-II, SPEA2) by introducing a dynamic population size. Thus, fewer individuals are wasted in the first generation on uninteresting regions. Instead, the population size increases when approaching the optimum (optima) [78]. The idea of dynamic population size, as introduced in FastGPA, has been adapted into HYENA as well.

Please note, that some multi-objective methods (e.g., NSGA-II, Omni-optimizer) work well for single-objective problems as well.

### 1.3.5 Advanced post-processing methods

Following an optimization, OPAL++ collects all results (all variables and objective values) into different CSV (Comma Separated Values) files to enable analysis. One CSV file contains all individuals encountered during the optimization. Additionally, for each generation one CSV file contains the actual population and one additional file the actual Pareto-front. This enables the analysis using many different software, e.g., Microsoft Excel. For more advanced analyses (Surrogate Models and Uncertainty Quantification), OPAL++ provides additional support, mainly through providing C++ libraries (some coding is required).

#### 1.3.5.1 Uncertainty Quantification

In order to support Uncertainty Quantification (UQ), Non-Intrusive Polynomial Chaos Expansion (NIPCE) was implemented into OPAL++. The implemented method supports UQ for independent uncertain input variables, of arbitrary distribution. The polynomial basis functions are evaluated based on the Gram-Schmidt orthogonalization [227]. Evaluation of the coefficients is done either by numeric integration [95, 101] or by Point Collocation. The detailed description and an example application of the method are given in Chapter 6.

#### 1.3.5.2 Surrogate models

OPAL++ provides support for creating different surrogate models based on evaluated design variables. Additionally, the generated Surrogate Models can be exported and saved from OPAL++. Without providing here a detailed description of the methods, the list of methods supported by OPAL++ is given below:

- **Least Squares Method (LSQ)**: LSQ is a fitting method, which tries to minimize the sum of the squares of error (between the interpolated and exact solution).

OPAL++ supports linear/non-linear LSQ methods as well, based on polynomials or on arbitrary parametric functions. Additionally, instead of uniform weighting, one can use Iteratively Reweighted LSQ as well, with Gaussian, inverse quadratic or inverse weighting method. Iteratively Reweighted Least Squares Method will be applied in Chapter 4.

- **Radial Basis Functions (RBF)**: RBF is an approximation method, which describes a response as a sum of radial functions, i.e., functions whose value depends only on the distance from the center of origin:

$$y(\mathbf{x}) = \sum_{i=1}^N w_i \phi(\|\mathbf{x} - \mathbf{x}_i\|), \quad (1.31)$$

where  $\mathbf{x}_i$  are the different RBF centers and  $\phi$  is the radial basis function. However, radial basis functions are dependent on a reference radius, e.g., for Gaussian  $\phi(r) = e^{-(r/R)^2}$ . OPAL++ supports both simple and augmented RBF method.

For choosing the  $R$  constant and the basis function, OPAL++ divides the points into centers (training set, 75% of points) and calibration points (testing set, 25% of points). Afterwards, the value of the constant and the basis function type (from Gaussian, multiquadratic, inverse quadratic, inverse multiquadratic, thin plate spline, cubic, compact1, compact2) is optimized with regard to the error computed for the testing set.

- **Kriging**: Kriging models have their origin in geostatistical application [198]. They belong to Gaussian process regression, which treats the response as if “it were a realization of a stochastic process” [39]. For Kriging, a semivariogram has to be provided by an expert, however, OPAL++ calibrates the semivariogram in an automated way, by optimizing a non-nuggeted, exponential power semivariogram. OPAL++ supports Simple Kriging, Ordinary Kriging, Moving Neighbourhood Kriging and Kriging with Detrending.

OPAL++ can save and load the computed surrogate models. This way, metamodel-based optimization becomes possible. In metamodel-based optimization, after training the metamodel [216], the objective functions are replaced with their metamodel-based approximation:

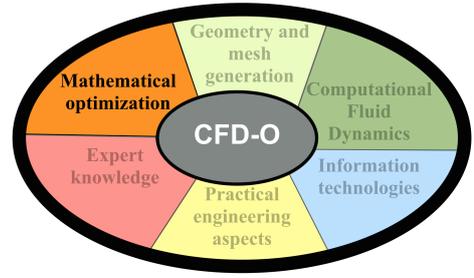
$$\mathbf{y}(\mathbf{x}) \approx \mathbf{y}^{meta}(\mathbf{x}), \quad (1.32)$$

updated in an on-line or off-line manner. As metamodels can be evaluated in a much faster way, optimization can be executed on this model within a couple of seconds.

$$\mathbb{O}^{meta} : \mathbf{y}^{meta}(\mathbf{x}) \xrightarrow{\mathbf{x}} \min \quad (1.33)$$

If the optimum of the virtual optimization is no optimum when evaluated with CFD, new training points have to be defined and added. Metamodel-based optimization will be used at the end of Chapter 4.

*Having defined all the basic concepts, it is now time considering the first, simple application of CFD-based optimization relying on OPAL++.*



## Chapter 2

# Optimization of a heat-exchanger - Analyzing the Pareto front

“In my opinion, no single design is apt to be optimal for everyone.”

---

Donald Norman

*The present chapter will especially focus on questions related to the theory of Mathematical Optimization, by analyzing the complex structure of the Pareto front for a simple Computational Fluid Dynamics case, to provide the necessary insights into CFD-O.*

### 2.1 Aim of the analysis

*In the previous chapter a short introduction was given to the theory of optimization and to OPAL++, which is the software developed and used in the present thesis.*

*In the followings, an aspect of the parameterization will be discussed, using a Computational Fluid Dynamics example. In many cases optimization problems are attacked by “brute force”, using a large number of parameters or a redundant description. Such setups may lead to slower convergence, incorrect or incomplete results, discouraging engineers from further use of optimization methods.*

*To present the most important aspects, a relatively easy CFD setup will be used: a theoretical two-dimensional cross-flow tube bank heat exchanger arrangement problem with internal laminar flow is considered in this chapter. The objective is to optimize the arrangement of tubes and find the most favorable geometries, in order to simultaneously maximize the rate of heat exchange while obtaining a minimum pressure loss. For this, a systematic study is performed involving a large number of simulations using the global optimization method NSGA-II. Special focus is given on the fundamental understanding of the structure of the Pareto front, comparing symmetric designs and asymmetric geometries, and on the effect of constraints concerning the speed of exploration. Altogether, more than 140 000 CFD simulations are used in this direct optimization.*

*The discussion is based on an extended version of the article:*

DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Systematic analysis of the heat exchanger arrangement problem using multi-objective genetic optimization. *Energy* 65

(2014), 364–373 [56].

### 2.1.1 Optimization of heat exchangers in the literature

The optimal placement of the heat sources or sinks in a channel, a cavity or a heat exchanger may affect dramatically the performance of the considered device. These problems are highly interesting, as they couple the questions relating to parameterization, topology and constraint handling. For this purpose, Computational Fluid Dynamics (CFD) coupled with Genetic Algorithms (GA) have a high potential to explore a large number of different configurations.

One efficient way to further speed up an optimization process is to use a Design of Experiment (DOE) with a limited number of evaluations as starting point, followed by the generation of a Response Surface using one of the available Response Surface Methods (RSM). Finally, a virtual optimization can be performed on this surface using, e.g., again GA. However, the global quality of such an advanced interpolation technique completely depends on the complexity of the problem. While it works well for simple configurations, it may completely fail for concurrent objectives involving local minima and stiff surfaces, as often found in practical applications. In order to avoid this issue and to eliminate any interpolation error resulting from the technique, all designs considered in this chapter are directly evaluated using CFD simulations.

Numerical investigation of heat exchangers and heat exchange processes in general is an intensive field of research due to its practical importance. For instance, Şahin et al. [189] investigated the flow around plate fin and tube heat exchanger for a single cylinder. Monteiro and Mello analyzed the thermal performance and pressure drop in ceramic heat exchangers [160].

The optimal location of heat sources was investigated by da Silva et al. [48] for forced convection and in [49] for natural convection. The optimal shapes of heat exchangers have been discussed by various authors [82, 83]. Bello-Ochendo et al. [18] performed gradient-based optimization of conjugate cooling channels. Pussoli et al. [181] optimized finned-tube evaporators. Arrangement problems have also been considered; for instance, Sudhakar et al. [209] analyzed the optimal arrangement of heat sources for a laminar, steady flow using ANSYS-Fluent for CFD.

Thanks to recent progress concerning multi-objective optimization problems, corresponding studies became increasingly popular. Lee et al. [138] performed multi-objective optimization of plate heat exchangers using MOGA, while Hämäläinen et al. studied papermaking processes with multi-objective methods based on simulations [107]. Ranut et al. [183] studied the optimal shape of tube bundles using multi-objective optimization. Seung-Hwan et al. [231] considered the optimization of radial heat sinks for weight and thermal resistance using a weighted sum method. Their approach relies on a response surface based on measurements, followed by a GA study. Hilbert et al. [110] performed a multi-objective shape optimization of 2D laminar tube bank heat-exchanger using GA. Copiello and Fabbri analyzed and optimized the heat transfer process considering longitudinal wavy fins [46] and using SPEA2 optimization method [233]. Foli et al. [89] and Okabe et al. [172] have obtained optimal results for a micro heat exchanger based on different multi-objective optimization methods. Iqbal et al. [116] determined optimal configuration for heat transfer processes under laminar conditions using GA. Nobile

et al. [169] performed a multi-objective optimization of convective periodic channels using again GA. Multi-objective genetic optimization of tube arrangement for cooling of prismatic bodies was analyzed by Robbe and Sciubba [186].

This long list of publications demonstrates the importance of this issue.

## 2.2 Setup of CFD model

### 2.2.1 Tube bank heat exchanger configuration

One possible simulated configuration of the analyzed design is exemplified in Fig. 2.1. Here,  $x_i$  and  $y_i$  denote the coordinates of the centre of tube  $i$ . Please note, that the dots over the outermost numerals represent repeating decimals.

Air enters the domain from the left at  $T_{\text{inlet}} = 293$  K and is warmed up by passing between tubes in which a warm fluid flows in the corresponding application. The tubes are supposed to have a constant outer wall temperature,  $T_{\text{wall}} = 343$  K. Walls 1-10 are considered to be fully adiabatic. In the followings the inlet is denoted by  $\Omega_{\text{inlet}}$ , and the outlet as  $\Omega_{\text{outlet}}$ . At the inlet a parabolic velocity profile is defined with average velocity of  $\bar{v}_{\text{inlet}} = 5$  cm/s. The outlet is defined as a pressure outlet with zero gauge pressure ( $p_{\text{outlet}} = 0$  Pa). The diameter of all tubes is  $D_{\text{tube}} = 10$  mm and the height

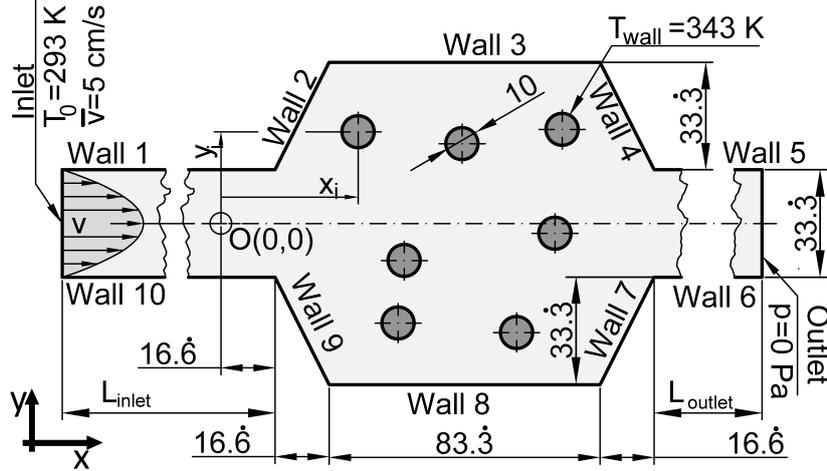


Figure 2.1: Schematic layout of optimal arrangement problem for a basic heat exchanger configuration [56]

of the main channel is  $H_{\text{channel}} = 33.3$  mm. In order to eliminate possible effects of the boundary conditions on the solution, the length of the inlet channel is increased to  $L_{\text{inlet}} = 4.5H_{\text{channel}}$ . For the same purpose at outlet, the condition  $L_{\text{outlet}} = 8H_{\text{channel}}$  is enforced.

To further avoid any impact of local discretization errors near the boundaries, the objectives are evaluated at  $\Omega_{\text{inlet}^*}$  and  $\Omega_{\text{outlet}^*}$ , which are at a distance of  $2.5H_{\text{channel}}$  and  $1.0H_{\text{channel}}$  from the inlet and outlet, respectively.

Due to the small range of temperatures involved in this problem, all material properties are regarded as constants:

$$c_p = 1006.5 \frac{\text{J}}{\text{kg} \cdot \text{K}} ; k = 0.026341 \frac{\text{W}}{\text{m} \cdot \text{K}} ; \rho = 1.1649 \frac{\text{kg}}{\text{m}^3} ; \mu = 1.868 \cdot 10^{-5} \frac{\text{kg}}{\text{m} \cdot \text{s}} . \quad (2.1)$$

The Reynolds number using the tube diameter as reference is

$$\text{Re} = \frac{\rho \bar{v} D}{\mu} = 31.18 < \text{Re}_1, \quad (2.2)$$

where  $\text{Re}_1 = 46$  according to Barkley and Henderson [15], or  $\text{Re}_1 = 47 \pm 1$  following Norberg [171]. Above this value, the fluid undergoes a supercritical Hopf bifurcation leading to a periodic, oscillatory flow and the physics becomes more complex [14]. However, in the present case, the Reynolds number is well below this limit. Therefore, a laminar and steady flow can be safely assumed, as done in all simulations. For flow past a stationary cylinder, 3D instabilities occur first at around  $\text{Re}=189$  and  $\text{Re}=259$  (Mode A and Mode B, respectively [15]). Therefore, 3D simulations are not required in the present case and only a 2D slice of the heat-exchanger will be simulated in what follows. The effect of buoyancy is neglected as well, since only small density differences will appear. Gravity is not included in the computational model.

## 2.2.2 Validation of the CFD model

The optimization takes place in a fully automatic manner, including geometry creation, unstructured mesh generation and CFD evaluation. The computational geometry and mesh were generated by ANSYS Gambit 2.4.6 using a prepared journal template file.

A meaningful optimization based on design comparisons requires a very high numerical accuracy. As a consequence, an appropriate emphasis must be set on mesh quality. However, when evaluating automatically hundreds or thousands of designs, a case-by-case analysis or the generation of structured meshes are both out of question. High-quality but automatic meshing is required, for which unstructured triangle mesh elements are recommended.

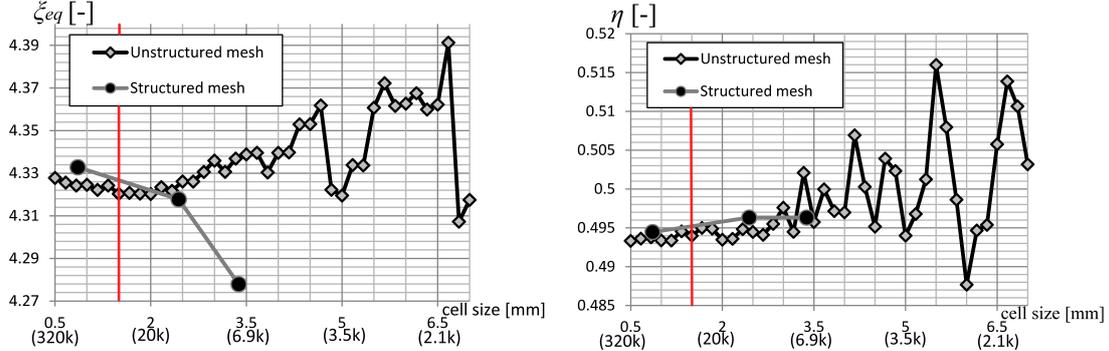


Figure 2.2: Mesh-independency test concerning (a) equivalent hydraulic resistance (b) efficiency

In a first step a mesh independency test was performed for different mesh sizes for a random design. Corresponding results can be seen in Fig. 2.2(a) and Fig. 2.2(b). On the  $x$ -axis the typical size of the mesh cells are shown (as defined in Gambit), together with the total number of computational cells in brackets. The rhombuses represent the automatically-generated unstructured triangle meshes, while the circles represent

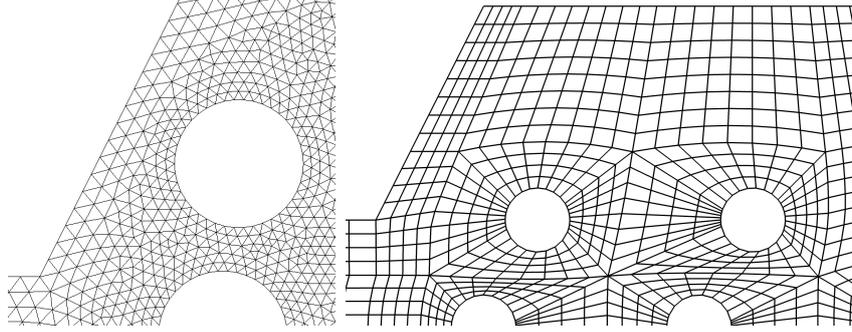


Figure 2.3: **(a)** Typical mesh for part of the domain with 36k cells **(b)** Topology of the block-structured mesh

block-structured meshes with quadrilateral cells created manually (the block-structure is exemplified in Fig. 2.3(b)).

As can be seen, when the mesh is getting finer (going toward the left side in the figures), the values obtained with both meshes converge. For a mesh size lower than 2.5 mm, the maximal relative difference between both approaches stays below 0.3 % for the equivalent hydraulic resistance, and below 0.6 % for the theoretical efficiency. This is small enough to allow a meaningful comparison of the designs.

Finally, choosing a mesh size of 1.5 mm (indicated with red) was the retained compromise between evaluation speed, needed memory storage and resulting accuracy. A zoom of such a typical mesh for one random design can be seen in Fig. 2.3(a). During the later optimization, all meshes contain between 32000 and 37000 finite volume cells.

All CFD simulations rely on the steady, 2D laminar, coupled solver of ANSYS-Fluent, neglecting gravity. For a higher numerical accuracy second-order spatial discretization is used for the pressure, and a second-order upwind scheme is activated for momentum and energy. The node-based calculation is used, since it is better suited for triangular meshes compared to the default, cell-based gradient scheme [6]. The flow domain is initialized using the velocity profile for the channel flow.

To check the possible impact of prescribed residual levels, all mesh-independency tests were run for very low thresholds of  $10^{-10}$ ,  $10^{-11}$  and  $5 \cdot 10^{-12}$  (relative) residual limits (with 1000 as maximal iteration number). Convergence was always achieved and no significant difference was found in the results or in the computing time. A  $10^{-11}$  criterion was finally applied for all evaluations. If a simulation should not reach this convergence level by iteration 1000, it is considered as problematic and becomes invalid.

After finishing the study every fourth element of the Pareto front was re-evaluated with a much finer mesh, and the results were compared against the results of the original mesh. The error in the case of efficiency always remained below 0.8 %, which is excellent. In case of pressure the relative error was for most designs in the  $[-2,0]$  % range, while for isolated designs (those with the largest pressure drop, i.e., with the smallest distance between walls) the obtained error was in the  $[-7,-3]$  % range. Since this error showed a monotonic trend (i.e., larger errors correspond to larger pressure drops), the structure of the Pareto front is not influenced by this uncertainty, and the precision of the obtained optimization is therefore appropriate.

## 2.3 Setup of the optimization

### 2.3.1 Design variables

Let us denote the  $x$  and  $y$  coordinate of the center of the seven cylinders by  $x_i, y_i$  ( $i = 1, \dots, 7$ ). Most designers would assume by instinct or experience that the optimal designs have to be symmetric. This is indeed a very interesting question to be analyzed. Thus, three different cases will be considered here. Case 1 (from now on referred as *Assym*) did not imply any restriction; thus, the resulting design might be either asymmetric or symmetric (if the convergence criteria are met). Case 2 (from now on referred as *Sym<sub>1</sub>*) and Case 3 (from now on referred as *Sym<sub>2</sub>*) were forced to be symmetric arrangements of different types, as explained below. Using such an approach it is possible to analyze quantitatively, if the symmetric designs (with fewer variables, thus easier and faster) are indeed superior, and if the differences between the two approaches are significant.

*Assym* is an optimization with 14 independent variables. This means that all the coordinates of the cylinders are optimized simultaneously. It is here the most difficult problem to solve due to the curse of dimensionality. The parameter space is:

$$x_i \in [21.\dot{6}; 128.\dot{3}], y_i \in [-45; 45], i = 1 \dots 7. \quad (2.3)$$

*Sym<sub>1</sub>* has only 7 independent variables:

$$x_i \in [21.\dot{6}; 128.\dot{3}], i = 1 \dots 5; y_4, y_5 \in [0; 45]. \quad (2.4)$$

The design was forced to be symmetric by

$$y_{1,2,3} = 0, x_{i+2} = x_i, y_{i+2} = -y_i, i = 4, 5. \quad (2.5)$$

*Sym<sub>2</sub>* has again 7 independent variables, namely

$$x_1, x_2, x_3, x_4 \in [21.\dot{6}; 128.\dot{3}]; y_2, y_3, y_4 \in [0; 45]. \quad (2.6)$$

The design was forced to be symmetric by

$$y_1 = 0, x_{i+3} = x_i, y_{i+3} = -y_i, i = 2, 3, 4. \quad (2.7)$$

All design variables were real-coded, thus SBX (Simulated Binary cross-over) had to be used [66].

As one can see, the employed representation can result either in "crowded" or "spacious" designs (i.e., small or large distances between the tubes). This gives us the possibility to analyze the assumption, that the best heat exchanger configurations are found for tubes placed close to each other. This idea corresponds to an implicit third objective function, the minimization of the size of the heat exchanger. Though beneficial for most practical purposes, less crowded designs could lead to significantly lower pressure loss while not degrading noticeably the heat transfer process. Such configurations might be particularly interesting for applications with low temperature and pressure differences, as considered more and more often to increase energetic efficiency of all existing systems.

### 2.3.2 Constraints

For further analysis, the variable-volume ( $VV$ ) metric is introduced, which is the volume of the  $n$ -dimensional region associated with the feasible domain. Although one cannot expect this simple metric to give a clear idea of the complexity of the optimization problem, for problems differing only in constraints or range of variables, greater values of  $VV$  mean a larger exploration space and therefore, a more considerable amount of work. On the other hand, comparing values of  $VV$  is impossible for problems involving different dimensions. Problems with smaller  $VV$  can obviously be explored faster.

In order to reduce the search domain, possibly to take into account practical aspects associated to later production and also to avoid any software problem during, e.g., the generation of the computational mesh, additional constraints have to be introduced. In the present case, mesh collisions between cylinders can be avoided if all designs satisfy

$$(x_i - x_j)^2 + (y_i - y_j)^2 > (2R)^2, \quad i, j = 1 \dots 7; i \neq j, \quad (2.8)$$

where  $R = D_{tube}/2$  is the cylinder radius. Moreover, collision of the tubes with walls must be avoided as well. Considering the already assigned parameter space, collision with wall can only appear at walls 2, 4, 7, 9. To avoid this problem, straightforward additional constraints have been coded using elementary coordinate geometry.

Applying all constraints finally reduces the search domains. The volumes of the domains were evaluated with Monte-Carlo integration using the Moivre-Laplace theorem for 99.9999% confidence intervals ( $\pm 0.003\%$ ):

$$VV_{constr.}^{Assym} = VV^{Assym} \cdot 24.8556\%, \quad (2.9)$$

$$VV_{constr.}^{Sym_1} = VV^{Sym_1} \cdot 28.8659\%, \quad (2.10)$$

$$VV_{constr.}^{Sym_2} = VV^{Sym_2} \cdot 38.8119\%. \quad (2.11)$$

The original search domains were thus reduced to  $1/3 - 1/4$  of the original extent. Additionally, it must be ensured that the optimization algorithm will not waste time on evaluating cyclic permutations of the same designs. As for *Assym* the variable space to be explored is relatively large, it had to be ensured that the optimization algorithm will not be wasted on evaluating cyclic permutations of the same designs. This was easily achieved by adding the constraint

$$x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq x_6 \leq x_7. \quad (2.12)$$

This way 7! permutation of the same design can be eliminated, thus reducing the search domain to

$$\begin{aligned} VV_{constr.}^{Assym, \emptyset cyclic} &= VV_{constr.}^{Assym} / 5040 = VV^{Assym} \cdot (4.9202 \cdot 10^{-3}\% \pm 5 \cdot 10^{-5}\%) \simeq \\ &\simeq VV^{Assym} / 20324. \end{aligned} \quad (2.13)$$

Similarly

$$x_4 \leq x_5; x_1 + R \leq x_2; x_2 + R \leq x_3 \quad (Sym_1), \quad (2.14)$$

$$x_2 \leq x_3 \leq x_4 \quad (Sym_2), \quad (2.15)$$

leading to a reduction of

$$VV_{constr.}^{Sym_1, \emptyset cyclic} = VV_{constr.}^{Sym_1} / 12 = VV^{Sym_1} \cdot (2.4 \pm 10^{-3}\%) \simeq V^{Sym_1} / 41.6, \quad (2.16)$$

and

$$VV_{constr.}^{Sym_2, \emptyset cyclic} = VV_{constr.}^{Sym_2} / 6 = VV^{Sym_2} \cdot (6.5 \pm 2 \cdot 10^{-3}\%) \simeq VV^{Sym_2} / 15.5. \quad (2.17)$$

It must be kept in mind that such reductions of the search space do not always speed-up a practical optimization problem. With more constraints, the optimization algorithm encounters increasing difficulties to stay in the feasible domain, since corresponding regions may become severely disconnected, and the freedom of movements in the variable space is lost. Although it is widely recommended in the literature to avoid the undue use of constraints, only very few examples are provided on real problems concerning the impact of constraints on the speed of exploration.

Consider a simple optimization problem with two indistinguishable variables,  $x_1, x_2$  (i.e., swapping the values of the variables does not result in different designs). Supposing, that an optimal solution is located at  $x_1 = 30; x_2 = 32$  and the current population contains an individual  $x_1 = 32; x_2 = 33$ , the individual can easily mutate with  $x_2 = 33 \rightarrow 30$  to the optimal solution. However, if we reduce the domain by imposing  $x_1 \leq x_2$  (50 % reduction), the same mutation would lead to an infeasible solution. To reach the optimal solution both variables would have to mutate simultaneously, or in two steps (however, if the intermediate design is inferior, it might be deleted). With this example one can see, that although the constraints reduce the search domain, flexibility is lost. At 7 or 14 variables the question is much more difficult. On one hand, the flexibility is greatly reduced. But the search domain is drastically reduced as well (in our case, by a factor  $7! = 5040$  for *Assym*). The two effects are thus competing. One could see this loss of flexibility in a different way: without constraints the optimum can be reached on many different ways, while with constraints only few exist.

In order to identify the best strategy, three additional optimizations were performed as well, where cyclic permutations were allowed. From now on, they are denoted as *Assym<sub>cyclic</sub>*, *Sym<sub>1,cyclic</sub>* and *Sym<sub>2,cyclic</sub>*.

### 2.3.3 Objective functions

Our obvious goal is to maximize the efficiency of the heat exchanger. However, this goal cannot be mathematically formulated as a single value or indicator; at least two objectives are needed, which are known from many previous works as being concurrent. On one side, the pressure loss (which is proportional to the power need) is to be minimized. Due to the employed boundary conditions, it reads:

$$\Delta p = \bar{p}|_{\Omega_{inlet*}} - \bar{p}|_{\Omega_{outlet}}. \quad (2.18)$$

The associated value can be in the range  $\Delta p \in [0, \infty)$ . For a more meaningful comparison, an equivalent, non-dimensional hydraulic resistance [46] is calculated for the computed system:

$$\Delta p = \xi_{eq} \cdot \frac{\rho}{2} \bar{v}^2 \rightarrow f_1 = \xi_{eq} = \frac{2\Delta p}{\rho \bar{v}^2}, \quad (2.19)$$

where  $\bar{v}$  is the average velocity at the inlet.

Other options could have been the power input per unit volume [139], friction factor [169] or simply the pressure drop [12, 110].

The second objective is related to the efficiency of the heat transfer process. Several different metrics have been proposed in the literature for this quantity, e.g., temperature difference between outlet and inlet [110], Nusselt number [46, 116], entropy resistance [42], overall heat transfer coefficient [163], thermal resistance [231], entropy generation number [46, 181], total heat transfer [139] or mean heat transfer coefficient [139].

In the present problem, the total heat transfer can be formulated along the outflow boundary as

$$Q = \rho \cdot c_p \int_{\Omega_{outlet*}} (T(y) - T_0) \cdot \mathbf{v} \cdot \mathbf{dA} = \rho \cdot c_p \int_{\Omega_{outlet*}} (T(y) - T_0) \cdot v_x dA . \quad (2.20)$$

In practice, this integral was calculated along a fixed, vertical line ( $x = \text{const.}$ ) placed slightly before the outlet. Taking into account the imposed wall temperature, the maximum possible value of the total heat transfer is  $Q_{max,th.} = 97.706$  W. For a more straightforward comparison, the theoretical efficiency of the heat-exchanger  $\eta$  is introduced, which can be calculated as  $f_2 = \eta = Q/Q_{max,th.}$ , and is non-dimensional. This value is to be maximized. The possible range of the second objective reads  $f_2 \in [0, 1]$ .

### 2.3.4 Automation

One important part of the optimization process is to define the evaluation chain for the individual designs. After a new generation has been created, the following approach was applied for every individual:

- If the individual is infeasible (i.e., constraints are violated), do not proceed further. Otherwise, send the individual to the next free worker node for evaluation.
- Modify the prepared Gambit (and/or Fluent) journal template according to the values of the design variables.
- Create computational mesh in Gambit and check quality. If the mesh cannot be successfully generated, the individual becomes invalid and no evaluation takes place.
- Perform the CFD computation with Fluent.
- Check if convergence criteria are met. If not, the individual becomes invalid and is not considered further. Otherwise, compute the objectives  $f_1$  and  $f_2$ .

### 2.3.5 Optimization settings

A simultaneous, multi-objective optimization is performed using the NSGA-II algorithm as implemented in OPAL++ for the previously discussed objective functions. All optimizations were carried out with  $N = 100$  individuals in each generation.  $Sym_1$ ,  $Sym_{1,cyclic}$ ,  $Sym_2$ ,  $Sym_{2,cyclic}$  were evaluated using 250 generations, while  $Assym$  and  $Assym_{cyclic}$  were evaluated using 480 generations (due to the higher dimensionality of the problem). Designs that would violate constraints were not evaluated. As a result,

20937 CFD-evaluations were performed for  $Sym_{2,cyclic}$ , 19825 for  $Sym_2$ , 18457 for  $Sym_1$ , 17616 for  $Sym_{1,cyclic}$ , 33054 for  $Assym$  and 34678 for  $Assym_{cyclic}$ . In this extensive test, 144 567 different CFD simulations have thus been finally carried out. The probability of mutation has been set to  $1/n$ , as recommended in [68], while the probability of crossover was set to 0.8. The first generation of the population was initialized with feasible individuals using a Sobol sequence [122]. Due to the different formulation (and different number of variables) of the six different variants of the problem, all six optimizations received different initial populations.

Using the parallelization of OPAL++, a very good speed-up was achieved with 12 nodes. An acceleration of the full optimization process by a factor between 9 and 10 was obtained, meaning roughly 80% parallel efficiency.

## 2.4 Results

### 2.4.1 Comparing Pareto fronts

Analyzing the results first confirms the findings in [136]; in the NSGA-II algorithm certain elements of the Pareto front can get lost during the iterative process. Hence, the front can deteriorate. Furthermore, visual analysis revealed, that the Pareto front returned by the algorithm is really close to the front that is obtained by merging all individuals ever encountered during the optimization process and deleting dominated individuals; no difference can be found during visual inspection.

A modified (i.e., domination is used instead of weak domination) Coverage of Set is introduced here based on the original one [65]:

$$C'(A, B) = \frac{|\{b \in B | \exists a \in A : a \prec b\}|}{|B|}, \quad (2.21)$$

i.e., it shows the fraction of  $B$  which is dominated by elements in  $A$ . If  $C'(A, B) = 0$ , then no elements exists in  $B$ , which are dominated by individuals in  $A$ . Moreover if  $C'(A, B) = 0$  and  $C'(B, A) > 0$  then it can be stated, that the set  $B$  is better than  $A$ , if they are in common regions. However, if  $A$  and  $B$  discovered different parts of the true front, such comparison has no meaning.

Although this metric does not give a perfect view of the optimality of sets, it can still reveal several characteristics about convergence without knowledge of the true Pareto front (diversity is not considered here, as the same algorithm was used for diversity preserving). The comparison confirms the findings of [136], i.e., for all six cases  $C'(front, totalfront) = 0$  was found (this is always true), but  $C'(totalfront, total) \in [0.32; 0.55]$  (where total front is assembled from non-dominated individuals in regard of all encountered elements in a specific optimization run). However, visual inspection reveals that the two fronts cannot be differentiated from each other due to their minimal distance. Therefore, from now on *front* will always mean the total front.

### 2.4.2 Cyclic vs. normal version

As mentioned previously, the comparison of cyclic and normal optimization is interesting. Although preventing cyclic permutations should simplify the problem considerably, the

resulting number of constraints can hinder exploration by the optimization algorithm. This trade-off is indeed an issue, and larger, but more flexible design spaces are indeed found to be favorable. This supports the recommendation, that although GAs are able to work with constraint handling mechanisms, an excessive number of constraints should be avoided; free movements within the variable space should be a priority.

The cyclic versions have a larger freedom (due to the mechanism described earlier), i.e., they can easier reach specific designs along different paths, while the normal version can get stuck, but can at the same time focus on other regions with a higher precision. The comparison of the results for  $Assym$ ,  $Sym_1$  and  $Sym_2$  when allowing or preventing permutations is shown in Figs. 2.4 and 2.5. Figure 2.4 represents a typical Pareto front. The shaded area denotes the feasible set in the objective space, i.e., designs that exist. The white area represents the unfeasible set, which cannot be reached for the problem considered here since they lie beyond the Pareto front. Our dream is to advance as far as possible towards the lower right corner.

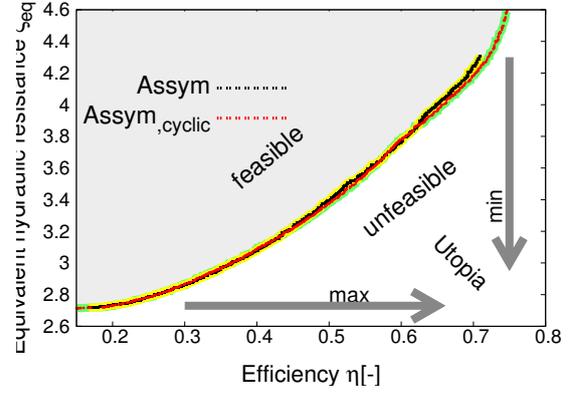


Figure 2.4: Comparing the Pareto-front of optimization problems  $Assym$  and  $Assym_{cyclic}$

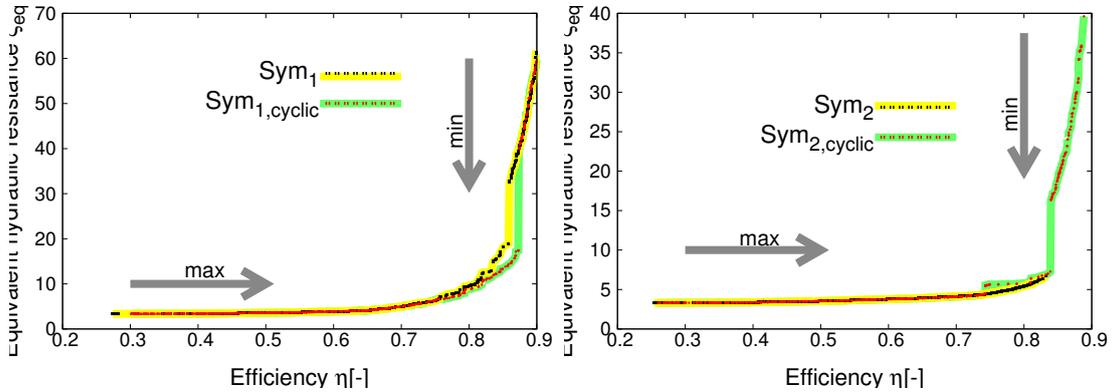


Figure 2.5: (a) Comparing the Pareto-front of optimization problems  $Sym_1$  and  $Sym_{1,cyclic}$  (b) Comparing the Pareto front of optimization problems  $Sym_2$  and  $Sym_{2,cyclic}$

Although the curves in Figs. 2.4 and 2.5 are completely different at first look, they indeed just cover different parts of the Pareto front (please note the different axis ranges).

As plotting all the curves on a single diagram would be too crowded to understand their relationship, the Approximated Continuous Generational Distance ( $ACGD$ ) is proposed for the analysis of two-objective Pareto fronts by modifying the original definition of Generational Distance ( $GD$ ). First, the best known approximation of the Pareto-front,  $\mathbf{X}_{P,f}$  is created (e.g., in this case by merging all simulations into a single archive) and is normalized into the  $[0, 1]^m$  range along with the analyzed Pareto-front  $\mathbf{X}_{P,f}^*$  using the same technique and ranges.

Next, a linear interpolation of the best known approximation of the Pareto-front is created by connecting all neighbouring points in the front  $\mathbf{X}_{P.f.}$ , and the arc coordinate along the interpolation is calculated for each point (the use of attainment surface might have been correct as well, or in many cases even better, but this would give unproportionally small distances measured from the Pareto-front at extremes). Supposing that  $\mathbf{x}_j \in \mathbf{X}_{P.f.}$  is closest (according to Euclidean distance in objective space) to  $\mathbf{x}_i \in \mathbf{X}_{P.f.}^*$ , where  $f_k(\mathbf{x}_{j-1}), f_k(\mathbf{x}_j), f_k(\mathbf{x}_{j+1})$  are all in monotonically decreasing or increasing order, the metric is defined as

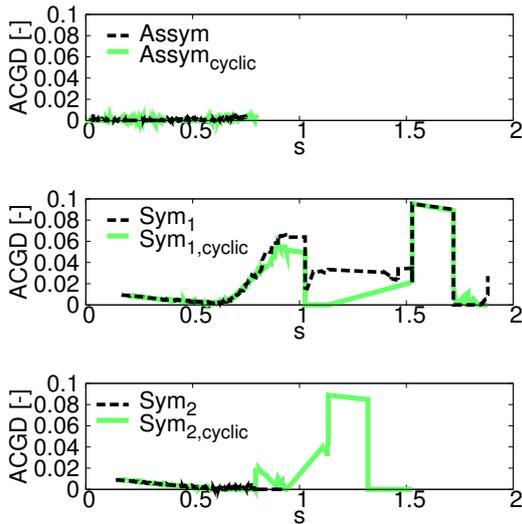
$$ACGD : s_i \rightarrow d_{ij}, \quad (2.22)$$

where the distance between  $\mathbf{x}_i \in \mathbf{X}_{P.f.}^*$  and  $\mathbf{x}_j \in \mathbf{X}_{P.f.}$  is

$$d_{ij} = \sqrt{\left( \sum_{k=1}^{m=2} (f_k(\mathbf{x}_j) - f_k(\mathbf{x}_i))^2 \right)}, \quad (2.23)$$

and the arc coordinate along the interpolated Pareto front is:

$$s_i = \sum_{l=2}^j \left( \sum_{k=1}^{m=2} (f_k(\mathbf{x}_l) - f_k(\mathbf{x}_{l-1}))^2 \right) \quad (2.24)$$



$ACGD = 0$  means that the analyzed Pareto front gave the points for the total front in the specific region, while horizontal lines will indicate discontinuities in the best known approximation of the Pareto front. Thus, this metric will not reveal information about the behavior of the analyzed Pareto front next to discontinuities, but will reveal the relationship between the analyzed fronts and the best known approximation of the Pareto front. Also, it can show not only how perfect the convergence was, but also which areas of the front were discovered.

Figure 2.6: ACGD metric for all six cases

Previous findings are supported by Fig. 2.6, but it is revealed as well, that *cyclic* variants (cyclic permutations allowed) are superior compared to the overly constrained (no cyclic permutations allowed) variant (except for some regions). Due to the stochastic nature of GA, for clear conclusions more optimizations had to be analyzed.

### 2.4.3 *Assym* vs. *Sym<sub>1</sub>* vs. *Sym<sub>2</sub>*

*Assym* and *Assym<sub>cyclic</sub>* discovered only the lower left part of the best known approximation of the Pareto front, both quite equivalently. However, they failed to find the corresponding part of the front for higher efficiencies. It is obviously more difficult to

reach this region when involving a larger number of design variables. These findings are confirmed by Fig. 2.7, which shows that NSGA-II reached the Pareto front quite quickly, but has spent the rest of the iterations refining it, without much progress concerning domains with higher efficiencies. This can be explained from one side with the curse of dimensionality. The complexity of a problem with 14 variables is tremendously larger than with 7 variables. Moreover, as the tubes can move independently, the number of constraints increases as well (in order to avoid collisions). The domain of *Assym* contains more “obstacles” or discontinuities compared to *Sym*<sub>1</sub> or *Sym*<sub>2</sub> (see Eqns. 2.9-2.11).

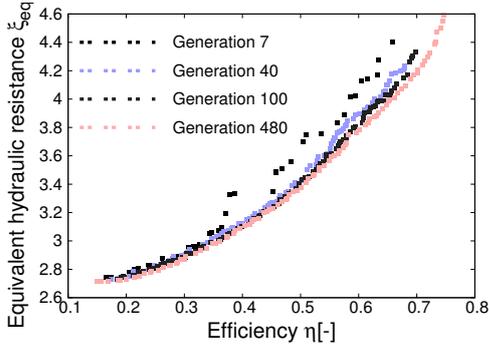


Figure 2.7: Convergence of Pareto front for *Assym*<sub>cyclic</sub> as a function of the generation number

On the other hand, both *Sym*<sub>1</sub> and *Sym*<sub>2</sub> managed to explore a considerably larger region of the front. Additionally, it can be seen that the cyclic and normal versions of the optimization have often concentrated on different parts of the Pareto front. Figure 2.8 shows the evolution of the Pareto front for *Sym*<sub>1,cyclic</sub>. The lower left part of the Pareto front is quickly found, as for *Assym*. But, thanks to the lower number of variables and constraints (and thus “obstacles”), the *Sym*<sub>1,cyclic</sub> optimization is able to expand progressively into the domain of higher thermal efficiency.

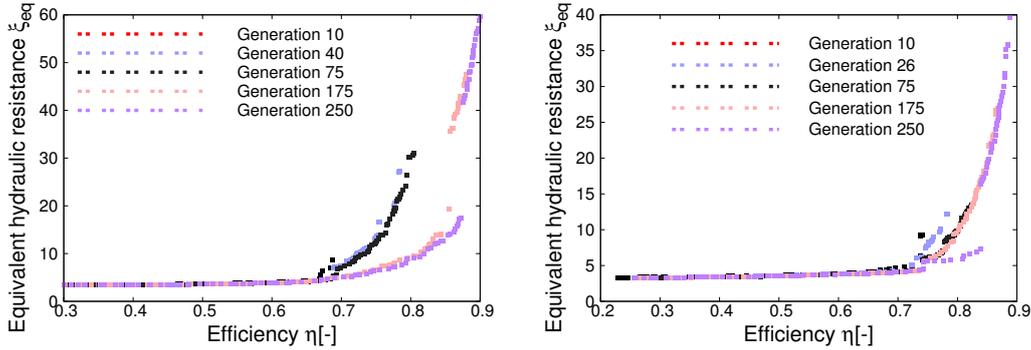


Figure 2.8: (a) Convergence of Pareto front for *Sym*<sub>1,cyclic</sub> as a function of the generation number (b) Convergence of Pareto front for *Sym*<sub>2,cyclic</sub>

## 2.4.4 Analysis of the Pareto front

By merging all individuals from all optimizations into a single data set and sorting dominated designs out, the best approximation of the true Pareto front can be derived. This Pareto front spans the range  $2.714 \leq \xi_{eq} \leq 59.687$ ,  $0.1497 \leq \eta \leq 0.8997$  and is shown in Fig. 2.9. The gray points represent dominated individuals encountered in the optimization process. Some selected optimal designs together with the computed temperature contours are also plotted using the same color scale as later (e.g., in Fig. 2.10 (a)). This figure clearly illustrates that different regions were successfully explored by the different optimization runs. Overall, an excellent exploration has been obtained.

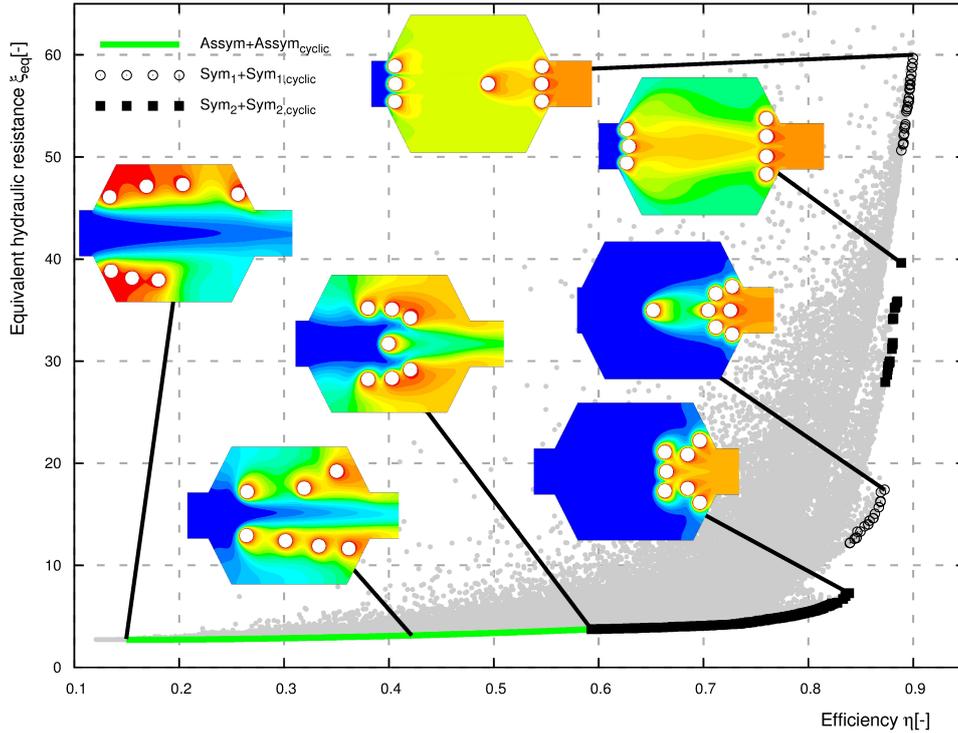


Figure 2.9: Best approximation of the true Pareto front from all optimizations [56]

The Pareto front consists of five clearly distinguishable segments. The lower left domain of the Pareto front is completely dominated by *Assym*. This can be easily explained. Since an odd number of cylinders have been retained, symmetric designs imply that at least one cylinder had to reside on the centerline, thus increasing immediately the pressure drop. Moving now from lower values of the thermal efficiency to higher values, the cylinders far from the centerline start to move slowly toward the central horizontal axis, interacting more and more strongly with the main flow. These designs provide low to moderate efficiency with extremely low pressure drop. Starting around  $\eta = 0.6$ , the *Sym<sub>2</sub>* optimization becomes essential to populate the Pareto front. It is perhaps interesting to note, that at this cross-over point, three completely different configurations leading to the same objective values have been identified, as shown in Fig. 2.10 (a). Several such points exist, i.e., the problem is multi-modal. Moreover, many relatively small regions contain designs, which are of completely different geometry, e.g., Fig. 2.11 (a). This justifies the need for optimization algorithms that preserve the diversity not only in the objective space, but in the variable space as well, as e.g., the Omni-Optimizer [69].

Fortunately, in our case a complete archive of all individuals was maintained, so that the diversity-preserving mechanism did not have an adverse effect. Moreover, this raises the interesting question, whether algorithms maintaining a full archive could be superior for practical engineering purposes. In such algorithms the diversity-preserving mechanisms would not focus on the archiving process, but on the selection of individuals for mating. These algorithms could not deteriorate with time, i.e., proof of convergence would be always provided. This question is discussed in more details in Chapter 8.

Going now above  $\eta = 0.84$ , the Pareto front appears to become discontinuous, and various optimization formulations are alternatively dominant. Starting at the first gap

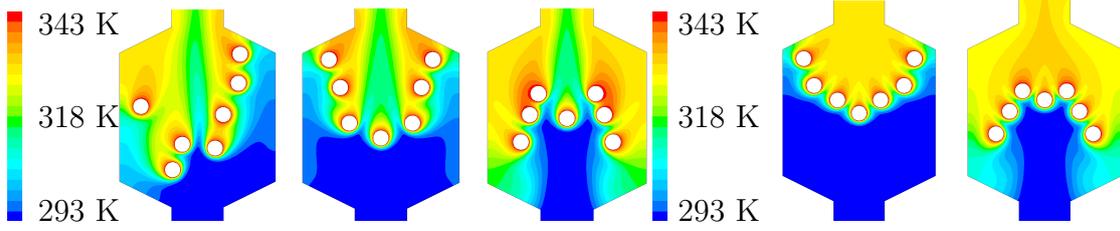


Figure 2.10: **(a)** Different designs all leading to  $\eta = 0.6$ ,  $\xi_{eq} = 3.77$  (from left to right, obtained from *Assym*, *Sym<sub>2</sub>*, *Sym<sub>2</sub>*, respectively) **(b)** Different *Sym<sub>2</sub>* designs for  $\eta = 0.736$ ,  $\xi_{eq} = 4.326$

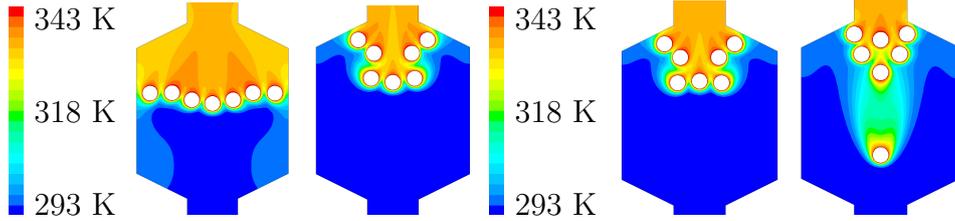


Figure 2.11: **(a)** Designs for  $\eta = 0.83$ ,  $\xi_{eq} = 6.7$  (*Sym<sub>2</sub>*) and  $\xi_{eq} = 7$  (*Sym<sub>2</sub>*) **(b)** Discontinuity in the Pareto front observed at  $\eta = 0.84$ , from  $\xi_{eq} = 7.3$  (*Sym<sub>2</sub>*, left) to  $\xi_{eq} = 12.2$  (*Sym<sub>1</sub>*, right)

( $\eta > 0.84$ ), the equivalent hydraulic resistance increases very rapidly ( $\xi_{eq} = 7.3 \rightarrow 60$ ), but the efficiency does not increase very much ( $\eta = 0.84 \rightarrow 0.9$ ). This means that, while compact heat exchangers will provide the best heat transfer rates, it is possible to find configurations with similar efficiency but significantly lower pressure drops. This could be an interesting direction for the design of low-power heat exchangers in applications with low pressure differences, as found sometimes for energy recuperation.

The first discontinuity is found around  $\eta = 0.84$ ,  $\xi_{eq} = 7.3$  (*Sym<sub>2</sub>*)  $\rightarrow$   $\xi_{eq} = 12.2$  (*Sym<sub>1</sub>*), see Fig. 2.11 (b). A second, even larger gap is present at  $\eta = 0.87$ ,  $\xi_{eq} = 17.4$  (*Sym<sub>1</sub>*)  $\rightarrow$   $\xi_{eq} = 29.9$  (*Sym<sub>2</sub>*) (Fig. 2.12 (a)). As usual for stochastic optimization methods, it cannot be proven, that no solutions exist in these gaps. Still, considering that more than 140 000 designs have been evaluated, we believe that these are either real discontinuities in the objective space, as observed for instance in the well-known optimization test case ZDT3 [45], or that only sensitive (i.e., not robust) solutions reside in those regions. Moreover, as seen in Figure 2.9, the important criterion in locating the discontinuities was not the distance in the objective space (as several smaller discontinuities exist), but the 'topological distance', i.e., the different rules used to enforce symmetry.

Finally, the last obtained discontinuity appears at  $\eta = 0.89$ , from  $\xi_{eq} = 39.6$  (*Sym<sub>2</sub>*)  $\rightarrow$   $\xi_{eq} = 50.6$  (*Sym<sub>1</sub>*), see Fig. 2.12. There, the observed pressure drop is very large, 22 times larger, than the smallest value found ( $\xi_{eq,min} = 2.714$ ). In this region of the Pareto front, most of the heat exchanger is left empty. Therefore, such designs could readily be transformed into compact configurations, as used in most practical tube bank heat exchangers.

As a whole, it is particularly interesting to see that the finally obtained, best approximation of the Pareto front contains results from different formulations of the problem. Therefore, it is indeed worth running an optimization involving different formulations

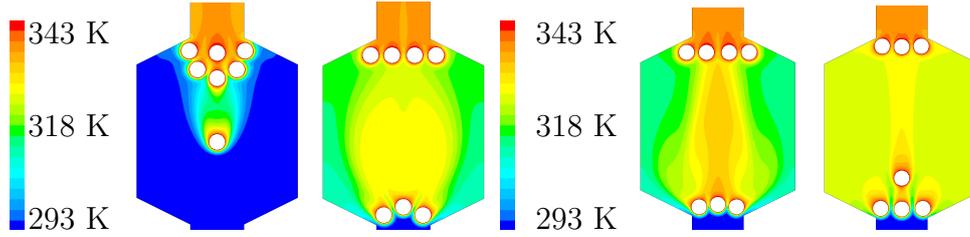


Figure 2.12: (a) Discontinuity in the Pareto front observed at  $\eta = 0.87$ , from  $\xi_{eq} = 17.4$  ( $Sym_1$ , left) to  $\xi_{eq} = 29.9$  ( $Sym_2$ , right) (b) Discontinuity in the Pareto front at  $\eta = 0.89$ ,  $\xi_{eq} = 39.6$  ( $Sym_2$ )  $\rightarrow$   $\xi_{eq} = 50.6$  ( $Sym_1$ )

and constraints in order to explore more widely the possible designs. An optimization relying only on the most general, *Assym* configuration could be misleading, since no design could be found for  $\eta > 0.74$ , while they do exist in reality. The complex impact of the constraints on the optimization process was illustrated as well. While adding additional constraints is attractive to reduce parameter space and in principle speed-up the computations, it might have a strong negative impact on the GA exploration, so that whole regions of the Pareto front might be missed. This is even true for very large numbers of evaluations, as considered here. Moreover, the current study points out the importance of the engineer carrying out the final analysis. Even such a simple problem can result in a complex Pareto front. A superficial analysis of the results can lead to premature, inaccurate or even wrong decisions.

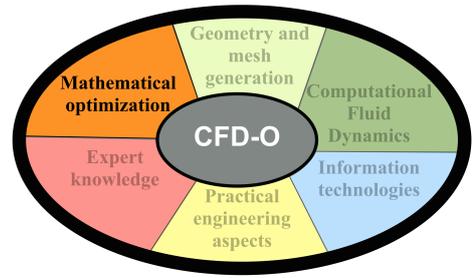
## 2.5 Conclusions and summary

In this chapter, a genetic algorithm has been applied to the multi-objective optimization of a two-dimensional heat exchanger arrangement problem. All evaluations rely on direct CFD simulations, after having checked mesh independency and avoiding any undue influence of the boundary conditions. Accepting a very large number of evaluations ( $> 140\,000$ ), the Pareto front connecting equivalent hydraulic resistance and efficiency of heat transfer is finally revealed.

It has been shown, that avoiding cyclic permutations, i.e., adding constraints to the optimization problem in order to reduce the search domain does not provide any advantage. The gain associated to the reduction in the feasible configurations is spoiled by the induced complexity associated to additional constraints, hindering the progress of the GA. Letting the algorithm freely search in the domain is computationally more effective.

The finally obtained Pareto front consists of designs showing alternating types of symmetry, with noticeable discontinuities between continuous regions, from low-power configurations to designs close to compact geometries. Symmetric designs are generally superior to asymmetric ones at high efficiency, but not necessarily in the low-efficiency region.

*After discussing symmetry and effect of constraints on the speed of exploration, in the following an attempt will be made to try to find a parameterization, which is even more efficient for the optimization of such arrangement problems.*



## Chapter 3

# Extending the heat exchanger example - What is the impact of parameterization?

“I do the very best I know how - the very best I can; and I mean to keep on doing so until the end.”

---

*Abraham Lincoln*

*In the previous chapter, questions related to symmetry of parameterization, number of constraints and curse of dimensionality were analyzed. Based on the experience made in the previous chapter, a second question, namely the speed of exploration depending on the parameterization will be analyzed. However, in this study, the original CFD study was extended to higher complexity, to be able to highlight the differences in convergence resulting from the parameterization. Thus, questions related to the theory of Mathematical Optimization will again be in the focus.*

*The results are partly based on a conference presentation:*

DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Genetic optimization of heat-exchanger arrangement problem. In *ProcessNet Jahrestreffen der Fachgruppen Computational Fluid dynamics, Mischvorgänge und Rheologie* (Würzburg, Germany, 2014) [55].

### 3.1 Setup of CFD model

The CFD setup and the analysis of the objectives will be only briefly presented in what follows, as the setup is based on the previous chapter.

#### 3.1.1 Tube bank heat exchanger configuration

A similar geometry was applied as in the previous study, but in this case the geometry was parameterized using 4 values, see Fig. 3.1 (a):

- $D_1$ : the diameter of the in- and outflow channel,

- $D_2$ : largest diameter of the heat-exchanger,
- $s$ : slope of the wall connecting  $D_1$  and  $D_2$  section,
- $L$ : length between the in- and outflow channels,
- $R_{tube}$ : radius of tubes.

As you can see, there were 8 tubes applied but the geometry was forced to be symmetric (as in the previous chapter symmetric optimizations turned out to be superior), with two tubes in the middle. The radius was fixed to  $R_{tube} = 10$  mm, the slope to  $s = 6$  and the channel diameter  $D_1 = 40$  mm. Additionally, the inlet and outlet channels had a length of  $L_{in} = 200$  mm and  $L_{out} = 320$  mm, respectively, to ensure fully developed flow profiles.

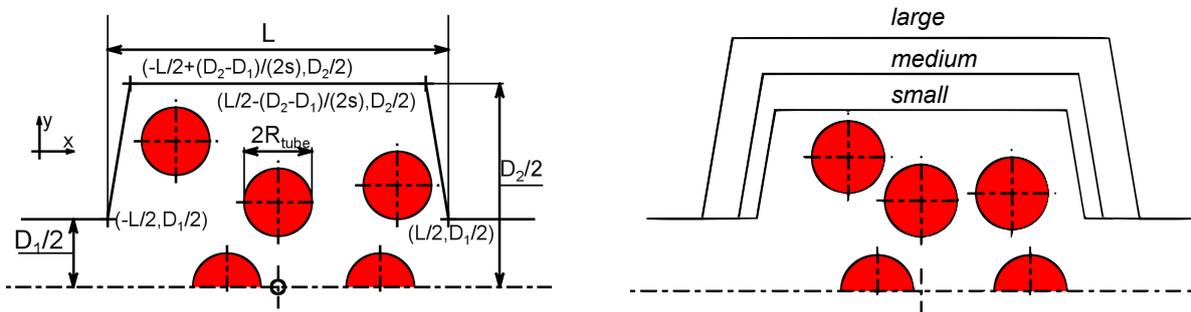


Figure 3.1: (a) Schematic layout of arrangement problem for the present heat exchanger configuration (b) *Small*, *medium* and *large* heat-exchanger configuration

In the study, three different sizes were analyzed (see Fig. 3.1 (b)):

- *small* configuration:  $D_2 = 100$  mm,  $L = 90$  mm,
- *medium* configuration:  $D_2 = 120$  mm,  $L = 100$  mm,
- *large* configuration:  $D_2 = 140$  mm,  $L = 120$  mm.

### 3.1.2 Physical modeling

The temperature of the tube walls was set to constant  $T_{tube} = 368$  K, while the inlet temperature was specified as  $T_{in} = 278$  K with  $\bar{v} = 3$  m/s average air speed (parabolic profile using User-Defined Function or shortly UDF). All other walls were considered to be adiabatic. As the setup is not anymore in the laminar flow regime,  $k$ - $\epsilon$  Realizable model with Enhanced Wall Treatment has been chosen. Computed from the channel diameter, turbulence intensity of 3.91% with length scale of 2.8 mm was applied to the inlet. Gravity is not included in the computational model.

Instead of using constant density and constant material properties, the temperature dependency of the different properties were modeled as well. For density ideal gas law was chosen, viscosity was specified using the Sutherland law while the thermal conductivity was specified using polynomial law. Only the specific heat capacity has been modeled as constant, which is an acceptable approximation for the considered temperature range.

Pressure and velocity equations have been solved in a coupled manner; all derivatives were discretized using second order or second order upwind method.

For temporal discretization steady-state simulation was assumed, as transient effects were negligible (for a single random case of *medium* configuration, both steady and transient computations were performed, resulting in identical objective function values, i.e.,  $\eta_{st.} = 0.2350909$ ,  $\eta_{trans.} = 0.2350985$ ,  $\xi_{st.} = 2.009210$ ,  $\xi_{trans.} = 2.009867$ ).

### 3.1.3 Spatial discretization

For the grid generation, ANSYS Gambit 2.4.6 was applied, with an unstructured quadrilateral mesh structure. A single parameter was defined, which controls the size of the mesh at different locations:  $s_0$ .

Afterwards, the mesh size at the walls was set to  $0.4s_0$ , at the tube walls to  $0.1s_0$ , everywhere else to  $0.5s_0$ . Four size functions ensured the smooth transition between the regions. Mesh near the wall was calibrated to ensure  $y^+ < 1$ . Following a mesh independency test (see section 3.1.4),  $s_0 = 1.2$  was chosen. This resulted in an average mesh size of 77.4k cells for *small*, 82.7k cells for *medium* and 90.6k cells for *large*.

The mesh structure can be seen in Fig. 3.2.

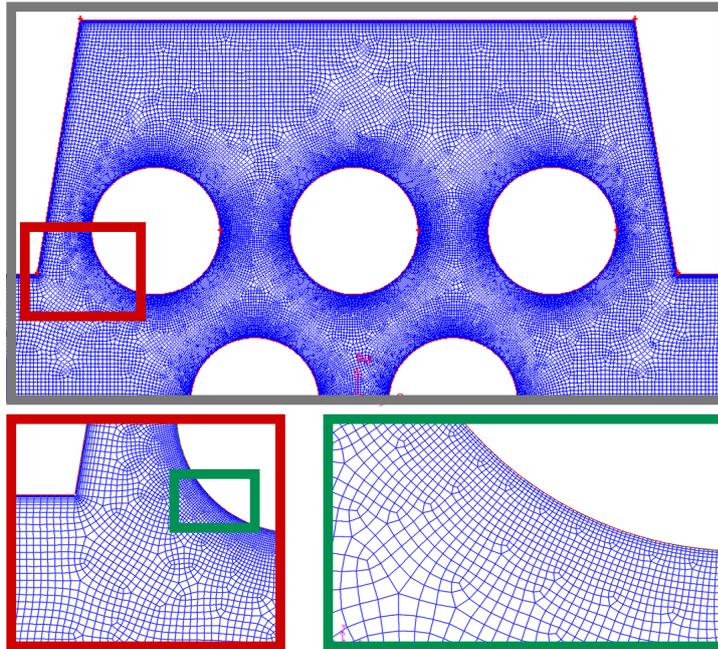


Figure 3.2: Mesh structure at different locations for an exemplary configuration of *medium* setup

### 3.1.4 Mesh independency

Afterwards, a mesh independency study was performed by varying the size of  $s_0$ . Figure 3.3 presents the number of cells, runtime,  $\xi$  and  $\eta$  (objective functions, see later) as a function of  $s_0$ . The green domain shows the final, chosen value  $s_0 = 1.2$ . As one can see, at this point,  $\eta$  and  $\xi$  are already quasi-constant.

Further decreasing  $s_0$  below 0.8, strange oscillations in the objective function values appear, which are the result of a convergence problem (however, if you look on the scale of the  $y$ -axis, they are indeed very small!). As the mesh size increases, more iterations would be needed to ensure convergence. Thus,  $s_0 = 1.2$  was retained for the followings as a compromise between runtime and accuracy.

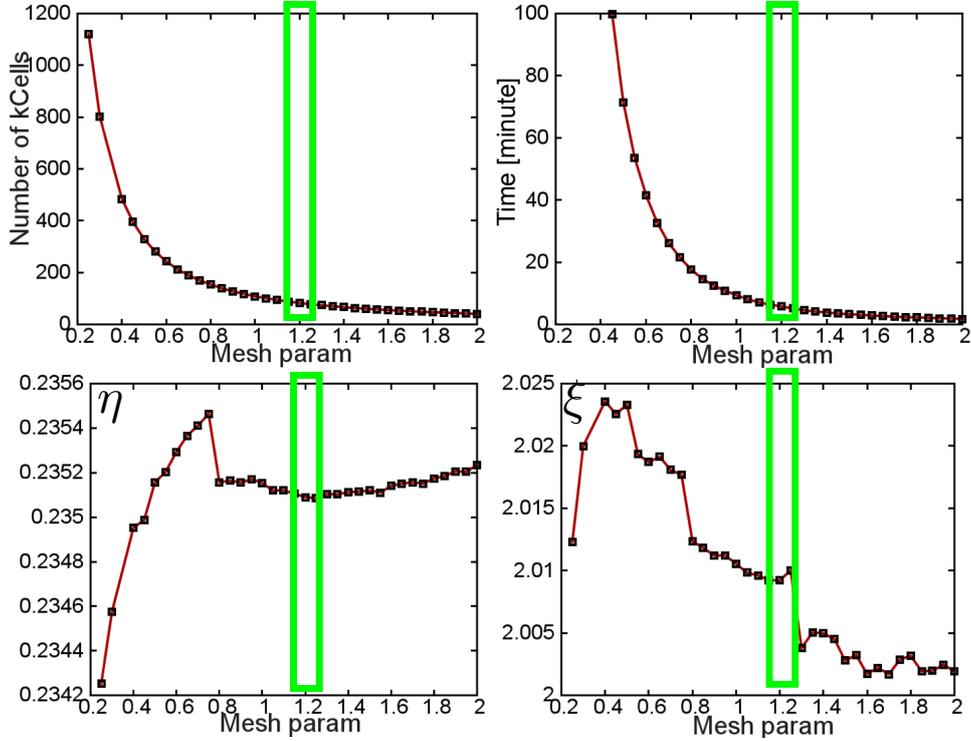


Figure 3.3: Mesh independency study for *medium* configuration

## 3.2 Setup of the optimization

As the aim of the present analysis is to find a better parameterization for the arrangement problem, two different versions were defined for the analysis with the same number of parameters.

### 3.2.1 Design variables

For the optimization, the coordinates of the cylinders ( $x_i, y_i, i = 1..5$ ) have to be varied; due to the symmetry  $y_1 = y_2 = 0$ , resulting in 8 Degrees-Of-Freedom (DOF). However, in the present case instead of using the coordinates directly, a non-dimensional form was defined with:

$$v_i \in [0, 1], i = 1..8 . \quad (3.1)$$

Afterwards, mapping functions were defined:

$$x_i = f_i(v_1, \dots, v_8), i = 1..5, \quad (3.2)$$

$$y_i = f_i(v_1, \dots, v_8), \quad i = 3..5, \quad (3.3)$$

which provide connection between the 8 non-dimensional variables and the real coordinates.

### 3.2.2 Constraints

In the followings,  $\delta_t$  and  $\delta_w$  denote the thickness of boundary layer for tubes and outer walls, respectively. These constants were automatically computed by OPAL++ and were necessary to ensure an appropriate mesh quality (mesh quality can be only ensured, if not only the tubes do not collide, but the prismatic boundary layers do not collide either). For the optimization, the following constraints were defined:

- *C1*: Collision of the tubes is not allowed:

$$(x_i - x_j)^2 + (y_i - y_j)^2 \geq (2R + 2\delta_t)^2 \quad i, j = 1..5, i \neq j, \quad (3.4)$$

- *C2*: Collision of the tubes and horizontal walls is not allowed:

$$\text{dist}((x_i, y_i), \text{wall}_h) \geq R + \delta_t + \delta_w, \quad (3.5)$$

- *C3*: Collision of tubes and sidewalls is not allowed:

$$\text{dist}((x_i, y_i), \text{wall}_s) \geq R + \delta_t + \delta_w, \quad (3.6)$$

- *C4*: The cylinders are not allowed to move into the symmetry plane (except the two fixed in the middle):

$$y_i > R + \delta_t \quad i = 3, 4, 5, \quad (3.7)$$

- *C5*: Triangle inequality for any three tube positions:

$$\text{dist}((x_i, y_i), (x_j, y_j)) < \text{dist}((x_i, y_i), (x_k, y_k)) + \text{dist}((x_k, y_k), (x_j, y_j)), \quad (3.8)$$

$$i, j, k = 1..5; i \neq j \neq k, \quad (3.9)$$

where *dist* denotes standard Euclidean distance on the plane. Please note, that with the tested mappings (parameterization 1 & 2), not all constraints are active during the optimization (will be discussed later).

#### 3.2.2.1 Parameterization 1

With parameterization 1, essentially the same method is followed, as earlier, i.e., the variables correspond to the coordinates directly. The non-dimensional variables are mapped using simple scaling to the following domain:

$$v_1, v_3, v_4, v_5 \in [0, 1] \rightarrow x_1, x_3, x_4, x_5 \in [-L/2 + R, L/2 - R] \quad (3.10)$$

$$v_6, v_7, v_8 \in [0, 1] \rightarrow y_3, y_4, y_5 \in [R + \delta_t, H/2 - (R + \delta_t + \delta_w)]. \quad (3.11)$$

An exception is variable 2, which was chosen to represent distance between the two cylinders on the symmetry plane:

$$v_2 \in [0, 1] \rightarrow \text{dist}(1, 2) \in [2R + 2\delta_t, L - 4R - 2\delta_t] \quad (3.12)$$

By this method, *C2*, *C4* and *C5* are automatically satisfied, thus only *C1* and *C3* are active. However, *C1* constraint is highly non-linear!

### 3.2.2.2 Parameterization 2

By parameterization 2, the variables do not correspond to the coordinates (except for  $v_1$ ), but to the distance between the cylinders (green segments in Fig. 3.4):

$$v_1 \in [0, 1] \rightarrow x_1 \in [-L/2 + R, L/2 - R] , \quad (3.13)$$

$$v_2 \in [0, 1] \rightarrow dist(1, 2) \in [2R + 2\delta_t, L - 4R - 2\delta_t] , \quad (3.14)$$

$$v_3 \rightarrow dist(2, 4), v_4 \rightarrow dist(1, 4), v_5 \rightarrow dist(4, 5) , \quad (3.15)$$

$$v_6 \rightarrow dist(2, 5), v_7 \rightarrow dist(1, 3), v_8 \rightarrow dist(3, 4) , \quad (3.16)$$

With this parameterization, the highly non-linear  $C1$  constraint is automatically satisfied, but other (easier) constraints  $C2 - 5$  remain active. As the collision of the tubes is impossible, the parameterization results in a more continuous search field. Furthermore, this parameterization is focusing on the *topology* of the problem. E.g., if  $v_1$  is varied, the same structure is shifted in  $x$ -direction, thus a relative arrangement can be retained even after mutation and cross-over.

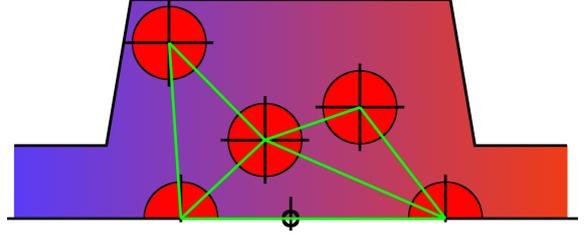


Figure 3.4: Parameters (length of green segments) of parameterization 2

### 3.2.3 Objective functions

Objectives were defined in agreement with the previous chapter, i.e., maximization of thermal efficiency and minimization of pressure drop is desired. However, slight modifications had to be done to take into account the variable density (but assuming a constant heat capacity):

$$f_1 = \eta = \frac{Q}{c_p \cdot \rho_{ref} \cdot (T_{tube} - T_{in}) \bar{v}_{in} A_{cross}} = \frac{\int_{\Omega_{outlet*}} \rho(T)(T(y) - T_{in}) \cdot v_x dA}{\rho_{ref} \cdot (T_{tube} - T_{in}) \bar{v}_{in} A_{cross}} \rightarrow \max \quad (3.17)$$

$$f_2 = \xi_{eq} = \frac{2\Delta p}{\rho_{ref} \bar{v}_{in}^2} = \frac{2(p|_{\Omega_{inlet*}} - p|_{\Omega_{outlet}})}{\rho_{ref} \bar{v}_{in}^2} \rightarrow \min \quad (3.18)$$

### 3.2.4 Automation

Due to the more complex physical models, the original workflow had to be extended with an additional criterion to ensure the quality of the solutions. The following approach was applied for every individual:

- If the individual is infeasible (i.e., constraints are violated), do not proceed further. Otherwise, send the individual to the next free worker node for evaluation.
- Modify the prepared Gambit (and/or Fluent) journal template according to the values of the design variables.

- Create computational mesh in Gambit and check quality. If the mesh cannot be successfully generated, the individual becomes invalid and no evaluation takes place.
- Perform the CFD with Fluent using 4 computing cores.
- Check if convergence criteria are met ( $10^{-5}$  residuals with no more than 2000 iterations). If not reached, the individual becomes invalid and is not considered further.
- A custom C++ library loaded by OPAL++ checks, if the monitored objective functions reached a constant value. If oscillations are encountered (i.e., RMS values are larger than 1% of the mean values), the configuration is considered invalid.

### 3.2.5 Optimization settings

For the optimization, the NSGA-II algorithm was applied again with  $N = 100$  individuals per generation. As not all configurations are feasible, optimization was run to approximately reach 10 000 full function evaluations with OPAL++ (i.e., CFD computations). This resulted in 177 generations (9842 CFD computations) for *medium* with parameterization 2 and 174 generations (10 034 CFD computations) with parameterization 1. For *small* and *large* only parameterization 2 was tested, resulting in 198 (10 006) and 160 (10 021) generations, respectively. One can see, that as the size of search domain increases, proportionally more individuals remain feasible, resulting in lower number of necessary generations to reach 10 000 CFD computations.

## 3.3 Results

In the followings, different questions will be analyzed using the performed optimizations with parameterization 1, 2 and configurations *small*, *medium* and *large*.

### 3.3.1 Efficiency of parameterizations

In the first step, the efficiency of the parameterizations was compared. The optimization of *medium* configuration was executed using both parameterizations; the Pareto-fronts are compared in Fig. 3.5, for generations 20, 75 and 175. As one can see, parameterization 2 has identified a significantly larger Pareto-front compared to parameterization 1. Furthermore, the tendency remains valid when comparing previous, unconverged generations as well (20 & 75). Theoretically, both parameterizations cover the same domain, but in practice not all methods exhibit the same efficiency. This highlights the importance of choosing a well-suited parameterization: equal care should be given by the engineers for the validity of the numerical model, for the choice of optimization algorithm and for the **choice of parameterization** as well.

### 3.3.2 Effect of domain size

In a second step, the effect of domain size was analyzed. As parameterization 2 proved to be superior, the optimizations were run using this method for *small*, *medium* and *large* configurations as well; results are compared in Fig. 3.6. As one can see, the methods agree on a large part of the Pareto-front, while *small* exhibits smaller efficiency regarding

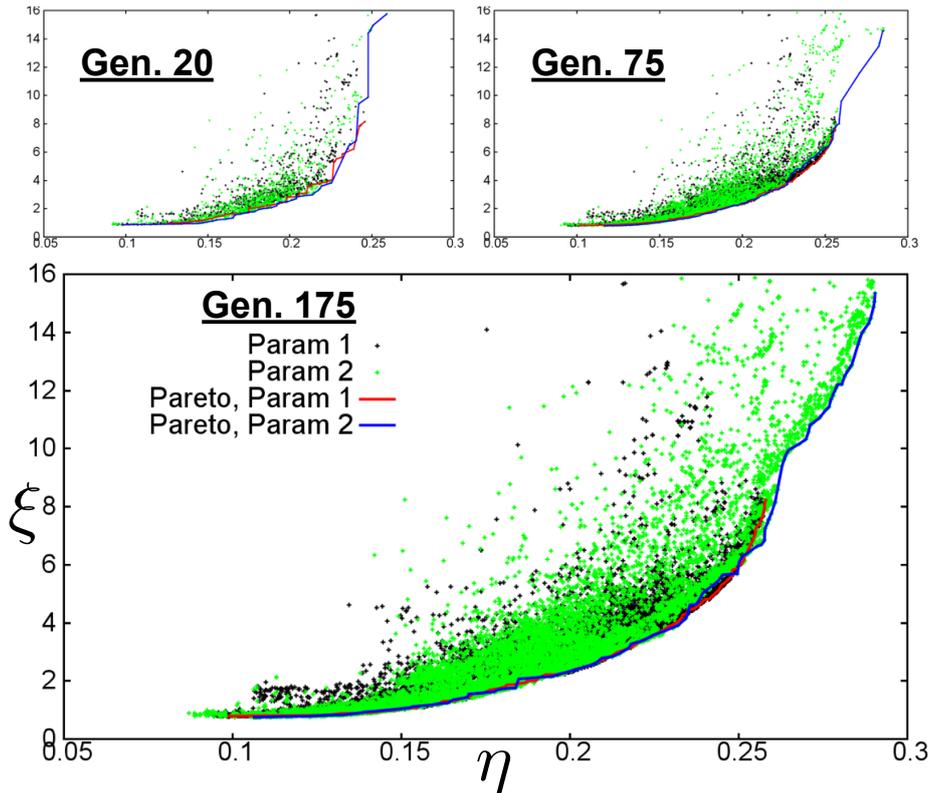


Figure 3.5: Comparison of optimization process with parameterization 1 & 2

pressure losses at higher thermal efficiencies (which is not surprising). However, *medium* and *large* agree almost completely.

This points out again, that when performing an optimization, one should not make unnecessary assumptions of the problem. E.g., here one would assume, that smaller configurations result in higher thermal efficiency but higher pressure drop as well. This is not the case, at least for *medium* and *large*. As a result, if one would try to perform the optimization based only on the *large* configurations, unnecessarily large heat exchanger designs would be achieved, when more compact alternatives exist. However, starting immediately with *small* would result in unnecessarily large pressure drops. Instead, the size of the heat-exchanger should be considered as an additional objective as well.

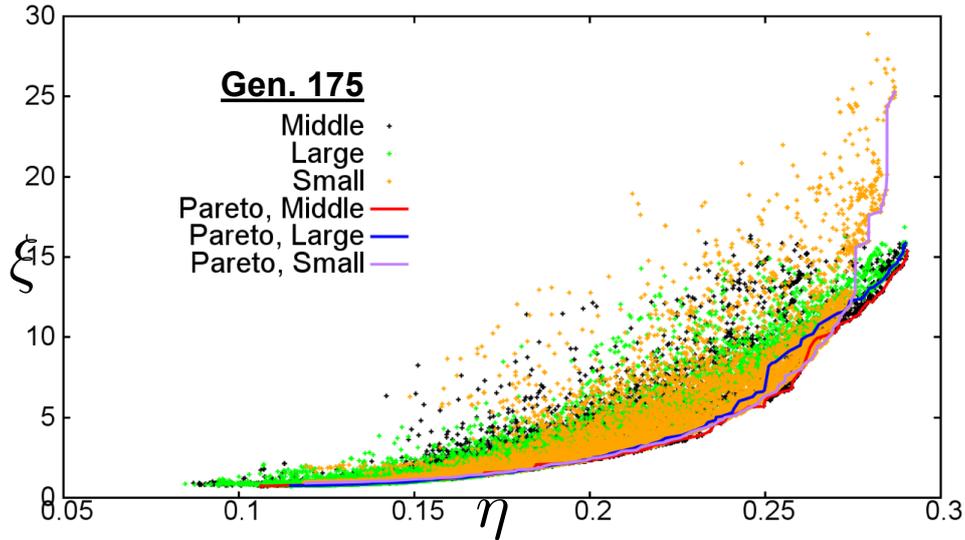


Figure 3.6: Comparison of optimization process with different sizes

### 3.4 Summary and conclusions

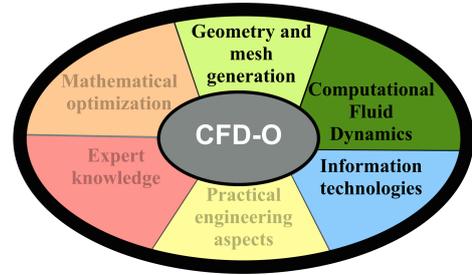
In this chapter, the importance of a correct parameterization was considered. First, the CFD model used in Chapter 2 was updated and validated. Afterwards, two different parameterizations and three different sizes were chosen for optimization. Altogether, 5 optimizations were performed, each with at least 10 000 direct CFD computations.

At first, one may think, that the two equivalent parameterizations result in the same Pareto-front. This is true, but one should not forget, that although the *problem is the same*, the *optimization methods are not well suited for all problems and formulations* (keep in mind the “no free lunch theory”!). In genetic optimizations, the probability of the appearance of a specific configuration decreases with an increasing distance, i.e., mutation generates configurations *near* the original configuration (although, one may define special operators for special problems as well). Thus, parameterizations, where the parameter space is highly disconnected due to the presence of constraints, are not desirable. A more efficient parameterization method for arrangement problems has been identified, resulting in better exploration. As a result, it may be advantageous for several problems to test different descriptions of the problem: this may result in configurations completely missed previously.

*This concludes the analysis of the heat exchanger configurations.*

*In what follows, even more difficult CFD problems will be considered, with much higher runtime requirements to analyze further questions of CFD-O. There, one cannot afford anymore thousands of computations. Instead, metamodels, Design-Of-Experiment methods or fewer parameters have to be applied with very robust automation methods.*

*The following chapter will start with the analysis of H-Darrieus rotors, where even 2D CFD simulations require many days in a parallel environment, as the problem is transient, turbulent and exhibits highly complex physics.*



# Chapter 4

## VAWT blade optimization

“The pessimist complains about the wind; the optimist expects it to change; the realist adjusts the sails.”

---

*William Arthur Ward*

*In the previous chapter a CFD-O problem with large number of parameters was analyzed. In this case, using metamodels proved to be impossible, due to the*

- (a) large number of discontinuities,*
- (b) large number of parameters,*
- (c) large number of global and local optima,*
- (d) complex response of the objectives.*

*It was presented that choosing the right parameterization and constraints can result in huge differences for the exploration.*

*The present chapter will especially focus on questions related to automated Geometry and mesh generation, validation of Computational Fluid Dynamics and to Information technology, i.e., actual implementation of the automation, based on wind turbines.*

*Due to the growing importance of wind energy, efficiency has become a crucial factor and optimization of wind turbines is being increasingly considered. The ultimate goal of the present chapter is to increase the performance of H-Darrieus rotors by choosing an optimal airfoil shape. The aerodynamic efficiency of commercially available rotors is already very high (large-size HAWTs can reach up to 50%). A further improvement is only possible using advanced optimization methods. Optimization in the literature itself is not unprecedented, but usually based on simplified models. For instance, Fischer et al. [87] performed multi-objective optimization of horizontal axis wind turbines with Blade Element Momentum (BE-M) code.*

*Even with CFD techniques, the resulting increase in performance cannot exceed more than a few percents. Therefore, it is essential to reach sufficient precision in corresponding CFD simulations, in order to be sure that the resulting improvement is not simply due to model uncertainties or numerical noise. Moreover, as computing time is an essential factor, one has to identify the really necessary mesh resolution and modeling approach,*

*delivering a high accuracy at acceptable computational cost. For this purpose, the model should be fast enough to perform at least 500 simulations with no more than 30 computers in no more than 3 months runtime. Besides, the model should provide mesh and temporal independency with no more than 2% error.*

*As a result, this chapter presents, how important it is for an optimization to identify the correct numerical setup. This step can be in some cases even more demanding, than the optimization itself! Following a very detailed numerical analysis, an optimization will be performed as well for an exemplary rotor.*

## **4.1 Introduction to wind energy**

*As several chapters of the current thesis focus on optimization of wind turbines, an introduction to the theory is given here.*

### **4.1.1 Importance of wind energy**

“The energy consumption of the world increases rapidly. Due to the increasing awareness of the population, to alarming reports concerning global warming, air pollution and nuclear accidents, these increasing needs have to be covered by clean and reliable energy sources. Wind energy can be a suitable answer to these needs. As a consequence, the gross production of wind energy in the EU27 (European Union with 27 members) increased from 80 PJ (2000) to 537 PJ (2010) just over a period of ten years [79], and an increase to 2300 PJ is prognosed for 2030 [80]. This would correspond to an increase in the share of gross production from 4.5% (2010) to 15.36% (2030). Meanwhile, in more developed countries the share of wind energy in the energy production is already high (e.g., Germany, 7.74% in 2012 [29])” [54]. For 2050, even larger increase is projected: altogether 3900 PJ production for the EU28 [81], this would correspond to 26% share.

Moreover, this increase is the result of the low necessary capital investments of wind energy [81]:

- The capital cost over time is expected to stay around 1100-1300 EUR/kW for onshore wind parks between 2015 and 2050.
- In contrast, the capital cost over time is expected to drop significantly for off-shore wind parks, from 4500 (2015) to 2800 (2050) EUR/kW.

Parallel to this decrease, the price of fossil fuels is expected to increase dramatically. As a result of this increase, markets can only stay competitive by increasing the share of wind energy. Parallel to the financial benefits, these changes will ensure a decreased CO<sub>2</sub> emission and reduced environmental pollution.

### **4.1.2 Description of wind**

As small Darrieus rotors do not have high rotational speed, the relative speed of the blades always stays well below  $M = 0.3$  ( $\approx 100$  m/s). As a result, air can be modeled as an incompressible medium, as widely done in the literature (e.g., [34]). For reference

density  $\rho = 1.225 \text{ kg/m}^3$  was chosen according to DIN EN 61400 [71], and the viscosity was chosen correspondingly as  $\mu = 1.7894 \cdot 10^{-5} \text{ kg m s}^{-1}$ .

Wind speed has spatial and temporal distribution as well, meaning, that it changes depending on the time and location. Wind speed can be classified e.g., using the Beaufort scale [152, 168], which specifies the wind speed from scale 0 (calm, 0-1 m/s) to scale 12 (hurricane, u=33- m/s). Wind speed has not only a horizontal, but a vertical distribution as well, i.e., wind speed near the ground does not exhibit a constant velocity due to the shear stress generated by the ground. This layer next to the ground is also called Atmospheric Boundary Layer (ABL), and the description of this layer is a separate scientific field [123]. Unfortunately, the precise description of the ABL is only possible through the full modeling of the surroundings of the wind turbine together with all natural and urban structures [40, 41].

For an optimization such an approach would be impossible due to the large computational requirements. In the engineering practice the ABL is replaced by simplified wind profiles, which are described by the different standards [71] and are presented in the followings. Exponential wind profile or wind profile power law [71] defines wind speed as:

$$u(z) = u_{ref} \cdot (z/z_{ref})^\alpha, \quad (4.1)$$

where  $z_{ref}$  is the reference height,  $u_{ref}$  is the reference wind speed (speed at the reference height) and  $\alpha$  is an empirical constant (usually 1/7 is a good approximation; DIN EN 61400 uses 0.2 [71]). Another widely used function is the logarithmic wind profile, defined as [9]:

$$u(z) = u_{ref} \cdot \ln \left( \frac{z/z_0}{z_{ref}/z_0} \right), \quad (4.2)$$

where  $z_0$  is the roughness length, depending on the surface of the actual location. Values for  $z_0$  are defined in tables (e.g.,  $z_0$  is for low crops 0.1, for forest and suburban 1, for offshore 0.0002 [225]).

However, when relying on 2D Computational Fluid Dynamics simulations, constant wind profile has to be assumed. If the rotor is positioned in an appropriately large distance from the ground, the simplification is valid, see Fig. 4.1. In this figure if the rotor is placed at 7 m height, the agreement will be very good.

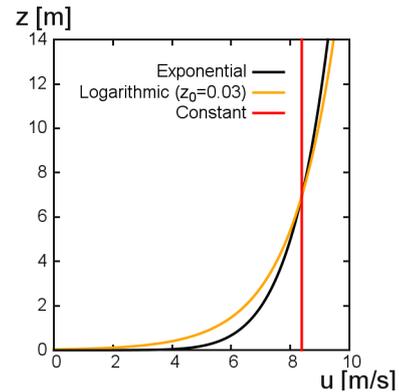


Figure 4.1: Comparison of exponential (black), logarithmic (orange) and constant (red) wind profiles ( $u_{ref} = 8.4 \text{ m/s}$ ,  $z_{ref} = 7 \text{ m}$ )

### 4.1.3 Classification of wind turbines

Without claim of completeness, the most important groups of wind turbines will be presented in the followings. Besides these groups, several other types of wind turbines exist, but these turbines are not well-spread or have no technical relevance.

#### 4.1.3.1 Axis of rotation

According to the position of the rotation axis, wind turbines can be classified into two groups:

- Horizontal Axis Wind Turbines (HAWTs): In these turbines the rotation axis is parallel to the wind direction, i.e., horizontal. As a result, these turbines have to be rotated always into the direction of the wind. These are the most common and most widely spread wind turbines (see Fig. 4.2).
- Vertical Axis Wind Turbines (VAWTs): In these turbines the rotation axis is perpendicular to the wind direction, i.e., vertical, see Fig. 4.3. Due to the vertical direction, the operation of these rotors is independent from wind direction, i.e., they are omnidirectional. The current project considers only VAWTs.



Figure 4.2: Horizontal Axis Wind Turbine (Photo courtesy of Warren Rohner (C), <https://www.flickr.com/photos/warrenski>)

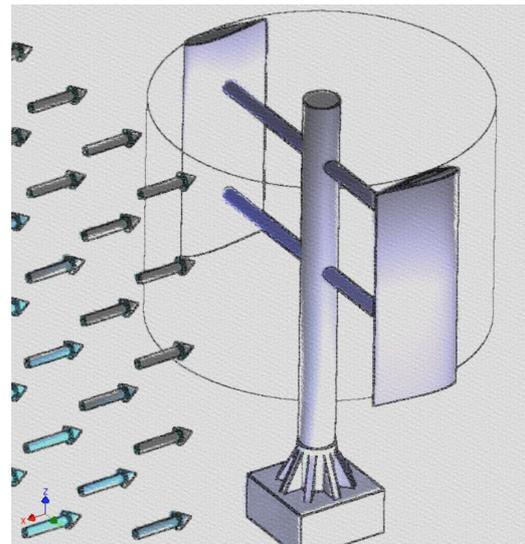


Figure 4.3: Schematical two-bladed H-Darrieus rotor

#### 4.1.3.2 Principle of operation

According to forces used for energy production, wind turbines can be classified into two groups:

- Drag-driven wind turbines: Drag-driven turbines use the aerodynamic drag to produce torque. These turbines usually have slower angular velocities, are small in

size and have small efficiencies. The most well known such turbine is the Savonius rotor [50], although horizontal drag-driven turbines exist as well.

- Lift-driven wind turbines: Lift-driven turbines use the aerodynamic lift to produce torque. These turbines usually have higher angular velocities. The two most common variants are the well known HAWTs (see Fig. 4.2), and the Darrieus rotors, see Fig. 4.4.

#### 4.1.4 Darrieus wind turbines

The original patent of the Darrieus rotor from Georges Jean Marie Darrieus dates back to 1927, but its very complex aerodynamics are still not completely understood. Darrieus wind turbines can be further classified into different groups based on the shape of their blades. The first turbines had the so-called eggbeater or Troposkien shape (see Fig. 4.4 (a)), due to mechanical considerations. Another possibility is the so-called H-Darrieus rotor, which is very popular due to its simple (thus cheap) blades (see Fig. 4.4 (b)). As Darrieus rotors tend to have very unsteady torque (i.e., torque ripple), the blades are sometimes twisted to reduce this effect (see Fig. 4.4 (c)).

The Darrieus rotor is a vertical-axis lift-type rotor. The flow physics is inherently unsteady due to the varying angle of attack, which leads to a higher complexity compared to standard HAWTs. In particular, dynamic stall is of major importance but is still not completely understood [3].

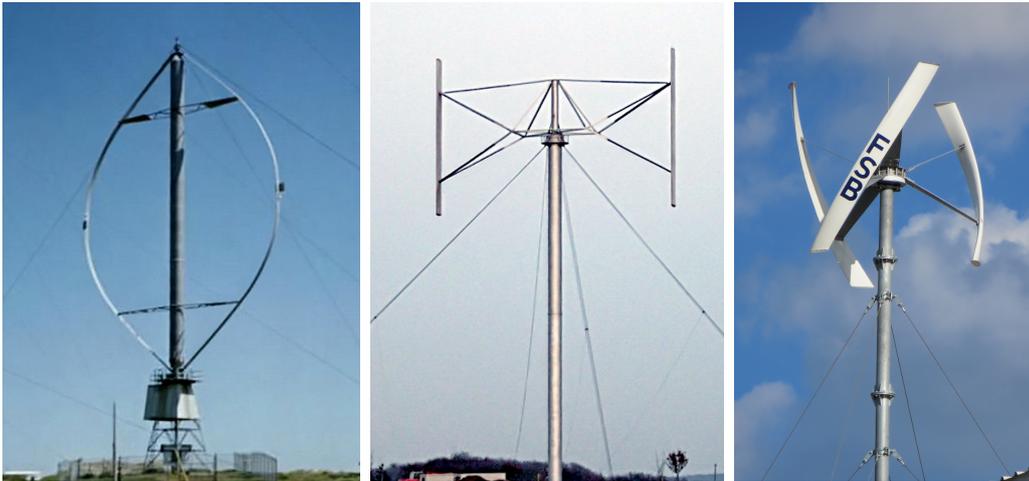


Figure 4.4: Darrieus rotors with (a) Troposkien (b) H-blades (Photo courtesy of Stahlkocher(C), <https://commons.wikimedia.org/wiki/File:H-Darrieus-Rotor.jpg>) (a) Twisted blades (Photo courtesy of Branko Radovanović (C), [https://commons.wikimedia.org/wiki/File:Wind\\_turbine\\_FSB\\_Zagreb\\_20130513\\_0424.JPG](https://commons.wikimedia.org/wiki/File:Wind_turbine_FSB_Zagreb_20130513_0424.JPG))

##### 4.1.4.1 Advantages and disadvantages

“Most installed turbines are Horizontal Axis Wind Turbines, but Vertical Axis Wind Turbines (VAWT), and especially the H-Darrieus wind turbines are very interesting as well, especially for small-scale applications, due to their low cost and easy installation at

almost any location, and due to more favorable interaction effects at low inter-turbine distances. As a yaw system is absent and since all electrical and mechanical components are close to the ground, it has a relatively simple design [155], which decreases installation and operation costs dramatically. Darrieus turbines show relatively low sound emissions and have increased performance in skew [85] as shown, e.g., by Bianchini et al. [21]. Furthermore, they are omni-directional and are thus better suited for areas with strongly inhomogeneous topology, like in urban and peri-urban environments [170]. Last but not least, Darrieus turbines can withstand larger windspeeds due to their aerodynamic behavior” [54].

## 4.1.5 Physics of the H-Darrieus wind turbine

### 4.1.5.1 Notation

As the current chapter considers only H-Darrieus wind turbines, in the followings the term "Darrieus" rotor will be used as short for "H-bladed Darrieus rotors". H-Darrieus wind turbines consist of 2, 3 or more straight blades, installed vertically around a rotating shaft (see Fig. 4.3). These blades use aerodynamic airfoil profiles in order to generate aerodynamic lift throughout the rotation.

In order to support the blades mechanically, the blades are mounted with struts to the rotating shaft. These struts have to be aerodynamic as well to minimize the parasitic drag of the struts. The struts can be installed horizontally (to minimize length) or with an angle (V-shaped). Except the blade tips and cross-sections with the struts, the rotor has quasi uniform cross-sections. As a result, in CFD H-Darrieus rotors are frequently simulated only in 2D, modeling a single cross-section of the rotor. The notations used in the current study are presented in Fig. 4.5. In the current project  $\phi = 0^\circ$  means, when the blade starts moving against the wind (leeward positions),  $90^\circ$  denotes the upwind position,  $180^\circ$  denotes the windward position,  $270^\circ$  denotes the downwind position.

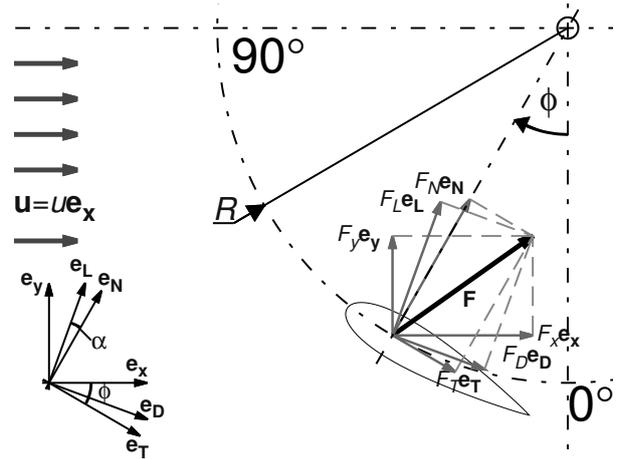


Figure 4.5: Schematic representation of the conventions used for the H-Darrieus rotor in the current study

### 4.1.5.2 Most important quantities

**Tip-speed-ratio** (or TSR; denoted by  $\lambda$ ) is the most important non-dimensional number used to describe the operation of wind turbines. The tip-speed-ratio is defined as the ratio of the highest velocity of the rotating blade (thus, the speed of the tip of the blades) compared to the wind speed ( $u$ ):

$$TSR = \lambda = \frac{v_{tip}}{u} = \frac{R\omega}{u}, \quad (4.3)$$

where  $R$  is the radius of the rotor,  $u$  is the wind speed and  $\omega$  is the angular velocity of the rotor. In case of the H-type Darrieus rotor the cross-section does not vary with height, i.e., the rotor has the same speed for each cross-section.

**The total performance of the wind** flowing through the swept area of the rotor is defined as:

$$P_{total} = \frac{1}{2} \rho A u^3, \quad (4.4)$$

where  $\rho$  is the density of air and  $A$  is the projected area of the rotor.

**The chord Reynolds number** (Re) describes the flow around the blade in a non-dimensional manner. As the relative velocity ( $W$ ) experienced by the blade in the rotating non-inertial coordinate system is not constant, it can be approximated using the rotational velocity:

$$\text{Re} = \frac{c v_{blade}}{\nu} = \frac{\rho c W}{\mu} \rightarrow \text{Re}_{mean} \approx \frac{c R \omega}{\nu}. \quad (4.5)$$

where  $c$  is the camber or chord length.

**Solidity** ( $\sigma$ ) is the ratio of the total rotor planform area compared to the total swept area [176]. Moreover, for Darrieus rotors the definition is slightly modified: the ratio of the perimeter occupied by blades compared to the whole perimeter multiplied by a constant. Unfortunately, no convention was reached for this constant yet and several different definitions exist, regarding the constant. In the current work the following definition will be used, if not stated otherwise:

$$\sigma = \frac{Nc}{2R}, \quad (4.6)$$

where  $N$  is the number of the blades. In several sources  $R$  or  $2R\pi$  is used for the denominator.

**Angle-of-attack** (AOA or  $\alpha$ ) changes throughout the rotation of the blades, resulting in a highly transient flow. The exact and precise evaluation of this quantity is not possible in CFD (as instead of a single local velocity a velocity field exist). Hence, a theoretical approximation is used:

$$\alpha_{th.} \equiv AOA_{th.} = \text{atan} \left( \frac{\sin(\phi)}{\lambda + \cos(\phi)} \right) - \beta, \quad (4.7)$$

where  $\phi$  is the angular position and  $\beta$  is the preset pitch angle.

The theoretical angle of attack is presented in Figure 4.6 for different tip-speed-ratios. As one can see, the angle of attack changes fast and far beyond static stall angles ( $12 - 13^\circ$  [147]). As a result deep stall and dynamic stall will be the dominating physical effects in such wind turbines.

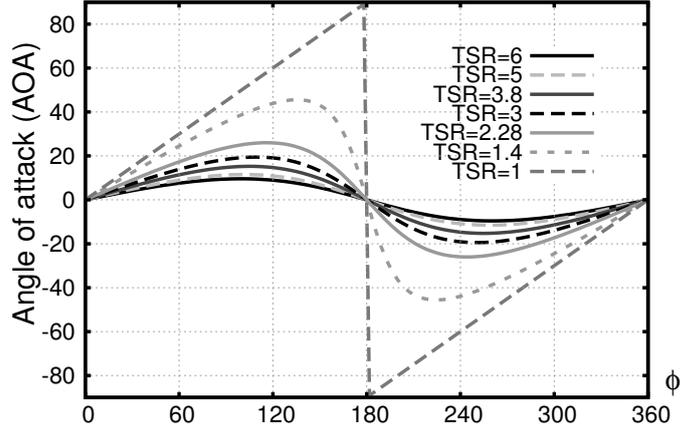


Figure 4.6: Theoretical angle of attack for different tip speed ratios

**Torque coefficient**, or moment coefficient ( $C_m$  or  $C_T$  or  $C_Q$  in the literature) of a wind turbine expresses the torque exerted on the rotor shaft in a non-dimensional manner. The coefficient is expressed as:

$$C_Q(\phi) = \frac{T(\phi)}{\frac{1}{2}\rho u^2 AL} = \frac{T(\phi)}{\frac{1}{2}\rho u^2 2R^2 H} \quad (4.8)$$

where  $T$  is the instantaneous torque on the shaft.

**Power coefficient** ( $C_p$ ) is the indicator of efficiency. It is the ratio of the power generated by the wind turbine compared to the available energy in the wind:

$$C_p = \frac{P_{mech}}{P_{total}} = \frac{\bar{T}\omega}{\frac{1}{2}\rho u^3 A} = \frac{\bar{T}\omega}{\frac{1}{2}\rho u^3 2RH} = -\lambda \cdot \bar{C}_Q, \quad (4.9)$$

where  $\bar{T}$  is the (average) torque generated by the rotor (average, because a torque ripple is present [226]). This value shows a theoretical upper limit ( $C_p = 1$  would mean, that the wind flowing through the rotor would stop moving, which is a violation of the continuity equation), which is known as Betz limit (59.3%). In reality, even modern HAWTs usually cannot reach more than  $C_p = 45 - 50\%$  [19, 77]. Please note, that the Betz limit is only valid, if the assumptions of the model hold (i.e., it is not valid anymore for rotors with guide vanes or in wind channels, as the streamlines cannot expand anymore). The negative sign applies due to the rotation direction used in the current CFD model: clockwise, i.e., negative rotation direction.

**Characteristic curve** is the  $C_p(\lambda)$  function. Although it is often regarded as a non-dimensional indicator of the performance, the same tip-speed ratio can correspond to different local chord Reynolds numbers. The dependence of the power coefficient on the Reynolds number can be experimentally measured [98, 195]. Thus, characteristic curves corresponding to constant wind speed and constant angular velocity are not the same!

As a result, for the complete description of the operation of a wind turbine one has to evaluate the  $C_p = C_p(\omega, u)$  function completely.

The optimal tip-speed ratio depends on the airfoil shape and on the solidity. Lower solidity rotors operate at higher tip-speed ratios, while higher solidity rotors typically have low optimal tip-speed ratios, as discussed also in [156], see Fig. 4.7. This dependence is the result of the changing Reynolds number.

As the camber length increases (thus the chord Reynolds number), the stall mechanism will be delayed, as the boundary layer flow becomes more resistant to separation [103]. Although the solidity can be decreased also with the number of blades (thus resulting in higher optimal tip speed ratio), this results in higher torque ripple and thus in mechanical problems [103].

The loss of performance at lower tip-speed ratios originates from the stall mechanism (when the angle-of-attack, AOA of the blade increases above  $12 - 13^\circ$  [147]). For higher tip speed ratios the loss of efficiency is the result of secondary losses: strut losses, wingtip losses, etc. These losses could only be properly computed with detailed 3D computations.

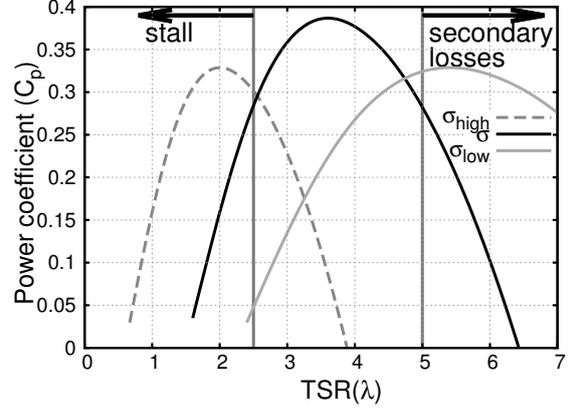


Figure 4.7: Effect of solidity on characteristic curves

**Global force coefficients** ( $C_x$  and  $C_y$ ) express the streamwise and perpendicular forces exerted ( $F_x$  and  $F_y$ ) on a rotor in a non-dimensional form:

$$C_x = \frac{F_x}{\frac{1}{2}\rho u^2 A} = \frac{F_x}{\frac{1}{2}\rho u^2 2RH}, \quad C_y = \frac{F_y}{\frac{1}{2}\rho u^2 A} = \frac{F_y}{\frac{1}{2}\rho u^2 2RH} \quad (4.10)$$

**Normal, tangential force coefficients** for the blades can be reconstructed using an approximation based on the theoretical angle-of-attack. Based on the definition of global force coefficients:

$$F_x = C_x \frac{1}{2}\rho 2RH u^2, \quad F_y = C_y \frac{1}{2}\rho 2RH u^2 \quad (4.11)$$

and on the definition of the blade force coefficients:

$$C_N = \frac{F_N}{\frac{1}{2}\rho c W^2}, \quad C_T = \frac{F_T}{\frac{1}{2}\rho c W^2}, \quad (4.12)$$

using simple geometrical considerations one gets:

$$F_N = F_x \sin(\omega t - \beta) + F_y \cos(\omega t - \beta) \quad (4.13)$$

$$F_T = -F_x \cos(\omega t - \beta) + F_y \sin(\omega t - \beta) \quad (4.14)$$

thus,

$$C_N = \frac{2RH u^2}{c W^2} (C_x \sin(\omega t - \beta) + C_y \cos(\omega t - \beta)), \quad (4.15)$$

$$C_T = \frac{2RHu^2}{cW^2}(-C_x \cos(\omega t - \beta) + C_y \sin(\omega t - \beta)), \quad (4.16)$$

where the relative speed is

$$W^2 = (R\omega + V_a \cos(\omega t))^2 + (V_a \sin(\omega t))^2, \quad (4.17)$$

where  $V_a$  is the induction speed inside the rotor. When not assuming an induction factor, one can use simply  $V_a \approx u$  as approximation. From these values one can estimate the drag and lift coefficient for the blades as well:

$$C_D = -C_T \cos(\alpha) + C_N \sin(\alpha), \quad (4.18)$$

$$C_L = C_T \sin(\alpha) + C_N \cos(\alpha). \quad (4.19)$$

Thus, the  $C_D(\phi)$ ,  $C_L(\phi)$  and  $C_D(\alpha)$ ,  $C_L(\alpha)$  curves can be estimated for further analysis.

**Strut losses** will be frequently considered in what follows. Thus, the most widespread analytical evaluation of strut losses will be presented here. The easiest analytical method was developed by Ion Paraschivou [176]. The approximation assumes, that tip-speed-ratios are large ( $\lambda \gg 1$ ), the drag coefficient is constant along the strut ( $C_{D0} = const.$ ), struts have a constant cross-section and the radius of the shaft is negligible compared to the rotor radius. With these assumptions the forces acting on the single strut-elements can be estimated as

$$dF_s = \frac{1}{2} \rho c_s C_{D0} W^2 dR \quad (W \approx R\omega) \quad (4.20)$$

The total torque acting on the shaft is given as:

$$T = N_b N_s \int_{R=R_{min}}^{R_{max}} dF_s R dR = \frac{1}{2} N_b N_s \rho c_s C_{D0} \omega^2 \int_{R=R_{min}}^{R_{max}} R^3 dR \quad (4.21)$$

Due to  $R_{min} \ll R_{max}$  assumption the equation can be simplified to:

$$T = \frac{1}{8} N_b N_s \rho c_s C_{D0} R_{max}^4 \omega^2 \rightarrow P_{loss} = \frac{1}{8} N_b N_s \rho c_s C_{D0} R_{max}^4 \omega^3 \quad (4.22)$$

Thus one gets finally:

$$\begin{aligned} P_{loss} &= \frac{1}{8} N_b N_s \rho c_s C_{D0} R_{max}^4 \omega^3 = \frac{1}{8} N_b N_s \rho c_s C_{D0} R_{max}^4 \left( \frac{\lambda u}{R_{max}} \right)^3 \\ &= \frac{1}{2} \rho A u^3 \frac{1}{4A} N_b N_s c_s C_{D0} R_{max} \lambda^3 = \frac{P_{total}}{4A} N_b N_s c_s C_{D0} R_{max} \lambda^3 \end{aligned} \quad (4.23)$$

As one can see, strut losses are cubic in nature ( $\sim \lambda^3$ ). A more advanced model can be built by assuming, that only the tangential component of the velocity has an influence on the strut, but not assuming  $\lambda \gg 1$  and  $C_D = const.$ :

$$P_{loss} = \frac{1}{2} N_b N_s c_s \rho \frac{\omega}{2 \cdot \pi} \int_{\phi=0}^{2\pi} \int_{R=R_{min}}^{R_{max}} C_D(\text{Re}, \phi) W(\phi, R)^2 dR d\phi \quad (4.24)$$

This method was first published by Goude et al. [104]. For this, numerical integration is needed.

## 4.2 Numerical validation of the model

The present section is based on:

DARÓCZY, L., JANIGA, G., PETRASCH, K., WEBNER, M., AND THÉVENIN, D. Comparative analysis of turbulence models for the aerodynamic simulation of H-Darrieus rotors. *Energy* 90, 1 (2015), 680–690 [54].

*As already pointed out, numerical validation is of crucial importance for CFD based optimizations. To present this, the first questions that must be answered in the present section concern mesh independency, turbulence modeling and boundary conditions, in order to ensure sufficient precision. Thus, a systematic numerical analysis will be performed in order*

- 1. to identify the necessary mesh resolution for the different turbulence models*
- 2. to choose a model suitable for optimization*
- 3. to compare the characteristic curves obtained with the different models for four different configurations.*

### 4.2.1 Performance evaluation of Darrieus wind turbines

Early models for the evaluation of wind turbines were based on the Glauert actuator disk theory [99]. Starting from the Single Streamtube Model [211], the Multiple Streamtube Model [208] and the Double Multiple Streamtube Model [177] were progressively developed. Following the development of vortex-based models, the next natural step was to rely on CFD simulations. There are three main families of turbulent CFD computations, namely the Direct Numerical Simulations (DNS), the Large Eddy Simulations (LES), and the simplified approach relying on Reynolds-Averaged Navier Stokes equations (RANS, or URANS for Unsteady RANS). DNS for Darrieus rotors is still infeasible, since resolving all scales at this Reynolds number is too expensive. LES has been employed only in a few studies (e.g., [84]), but due to the very high computational costs compared to RANS, LES is clearly incompatible with any optimization process and can only be used to investigate specific aerodynamic aspects. As a result, the present manuscript considers only URANS simulations.

### 4.2.2 CFD evaluation of Darrieus wind turbines in the scientific literature

Recently, a few articles dealt with the analysis and optimization of Darrieus turbines. Using CFD techniques, Mohamed [155] investigated different airfoils for performance improvement of an H-Darrieus rotor. Bedon et al. [16] used a BE-M model for the optimization of a Darrieus rotor, while Castelli et al. [35] proposed a new performance prediction model for Darrieus turbines. Gosselin et al. analyzed the flow around an H-Darrieus rotor while varying solidity, pitch angle or number of blades [103].

Due to the varying angle of attack the physics of these turbines is inherently complex, e.g., dynamic stall is still not completely understood [3]. As a result, the turbulent flow field is extremely complex and there are still contradictory statements in the literature concerning the most appropriate turbulence model. For instance, Mukinovic et al. [161]

have used Spalart-Allmaras and  $k-\omega$  SST model, while Lanzafame et al. used  $k-\omega$  SST and Transitional SST model in 2D CFD simulations to evaluate the performance of H-Darrieus rotors [135], and concluded that Transitional model is the best. The same model was applied in the mesh independency analysis for the rotor of Bravo et al. [28] by Almohammadi et al. [3]. In contrast, Gosselin et al. have tested Spalart-Allmaras (with modified strain-based formulation),  $k-\omega$  SST with low Reynolds corrections and Transition SST and concluded that the  $k-\omega$  SST model was the most appropriate [103]. The  $k-\omega$  SST model was also applied by Jericho et al. for a novel wind turbine concept [121] or by Campobasso et al. for HAWTs with harmonic balance method [30]. Wang et al. [221] found, that the  $k-\omega$  SST model provides appropriate agreement with experiments for the simulation of low-Reynolds flow around pitching blades in stall.

Ferreira et al. analyzed a single bladed H-Darrieus rotor with NACA0015 blade and compared the vortex structures to PIV measurements [84]. They have used 3305 nodes on the blade, giving totally 1.6 million cells. They have tested laminar, Spalart-Allmaras,  $k-\epsilon$ , DES and LES model, using  $\Delta\phi = 1/2; 1/4; 1/8; 1/16^\circ$  angular time-steps. They concluded, that URANS models proved to be inaccurate in modeling the large eddies, and the LES performed even worse compared to the DES. However, the obtained performances were not compared.

Castelli et al. found that  $k-\omega$  SST model is the most appropriate in 3D, while  $k-\epsilon$  Realizable model is more accurate in 2D [34]. Maître et al. have also applied  $k-\omega$  SST model for Darrieus water turbines [147].

### 4.2.3 Limitations in the comparison of numerical and experimental investigations

Although in some articles a nearly perfect agreement between CFD and experiment is discussed [135, 155], such an observation should not indeed be expected. Even if CFD can be used to resolve the flow field around the rotating turbine, important details will always be missing in the CFD. As a whole, the resulting differences can be quite large. In many studies the predicted efficiency exceeds the measured one by up to 75-95% [3].

Minor issues encompass the uncertainty of the measurement devices in the experiment, geometrical tolerances during manufacturing, insufficient temporal or spatial resolution in the simulation, inadequate level of convergence, rounding errors, missing fluctuations in the rotation speed (present in the reality, but not in the simulation), unknown turbulence conditions (cannot be exactly measured), lacking three-dimensional instabilities and three-dimensional expansion. For 2D simulations, the lack of end-plate friction and the missing exponential wind profile additionally lead to mistakes as well.

More significant terms are the generator losses (if only the power output is measured) and bearing losses. However, these terms can be in principle measured and taken into account.

The major source of discrepancy is connected to the strut losses and strut-junction wake interaction. Maître et al. came to the conclusion, that the main losses do not come from the arms friction, but from the tip and arm-blade junction vortices [147]. In contrast with this observation, the 3D CFD study of Castelli et al. indicated that, for small aspect ratio rotors the arm friction losses can result in more than 20% loss of performance [34]. Fortunately, the strut losses can be analytically estimated with a good accuracy [104, 176],

as discussed previously.

The largest and most difficult terms are the wing-tip vortex losses.

Airfoils generate lift through the pressure difference on the two sides of the airfoil. Moreover, the blade has an end and due to the pressure difference airflow is introduced from the higher pressure side to the lower pressure side along the end of the blade. As a result, a wingtip vortex is generated (see Fig. 4.8). In wind turbines with small aspect ratios these losses can be large (up to 25% [34]), while at high aspect ratio these effects are small. As shown in the study of Gosselin et al., the last 20-30% of the blade height are less effective [103]. Amet also came to a similar conclusion, with 22% loss of performance due to the tip vortices and blade/arm junctions [5], while Qin et al. reported 40% losses compared to 2D simulations [182].



Figure 4.8: Wingtip vortex generated by an airplane; Source: NASA; Photo ID: EL-1996-00130; Alternate ID: L90-5919

Finally, the uncertainty of the turbulence models can also result in large errors and will be considered more specifically in what follows. As a result, at least 5-25% differences in  $C_p$  are expected. This error could only be eliminated (or at least reduced) using 3D CFD computations.

One final difference originates from the way performance is evaluated experimentally and analytically. In the reality performance is measured and averaged for many revolutions, thus averaging out the effects of vibrations, vortex shedding and turbulence gusts. In a simulation, computing such a large number of revolutions is very time consuming and thus not feasible. Some sources estimate [20...90] for the minimal necessary number of revolutions [11].

## 4.2.4 Computational method

### 4.2.4.1 Analyzed rotor

This section describes the complete workflow of the simulation. For the current study a geometry was chosen, which is currently under development for an industrial project. The rotor has three blades with  $R = 1.5$  m radius, each blade has a chord of  $c = 160$  mm, thus solidity is  $\sigma = 0.32$  and the optimal tip-speed ratio is  $3 < \lambda < 4$ . The exact airfoil profile cannot be disclosed due to confidentiality requirements from the industrial partners of the project.

### 4.2.4.2 Automated workflow

Systematic studies are always subject to human errors. Thus, in order to eliminate this problem, the complete workflow has been fully automated. The analysis and the evaluation of the different configurations have been carried out using OPAL++.

For the present study a module has been created for OPAL++ in C++. In this module (approx. 5500 lines of code) the complete CFD workflow was parameterized, from the geometry creation (number of blades, geometry of the blade, radius of the rotor, blunt/rounded/sharp edge, etc.), through the meshing (mesh size, size of domain, etc.) to the CFD setup (choice of CFD software, choice of solver, turbulence model, boundary conditions, etc.). By keeping all settings as parameters, extensive studies could be performed in a fast and efficient manner, without the risk of errors. Additionally, the module collected the most important results (number of cells in the mesh, orthogonal quality of the mesh, instantaneous maximal and average  $y^+$  values for all blades, thickness of the blade, performance coefficient, instantaneous lift-, drag- and torque coefficients, maximal and average lift-, drag- and torque coefficients of the last revolution) in an Excel file, enabling extensive analysis based on the data.

The workflow is presented in Fig. 4.9. When starting an optimization, for each configuration the same OPAL++ script was repeated automatically according to the specified parameters (blue section) from the Genetic Algorithm. If a numeric study was executed, instead of a Genetic Algorithm, directly the configurations listed in an input listfile were processed.

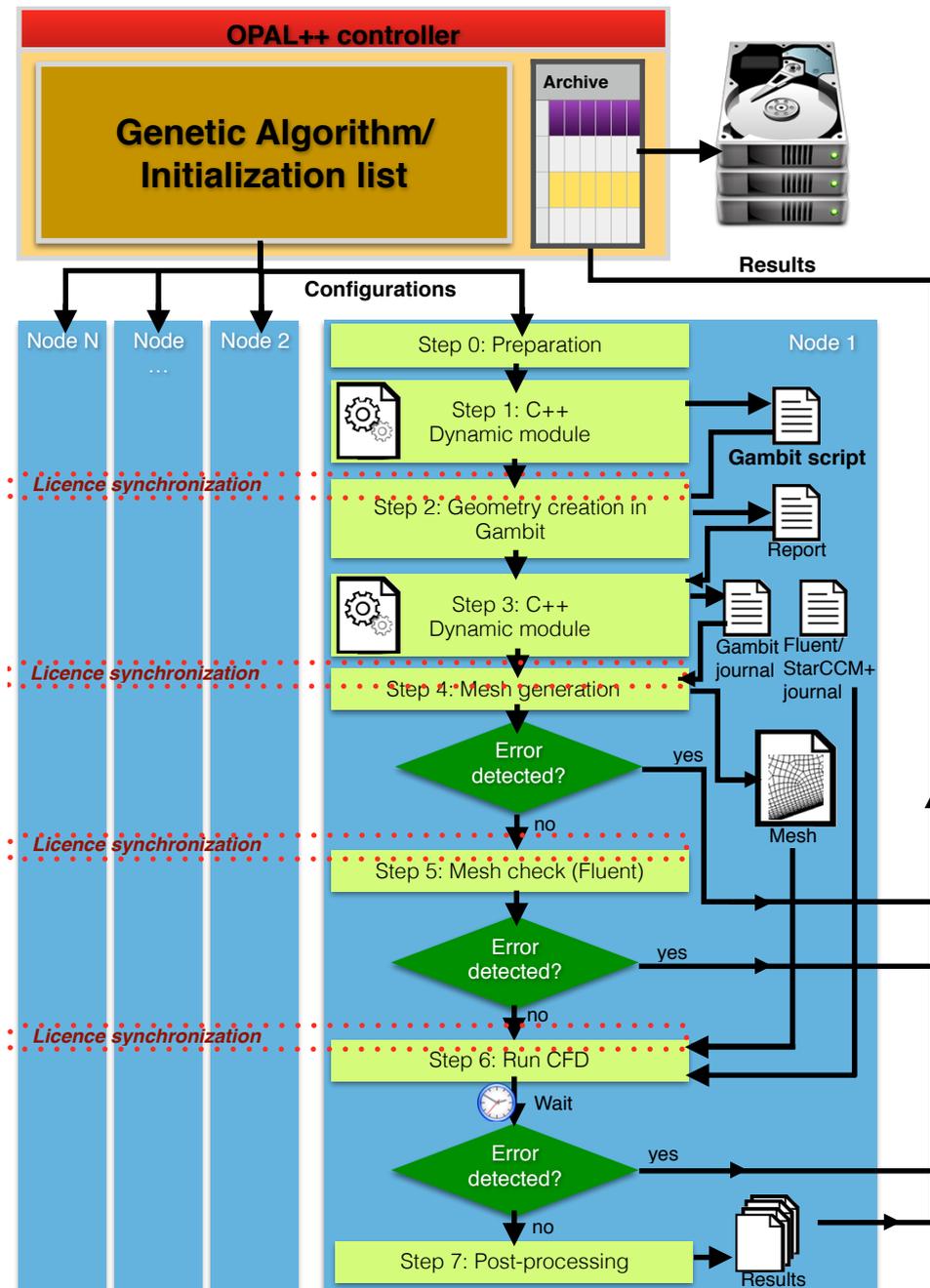


Figure 4.9: Loop of evaluation in the OPAL++ H-Darrieus module

As proprietary software is used for mesh generation and CFD computation, the number of available licenses becomes an issue. To solve this problem, synchronization barriers are defined, which can be entered only by one node at a time. The defined loop of evaluation will be blocked on each worker node in each step, when a license is needed (starred steps in the list below, dashed red box in the figure). A single node will check the license server, and if enough licenses are available, will proceed with the computation. After the node has acquired the license, other nodes will start to check the license server after each other. This blocking (or synchronization) is necessary to avoid race conditions (i.e., only

1 license is available, but all parallel nodes detect available licenses). The following steps were executed:

- (0) Preparation: OPAL++ looks for a free node (OPAL++ has a built-in separate job manager), which receives the actual configuration and prepares the different data files and temporary working directory.
- (1) Preparation of script files: OPAL++ calls the OPAL++ dynamic library, which creates the necessary script files for the meshing software and for the CFD software.
- (2\*) Geometry preparation: The worker calls the meshing software (ANSYS Gambit) to prepare the geometry using the generated script files and to create the report.
- (3) Preparation of script files/2: OPAL++ calls the compiled dynamic library again, which will read the output generated by the meshing software, determines the IDs of the different edges and faces, and prepares the final script for meshing. This is necessary, as during the different subtraction operations the different geometrical entities receive their names in an automatic manner, depending on the geometry of the current configuration.
- (4\*) Meshing: OPAL++ calls the meshing software (ANSYS Gambit) to prepare the mesh using the generated script (journal) files. If any fatal error (failed mesh generation process; distorted cells; etc.) is detected, the loop of evaluation is broken and the individual is considered invalid.
- (5\*) Mesh check: OPAL++ calls ANSYS Fluent/CD-Adapco StarCCM+ to check the quality and validity of the mesh. The mesh size and mesh quality indicators are then read and stored in OPAL++. If a mesh with inadequate quality is detected, the loop of evaluation is broken and the individual is considered invalid.
- (6\*) CFD computation: After the mesh is prepared, the CFD computation is executed using all cores of the actual node. If divergence is detected, the individual becomes invalid.
- (7) Post-processing: Finally, the results saved by the CFD software are post-processed. OPAL++ computes the average and maximal values of the different coefficients in the last revolution. The values are read and stored by OPAL++ and the results (and important files) are given to the master node for archiving.

#### 4.2.4.3 Domain decomposition

As the flow around a Darrieus rotor is highly transient, it has to be resolved not only in space, but in time as well: a moving mesh has to be created with a sliding mesh technique (although there are some groups, which try to simulate the flow in a stationary manner, which is a principally wrong approach [148]). This means, that the domain of the computation has to be decomposed into a stationary domain, which will be used to simulate the free stream flow around the rotor (Fig. 4.10 (a)), and a moving domain, which will rotate with the blades (Fig. 4.10 (b)). Between the two domains an interface is used in order to interpolate the flow quantities. In this study the same circular decomposition was applied as e.g., by Castelli et al. [34]. Another possible decomposition could have

been a ring structure [135, 170]. Preliminary tests have shown no significant differences between both.

The mesh was created with ANSYS Gambit 2.4.6 [88]. Unstructured parts are plotted with a brick pattern in Figs. 4.10, the density of the pattern indicating the mesh resolution. If a shaft is not present in the computation, the inner structured domain can be simply removed and replaced with an unstructured mesh. For the structured parts the Map algorithm is applied, while for the unstructured parts the quadrilateral Pave method is used. In the current study only quadrilateral cells are used (following the recommendation of [3]), as triangular cells are well-known to be dissipative in nature. However, tests have indicated in the present study no large differences between triangular or quadrilateral cells for the mesh-independent resolution.

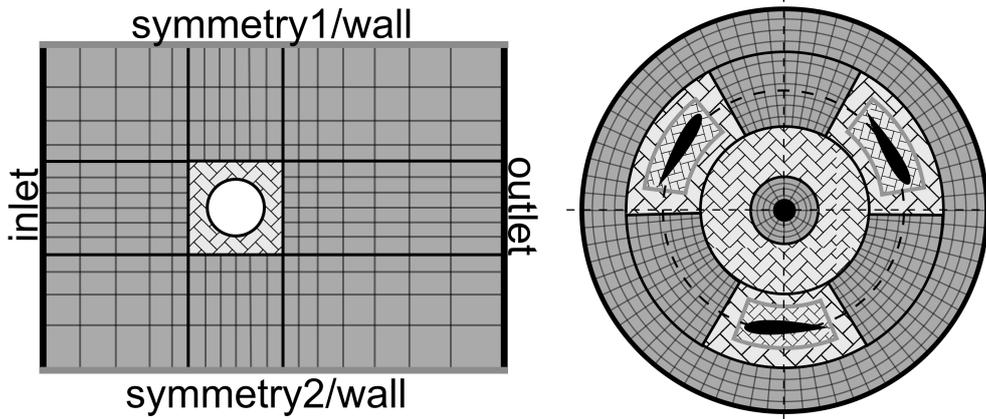


Figure 4.10: (a) Stationary domain of the mesh (b) Rotating domain of the mesh

The employed hybrid structured-unstructured grid (as shown schematically in Fig. 4.10) provides the advantage, that it increases the accuracy near the interface (where interpolation errors can be large) and in the wake of the blades, while being still very flexible, thus enabling automatic mesh generation of different blades. A completely structured approach was tested as well, but tests did not indicate significant differences either [158].

#### 4.2.4.4 Mesh size and mesh independency

In published studies very different mesh resolutions and temporal resolutions have been applied. For a corresponding review please refer to Trivellato and Castelli [214].

The mesh size is very important for the turbulence as well. If the mesh spacing is too coarse, the decay of turbulence will be grossly overestimated [203]. The reference mesh sizes are defined at seven different locations ( $S_1, \dots, S_7$ ): on the sides of the middle rectangle in the stationary domain (90 mm), inside the middle rectangle in the stationary domain (60 mm), on the interface (20 mm), on the wall of the shaft (2.5 mm), on the blade (0.375 mm), in the middle of the rotating domain (13.5 mm) and around the blade in the additional control area (4.8 mm). The C++ module uses additional size functions to ensure a smooth transition between the different sizes. The nodes on the surface of the blade were uniformly distributed, except at the trailing edge, where a refinement was used with an exponential growth factor.

Throughout the studies, a calibration coefficient ( $c_{calib}$ ; with tested values of 1.5, 1.25, 1.0, 0.75, 0.5) was used to compute actual mesh sizes from the reference size ( $S_{act,i} = c_{calib}S_{ref,i}$ ). In case of the boundary layer the refinement is modified to  $y_{act} = y_{ref}\sqrt{c_{calib}}$ . The quality of the mesh was checked and recorded in all cases by the software and reported in the output log file. Orthogonality (as defined by Fluent) was always above 0.7, and maximum skewness angle (as reported by StarCCM+) was always below  $40^\circ$  during the validation process.

For a precise computation the boundary layers have to be resolved up to the laminar sublayer. The number of layers and thickness of the first layer were calibrated by the C++ module so, that the transition between the size of the boundary layer and the normal mesh has to be smooth (see Fig. 4.11) and  $y^+ < 1$  has to be maintained. Then,  $y_{avg}^+ \approx 0.2$ ,  $y_{max}^+ \approx 1.0$  was maintained for all computations (except for the test with the  $k-\epsilon$  Realizable model with Wall function). This strong criterion was applied following the recommendations of Maître et al. [147].

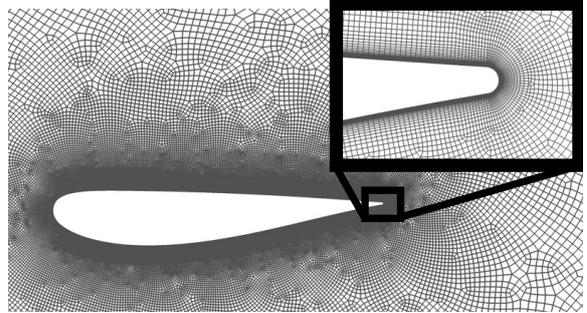


Figure 4.11: Example of the obtained mesh resolution near a random blade geometry

This was ensured by using the Schlichting correlation [192] with the camber of the blade as reference length and  $W = (\lambda + 1)u_{wind}$  as the relative speed.

## 4.2.5 CFD setup

### 4.2.5.1 Material properties & discretization

Flow was chosen, as described earlier, to be incompressible and properties according to DIN EN 61400 [71] ( $\rho = 1.225 \text{ kg/m}^3$ ,  $\mu = 1.7894 \cdot 10^{-5} \text{ Pa s}$ ), as discussed previously.

In the present work a second-order implicit temporal discretization is always applied, together with second-order (central or upwind) derivatives, if not stated otherwise. If for a specific case or model divergence is detected as observed with  $k-\epsilon$  RNG always, the simulation is restarted using first-order discretization and Segregated solver.

### 4.2.5.2 Segregated vs Coupled solution

When solving the fluid dynamics equations, there are two possibilities for the pressure-velocity coupling: the segregated approach (SIMPLE, SIMPLEC or PISO in ANSYS Fluent and Segregated in CD-Adapco StarCCM+), where a predictor-corrector approach is applied, and the Coupled solver, where the momentum and continuity equations are directly coupled. In transient cases the coupled algorithm is necessary for poor mesh quality or large time steps [6]; this solver can dramatically increase the performance even for incompressible flows [125]. According to the CD-Adapco StarCCM+ User Guide [37], the Coupled solver is always recommended when sufficient resources are available due to its increased robustness. Another advantage of the Coupled solver is that the CPU time scales linearly with the number of cells, i.e., the convergence rate will not deteriorate when refining the mesh resolution.

Again, different authors do not agree concerning this topic. The SIMPLE method was applied, e.g., in the study of Lain and Osorio [133], while the PISO method was finally retained by Lanzafame et al. [135]. Instead, Qin et al. [182] or Balduzzi et al. [10] have applied the Coupled solver due to its increased robustness.

For the geometry discussed later for numerical analysis (Operating condition 2,  $k-\omega$  SST model), the PISO and Coupled solvers were compared using ANSYS Fluent 14.0. As a convergence criterion, the number of inner iterations per time step was fixed to 15, 24, 50 and 100 for both solvers.

Figure 4.12 shows the torque coefficient for blade 1, for the 15<sup>th</sup> revolution. One can see, that the results with the Coupled solver stay identical when enforcing more than 24 iterations, while for PISO the agreement is only achieved for more than 100 iterations. Thus, the Coupled solver was always employed with 24 inner-iterations in what follows. This is the faster solution. These results support the conclusions of Maître et al. [147], where at least 75 inner iterations were necessary for convergence with the SIMPLE method.

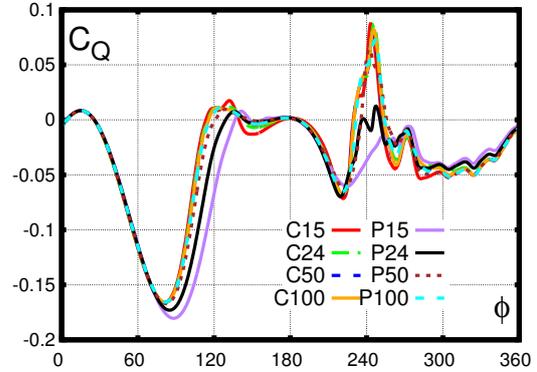


Figure 4.12: Torque coefficient with different solvers and number of inner iterations in ANSYS Fluent (C denotes Coupled Solver, P denotes PISO solver, the number denotes the number of inner iterations)

This proved to be a very strict convergence criterion. The different residuals, as reported by Fluent, generally dropped to  $10^{-5} - 10^{-10}$  depending on turbulence model and angular position. All residuals always were reduced at least to  $10^{-3}$  in the numerical validation.

#### 4.2.5.3 Boundary conditions

The four sides of the computational domain are defined as velocity inlet, pressure outlet and symmetry boundaries. Proper inlet conditions are clearly the most important ones. Unfortunately, the specification of inlet turbulence properties is a very difficult question, particularly so for wind turbines. Different sources propose incompatible values and procedures. Although one could use values derived from corresponding experimental measurements, turbulent variations are usually post-processed from measured wind fluctuations during 10-60 min [71, 188]. As noted by Spalart and Rumsey [203], in the atmospheric boundary layer the typical length scales are around 100 m and the eddy viscosity can reach  $50 \text{ m}^2/\text{s}$  on windy days (corresponding to a turbulent viscosity ratio  $\text{TVR} = 3.3 \cdot 10^6$ ). However, the boundary layer will couple only with length scales in the order of 1 cm [203] and, due to the large rotation speed, only a part of the turbulent spectra will really interact with the blades in the CFD [129].

In the first numerical studies different values were tested. After confirming that small values are almost equivalent to each other, final values of  $I = 0.1\%$  and  $\text{TVR} = 10$

were retained for all further configurations, in agreement with recommendations from the literature [134, 135]. Such low intensities are not rare in wind tunnels. E.g., the wind tunnel of TU Delft has  $I = 0.015\%$  at 10 m/s [85].

#### 4.2.5.4 Temporal discretization - Size of time step

To determine the time step size, the Courant-Friedrichs-Lewy (CFL) number can be used. In CFD,  $CFL = 1$  is usually recommended for stability. However, with an implicit time integration scheme, much larger values are allowed. In the current study the time-step was thus defined as

$$\Delta t = \frac{2\pi}{\omega \Delta \phi}, \quad (4.25)$$

where  $\Delta \phi = 2^{-k}$  ( $k = 0, 1, 2$ ) is the angle of rotation for each tested time step. This criterion is strongly correlated with the CFL number. Then, the post-processing becomes very simple, as a single revolution will be  $360 \cdot 2^k$  time steps. Of course, the independence from the temporal resolution still has to be analyzed.

This strategy offers another advantage: if the nodes on the interface are equidistant, and equal in number to  $n \cdot 360 \cdot 2^k$ , they will overlap at each time step. Instead of interpolation, the values can be copied directly between the moving and stationary domain (conformal match). This is automatically detected by some solvers (e.g., CD-Adapco StarCCM+). In the experimental validations 720 points were always used. The study of Gosselin et al. proposed similar values, i.e., 1000 time-steps [103], while Castelli et al. [34] or Maître et al. [147] used 360 time-steps per revolutions.

#### 4.2.5.5 Temporal discretization - Number of revolutions

Reaching a quasi periodic solution can require a large number of revolutions, leading to very long computations. Moreover, when starting the computation, the flow field is not realistic around the blade. To solve this problem, a two-level approach was applied. In the first part the flow field around the blade does not necessarily have to be computed precisely, thus  $\Delta \phi = 4 \dots 6^\circ$  is used to speed up the initialization of the flow field. Later, in the second part  $\Delta \phi = 1^\circ$ ,  $\Delta \phi = 1/2^\circ$ ,  $\Delta \phi = 1/4^\circ$  or an even smaller value is employed.

In the current study at least 10 (coarse)+5 (detailed) revolutions were always used (for slower wind speed or faster rotation, the number of revolutions were increased). Although this seems to be large at first look, similar recommendations are found in the literature (e.g., Gosselin et al. used 20 revolutions [103], Maître et al. considered 11 revolutions [147]). To evaluate the average performance, only the last revolution is post-processed.

Although in the present study such a case was never encountered, theoretically, it is not guaranteed, that a quasi-periodic solution exists: the vortex shedding frequency of the blades, shaft and rotor might be very different, resulting in a non-periodic torque signal. E.g., the vortex shedding frequency of the rotor depends on the Strouhal-number; if no lock-in condition is achieved [13], the torque signal will not be periodic.

## 4.2.6 Systematic numerical analysis

In order to identify the optimal setup and mesh for the CFD computations, an extensive numerical analysis was performed, involving many different tests. In particular, the domain size, boundary layer resolution, mesh resolution, temporal resolution, different turbulence specifications and different tip-speed-ratios were analyzed. Unless otherwise specified, all comparisons have been done with ANSYS Fluent 14.0.

In order to enable a short and quick description of the turbulence models, the following notations are used in all Tables: T0= $k-\epsilon$  Realizable with Enhanced Wall Treatment; T0\*= $k-\epsilon$  Realizable with Standard Wall Function ( $y^+ > 30$ ); T2= $k-\epsilon$  RNG (Re-Normalization Group), T3=Spalart-Allmaras, T4=SAS, T5= $k-\omega$  SST, T6= $k-\omega$ ; T7=Transitional  $k-k_l-\omega$ ; T8=Transitional SST in the standard formulation implemented in ANSYS Fluent [6]. For selected cases the simulations were performed using StarCCM+ as well to allow for comparison: all models starting with S denote the equivalent model in StarCCM+ with All- $y^+$  Wall Treatment. As different turbulence intensities and tip speed ratios result in different stall mechanisms and angle of attacks, four different cases were tested.

### 4.2.6.1 Impact of domain size

In order to analyze the effect of the domain size on the performance coefficient, four different sizes were tested (D1-D4). Table 4.1 presents the performance coefficients for the corresponding domain size ( $L_{inlet}$ ,  $L_{outlet}$  and  $L_{sym}$  represent the distance from the central rectangular area to the inlet, outlet and symmetry boundary, respectively;  $L_{rect} = 8$  was kept constant in all cases). The computations were done with  $\lambda = 2.28$ ,  $I = 0.1\%$ ,  $TVR = 10$  and  $k-\omega$  SST model. All coefficients agreed and the hysteresis curves did not reveal significant differences either. Choosing the first variant might be inappropriate for some cases, as it could be on the boundary of the acceptable domain. Hence, D2 was retained for all further simulations.

Table 4.1: Comparison of performance coefficients for four different domain sizes ( $\lambda = 2.28$ ,  $k-\omega$  SST model)

	D1	D2	D3	D4
$L_{inlet}[m]$	15	20	30	45
$L_{outlet}[m]$	30	40	60	90
$L_{sym}[m]$	18.75	25	37.5	56.25
$C_p[\%]$	23.58	23.46	23.46	23.75

### 4.2.6.2 Impact of boundary layer resolution

In order to analyze the effect of the boundary layer resolution, the thickness of the first layer and the growth rate were varied. A growth rate of 1.1, 1.175 and 1.225 was tested with different thicknesses resulting in  $y_{max}^+ \in [0.23 - 3.65]$  and  $y_{ave}^+ \in [0.04 - 0.63]$  for the last revolution. With  $\lambda = 2.28$ ,  $I = 0.1\%$ ,  $TVR = 10$  and  $k-\omega$  SST model the resulting coefficients were all within the range of  $C_p \in [23.41 - 23.70]$ , and the hysteresis curves indicated only very small differences as well. This confirms that when using the default values (growth rate  $\approx 1.2$ ,  $y^+ < 1$ ), the models will deliver suitable results in all cases.

#### 4.2.6.3 Effects of distance of the interface

In order to analyze the effect of the distance between the interface and blades, four different distances were tested (see Table 4.3). The computations were done with  $\lambda = 2.28$ ,  $I = 0.1\%$ ,  $TVR = 10$  and  $k-\omega$  SST model. Results have agreed almost completely for all cases, but choosing the first variant might be inappropriate for some cases, as it could be on the boundary of the acceptable domain. Hence, I2 was retained for all further simulations.

Table 4.2: Comparison of performance coefficients for four interface locations ( $\lambda = 2.28$ ,  $k-\omega$  SST model)

	I1	I2	I3	I4
$d_{int}[m]$	0.5	0.7	1.1	1.5
$L_{rect}[m]$	8	8	8.8	9.6
$C_p[\%]$	23.46	23.48	23.52	23.99

#### 4.2.6.4 Effects of the presence of the shaft

In a next test the effect of the shaft was analyzed using the  $k-\omega$  SST model with  $I = 0.25\%$  and  $L = 0.15$  for  $\lambda = 2.28$ . The power coefficients show similar tendencies with and without the shaft ( $M4 : 21.48\%$ ;  $M3 : 23.63\%$ ;  $M2 : 23.52\%$ ;  $M1 : 24.77\%$ , for value with shaft see Table 4.4). The torque coefficient ( $C_Q(\phi)$ ) without the shaft was smoother due to the lack of the Kármán vortex street - blade interaction, but this did not help improving mesh independency due to dynamic stall and wake-blade interaction.

We nevertheless decided to keep the shaft in the CFD computations.

#### 4.2.6.5 Operating condition 1 - Low turbulence intensity, $\lambda = 3.8$

In the first numerical study the turbulence models were analyzed for  $\lambda = 3.8$  and low turbulence,  $I = 0.25\%$ ,  $L = 0.15$  m. In this case the angle of attack was rather low resulting in only weak stall effects and in smooth torque curves (see Fig. 4.13). Analysis of the vortex structure revealed a wide filament, as also found by Maître et al. [147]. For all cases five different mesh configurations were analyzed, with  $c_{calib} = 1.5, 1.25, 1.0, 0.75, 0.5$ . For the time-steps  $\Delta\phi = 1^\circ$  was applied. When changing the time-step to  $\Delta\phi = 0.5^\circ$ , the total change in the performance coefficient with the  $k-\epsilon$  Realizable model was only  $\Delta C_p = 0.02\%$ . Investigating the performance and torque not only in an integral manner, but also in a time-dependent sense, revealed no significant changes either. This is in agreement with the study of Maître et al. [147], where  $\Delta\phi = 1^\circ$  was found to be sufficient for appropriate accuracy.

As strong dynamic effects are missing, in Fluent solver  $CFL = 200$  (default value) was used. In StarCCM+ lower values (between 25 and 35) were applied to ensure stability. The results are presented in Table 4.3, together with the number of cells and with the number of nodes on the interface (int.).

Table 4.3: Comparison of performance coefficients for the last revolution for Operating Condition 1 with 8 different turbulence models  $Ti$ , 5 mesh resolutions  $Mi$  and 2 CFD codes.

	M5	M4	M3	M2	M1
$cell$	201k	296k	368k	588k	1089k
$int.$	360	720	720	1080	1440
$C_p^{T0}$	40.04	40.14	40.19	40.22	40.23
$C_p^{T0*}$	36.86	37.84	38.93	39.72	40.50
$C_p^{T2}$	35.66	35.24	35.18	34.89	34.38
$C_p^{T3}$	34.49	34.45	34.50	34.47	34.43
$C_p^{T4}$	42.44	42.53	42.66	42.83	43.00
$C_p^{T5}$	42.26	42.36	42.48	42.52	42.54
$C_p^{T7}$	49.40	49.50	49.66	49.78	49.87
$C_p^{T8}$	45.47	45.70	46.01	46.51	46.42
$C_p^{ST0}$	40.89	40.99	41.09	41.10	41.44
$C_p^{ST3}$	35.17	35.27	35.34	35.36	35.41
$C_p^{ST5}$	42.50	42.56	42.82	42.95	43.38

The standard  $k-\omega$  model was tested as well, but resulted in unphysical results. This is no surprise, since the  $k-\omega$  standard model is intended only for fully turbulent flows. In StarCCM+ the AKN (Abe-Kondoh-Nagano)  $k-\epsilon$  Realizable model resulted also in unacceptable oscillations. Model T0\* ( $k-\epsilon$  Realizable with Wall Function) did not provide appropriate results either (in this case the boundary layer was calibrated for  $y^+ > 30$ ). The reason for this failure was that maintaining  $y^+ > 30$  and a cell aspect ratio of 1 was impossible for the finer mesh resolutions.

All models managed to reach  $\Delta C_p = |C_p^{M1} - C_p^{M3}| < 1\%$ , and most of them provided even stricter results. Thus, the grid corresponding to M3 provides an appropriate resolution. The agreement between StarCCM+ and Fluent is satisfactory as well, although StarCCM+ always overestimated the power coefficient by approximately 1% compared to Fluent. However, the hysteresis curves did not reveal significant differences between StarCCM+ and Fluent. The  $\phi - C_Q(\phi)$  curves are presented for blade '1' in Figure 4.13 for all turbulence models computed in Fluent.

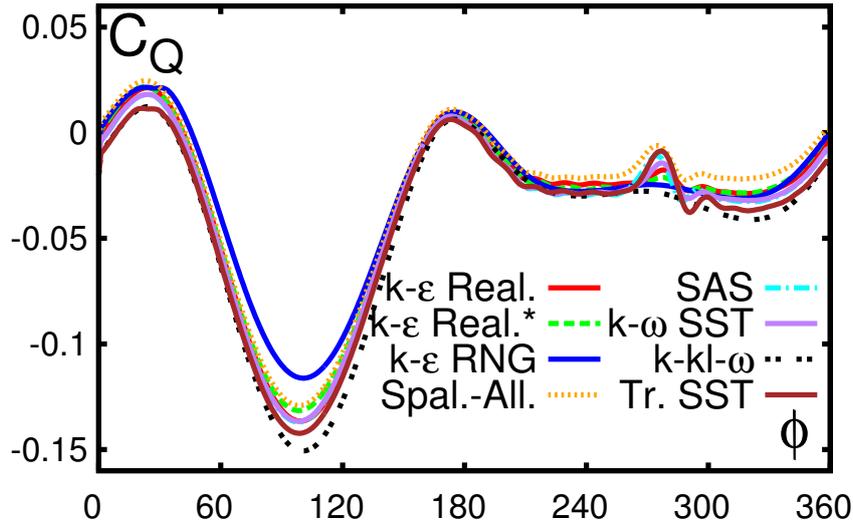


Figure 4.13: Comparison of torque coefficients for the last revolution for Operating Condition 1 with 8 different turbulence models for a single blade

#### 4.2.6.6 Operating condition 2 - Low turbulence intensity, $\lambda = 2.28$

For Operating Condition 2 the turbulence was still kept low ( $I = 0.25\%$ ,  $L = 0.15$  m), but the tip-speed-ratio was decreased to increase the angle of attack, and thus stall effects. Due to their inappropriate performance in the first test, following approaches have not been considered further:  $k-\epsilon$  Realizable (T0\*) with Wall Function;  $k-\omega$  (T6), and Transitional  $k-k_l-\omega$  (due to instability problems with the tested Fluent version when running it in parallel on Linux). The results are shown in Table 4.4.

For the temporal resolution  $\Delta\phi = 0.5^\circ$  was used. Coarser resolutions did not result in temporal independency (the hysteresis curves indicated noticeable differences). Using  $\Delta\phi = 0.125^\circ$  resulted in only 1.5% difference for  $k-\epsilon$  Realizable model, 0.8% for Spalart-Allmaras and 2.2% for  $k-\omega$  SST in the performance coefficient. As the maximal allowable error was chosen as  $\approx 2\%$ ,  $\Delta\phi = 0.5^\circ$  was deemed acceptable. One has to keep in mind, that this excellent temporal independency has been achieved due to the robustness of the Coupled solver, even at large CFL numbers.

Table 4.4: Comparison of performance coefficients for the last revolution for Operating Condition 2 with 6 different turbulence models  $Ti$ , 5 mesh resolutions  $Mi$  and 2 CFD codes

	M5	M4	M3	M2	M1
<i>cell</i>	226k	266k	370k	547k	1091k
<i>int.</i>	540	540	720	900	1440
$C_p^{T0}$	35.49	36.30	37.66	38.0	38.65
$C_p^{T2}$	33.54	33.89	34.77	35.18	35.83
$C_p^{T3}$	25.27	25.54	26.21	26.27	26.59
$C_p^{T4}$	21.01	20.80	22.60	22.05	19.90
$C_p^{T5}$	21.62	21.77	23.63	23.95	24.05
$C_p^{T8}$	16.53	17.69	18.44	20.29	21.4
$C_p^{ST5}$	19.83	19.68	23.35	22.93	24.65

All models were able to satisfy  $\Delta C_p = |C_p^{M1} - C_p^{M3}| < 2\%$ , except SAS and Transitional SST, where M2 resolution is necessary (in case of T4 (SAS) model the SIMPLE solver was used due to convergence problems).

The phase-angle dependent torque values are presented in Figure 4.14. As one can see, the predictions are very different. The largest differences are seen at  $\phi = 60 - 160^\circ$  and  $\phi = 200 - 300^\circ$ , which correspond to the appearance of dynamic stall and wake interaction. The  $k-\omega$  based models result in higher fluctuations, while  $k-\epsilon$  based models stay relatively smooth.

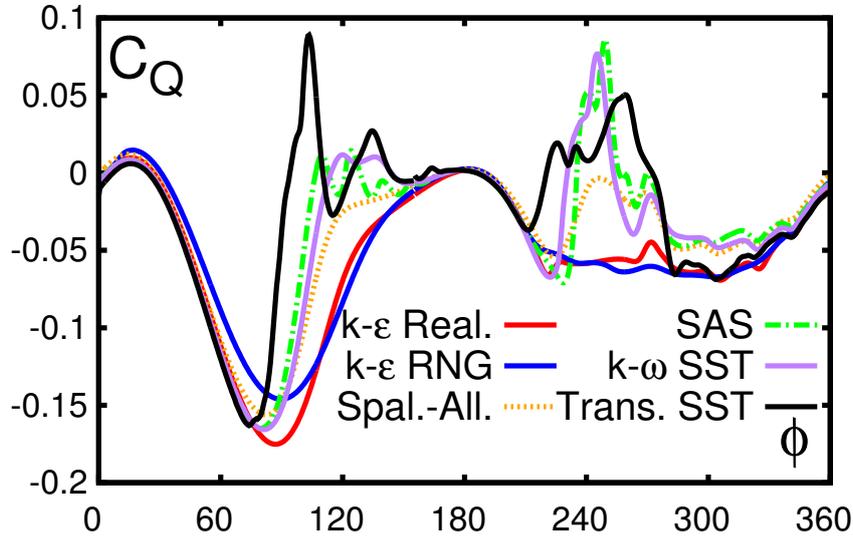


Figure 4.14: Comparison of torque coefficients for the last revolution for Operating Condition 2 for 6 different turbulence models for a single blade

**Comparison of vortex shedding mechanism** In order to analyze the source of the difference for the turbulence models, the vortex shedding structure as obtained with the

different turbulence models are compared in Figures 4.15-4.21. The Figures show the results as obtained with the  $k-\epsilon$  Realizable, Spalart-Allmaras and  $k-\omega$  SST models, from left to right, respectively. The shape of the blade was censored due to confidentiality. As one can see, the three models show a high similarity, despite the very different performance predictions.

In the first part of the revolution ( $\phi = 0 - 90^\circ$ ) there is only a wide filament generated behind the blade. A vortex is formed at the trailing edge at  $\phi = 90^\circ$  by  $k-\omega$  SST, at  $\phi = 100^\circ$  by Spalart-Allmaras and at  $\phi = 110^\circ$  by  $k-\epsilon$  Realizable. The leading edge vortex starts to roll up and is shed from the blade at  $\phi = 120^\circ$  by Spalart-Allmaras, at  $\phi = 130^\circ$  by  $k-\epsilon$  Realizable and at  $\phi = 110^\circ$  by  $k-\omega$  SST.

Reattachment happens approximately at  $\phi = 170^\circ$  by  $k-\omega$  SST, at  $\phi = 170^\circ$  at Spalart-Allmaras and at  $\phi = 180^\circ$  by  $k-\epsilon$  Realizable. The upstream wake propagates down to the blade between  $\phi = 210 - 220^\circ$ . At this point, the difference between the predictions of the different models is significant, which explains the different prediction of the oscillations encountered in the simulations. Around  $\phi = 230^\circ$  a vortex is shed from the leading edge (by all models) and reattachment does not happen until  $\phi = 290^\circ$ . After this point, the vortex structure is a wide filament again.

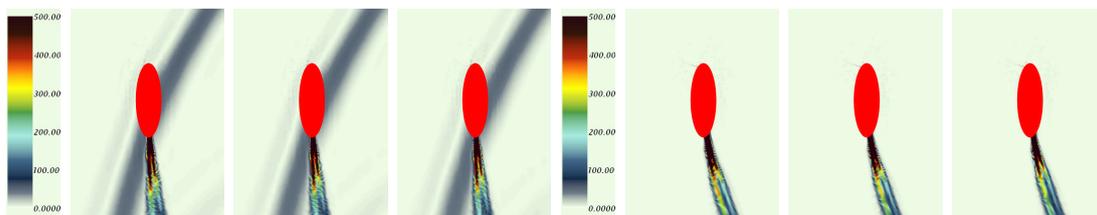


Figure 4.15: Vortex shedding ( $\phi = 0^\circ$  and  $\phi = 60^\circ$ ; StarCCM+)

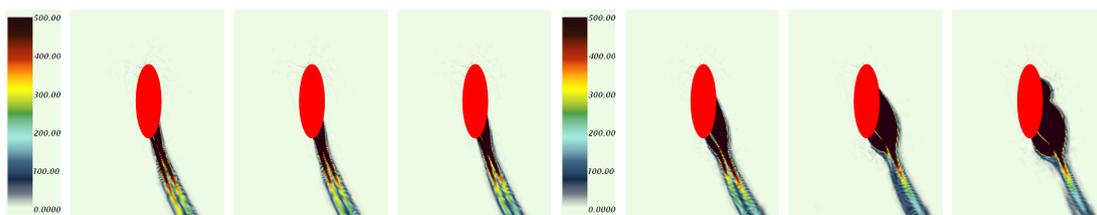


Figure 4.16: Vortex shedding ( $\phi = 90^\circ$  and  $\phi = 110^\circ$ ; StarCCM+)

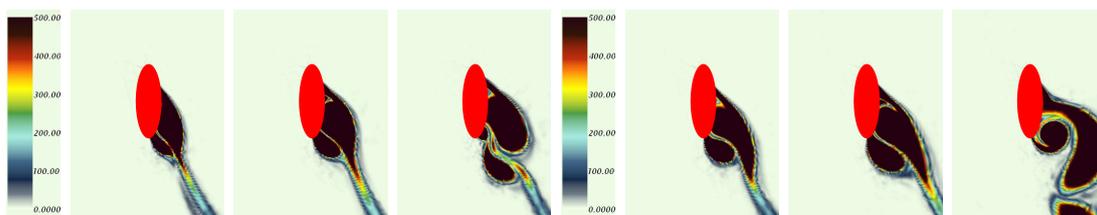


Figure 4.17: Vortex shedding ( $\phi = 120^\circ$  and  $\phi = 130^\circ$ ; StarCCM+)

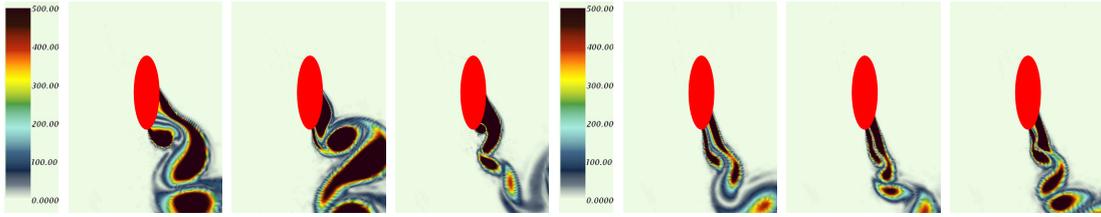


Figure 4.18: Vortex shedding ( $\phi = 150^\circ$  and  $\phi = 170^\circ$ ; StarCCM+)

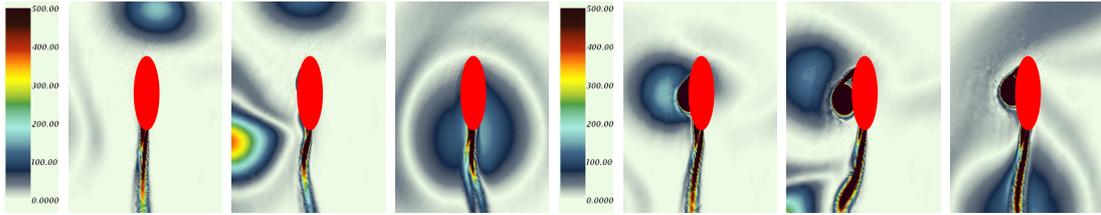


Figure 4.19: Vortex shedding ( $\phi = 220^\circ$  and  $\phi = 230^\circ$ ; StarCCM+)

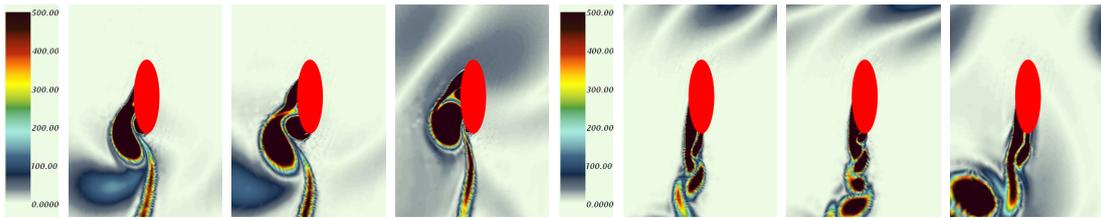


Figure 4.20: Vortex shedding ( $\phi = 240^\circ$  and  $\phi = 260^\circ$ ; StarCCM+)

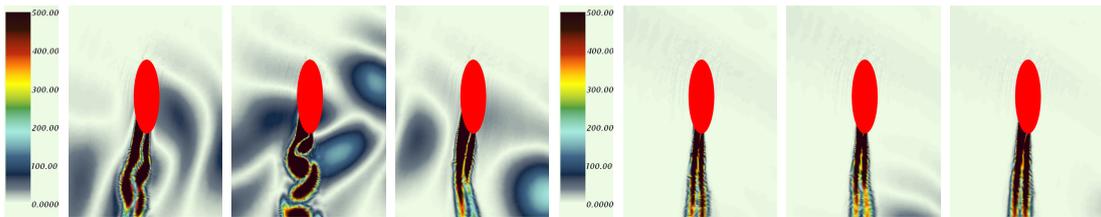


Figure 4.21: Vortex shedding ( $\phi = 280^\circ$  and  $\phi = 300^\circ$ ; StarCCM+)

**Impact of turbulence intensity ( $I$ )** When comparing different, but low turbulence specifications, the results were found to be very similar. The comparison was done for the  $k$ - $\omega$  SST model, which provided the most fluctuating but still converged solution for the previous, stall conditions. When varying  $I$ , no significant changes were detected in the time-dependent curves. The performance coefficients were very similar as well, for instance 23.50% for  $I = 0.1\%$ ,  $\text{TVR} = 10$ , compared to 23.63% for  $I = 0.25\%$ ,  $L = 0.15$  m. As the former set of values has been widely used in publications [134, 135], it was retained for the experimental validations.

#### 4.2.6.7 Operating condition 3 - High turbulence intensity, $\lambda = 2.28$

In the third case the same tip-speed-ratio is retained as in the previous test but with increased turbulence ( $I = 7\%$ ,  $L = 0.3$  m), based on the NRC wind tunnel [218]. The results are shown in Table 4.5.

Table 4.5: Comparison of performance coefficients for the last revolution for Operating Condition 3 with different turbulence models

	M5	M4	M3	M2	M1
<i>cell</i>	224k	264k	367k	544k	1085k
<i>int.</i>	540	540	720	900	1440
$C_p^{T0}$	35.29	35.75	36.67	36.83	37.16
$C_p^{T2}$	26.85	28.83	29.28	29.47	-
$C_p^{T3}$	16.65	16.69	16.92	16.96	-
$C_p^{T4}$	-	-	-	-	-
$C_p^{T5}$	21.00	21.24	23.02	23.17	24.15
$C_p^{T8}$	-	-	-	-	-
$C_p^{ST0}$	34.19	34.85	36.08	36.44	37.23
$C_p^{ST3}$	15.76	15.96	16.39	16.52	17.00
$C_p^{ST5}$	19.03	19.98	22.00	22.44	24.27

As one can see, the performance coefficients drop down due to the increased turbulence, especially for the Spalart-Allmaras model. Furthermore, the different torque curves were very smooth in this case. This is no surprise; it was also noted by Wang et al. [221], that with increasing turbulence intensity the oscillations in the stall phenomenon are effectively stabilized (damped).

SAS and Transitional models failed to converge, Spalart-Allmaras and  $k-\epsilon$  RNG models have experienced significantly decreased performances, while other models delivered similar results. The models were again able to reach for all cases  $\Delta C_p = C_p^{M1} - C_p^{M3} < 2\%$ , thus  $M3$  was deemed as the appropriate resolution, once again. The agreement between StarCCM+ and Fluent is again satisfactory.

#### 4.2.6.8 Operating condition 4 - Very high turbulence intensity, $\lambda = 2.28$

When choosing an extremely large turbulence intensity, as specified by the DIN EN 61400 standard ( $> 20\%$ ), most turbulence models start to have problems. The  $k-\epsilon$  Realizable, RNG and Spalart-Allmaras lead to significantly reduced performance ( $\approx 27\%$ ,  $22\%$  and  $8\%$ , respectively), while the curves become very smooth and the strong oscillations and peaks in the hysteresis curves completely disappear. SAS and Transitional SST models fail to converge and  $k-\omega$  SST model provides completely unrealistic values. The same is true for StarCCM+ as well.

#### 4.2.6.9 Final recommendations

Considering all the previous results,  $M3$  resolution was finally chosen, with 367k cells. This corresponds to at least 720 nodes along the interface and at least 720 nodes along the blades. A coarser resolution is not sufficient.

Concerning turbulence intensity,  $I = 0.1\%$  and turbulent viscosity ratio  $TVR = 10$  is recommended. At least 10+5 revolutions with  $\Delta\phi = 0.5^\circ$  are needed for proper time-resolution. Resolving 720 time steps per revolutions together with at least 720 points on the interface ensures  $CFL \leq 1$  for stability.

## 4.2.7 Experimental validation process

Although most standard turbulence models managed to reach mesh independency at M3-level, they delivered significantly different results concerning performance. In what follows, four experimental measurement campaigns are compared with the results of 2D CFD computations in order to help identifying the most suitable model.

For all comparisons the previous recommendations have been taken into account. Although the integral  $C_p$  values are not able to capture all aspects of the flow (e.g., the time-dependent distribution of forces and torque in a single rotation), PIV or LDV measurements are rarely available in the literature and difficult to obtain. Using such data would make it impossible to compare a wide range of rotors and Reynolds numbers. Due to the lack of data, in the followings only performance coefficients will be compared.

The main purpose of the validation process was to see, if the differences between experimental measurements and 2D simulations show a systematic tendency, then opening the door for quantitative predictions based on 2D simulations. At the end this was indeed observed, but only for the  $k-\epsilon$  Realizable model and the  $k-\omega$  SST model. The differences compared to experiments were found to be constant, or cubic with the tip-speed-ratio, respectively. A constant offset was already observed for the  $k-\epsilon$  Realizable model in other works as well [34]. The cubic offset found for the  $k-\omega$  SST model is probably associated to the strut losses, though further studies are needed to prove this point.

In all cases the original experimental measurements and the characteristic curves of the 2D simulations are first shown unaltered, along with the modified curves taking into account the offset (constant for  $k-\epsilon$  Realizable, cubic for  $k-\omega$  SST) obtained from a best fit.

### 4.2.7.1 Validation 1

The first validation is based on the investigated geometry of the industrial project. For this, the rotor was placed on a platform moving at constant speed (similar, e.g., to Gorle et al., who analyzed the flow around a Darrieus water turbine in a towing tank [102]) and an anemometer was fixed at midheight in front of the rotor. The wind speed was recorded every second.

In a first step the turbulence spectrum was analyzed by defining the turbulence intensity as a function of the time window ( $N\Delta t$ , i.e., the duration used to compute the turbulence intensities). If the velocity with  $i$  samples after the arbitrary  $t_0$  time is denoted as  $u(t_0, i) = u(t_0 + i\Delta t)$ , the turbulence intensity for a time window of  $N$  steps ( $t_{window} = N\Delta t$ ) after  $t_0$  can be defined as:

$$I(t_0, N) = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N \left( u(t_0, i) - \frac{1}{N} \sum_{i=1}^N u(t_0, i) \right)^2}}{\frac{1}{N} \sum_{i=1}^N u(t_0, i)}. \quad (4.26)$$

Then, if the total time domain of the measurement is denoted with  $\mathbb{T}$ , the average turbulence intensity with a time window of  $t_{window} = N\Delta t$  can be defined as:

$$I(N\Delta t) = \overline{I(t_0, N)} \quad \forall t_0 \in \mathbb{T} \quad (4.27)$$

The  $I(N\Delta t)$  curve was computed and plotted in Fig. 4.22 for all the measurements (denoted as all), for a single measurement (stat.; length of the measurement: 2821 s), where the mobile platform was standing and for a single measurement, where the mobile platform was moving (mov.; length of the measurement: 1169 s). As one can see, the standing and moving spectra are significantly different; in the moving case the vibrations of the platform during movement increase significantly the measured intensities.

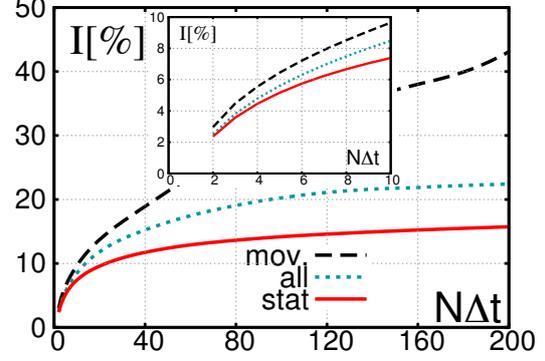


Figure 4.22: Measured turbulence spectra for different time windows

However, although the turbulence is high, the turbines are operating at smaller time scales than the atmospheric turbulence. Thus, only a part of the spectra will interact with the blades, where the frequency of the wind speed oscillations is similar to the frequency of the rotation [129]. This means, that the rotor will feel turbulence and gusts rather as a quasi-steady flow. In the  $N\Delta t < 4$  s range (inlaid plot) the curves collapse with very low turbulence intensities. This explains and justifies the common practice of using low-turbulence intensities in URANS-based simulations of wind turbines, even under real wind conditions.

As measurements themselves are very unsteady concerning wind speed, rotation speed and torque, a filtering method was applied. Only measurement intervals ( $N\Delta t$ ) were accepted, where the root-mean-square of the rotation speed and wind speed stayed below a specified limit compared to the average. Uniformity factor was defined as

$$\Psi_\omega = \frac{\omega_{RSM}}{\omega_{mean}}, \quad \Psi_u = \frac{u_{RSM}}{u_{mean}} \equiv I. \quad (4.28)$$

Afterwards, the corrected efficiency can be determined using the energy balance:

$$\eta_{2D} \approx \frac{E_e - E_s + E_{output} + E_{bearing} + E_{strut}}{E_{wind}}, \quad (4.29)$$

where  $E_s = 0.5I_z\omega(t_0)^2$ ,  $E_e = 0.5I_z\omega(t_0 + N\Delta t)^2$

$$E_{wind} = 0.5\rho A \int_{t=t_0}^{t_0+N\Delta t} u(t)^3 dt, \quad (4.30)$$

$$E_{bearing} \approx M_{bearing} \int_{t=t_0}^{t_0+N\Delta t} \omega(t) dt, \quad (4.31)$$

Finally, all measurement points were formed with  $N = 8$  length (see Eq. 4.30), and these measurement points were filtered to include only points with  $6 < u < 11$  m/s and with uniformity factor  $\Psi_u^{max} = 0.16$ ,  $\Psi_\omega^{max} = 0.06$ .

From these points, tip-speed-ratio intervals were formed ( $\lambda_{min,i} \leq \lambda \leq \lambda_{max,i}$ ), and for each interval a single point was computed by averaging the tip-speed-ratio and the performance for the bin, forming a characteristic curve. The filtered measurements points, together with the obtained characteristic curve are presented in Figure 4.23. The characteristic curve has been already compensated for strut losses (based on an analytical model [176]) and bearing losses.

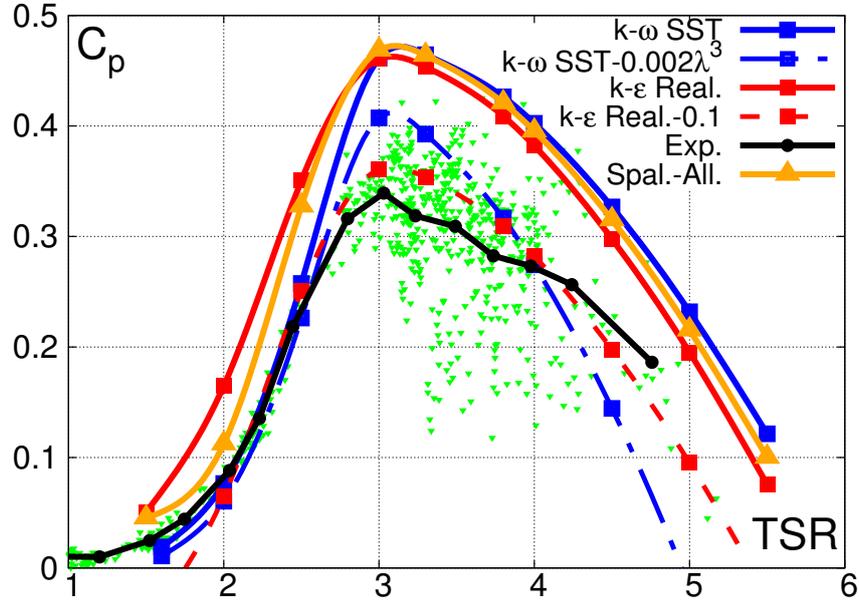


Figure 4.23: Experimental validation (1) based on own measurements

For this experimental comparison the curves were computed using CD-Adapco Star-CCM+, with  $k - \epsilon$  Realizable,  $k - \omega$  SST and Spalart-Allmaras model. As one can see, the overall prediction of the location of the maximal performance is excellent, but the turbulence models exhibit a very interesting tendency. With the  $k - \epsilon$  Realizable model, the numerically evaluated curve shows a constant offset of  $\Delta C_p = 0.1$  compared to the experiment. Alternatively, the  $k - \omega$  SST model exhibits roughly a cubic offset with  $\Delta C_p = 0.002\lambda^3$ . As the curves were compensated for bearing and strut losses, the differences can only originate from junction wake interaction, wing-tip losses and turbulence effects.

#### 4.2.7.2 Validation 2

The second validation from the literature is based on the experimental work of Bravo et al. [28], where a small ( $H = 3$  m,  $D = 2.5$  m), high-solidity ( $N = 3$ ,  $c = 0.4$  m) rotor was tested in the NRC 9 m x 9 m Low Speed Wind Tunnel in Ottawa ( $I < 2\%$  [129]). The rotor used NACA0015 profiles. The mount position was at  $0.5c$  [86] and the trailing edge rounding was 4 mm [129].

Experiments have confirmed, that the characteristic curves of the Cleanfield rotor were almost invariant for  $8 \text{ m/s} \leq u \leq 16 \text{ m/s}$  ( $\omega = const.$ ). The experiments were performed for the  $0.8 < \lambda < 2.2$  domain, the maximum power coefficient achieved was

$C_p = 0.3$  for  $\lambda = 1.6$ . It was further revealed in the work of Fiedler and Tullis [86], that the frictional and strut losses were quite large, 50-500 W between 40-200 rpm. The study has also revealed, that the mounting position (i.e., location of the chord-strut connection) has quite a large effect on the rotor. Mounting at 145 mm instead of 200 mm ( $c = 400$  mm) resulted in a performance loss of almost  $\Delta C_p = 0.086$ .

The performance curve in the wind tunnel and on the rooftop (thus in turbulent urban environment) were very close to each other [129]. Even with 25 % fluctuations the performance dropped only by 10%, showing that the urban environment has very little impact on the performance.

In the simulation the rotor was placed in an open field. The curves were computed using ANSYS Fluent. About 1700 points were used to resolve the blade with  $u = const. = 10$  m/s. The resulting number of cells was around 530,000.

Figure 4.24 shows the characteristic curves with "Exp." as obtained in [28]. The shaft losses were added to this curve as measured in [86] (denoted with Exp. (+shaft)). As the struts were not aerodynamic, the assumption must hold that large cubic losses are present. The results with the different turbulence models are shown as well. In case of the Transitional model the trend is not correct as the optimal tip-speed ratio is shifted to larger values. Interestingly, for the present geometry all other models agree approximately with each other. All models predict the optimal tip-speed ratio almost correctly.

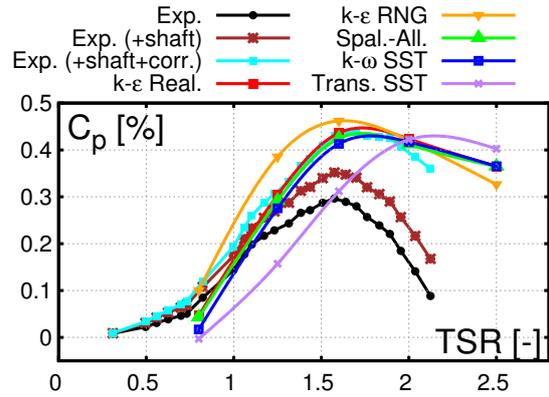


Figure 4.24: Experimental validation (2) based on [28]

When using a cubic correction ("Exp. (+shaft+corr.)",  $\Delta C_p = 0.02\lambda^3$ , computed for the best fit), the agreement is even better. Unfortunately, without knowing the exact geometry it is not possible to determine more accurately the corresponding cubic losses.

### 4.2.7.3 Validation 3

The third validation is based on the experimental work of Kjellin et al. [127]. In this field test a three-bladed Darrieus rotor with 12 kW rated power output ( $u = 12$  m/s),  $H = 5$  m height and  $D = 6$  m diameter was tested and measured for around 350 h. The blades were NACA0021 airfoils, with  $c = 0.25$  m chord length and tapered end to reduce the wingtip losses (the mounting position was not indicated in the original article, but the difference in the pitch angle between  $0.25c$  and  $0.5c$  corresponds only to  $\Delta\beta = 0.6^\circ$ ). The rotor used passive stall regulation with a direct drive. The test was conducted at 48 - 57 rpm. The blades were mounted with two struts at  $17.6^\circ$ , the struts having NACA0025 profile with 280-320 mm chord length. With the tested constant rotational speeds, the optimal performance of  $C_p = 0.29$  was found at  $\lambda = 3.30$ . Comparisons are presented in Fig. 4.25. The computations were carried out using ANSYS Fluent. About 1200 points were used to resolve the blade. A shaft with 200 mm was added to the simulation as well. The total number of cells was around 815,000.

In Figure 4.25, the typical behavior of the turbulence models can be seen again. The Transitional SST model overestimates the performance coefficient and the optimal tip-speed-ratio as well, due to the lack of calibration of the empirical correlation function. The  $k-\epsilon$  Realizable model is able to predict the shape of the characteristic curve and the location of the maximal performance, but not the exact values. Instead, it shows a constant offset ( $\Delta C_p^{corr} = 0.14$ ). The modified curve (see  $k-\epsilon$  Real. corr.) shows a very good agreement. A similar behavior was experienced by Castelli et al. for a different wind turbine [34].

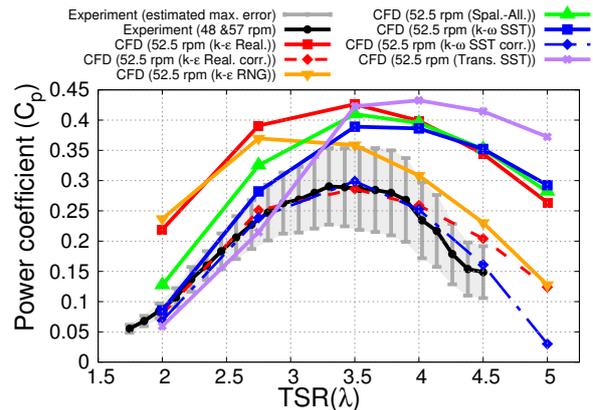


Figure 4.25: Experimental validation (3) based on [127]

For  $k-\epsilon$  RNG neither the values nor the optimal tip-speed-ratio are predicted correctly. In case of the Spalart-Allmaras and  $k-\omega$  SST models the tendency is completely different. At low tip-speed-ratios the prediction is rather accurate; but, at higher  $\lambda$ -values the differences increase. It is interesting to note, that the difference is again cubic in nature ( $\propto \lambda^3$ ). The same tendency can be found, e.g., in [3]. When using a correction factor of  $\Delta C_p^{corr} = -0.0021\lambda^3$ , the resulting curve ( $k-\omega$  SST corr.) shows a very good agreement for the whole domain. However, estimating the strut losses analytically ( $C_{D0} = 0.01288$ ), they can account only for about 30 % of these cubic losses. It remains unclear yet where the further difference finds its explanation.

#### 4.2.7.4 Validation 4

The 4th and last validation was based on the work of Castelli et al. [34]. The authors have numerically and experimentally analyzed a three-bladed Darrieus-rotor with  $D = 1030$  mm diameter and  $c = 85.8$  mm camber length using a NACA0021 airfoil. The rotor had a small aspect ratio, with only  $H = 1456.4$  mm height. For the location of the spoke-blade connection  $0.5c$  was used [35].

The blunt edge (0.38 mm) of the airfoil was disregarded. In the experiment, 9 m/s wind speed was tested in a wind tunnel with 4000x3840 mm cross-section. The blockage was not considered in the simulation. About 840 points were used to resolve the blade, the shaft was disregarded. The total number of cells was around 330,000, ANSYS Fluent was used for the simulations. The results can be seen in Fig. 4.26.

In their study, Castelli et al. have found, that  $k-\omega$  SST model is the most appropriate model in 3D, while  $k-\epsilon$  Realizable model is the most accurate in 2D. One can see, that the present study further confirms this observation. Indeed,  $k-\epsilon$  Realizable shows an acceptable agreement. In spite of an overestimation at lower tip speed ratios, the optimal tip speed ratio is correctly predicted. The overproduction at low  $\lambda$  might be a consequence of the mounting position, since T0\* shows the characteristic curve obtained for  $0.25c$  mounting (as computed with StarCCM+).

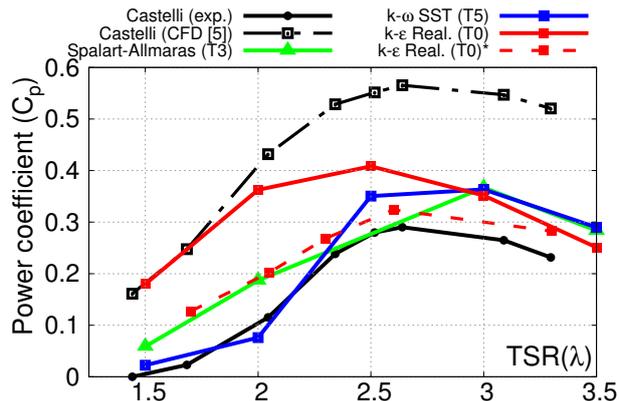


Figure 4.26: Experimental validation (4) based on [34]

In case of the  $k-\omega$  SST model the performance curve is shifted slightly to higher tip-speed-ratios. The underestimation at low TSR values for this model was observed as well in many different studies [103,147], but remains unexplained yet. The agreement between the present computations and the computations of Castelli et al. is very good concerning the location of the maximal value, but the absolute values do not agree, though all models are identical. The origin of this discrepancy could not be explained yet.

## 4.2.8 Conclusions and summary

In the present section, URANS-based 2D computations of H-Darrieus wind turbines were analyzed and compared. Deriving a proper URANS tool is of considerable importance, since such an approach opens the door to performance optimization, as corresponding simulations are relatively fast. For this purpose mesh independency, temporal discretization, the effect of turbulence intensity and the impact of different turbulence models have been checked. After identifying an optimal value for these parameters, an extensive validation campaign was performed by comparison with independent experimental measurements. Experimentally measured characteristic curves were compared with the results of different turbulence models using CFD computations with StarCCM+ and Fluent. The results with the different software showed a very good agreement.

Finally, two turbulence models were identified that lead to the best systematic agreement:

1. The  $k-\epsilon$  Realizable model was able to predict correctly the location of the optimal tip-speed-ratio in all four cases, with an almost constant offset of  $C_p$  compared to the experimental measurements.
2. The  $k-\omega$  SST model with a cubic correction delivers also an excellent prediction for all four cases.

Furthermore, at higher tip-speed-ratios (where losses are significant)  $k-\epsilon$  Realizable and  $k-\omega$  SST models agreed, but at lower tip-speed-ratios (where losses are less dominating), the  $k-\epsilon$  Realizable model showed a significant overestimation, while  $k-\omega$  SST model

indicated agreement. Thus, for following studies the  $k$ - $\omega$  SST model will be recommended (it is able to predict the performance curves at least for one region correctly). The  $k$ - $\epsilon$  Realizable model seems to underestimate stall effects. Additional 3D computations were performed confirming indeed, that strut losses and endplate friction losses are really cubic in nature and significant (results are not presented due to confidentiality).

*In the present section, it was presented through a very detailed example that numerical validation is an essential part of an optimization process, sometimes even more demanding than optimization itself. In the next section, an optimization will be executed based on an exemplary geometry to show the effectiveness of the chosen method.*

## 4.3 Optimization of a H-Darrieus rotor blade

*In the previous section the numerical validation of the CFD setup was presented. This pointed out the importance of the appropriate validation of models. In fact, such validations can require in some cases significantly longer time than the optimization itself. This is the case here as well: the validation took 10-12 months, while the optimization required “only” 2-3 months of run-time.*

*In the followings, the optimization process will be presented with the parameterization and detailed analysis of results.*

### 4.3.1 Introduction to the current analysis

As commercially available rotors already exhibit large performance (large-size HAWTs can reach up to 50%) further improvements are unlikely to be achieved by the usual trial-and-error engineering design approach. Instead, systematic optimization approach is needed. For this reason, the aim of the current study is to optimize the shape of the airfoil to achieve higher performance ( $C_p$ ).

#### 4.3.1.1 Optimization of wind turbines in the scientific literature

There are already a few studies dealing with optimization of wind turbines, but most of these studies lack high-fidelity performance evaluation methods or systematic optimization algorithms. E.g., Fischer et al. [87] used a Blade Element Momentum (BE-M) code instead of CFD for the multi-objective optimization of HAWTs. Although the present study considers only performance, annual energy yield [20], aero-elastic behavior and loading [32, 33], noise [43, 63, 157], optimal layout [92] or robustness [143], cost [179, 215], or structural optimization [25] could also be considered. Simultaneous consideration of several objectives, i.e., real multi-objective optimizations are even scarcer (Göçmen and Özerdem [100] or Slesongsom and Bureerat [199]).

#### 4.3.1.2 Aim of the analysis

The ultimate goal of the present study is to perform a blade shape optimization for VAWT rotors. Besides the obvious question (what is the optimal geometry), additional questions to be answered are:

- Are there more efficient geometries than the classical airfoils used in many studies?
- Are the optimal blade geometries cambered and/or pitched?
- Are the results robust regarding the turbulence modeling (i.e., does the blade maintain its efficiency, when performance is evaluated with a different turbulence model)?
- How could significant improvements be achieved?

## 4.3.2 Setup of CFD Model

### 4.3.2.1 Exemplary rotor geometry

In the present study an *exemplary* rotor was analyzed and optimized. The exemplary geometry was chosen to avoid any confidentiality requirements, so that the full specification of the rotor geometry can be given without industrial restrictions. Thus, this study can be repeated or continued by any group and can be used as a good reference for further studies. Furthermore, a similar size was chosen as by Mohamed [155], enabling comparisons.

The rotor has three straight blades ( $N = 3$ ) with  $R = 1$  m rotor radius. All airfoils have a sharp trailing edge (instead of blunt or rounded trailing edge) with  $c = 100$  mm camber length, mounted at  $0.25c$ . The resulting solidity is  $\sigma = Nc/(2R) = 0.15$ . Shaft was neglected in all CFD simulations.

### 4.3.2.2 Spatial discretization

The mesh was created with ANSYS Gambit 2.4.6 using a hybrid (structured-unstructured) quadrilateral mesh, following the recommendations of [54], as presented earlier. However, as the rotor is smaller compared to the original study, mesh sizes were reduced proportionally: on the blades 0.25 mm, in the middle of the rotating domain 9.17 mm, around the blades 3.2 mm, in the stationary domain 60 mm, around the interface 40 mm mesh size was applied, resulting in approximately 500,000-520,000 two-dimensional quadrilateral finite-volume cells and 700-750 nodes on the airfoil walls. The boundary layer was calibrated using the Schlichting correlation to ensure  $y^+ \approx 1$ , following the recommendations of Maître et al. [147]. The interface contained 720 nodes to ensure conformity throughout the computations and to minimize interpolation errors. Finally, to eliminate the effect of boundaries, a relatively large domain with  $68 \text{ m} \times 58 \text{ m}$  was applied.

### 4.3.2.3 Simulation settings

Based on the previous section, the settings are summarized as follows:

- The flow around Darrieus rotors was modeled as incompressible ( $M < 0.3$ ), wind speed is fixed to 8 m/s.
- Reference density  $\rho = 1.225 \text{ kg/m}^3$  was chosen according to DIN EN 61400 [71], the dynamic viscosity was correspondingly  $\mu = 1.7894 \cdot 10^{-5} \text{ Pa s}$ .
- Second-order implicit temporal discretization is applied with second-order (central or upwind) derivatives.
- In the first step at least 10 revolutions were computed with lower temporal resolution (72 time steps per revolution) to initialize the flow field.
- In the second step at least 5 high-resolution revolutions were computed, with 720 time steps per revolution ( $\Delta\varphi = 0.5^\circ$ ).
- For each time step, 24 inner iterations were computed.
- Only the last quasi-periodic revolution was used for post-processing.
- $k$ - $\epsilon$  Realizable and  $k$ - $\omega$  SST models were considered, with the Coupled solver.

- The sides of the domain were defined as velocity inlet, pressure outlet and symmetry boundaries. At the inlet,  $I = 0.1\%$  and turbulent viscosity ratio of 10 was retained in agreement with recommendations from the literature [54, 134, 135].

#### 4.3.2.4 Characteristic curve of the reference rotor

In order to determine the optimal tip-speed-ratio for the chosen solidity, a NACA0021 profile was retained and the complete characteristic curve was determined (based on the points  $\lambda = 2; 2.5; 3; 3.5; 4; 4.5; 5$ ), see Figure 4.27. As one can see, the rotor has its optimal tip-speed-ratio at 3.5 with  $C_p \approx 36.98\%$  ( $k-\epsilon$  Real.) and  $C_p \approx 38.72\%$  ( $k-\omega$  SST). Thus, this tip-speed-ratio was retained for the further optimization.

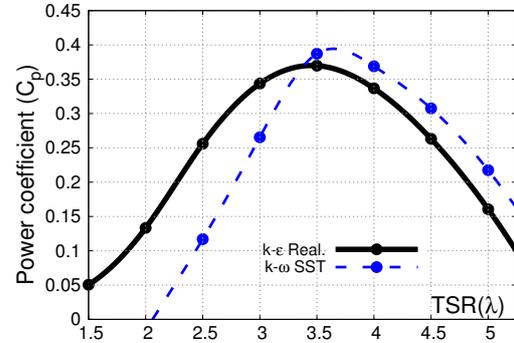


Figure 4.27: Characteristic curve of the exemplary rotor

#### 4.3.2.5 Spatial and temporal independency

In order to test if the setup is correct, mesh, temporal independency and convergence were analyzed for the reference geometry with the  $k-\omega$  SST model. Analyzing only the optimal tip-speed-ratio may not be enough, as several airfoils may move into the stall region during the optimization. Figure 4.28 shows the torque coefficient for doubled temporal resolution (1440 time steps per revolution) and refined mesh (all mesh sizes were reduced to 75 %, resulting in 990,000 cells). The figure additionally shows the results for the penultimate revolution, demonstrating appropriate convergence toward periodic steady-state.

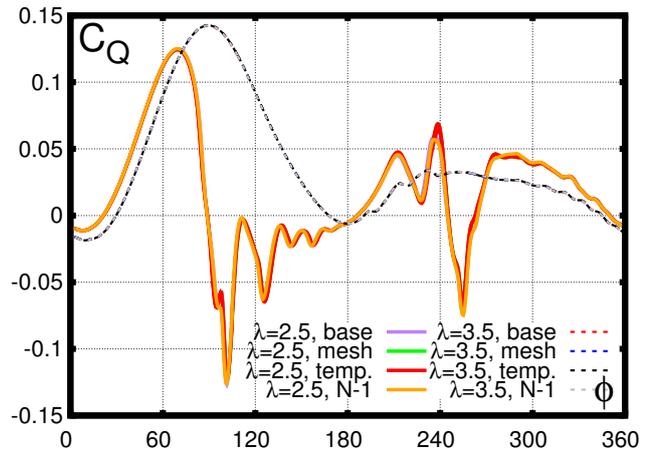


Figure 4.28:  $C_Q(\varphi)$  curves for a single blade for  $\lambda = 2.5$ & $3.5$ , improved *mesh* and *temporal* resolutions are compared to the original simulation (*base*) and for the penultimate revolution ( $N-1$ ) with  $k-\omega$  SST model

When looking at the performance coefficients, appropriate convergence was confirmed again, i.e., differences stayed below the predefined  $\Delta C_p = 2\%$  accuracy range of the optimization: for  $\lambda = 3.5$  values changed from  $C_p = 38.72\%$  to  $C_p = 38.75\%$  and  $C_p = 38.93\%$ , for  $\lambda = 2.5$  from  $C_p = 11.69\%$  to  $C_p = 13.46\%$  and  $C_p = 12.14\%$ , for the base, refined temporal and refined spatial simulations, respectively.

As the mesh resolution around the blades might be slightly different for the different geometries, the mesh size along the blade walls was further reduced by 5 % (to 0.2375 mm) in the final optimization.

### 4.3.3 Setup of the optimization

The complete optimization process was implemented in the OPTimization Algorithm Library++, as presented earlier. One additional modification was added, to increase the robustness of the process. Should the mesh generation process fail in Gambit (due to unspecified error), OPAL++ will try to generate a second mesh with 0.225 mm resolution of the blades (additional 5% refinement).

#### 4.3.3.1 Parameterization

For an optimization the appropriate choice of the parameterization is at least as important as choosing the right algorithm or a stable automation (as presented in Chapter 3). When using too many parameters, methods will converge much slower; when using not enough parameters, one might miss the optimum. For the description of an airfoil geometry, many different options are available. Popular approaches are to use NACA4 parameterization [73], PARSEC-11 parameterization [149, 200] or different polynomials, like Bézier-curves or NURBS [217]. Alternatively, instead of a parameterization one could use a direct topology optimization as widely done in structural optimizations [61].

In the current study, three different methods were tested in a preliminary optimization: a) Bézier-curves, b) description based on the radius of inscribed circles and c) an extended NACA4 method [57] (called *exNACA* from now on). For the optimization, the latter method was finally retained, as it produced very realistic airfoils and could describe NACA4 and other classical airfoils with high accuracy as well. When using Bézier-curves the convergence was slower, as the variables of the optimization (i.e., the coordinates of the control points) had no direct physical meaning. Method b) produced many unrealistic or distorted geometries.

With *exNACA* parameterization, the non-dimensional shape of the airfoil is defined similarly to NACA4, by the thickness and camber. The thickness of the airfoil is defined as for NACA4:

$$t(x) = a_0\sqrt{x} + a_1x + a_2x^2 + a_3x^3 + a_4x^4. \quad (4.32)$$

This equation automatically satisfies  $dt(x)/dx|_{x=0} = \infty$  and  $t(0) = 0$ . The coefficients are determined using the following criteria:

$$t(1) = 0, \quad t(p_t) = \frac{t_{\max}}{2} \quad (4.33)$$

$$\left. \frac{dt(x)}{dx} \right|_{x=1} = -s_{t,2} \frac{t_{\max}}{0.2}; \quad \frac{1}{2} \left( \frac{t_{\max}}{0.2} a_0 \right)^2 = r_L \quad (4.34)$$

where  $r_L$  is the leading edge radius,  $t_{\max}$  is the maximal thickness,  $p_t$  is the location of the maximal thickness and  $s_{t,2}$  is the first derivative of the thickness at the trailing edge. NACA4 uses in comparison  $p_t = 0.3$ ,  $t(1) = 0.002$ ,  $t(0.1) = 0.078$  (corresponding to  $1.1t_{\max}^2$ ) and  $s_{t,2} = 0.234$  (for  $t_{\max} = 0.2$ ). However, in the present formulation,  $p_t$ ,  $t_{\max}$ ,  $s_{t,2}$  and  $r_L$  all remain parameters, resulting in an increased flexibility.

The camber-line is defined in a more flexible form compared to NACA4 (with cubic functions instead of quadratic):

$$c(x) = \begin{cases} \sum_{i=0}^3 b_i x^i & \text{if } 0 \leq x \leq p_c \\ \sum_{i=0}^3 c_i x^i & \text{if } p_c \leq x \leq 1. \end{cases} \quad (4.35)$$

The coefficients can be determined using the following equations:

$$c(0) = 0, \quad c(1) = 0, \quad c(p_c) = c_{\max}, \quad (4.36)$$

$$\left. \frac{dc(x)}{dx} \right|_{x=0} = s_{c,1} \frac{c_{\max}}{0.2}, \quad \left. \frac{dc(x)}{dx} \right|_{x=1} = -s_{c,2} \frac{c_{\max}}{0.2}, \quad (4.37)$$

where the variables of the parameterization are the derivatives at the extremes ( $s_{c,1}, s_{c,2}$ ) together with the value and location of the maximal deviation from the axis ( $p_c, c_{\max}$ ).

The dimensionless airfoil is computed afterwards by using the same transformation as NACA4 [73], i.e., the  $x$ - and  $y$ -coordinates of the lower(-) and upper (+) contours are computed as:

$$\mathbf{p}^{\pm} = \begin{pmatrix} x^{\pm} \\ y^{\pm} \end{pmatrix} = \begin{pmatrix} c(x) \mp t(x) \sin(\Theta) \\ c(x) \pm t(x) \cos(\Theta) \end{pmatrix}, \quad (4.38)$$

where the rotation of the camberline is

$$\Theta = \arctan \frac{dc(x)}{dx}. \quad (4.39)$$

Finally, the airfoil is scaled up to the camber length ( $c$ ), moved to the mounting position ( $0.25c$ ) and rotated around the mounting point with the pitch angle ( $\beta$ ). Fixing the mount position is necessary, as the pitch angle and the mounting position are not independent [86]. Moreover, the pitch angle is a very important optimization parameter, as studies have confirmed that a correctly chosen pitch angle leads to improved performance for VAWTs [86]. Throughout the study, the parameter space presented in Table 4.6 is applied for the optimization.

Table 4.6: Parameter space for the optimization

Param.	$\beta$	$p_t$	$t_{max}$	$r_L$	$s_{t,2}$	$p_c$	$c_{max}$	$s_{c1}$	$s_{c2}$
Min.	$-5^\circ$	0.25	0.06	0.005	0.01	0.25	-0.15	0.7	0.01
Max.	$5^\circ$	0.35	0.3	0.08	0.3	0.35	0.15	2.0	0.2

### 4.3.3.2 Constraints

Unfortunately, not all generated profiles are correct and can contain different errors. In order to ensure, that only realistic profiles are evaluated, several constraints were applied. Thus, only airfoils were accepted, that ensured that (C1) the thickness function is always positive, (C2) the camber function does not change sign, (C3) the thickness and (C4)

camberline functions have only one extremum and (C5) the contours cannot contain any fracture:

$$(C1) : t(x) > 0 \quad \forall x \in [0; 1]. \quad (4.40)$$

$$(C2) : c(x_1)c(x_2) \geq 0 \quad \forall x_1, x_2 \in [0; 1]. \quad (4.41)$$

$$(C3) : \left. \frac{dt(x)}{dx} \right|_{x_1} = 0; \left. \frac{dt(x)}{dx} \right|_{x_2} = 0 \rightarrow x_1 = x_2, \quad (4.42)$$

$$(C4) : \left. \frac{dc(x)}{dx} \right|_{x_1} = 0; \left. \frac{dc(x)}{dx} \right|_{x_2} = 0 \rightarrow x_1 = x_2, \quad (4.43)$$

$$(C5) : \lim_{\delta x \rightarrow 0} \left( \arctan \frac{dp_y^\pm(x + \delta x)}{dx} - \arctan \frac{dp_y^\pm(x)}{dx} \right) = 0 \quad \forall x \in [0; 1]. \quad (4.44)$$

The coordinates of the contours were computed at 600 different locations. Then, the generated file was forwarded to Gambit 2.4.6 for mesh generation. If (C1-C5) were not satisfied, the configuration was marked *invalid* in OPAL++.

#### 4.3.3.3 Objective functions

The goal was to maximize the performance coefficient ( $C_p \rightarrow \max$ ) corresponding to the optimal tip-speed-ratio ( $\lambda = 3.5$ ), computed from the ultimate revolution.

#### 4.3.3.4 Optimization with Genetic Algorithm

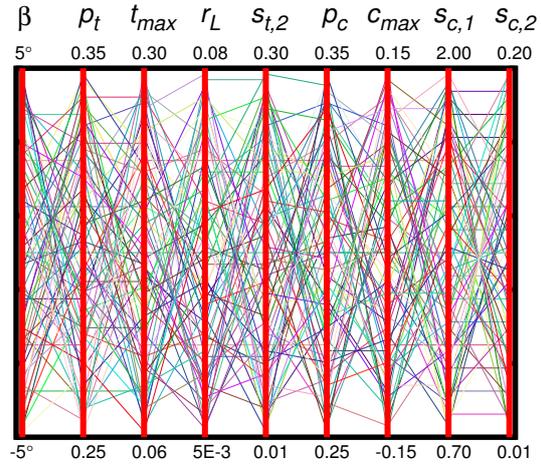
For the optimization a single-objective genetic algorithm (called *GENETIC1* in OPAL++) was applied. The properties of this method (already described in Chapter 1) and the applied settings are summarized below:

- All variables have a real representation.
- Each generation contains  $N$  individuals.
- Tournament with 2 cycles is used to select parents.
- SBX is used for cross-over with distribution index of  $\eta_c = 20$  and probability of  $p_c = 0.8$  [66].
- Polynomial mutation with distribution index of  $\eta_m = 10$  and probability of  $p_m = 1/9$  is applied [67, 68].
- Elitist selection is applied (individuals are immortal to preserve information, but only the best  $N$  individuals survive).

Although there is no guarantee that the method will converge to the global optimum (far more evaluations would be needed), the simple Genetic Algorithm managed to find significantly improved configurations, demonstrating its efficiency.

## Initialization of the first generation

To provide a better coverage of the search domain, a low-discrepancy quasi-random sequence, the Sobol sequence, was applied [122] for the initial generation instead of using a random population. This not only ensures that the domain is well covered, but additionally, when starting the optimization with different turbulence models, it is guaranteed that initial generations will contain the same configurations. The coverage of the parameter space (see Table 4.6) is shown using a parallel plot for  $N = 72$  in Fig. 4.29.



To speed-up convergence, some classical airfoils, e.g., NACA0021, were added to the first generation.

Figure 4.29: Design-Of-Experiment with Sobol method for  $N = 72$

However, to be able to include these profiles in the optimization efficiently (i.e., with cross-over and mutation operators as well), the airfoils had to be reproduced using the *exNACA* parameterization. A preliminary optimization was executed for each profile with the objective to minimize the distance between the points of the reference profile and the newly computed profile. This was successfully achieved with maximally  $70 \mu\text{m}$  error, which is well below manufacturing tolerances.

## 4.3.4 Results

The optimization was executed on the Neumann High Performance Cluster of the Otto-von-Guericke University. Each CFD simulation was executed by using all 16 cores of a single node ( $2 \times$  Intel Xeon E5-2630 v3), and  $N$  nodes were used for the optimization process simultaneously ( $N$  being the number of individuals in a single generation). With both turbulence models, a single computation took approximately 16-18 hours, depending on the configuration.

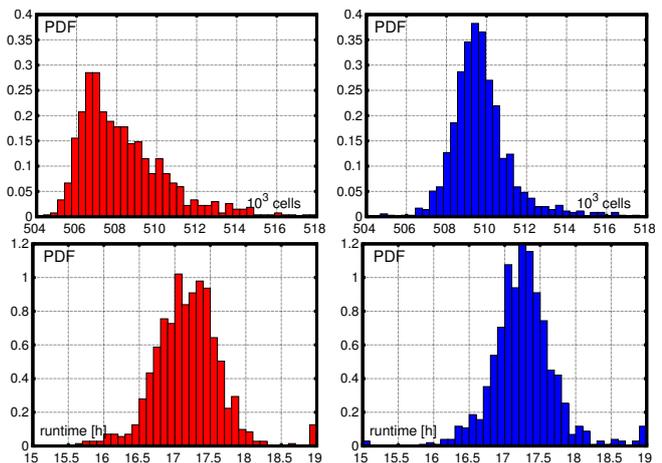


Figure 4.30: Distribution of mesh sizes and runtime during the optimization (red= $k-\epsilon$  Realizable; blue= $k-\omega$  SST)

The histogram of the necessary runtime (after removing invalid individuals and a few

outliers above 24 hours) and the mesh size distributions are presented in Figure 4.30. As one can see, the runtime shows a very similar distribution to the Gaussian. This also highlights, why one cannot achieve 100% efficiency in the parallelization of CFD-based optimizations.

In the followings, as they are of special interest, the following geometries have a special notation:

- NACA0021: NACA0021 without pitch and with a sharp trailing edge;
- NACA0021\*: NACA0021 with pitch;
- $\text{OPT}_\epsilon$ : best configuration found in the optimization process with  $k-\epsilon$  Realizable model;
- $\text{OPT}_\omega$ : best configuration found in the optimization process with  $k-\omega$  SST model;
- $\text{OPT}_\omega^S$ : best configuration found in the optimization process with  $k-\omega$  SST model with the camber removed (S means **S**ymmetric,  $c_{max} = 0$ ).

#### 4.3.4.1 Optimization with $k-\omega$ SST model

**Optimization process** In this optimization  $N = 72$  individuals were applied for each generation, resulting in the simultaneous use of 1152 processor cores. Altogether 14 generations were computed. Out of the 1008 generated individuals, 888 resulted in valid evaluations (valid geometry, mesh & CFD computation). The optimization process is presented in Fig. 4.31.

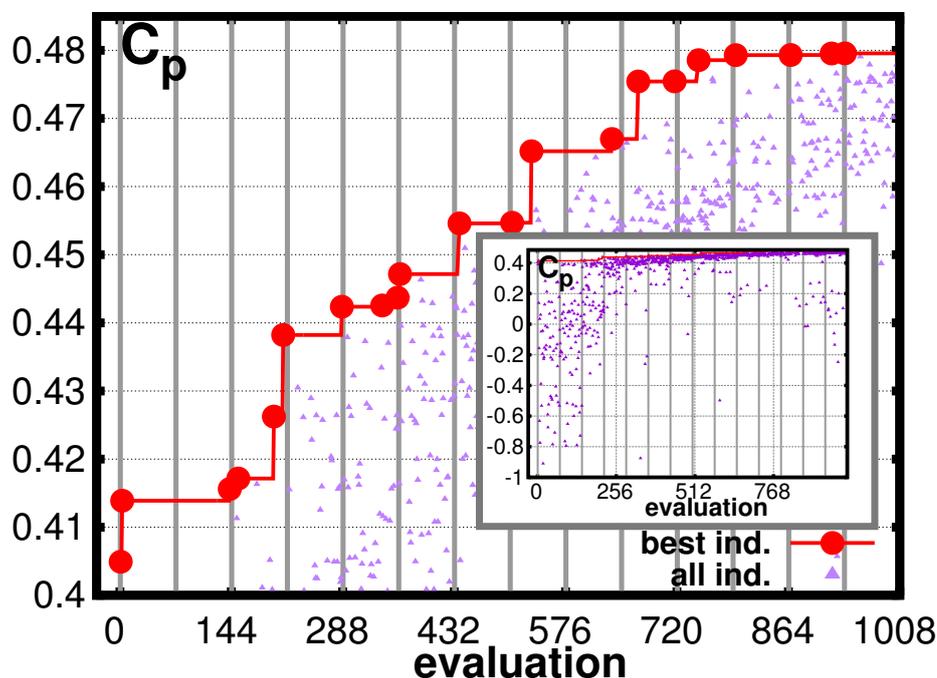


Figure 4.31: Optimization with the  $k-\omega$  SST model

The configuration, which is denoted as  $\text{OPT}_\omega$  (see Fig. 4.32), had an impressive performance coefficient of  $C_p = 48\%$ . With the increasing number of generations the improvements became smaller in the subsequent generations, i.e., the optimization converged. In Figure 4.32 one can compare the different airfoil shapes. Comparing  $\text{OPT}_\omega$  and  $\text{OPT}_\omega^S$  reveals, that the camber is not very significant. Interestingly, the optimal profile is not curved towards the centre of the circle, but outwards (green arrow shows the location of the axis of rotation). When applying the pitch angle of  $\text{OPT}_\omega$  to NACA0021, the resulting NACA0021\* profile seems to be significantly thicker compared to  $\text{OPT}_\omega$ .

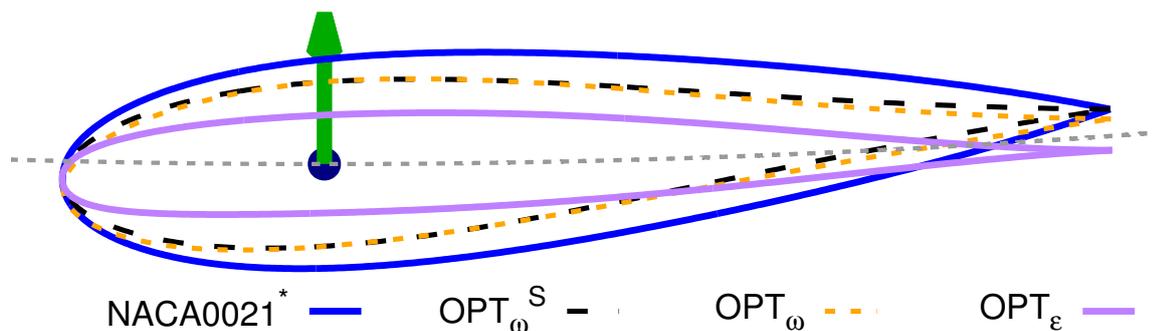


Figure 4.32: Comparison of the different geometries obtained by optimization

## Modality

After the optimization, solutions near the optimum were analyzed. All solutions were filtered, for which the CFD simulation predicted  $C_p(\text{OPT}_\omega) - C_p < 2\%$ , i.e., all solutions with  $46\% < C_p < 48\%$ . These configurations are displayed in Fig. 4.33 using a parallel plot [117]. As one can see, the optimal solutions all share similar pitch angles, thickness and camber. However, regarding the first derivative of the thickness at the trailing edge ( $s_{t,2}$ ), two dominant values are present. This indicates that the present problem may show multi-modality, instead of having one single global optimum. For a more efficient exploration of the multi-modal solutions, Firefly method could be used in future research [230].

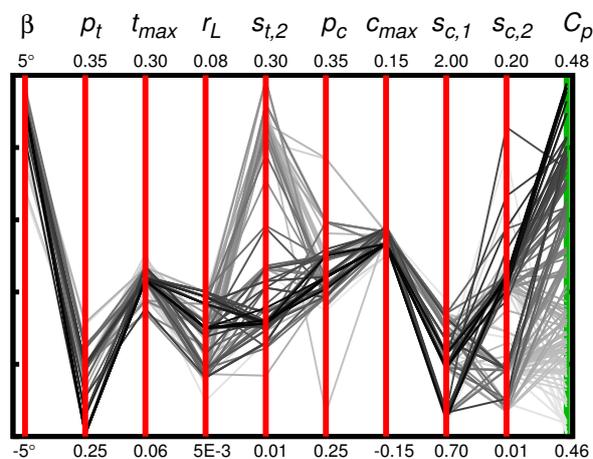


Figure 4.33: Airfoils with  $C_p > 46\%$  (darker lines represent configurations with higher performance)

**Multi-fidelity analysis** When comparing the performance coefficients computed from the ultimate ( $C_p$  in Table 4.7) and penultimate revolutions ( $C_{p,N-1}$  in Table 4.7), very sur-

prising findings are to be seen. For NACA0021, the differences are very small ( $\Delta C_p < 0.13\%$ ), while for  $\text{OPT}_\omega$  the differences are quite large ( $\Delta C_p < 0.92\%$  and  $\Delta C_p < 0.42\%$  for  $k-\omega$  SST and  $k-\epsilon$  Realizable model, respectively). This indicates, that although for several geometries the validation in the previous chapter may be appropriate, but for others more revolutions would be necessary, as recommended by Balduzzi et al. [11].

Using a very strict criteria would increase the computational time several times. However, one should not forget, that even in the presence of an error the optimization process and its result still remain valid, as long as there is a clear trend (i.e., the error exhibits a systematic bias). This means that the optimization only has to show the right direction! Such an approach is widely used in multi-fidelity optimization processes.

To reduce the effect of this convergence problem, the temporal resolution was increased to 10 ( $5^\circ/\text{time step}$ )+10 ( $0.25^\circ/\text{time step}$ ), halving the size of the time step and doubling the number of revolutions. The results obtained with this second fidelity are denoted as  $C_p^2$ . The performance coefficient of NACA0021,  $\text{OPT}_\omega$ ,  $\text{OPT}_\omega^S$  and NACA0021 with the pitch angle of  $\text{OPT}_\omega$  (denoted as NACA0021\*) were recomputed using  $k-\omega$  SST and  $k-\epsilon$  Realizable models as well; values are presented in Table 4.7. With 10+10 revolutions, the total physical time computed was 4.55 s, requiring 62 hours of runtime with 16 cores. To analyze the convergence further, the computation of NACA0021 and  $\text{OPT}_\omega$  was extended to simulate in total 27 seconds ( $> 120$  revolutions). Performance coefficients were computed again (denoted as fidelity level 3:  $C_p^3$ ), resulting in 37.91% and 41.80% for the ultimate and penultimate revolution as well for NACA0021 and  $\text{OPT}_\omega$  respectively. Hence, at this point, results were completely converged.

Table 4.7: Performance coefficients of the different airfoils with different models and fidelity

Model	NACA0021	NACA0021*	$\text{OPT}_\omega$	$\text{OPT}_\omega^S$
$C_p(k-\epsilon \text{ Real.})$	37.22%	-	42.8%	-
$C_{p,N-1}(k-\epsilon \text{ Real.})$	37.32%	-	43.22%	-
$C_p(k-\omega \text{ SST})$	39.06%	-	47.96%	-
$C_{p,N-1}(k-\omega \text{ SST})$	39.19%	-	48.88%	-
$C_p^2(k-\epsilon \text{ Real.})$	36.95%	37.56%	41.19 %	41.00%
$C_{p,N-1}^2(k-\epsilon \text{ Real.})$	37.00%	37.64%	41.33%	41.14%
$C_p^2(k-\omega \text{ SST})$	38.76%	39.86%	45.74%	44.82%
$C_{p,N-1}^2(k-\omega \text{ SST})$	38.82%	39.97%	46.01%	45.03%
$C_p^3(k-\omega \text{ SST})$	37.91%	38.70%	41.80 %	-
$C_{p,N-1}^3(k-\omega \text{ SST})$	37.91%	38.70%	41.80%	-

The instantaneous torque coefficients of  $\text{OPT}_\omega$  with  $k-\omega$  SST (fidelity level 1,2,3) and  $k-\epsilon$  Realizable (fidelity level 1,2) models are compared in Fig. 4.34. As one can see, the differences are small and both models agree to a large extent, showing only smaller offsets at the highest angle-of-attack ( $60^\circ < \phi < 120^\circ$ ) in the upstream zone and at wake-blade interaction ( $210^\circ < \phi < 330^\circ$ ) in the downstream zone. As  $C_p$  is an integral quantity, the small offset still results in noticeable differences.

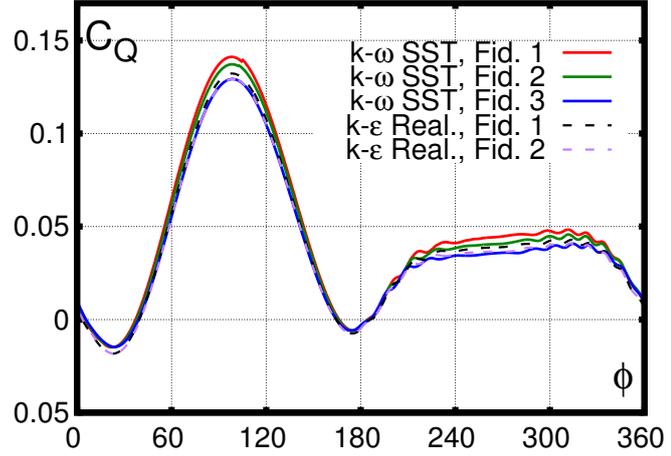


Figure 4.34:  $C_Q(\varphi)$  curves for a single blade ( $\lambda = 3.5$ ) with two different turbulence models for  $\text{OPT}_\omega$

**Source of improvement** Using fidelity level 2 and  $k-\omega$  SST model, the source of the improvements was analyzed. As one can see in Table 4.7, the improvement originates from many different sources. Comparing  $\text{OPT}_\omega$  and  $\text{OPT}_\omega^S$  reveals, that the cambered geometry is indeed slightly superior compared to the symmetric variant ( $\Delta C_p = 0.92\%$ ). This means, that asymmetric geometries could be superior for practical applications.

Furthermore, the optimized blades have a significant pitch angle ( $\beta > 4^\circ$ ). When applying this pitch angle to NACA0021, an improvement is achieved for this profile as well ( $\Delta C_p = 1.1\%$ ). However, the pitched geometry still does not have the same performance as  $\text{OPT}_\omega^S$  ( $\Delta C_p = 5.88\%$ ). This means that actually the new airfoil shape helps in providing a superior performance. Summarizing, using a camber, a pitch angle or a different airfoil shape alone can provide an improved performance. However, using an optimal camber ( $\Delta C_p = 0.92\%$ ), airfoil ( $\Delta C_p = 5.88\%$ ) and pitch ( $\Delta C_p = 1.1\%$ ) together results in a significantly improved performance. Although it could be tempting to use the optimal values of the present case for a different wind turbine, one should not forget, that these parameters have a strong interaction with each other. Therefore each turbine should have a (slightly) different optimal pitch angle, camber and airfoil shape. When checking fidelity level 3,  $\text{OPT}_\omega^S$  is still significantly better compared to NACA0021.

Looking at the instantaneous torque values (see Fig. 4.35), the mechanism behind the improvement becomes clear. Applying the pitch angle and modified airfoil shape results in decreased torque values and thus energy extraction in the upstream zone ( $30^\circ < \phi < 120^\circ$ ). This reduced energy extraction results however in an increased wind speed in the downstream region. Normally, the rotor would experience in the downstream zone smaller angle-of-attack due to the decreased wind speed. However, with increased downstream wind speeds angle-of-attack and energy extraction increases in the downstream zone as well, resulting in an increased energy extraction in total. This means, that  $\text{OPT}_\omega^S$  balances angle-of-attack and the energy extraction between the upstream and downstream zones. Finally, adding the camber keeps energy extraction almost constant in the downstream zone, while providing an increase in the upstream zone, where angle-of-attack is large ( $30^\circ < \phi < 120^\circ$ ).

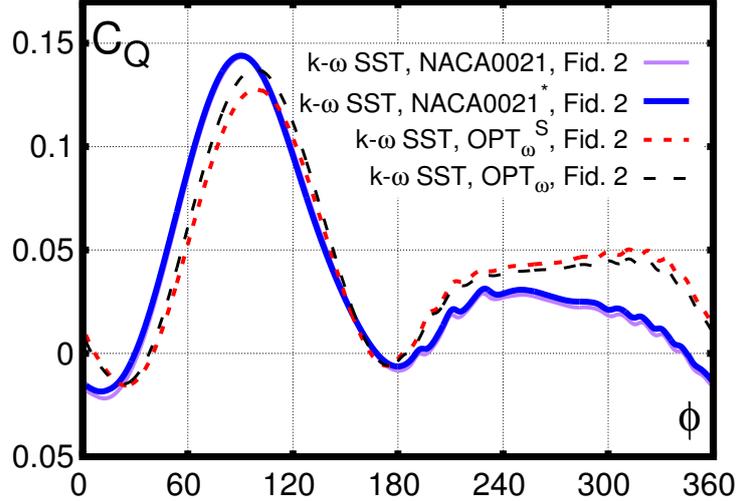


Figure 4.35:  $C_Q(\varphi)$  curves for a single blade ( $\lambda = 3.5$ ) for different airfoils with  $k-\omega$  SST model

**Comparison with  $k-\epsilon$  Realizable** When comparing the performances with fidelity level 2 as computed with the  $k-\epsilon$  Realizable model, the improvements are unfortunately different, but the trend still holds: i.e., modifying pitch ( $\Delta C_p = 0.61\%$ ), airfoil shape ( $\Delta C_p = 3.44\%$ ) and camber ( $\Delta C_p = 0.19\%$ ) results in an improved performance. The new airfoil shape seems to be very robust. Independently from the turbulence modeling and level of fidelity, it still provides an improvement.

#### 4.3.4.2 Optimization with $k-\epsilon$ Realizable model

The previous validation section stated, that  $k-\epsilon$  Realizable model seems to underestimate stall effect (see Section 4.2.8). This makes the model incompatible with an optimization approach due to robustness considerations.

In order to test this assumption (i.e., that  $k-\epsilon$  Realizable model is less suited for an optimization process), a second optimization was performed using  $N = 64$  individuals in each generation (as the information gained from the previous optimization was used, the number of evaluations could be reduced).

**Optimization process** For the initialization the same Sobol series was applied with the same classical airfoils as earlier, additionally adding  $\text{OPT}_\omega$  and  $\text{OPT}_\omega^S$ . 13 generations were computed. From the 832 tested configurations, 772 individuals were valid (correct geometry, mesh & CFD computation). The results of the optimization process are presented in Fig. 4.36. As one can see, the results seem to be very different. The process is able to quickly find (in the first generation from the Sobol initialization) a configuration with  $C_p \approx 44\%$ . From this point on, only slight improvements are found. After 13 generations, the best configuration still has  $C_p = 44.2\%$  performance. Thus, no further computations were needed.

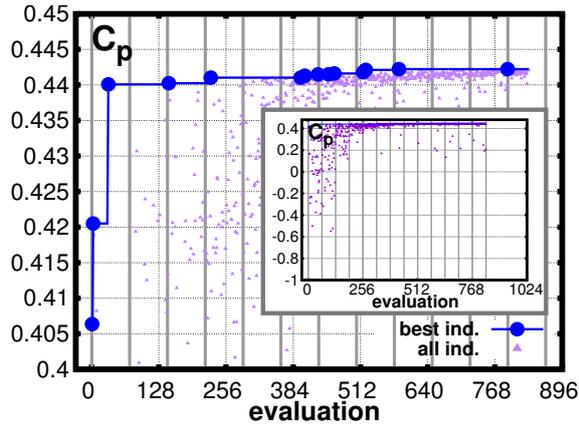


Figure 4.36: Optimization with the  $k-\epsilon$  Realizable model

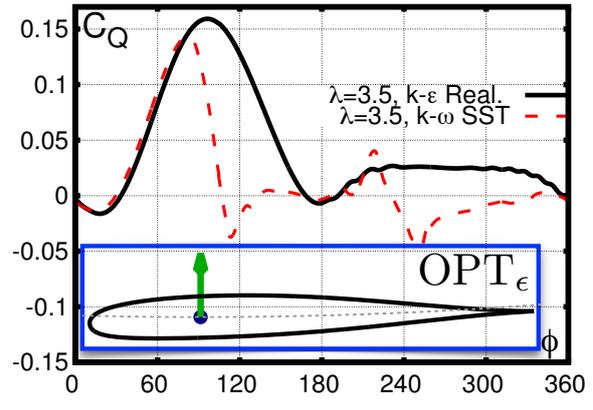


Figure 4.37:  $C_Q(\varphi)$  curves for a single blade for  $\lambda = 3.5$  with two different turbulence models for  $\text{OPT}_\epsilon$

**Recomputing optimum with  $k-\omega$  SST** However, when computing the performance of  $\text{OPT}_\epsilon$  with  $k-\omega$  SST, the result is surprisingly only  $C_p = 13.21\%$ ! Looking at the instantaneous torque coefficient reveals the reason for this surprising behavior: as previous studies have already indicated, the  $k-\epsilon$  Realizable turbulence model underestimates stall effects at large angle-of-attacks and wake-blade interactions (see Fig. 4.37). Although without experimental measurements it is difficult to tell, which model predicts the correct values for the **present** geometry,  $\text{OPT}_\omega$  seems to be a much more promising geometry. Due to its larger thickness,  $\text{OPT}_\omega$  is superior when considering mechanical aspects as well (e.g., fatigue).

### Correlation of turbulence models

This underestimation can be further confirmed by looking at the correlation of the power coefficients computed with  $k-\omega$  SST and  $k-\epsilon$  Realizable models based on 60 configurations from the initial generation, see Fig. 4.38. As one can see, the correlation is strong (i.e., geometries exhibiting high performance with one model result with high probability in superior configurations for the other model). However, there are many outliers. More importantly, geometries, which exhibit high performance with the  $k-\omega$  SST model always show high performance with the  $k-\epsilon$  Realizable model.

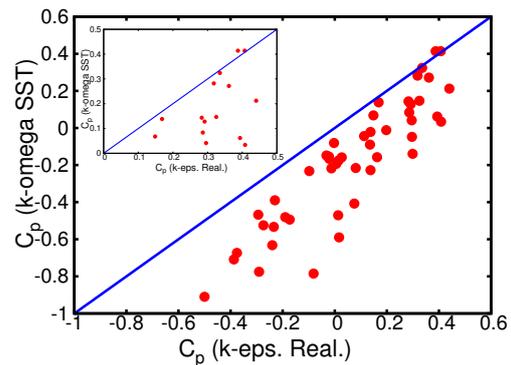


Figure 4.38: Correlation of the performance coefficient for 60 configurations of the initial generation

Unfortunately, inverted, the statement does not hold, i.e., superior configurations found with the  $k$ - $\epsilon$  Realizable model can lead to inferior configurations with the  $k$ - $\omega$  SST model, due to the underestimated stall. These conclusions agree with the findings in [57]. The outliers in the present study, however, are even more pronounced, due to the significantly larger parameter space ( $0.06 < t_{max}$  instead of  $0.17 < t_{max}$  in [57]). Hence, the  $k$ - $\epsilon$  Realizable model is not appropriate for optimization.

#### 4.3.4.3 Analysis with surrogate model

As a large number of configurations were computed, it becomes possible to analyze the problem with a Surrogate model. For this, a Response Surface Method was chosen with a polynomial containing the constant, all linear, quadratic and first-order interaction terms. Altogether, 55 coefficients had to be evaluated. For the fitting all configurations exhibiting  $C_p > 30\%$  were chosen to focus on the optimal region (containing 680 configurations). The coefficients were evaluated in OPAL++ by first normalizing all variables and objectives to the  $[-1; 1]$  domain and afterwards using an Iteratively Reweighted Linear Least Squares (IRLSQ) method, relying on Gaussian weights.

In order to test the quality of the model, the performance of the original points were recomputed with RSM. The correlation is presented in Fig. 4.39. The performance coefficient obtained from CFD using the  $k$ - $\omega$  SST model is denoted as  $C_p(CFD)$ , from the surrogate model as  $C_p(RSM)$ . One can see that the correlation is very good in general; almost all points reside in the  $C_p(CFD) \pm 2\%$  (yellow) zone. Only a small number of outliers are found (28 out of 680), where the RSM fails. As outside the  $\pm 2\%$  zone the outliers were always the result of overestimation of the performance with RSM, the response surface is safe for additional analysis (i.e., it would not underestimate the performance of promising configurations).

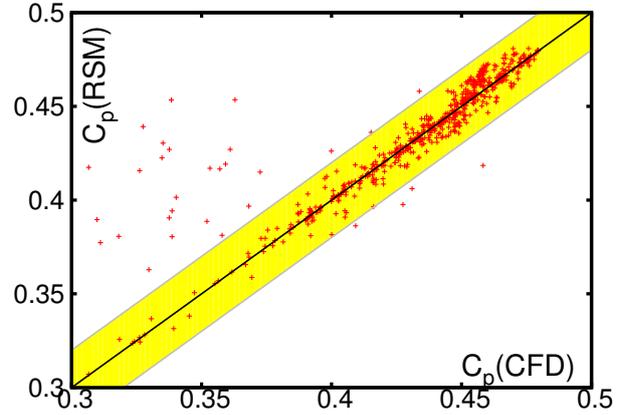


Figure 4.39: Correlation of RSM and CFD for 680 configurations with  $C_p(CFD) > 30\%$

Thus, a second analysis was performed in OPAL++. First, twenty billion points were generated randomly (using a Mersenne-Twister random generator [150]) in the parameter space spanned by the 680 configurations. Then, performance was evaluated based on the Response Surface Model. Additionally, GENETIC1 was started with 200 generations and 2000 individuals per generations to search for the best configuration in the parameter space spanned by the 680 configurations (performance was evaluated again with the Response Surface Model). Due to the  $\pm 2\%$  error zone and the  $C_p = 48\%$  optima found in CFD-based optimization, OPAL++ was looking for configurations with  $C_p > 48 + 2\%$ . However, no configuration was found satisfying this criterion, neither with the brute force search method, nor with the metamodel-based optimization. This indicates, that the

CFD-based optimization was already able to explore the parameter space appropriately.

### 4.3.5 Possible improvements

Although the present method proved to be very efficient and already managed to identify superior configurations, there is still place for further improvements:

- One obvious improvement would be to use CFD models with higher fidelity, e.g., 3D LES computations. This would result in significantly larger runtime.
- Additional objective functions could be added, e.g., to minimize the cost or to minimize the noise emission. This would result in a multi-objective approach.
- Instead of maximizing the performance coefficient for the optimal tip-speed-ratio, one could maximize the annual energy yield based on the Rayleigh distribution of the wind speed and the complete characteristic curve.
- Airfoil shapes might be sensitive to operating conditions [174]. For a robust design, Robust Design Optimization (RBO) or Reliability-Based Design Optimization (RBDO) could be performed. For this, Uncertainty Quantification (UQ) is required (see Chapter 6).

### 4.3.6 Conclusions and summary

In the present study, the optimization of H-Darrieus rotor airfoils was considered based on two-dimensional CFD computations. First, an exemplary rotor geometry was chosen and the characteristic curve of the rotor was computed with  $k-\omega$  SST and  $k-\epsilon$  Realizable models. Afterwards, the parameterization was presented, followed by the optimization with two different turbulence models. The optimization confirmed again, that  $k-\epsilon$  Realizable model underestimates stall and  $k-\omega$  SST is better suited. Afterwards, the identified optimum was analyzed with increased fidelity. Detailed simulations with both turbulence models confirmed the increased performance.

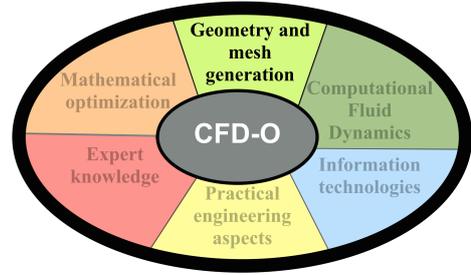
At last, based on the computed points a Response Surface was created and analyzed. No further promising configurations were found using this method, confirming that the Genetic Algorithm provided an appropriate exploration. The following conclusions were made:

- The optimum achieved in the optimization with  $k-\epsilon$  Realizable model leads to inferior results compared with other turbulence models.
- The optimum achieved in the optimization with  $k-\omega$  SST model shows increased performance with each tested turbulence model and temporal resolution.
- Non-zero camber, pitch and modified shape contribute to improved performance.
- The improved design has a more balanced energy extraction in the upstream and downstream zones.

*In the current section, the optimization of Darrieus wind turbines blades was considered. First, an extensive validation was presented to point out the importance of an appropriate computational model. Additionally, the automatization was presented in detail, which can be a very long and complex process to provide robust mesh generation and*

*evaluation. Following the numerical validation, the optimization itself was presented with the parameterization and analysis of the results. Even Response Surface method could not identify configurations that could beat the performance of the optima found by the CFD-based optimization; Genetic Algorithm provided an appropriate exploration.*

*Although significant improvements were achieved, this study was still based on 2D CFD models. For realistic, practical optimization problems, this is not always enough. Thus, the next chapter will present a proof-of-concept to show that implementing a similar optimization in 3D is not necessarily more difficult, but “only” requires more computational power.*



## Chapter 5

# Optimization of VAWT blades revisited - Extension to 3D

“The fuel in the earth will be exhausted in a thousand or more years, and its mineral wealth, but man will find substitutes for these in the winds, the waves, the sun’s heat, and so forth.”

*John Burroughs (1916)*

*The present chapter will especially focus on questions related to automated, 3D geometry and mesh generation.*

*In the previous chapter the optimization of H-Darrieus wind turbines using 2D CFD simulations was considered, with success. Although the results are already very promising, the approach is unfortunately very limited, as only straight bladed Darrieus-rotors can be considered with the 2D method. Popular designs, such as Troposkein blades, or innovative new approaches, such as twisted blade rotors or winglets cannot be approached with the previous method, unfortunately.*

*In the present chapter, the question of the optimization is revisited to answer the questions:*

- (a) Is it possible to extend the Darrieus optimization to 3D?*
- (b) What is an efficient way for the parameterization of 3D rotor blades?*
- (c) Can the parameterization be implemented without additional softwares (e.g., CAD licenses)?*
- (d) Is it possible to generate 3D meshes in a fast, robust and automated way?*

*Fortunately, for all questions a positive answer will be given. The intention of the current chapter is not to optimize a complete three-dimensional Darrieus rotor (although in principle it would be possible), but to present a proof-of-concept using winglets. The following sections are partly based on a conference presentation:*

DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Towards the optimization of winglets for H-Darrieus rotors: Parameterization and automatization for performance evaluation based on 3D-URANS. In *European Congress on Computational Methods in Applied Sciences and Engineering* (Crete Island, Greece, 2016) [60].

## 5.1 Blade tips of Darrieus rotors

Darrieus rotors are lift-driven rotors equipped with airfoils. Airfoils generate lift through the pressure difference on the two sides of the airfoil. As all blades have a finite aspect ratio, an airflow is induced from the higher pressure side to the suction side along the end of the blade and a wingtip vortex is generated, which results in losses. In wind turbines with small aspect ratio these losses can be large (even up to 25% [34]), while at high aspect ratio the corresponding losses become negligible.

This aerodynamic problem has been known for a long time, mostly in connection with airplanes, and several blade tip modifications were already proposed to overcome it, e.g., vortex diffuser, tip sails [130], wing-grids, raked tip [106] and many different winglet designs. Some of them are presented later in Figure 5.7. Winglets are now widely used for airplane wings [223] and, to a less extent, for wind turbines.

Only few studies have considered winglets on wind turbines in the scientific literature. Taborda et al. [210] analyzed a cross-flow vertical axis wind turbine based on 3D transient turbulent simulations using the Transitional SST model. The original rotor without winglet, with an asymmetric winglet and with a symmetric winglet was analyzed. They came to the conclusion, that the performance can be increased using winglets, the optimal solution being to use a symmetric winglet. Berlferhat et al. experimentally analyzed the polars for different winglets [17]. The aerodynamic efficiency of HAWTs with blade tip modifications was also discussed in the work of Gaunaa and Johansen [94] and in the work of Imamura et al. using a vortex lattice method [114].

Winglets on VAWTs have been even less considered. Amato et al. investigated Darrieus rotors equipped with different blade tips [4]. The original configuration, an aerodynamic bulkhead, elliptic termination and three different winglets were evaluated using numerical simulations, validated by the measurements of Castelli et al. [34]. This study indicated that using a correct wingtip configuration, the performance can be increased. However, a poorly chosen solution can further decrease it.

The present study is even more challenging compared to the blade shape optimization of Darrieus wind turbine, since the evaluation of the winglet performance can only be achieved using three-dimensional, transient, turbulent, large-scale Computational Fluid Dynamics (CFD) simulations. As a result, only very few articles deal with the CFD simulation of winglets for VAWTs. The optimization of such winglets has never been discussed in the literature according to the best knowledge of the author.

The first step of the current study is to derive a fully automatic procedure to optimize winglets for H-Darrieus rotors. To show the viability of this proof-of-concept, the general geometrical description of parametric winglets is first given, followed by a simplified model parameterized by only two parameters. Afterwards, a very simple Design-of-Experiment is created with 6 different configurations and CFD simulations are carried out for a single tip-speed-ratio. Finally, results are analyzed and conclusions are drawn. The evaluation of the cases is executed in a completely automatic manner in an optimization framework, proving the viability of the developed approach.

## 5.2 Setup of CFD Model

In the followings the applied CFD setup will be presented.

### 5.2.1 Physical model

As many different studies found, that 2D and 3D computations may require different turbulence models [34], both  $k$ - $\epsilon$  Realizable and  $k$ - $\omega$  SST models are retained for the present study, despite the conclusions of the previous chapter. The flow around the rotor was modeled as incompressible (this is a widely adopted simplification in the literature [34], as the relative speed of the blades stays well below  $M = 0.3$  due to the relatively small rotation speed of small Darrieus rotors). Reference density  $\rho = 1.225 \text{ kg/m}^3$  was chosen according to DIN EN 61400 [71], and the dynamic viscosity was chosen correspondingly as  $\mu = 1.7894 \cdot 10^{-5} \text{ Pa s}$ .

The system of equations was solved in a coupled manner, as the solver was revealed to be superior in earlier chapters. Additionally, Cell Quality Remediation was activated [37]. This model can efficiently improve stability locally if divergence is detected in a couple of deformed cells, which is unavoidable in an automatic 3D mesh generation process. To promote precision, second-order implicit temporal discretization is always applied in the current work together with second-order (central or upwind) derivatives.

### 5.2.2 Boundary conditions

The sides of the computational domain are defined as velocity inlet, pressure outlet and symmetry boundaries (except for the validation in the wind tunnel, where no-slip conditions are applied except for the symmetry plane). Proper inlet conditions are clearly the most important ones:  $u = 9 \text{ m/s}$ ,  $I = 0.1\%$  and  $\text{TVR} = 10$  were retained for all configurations, in agreement with recommendations from the literature [134, 135] and as a result of preliminary studies [54], as already discussed in Chapter 4.

For the computations with winglets,  $\lambda = 2.6$  was used in all cases. For the experimental validation the angular velocity was varied while keeping the wind speed constant.

### 5.2.3 Spatial discretization

After the geometry was imported into StarCCM+ v10, the mesh generation is executed in a completely automatic manner based on a custom JAVA script developed for the present case. Relevant parts of the geometry are retained and combined with the surroundings with the help of the ‘Surface Wrapper’. Mesh for the stationary and rotating domain is created using polyhedral mesh, additional refinements are created with several volume controls, surface and edge controls to ensure an appropriate resolution. In the stationary domain, refinement is added for the wake, in the rotating domain for the blade, wingtip and trailing edge (see Fig. 5.1). Extreme care was given to resolve the flow around the wingtips. The most important settings are summarized in Table 5.1.

The applied script can be used with any 3D Darrieus blade geometry.

The inlet and outlet domains are extended to allow development of the flow and to ensure an appropriate distance from the boundaries with the help of ‘Directed Mesh’.

### 5.2.4 Multi-fidelity approach

In order to speed-up the CFD computation, a multi-level approach was adopted. In this multi-fidelity computational approach not only the temporal, but the mesh resolution

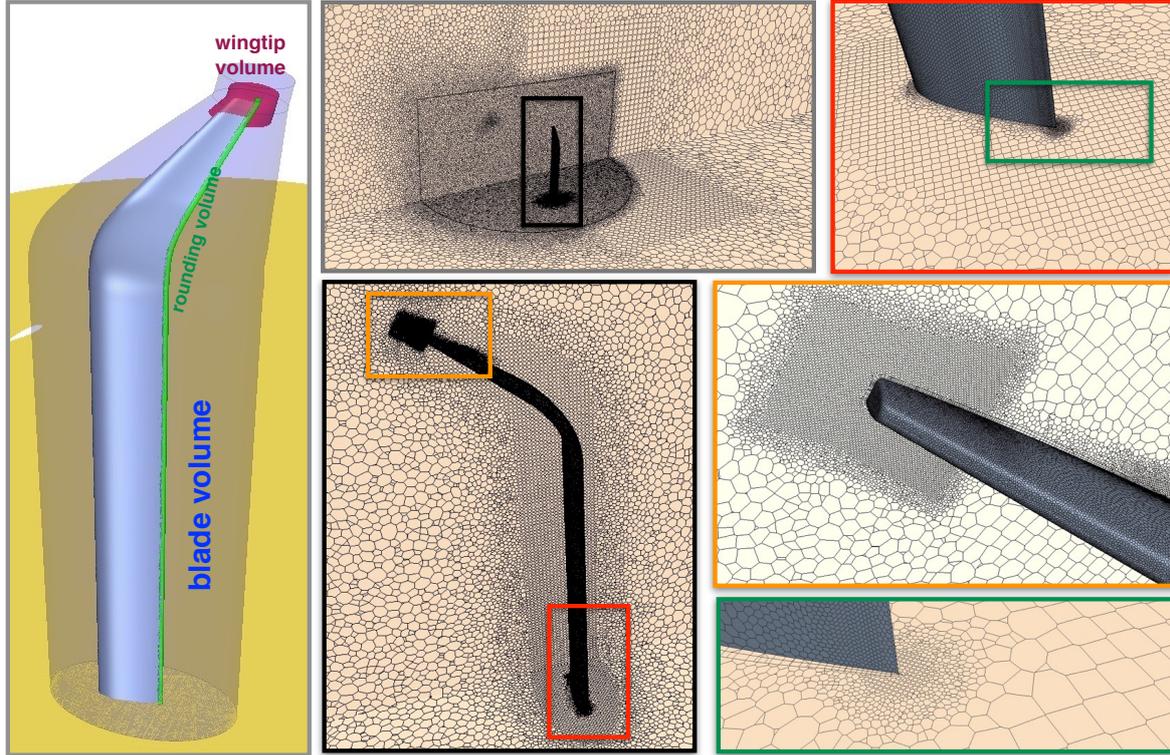


Figure 5.1: Cross-section of mesh at different locations

was varied as well. The following three steps were defined for the computations:

1. Level 1 - *Initialization of flow field*: Coarse mesh resolution ( $s_{act,i} = s_{calib,i} \cdot 1.5$ ), large time step ( $\Delta\varphi = 5^\circ$ ),  $N = 6$  revolutions, Solver Courant number=15.
2. Level 2 - *Initialization of flow field near the blades*: Fine mesh resolution ( $s_{act,i} = s_{calib,i}$ ), intermediate time step ( $\Delta\varphi = 1^\circ$ ),  $N = 1$  revolution, Solver Courant number=30.
3. Level 3 - *Precise computation of torque*: Fine mesh resolution ( $s_{act,i} = s_{calib,i}$ ), small time step ( $\Delta\varphi = 0.5^\circ$ ),  $N = 1$  revolution, Solver Courant number=30.

In all cases  $N = 30$  inner iterations were performed, in order to ensure very low residuals (e.g., for B1 configuration discussed later all residuals stayed below  $10^{-7}$  in absolute value). Although it would be advantageous to increase the number of revolutions, this is unfortunately not possible, as the detailed mesh and temporal resolution already lead to huge computational requirements. Additionally, 200 would be better suited for the solver Courant number, but this resulted in instability and divergence. With the current settings, an analysis of the coefficients revealed an appropriate convergence, and the results did not change significantly between Level 1, 2 and 3, see Figure 5.3. Thus, additional revolutions were not considered to be necessary for the computations. As the periodic signal generated by the simulation did not change significantly between Level 1-3, the mesh resolution seems to be appropriate as well.

Due to the huge computational requirements, mesh independency was not analyzed (Please note, that previous 2D simulations have indicated, that for complete mesh independency  $y^+ < 1$  would be required with a resolution  $s < 0.4$  mm on the surface of the

Table 5.1: Settings for mesh generation

Description	Value
base size in stator	0.12 m
size on the rotating interface	14 mm
size in the wake of the rotor	55 mm
size on the edge of straight blade	1.25 mm
target maximal $y^+$ value	100
base size in the rotating domain	14 mm
size around the blades	5.5 mm
size on the surface of the blades	2 mm
size near the trailing edge	0.5 mm
size around the wingtips	4 mm
size on the surface near the wingtips	0.5...0.75 mm
prism growth rate	1.3
volume growth rate	1.06
surface growth rate	1.075
volume blending	0.68
volume density	0.8
length of extruded inlet domain	13 m
length of extruded outlet domain	17 m
length of stator domain	4+6 m
width of the domain	12 m
height of domain	7.68 m

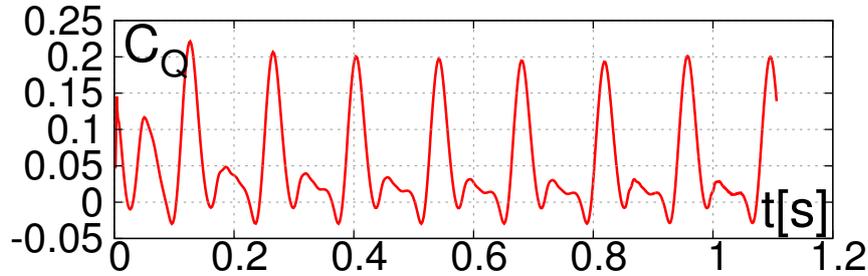


Figure 5.2: Torque coefficient for C1 configuration for a single blade

blades; this would require at least 100 million cells only in the boundary layer). Mesh sizes were calibrated so that the number of cells stays in the acceptable range for the available computational system (below 12 million polyhedral cells).

### 5.2.5 Validation of the model

The experimental validation relies on measurements from the literature. In the work of Castelli et al. [34] a small H-Darrieus rotor was tested. The rotor has NACA0021 blades

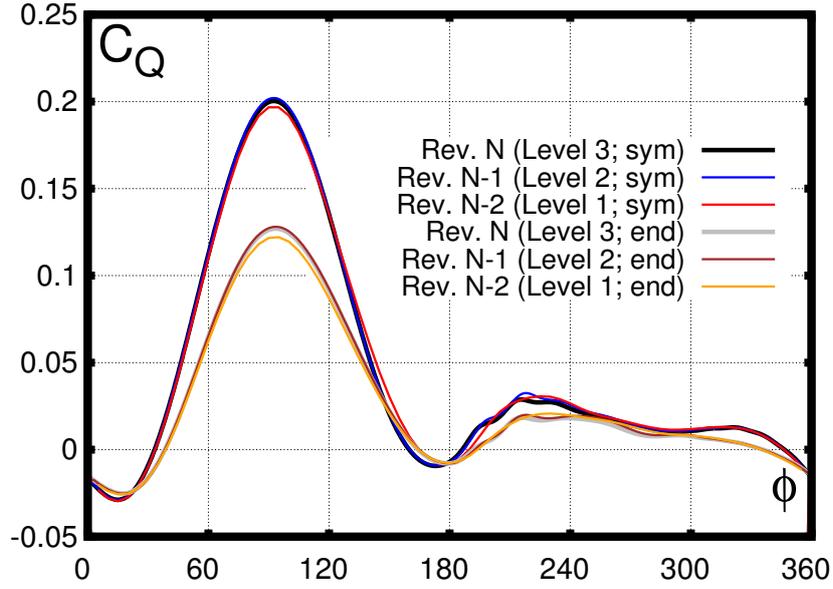


Figure 5.3: Torque coefficient for C1 configuration for a single blade for a single section near the symmetry and near the end (scaled by the projected area)

with 0.38 mm blunt edge,  $c = 85.8$  mm chord length and was mounted at  $0.5c$ . The radius and height were 1030 mm and 1414 mm respectively. The geometry is summarized in Table 5.2. The measurement was conducted in the Milan-Bovisa low turbulence wind tunnel with 4 m x 3.8 m cross-section at 9 m/s wind velocity. In the original experiment the results were not corrected for wind tunnel blockage. Castelli et al. [34] found that the best agreement was delivered by the  $k-\omega$  SST model, with  $y^+ > 30$  and a resolution of 2 mm on the surface of the blades.

Table 5.2: Geometry for validation & optimization

Parameter	Valid.	Optim.
number of blades	3	3
airfoil	NACA0021	NACA0021
radius of the rotor	515 mm	515 mm
camber	85.8 mm	85.8 mm
mounting	$0.5c$	$0.5c$
(half)-length of blades	707 mm	707+200 mm
wind speed	9 m/s	9 m/s
TSR	2.05/2.35/2.6/3.1	2.6

To reduce the mesh size, only half of the rotor and wind tunnel were retained. The mesh was created with the same setup as the optimization process itself (except the wall of the wind tunnel, where prism layers were added and no-slip condition was set), see Fig. 5.4 for the geometry.

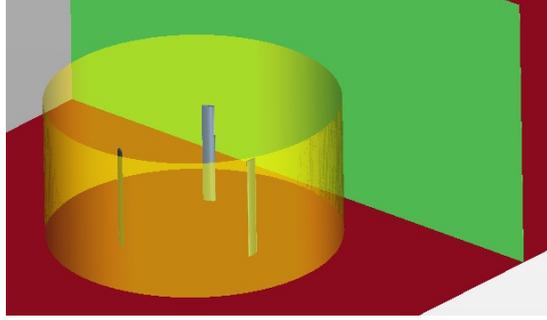


Figure 5.4: 3D geometry of experimental validation

The recommendation of Castelli et al. was retained, i.e.,  $y^+ > 30$  and 2 mm cell size on the blades. The resulting mesh contained 7.05 million polyhedral cells. If the boundary layer would be resolved up to  $y^+ < 0.8$  and the mesh would be refined with additional 30% size reduction, the resulting resolution would be 24 million polyhedral cells, leading to  $4\times$  longer computational times.

The simulations were performed with the  $k-\epsilon$  Realizable and  $k-\omega$  SST model for tip-speed-ratios for  $\lambda = 2.05$ ,  $\lambda = 2.35$ ,  $\lambda = 2.6$ ,  $\lambda = 3.1$  (near the optimal tip-speed-ratios; the curves between the operating points have been interpolated with cubic spline); the blunt edge was disregarded and modeled instead as being sharp (resolving such a small geometric detail would have required an even larger resolution making the optimization impractical). The characteristic curve of the complete rotor is compared in Fig. 5.5 to the measurement of Castelli et al.; this Figure also contains the characteristic curve far away from the wingtip, near the symmetry plane.

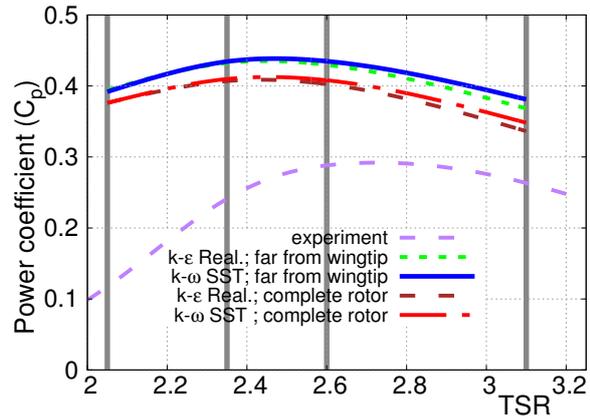


Figure 5.5: Experimental validation

As one can see, the agreement in the location of the optimal tip-speed-ratios is quite good (exp.:  $\lambda_{opt} = 2.7$ ; sim.:  $\lambda_{opt} = 2.5$ ), but the exact values are not reproduced. As struts were not modeled and gear losses were not computed, this is not really a surprise. Furthermore, the large deviation in the trend at small tip-speed-ratios is most likely due to the application of the Wall Function approach ( $y^+ > 30$ ). In this domain the flow is very complex due to the presence of dynamic stall and a finer grid resolution would be required.

The two turbulence models deliver very similar (but not identical) results. This is again not a surprise: the  $k-\omega$  SST model is a blended model, which falls back to  $k-\epsilon$  in the free stream and to  $k-\omega$  near the wall. Since  $y^+ > 30$  was employed, the benefits offered by the  $k-\omega$  model are replaced virtually with the Wall Function, except for the places, where the blade interacts with vortex structures shed from the previous blade,

thus, where  $y^+$  values can drop down locally. As a result, the  $k-\epsilon$  Realizable model was retained for all further computations.

## 5.3 Setup of the optimization

### 5.3.1 Parameterization

Winglets are essentially the composition of one or more wings, with (usually) blended center line and non-uniform cross-section. Thus, the geometry of the winglet can be defined based on several support sections and/or by varying all parameters along the span. This approach is described, e.g., in the work of Sobieczky [200], and can be applied to an arbitrary blade geometry.

#### 5.3.1.1 General parameterization

In the current study variation of parameters along the span were defined. The different geometrical parameters of the winglet are defined along a non-dimensional parameter ( $\psi = 0 \dots 1$ ). First, a guiding line (see Fig. 5.6(a)) of the winglet is defined by

$$\mathbf{x}(\psi) = (x(\psi), y(\psi), z(\psi)). \quad (5.1)$$

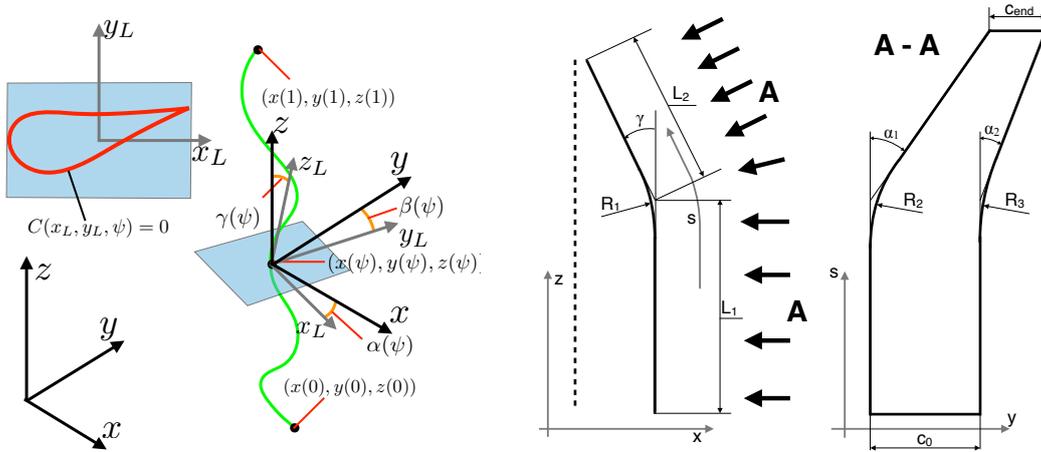


Figure 5.6: Parameterization (a) general; (b) specific variant

This is equivalent to the definition of the parameters along the span as the arc length can be simply computed as

$$s(\psi) = \int_0^\psi \sqrt{\left(\left(\frac{dx}{d\hat{\psi}}\right)^2 + \left(\frac{dy}{d\hat{\psi}}\right)^2 + \left(\frac{dz}{d\hat{\psi}}\right)^2\right)} d\hat{\psi}. \quad (5.2)$$

For each  $\psi$  a local coordinate system  $(x_L, y_L, z_L)$  is defined with  $(x(\psi), y(\psi), z(\psi))$  as center and with  $\alpha(\psi), \beta(\psi), \gamma(\psi)$  angle rotation around the  $x$ -,  $y$ - and  $z$ -axis, respectively. In this coordinate system, a profile is mounted on the  $(x_L, y_L)$  plane, defined by the

equation  $C(x_L, y_L, \psi) = 0$ . Using this method, the contour of the winglet is completely defined.

This parameterization is implemented in C++ and the centerline positions and profiles are exported at the  $\mathbf{x}(\psi_1)$ ,  $\mathbf{x}(\psi_2)$ , ...,  $\mathbf{x}(\psi_n)$  positions as support sections. StarCCM+ can create the surface of the winglets as a NURBS based on these data. Alternatively, one could produce the geometry as an STL surface using in-house code as well.

To illustrate the appropriate flexibility provided by the present model, a couple of winglets are presented in Fig. 5.7, as created by the current workflow. As one can see, the most common types of winglets can be described and fully parameterized.

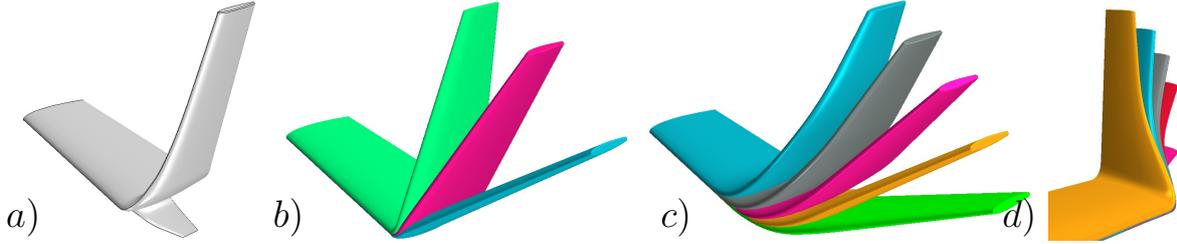


Figure 5.7: Example of different winglet designs: **(a)** Whitcomb winglet as a composition of two single centerline winglets; **(b)** Canted winglets; **(c)** Simple blended winglets; **(d)** Blended winglets with more complex leading edge

Actually, the present description is not only valid for winglets, but for any geometry which can be described as a geometry along a centerline (e.g., blades, pipes, channels, etc.)!

### 5.3.1.2 Specific parameterization

Although the previously presented (and widely used) parameterization is very flexible, it is too general to be used for an optimization (as all parameters are arbitrary functions). As a result, a simplified form was defined, see Fig. 5.6 (b). In this method the frontal projection of the centerline of the winglet is composed of a vertical segment with length  $L_1$  and an inclined segment with  $L_2$  length, the angle between the two segments is denoted as  $\gamma$ . The two sections are connected by a circular arc with radius  $R_1$ . Besides the frontal projection, the side view of the winglet is defined so, that its height is equal to the length of the frontal projection: this way the same form can be bended on the frontal projection independently from the size of  $\gamma$ . The side view is defined by the trailing and leading edge, which are both created by two linear segments connected by circular arcs. The lower camber has a length of  $c_0$ , the winglet ends with a camber of  $c_{end}$ .

In the current study  $R_{1,2,3}$ ,  $c_0$ ,  $c_{end}$ ,  $L_1$ ,  $L_2$  were fixed and  $\gamma$  and  $\alpha_2$  were varied.

### 5.3.1.3 Analyzed configurations

Following the setup of the workflow, 8 different configurations were created and meshed. 'Base' denotes a straight three-bladed rotor with  $H_{half}=707+200$  mm height, 'Endplate' denotes the 'Base' configuration equipped with a NACA0030 endplate ( $c = 150.15$  mm,  $r_T = 4$  mm rounding,  $t = 2$  mm thickness).

Configurations A1, A2, B1, B2, C1 and C2 denote winglets with length  $L_1 + L_2 = 707 + 200$  mm. For the transition  $R_{1,2,3} = 100$  mm was applied. For the cambers  $c_0 = 85.8$  mm,  $c_{end} = 85.8/3$  mm was retained, A, B and C denote respectively  $\alpha_2 = 0^\circ$ ,  $15^\circ$  and  $30^\circ$ . Numbers 1 & 2 denote configurations with  $\gamma = 60^\circ$  and  $\gamma = 30^\circ$ , respectively.

The configurations can be seen in Figure 5.8.

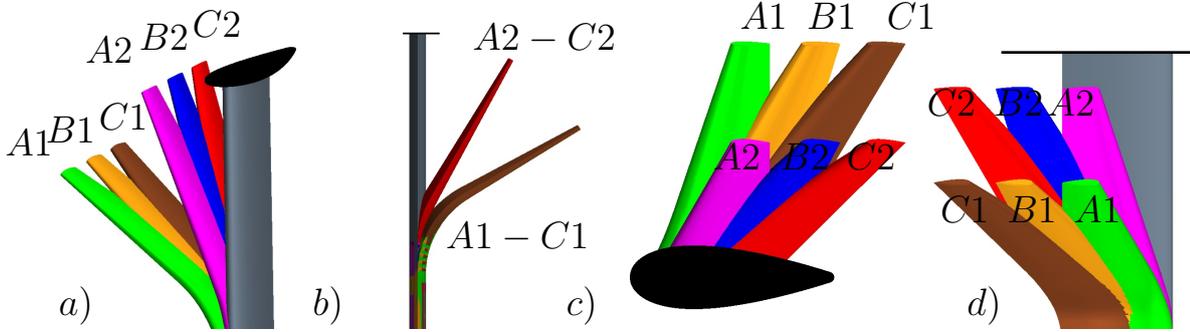


Figure 5.8: Tested configurations from (a) isometric; (b) front; (c) top; (d) side view (gray='base'; black='endplate'; red='C2'; blue='B2'; purple='A2'; brown='C1', yellow='B1'; green='A1')

### 5.3.2 Automated workflow

For the optimization of winglets, the whole process (mesh generation, CFD computation, post-processing) has to be automated. This was achieved using OPAL++. OPAL++ is able to operate in parallel on many PCs to speed up the optimization process. For a single configuration, the followings steps are performed:

1. The geometry is created using a custom C++ code (approximately 1200 lines), which is called by OPAL++ as an ODL (OPAL++ Dynamic Library); the surface geometry is exported in CSV format.
2. The geometry is read in CD-Adapco StarCCM+ and the surface is prepared using a NURBS surface. Afterwards, polyhedral mesh is generated. All operations are performed by a custom JAVA script (4000 lines).
3. Mesh quality is checked.
4. The simulation is started in parallel with CD-Adapco StarCCM+ (in the present study each simulation was performed with 24 cores using Intel(R) Xeon(R) E5-1650v3 3.5 GHz CPUs; each simulation required 1-2 weeks to finish). Upon completion, StarCCM+ exports the  $y^+$  values, forces and moments for the different parts of the rotor and for the whole rotor.
5. Post-processing is performed using a small bash script through OPAL++.
6. Results are collected in a Microsoft Excel file for all the different winglets.

## 5.4 Results

The direct comparison of the performance of the different rotors is not possible, as the projected area (see Table 5.3) and height of the rotor changes from configuration to

configuration. As a result, four different analysis were performed, as discussed in the followings.

### 5.4.1 Performance coefficient far away from the wingtip

In the first step the performance coefficient of the first 0.1 m blade section (Section 0) was analyzed, far away from the wingtip (near the symmetry plane). Although far away from the wingtip it is expected, that the different rotors will deliver a similar performance, this is not necessarily true. The wingtip vortex changes the flow structures near the end of the blades, and through 3D instabilities it can have an effect on the flow structure of the whole rotor, see Table 5.4 and 5.5. As one can see, the performance coefficients remain very similar to each other; only small differences can be seen. Nevertheless, more than 1% (relative) improvement is detected for the B2 and C2 configurations.

This can be either the effect of the decreased energy extraction of B2 and C2 configurations near the wingtip, which results in an increased energy flux, or could be a result of the different blockage, since the wingtips restrict the 3D expansion in the vertical direction, leading to an increased efficiency. For rotors with high aspect ratios (AR) this effect could provide an advantage.

Table 5.3: Projected area of the different configurations

	Proj. area [mm <sup>2</sup> ]
Base/Endplate	934210 mm <sup>2</sup>
[A/B/C]1	812814 mm <sup>2</sup>
[A/B/C]2	889168 mm <sup>2</sup>

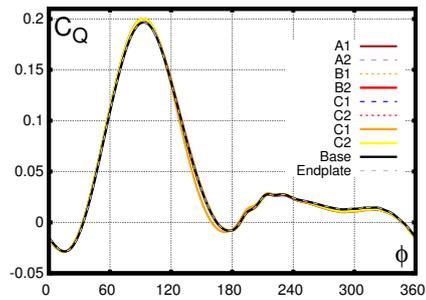


Figure 5.9: Instantaneous torque coefficients for all configurations for a single blade for a single section near the symmetry

The instantaneous torque coefficients for Section 0 (first 0.1 m near the symmetry plane) for a single blade are shown in Fig. 5.9. As one can see, the difference is small and for most angular positions they overlap. For configuration C1 (yellow), a small increase can be seen at the location of the maximal energy extraction ( $60^\circ < \phi < 120^\circ$ ), as expected from the  $C_p$  values.

Table 5.4: Performance coefficients far away from the wingtips

	$C_p^{first}$ [%]	Improv.
Base	34.43%	–
Endplate	34.45%	0.058%
A1	34.54%	0.319%
A2	34.65%	0.639%
B1	34.71%	0.813%
B2	34.81%	1.103%
C1	34.42%	–0.029%
C2	34.89%	1.336%

Table 5.5: Performance coefficients for the whole rotor

	$C_p^{rotor}$ [%]	Improv.
Base	32.20%	–
Endplate	32.40%	0.62%
A1	31.38%	–2.55%
A2	31.33%	–2.70%
B1	31.48%	–2.24%
B2	31.03%	–3.63%
C1	31.09%	–3.45%
C2	29.97%	–6.93%

### 5.4.2 Performance coefficient of the complete rotor

In the second step the total performance coefficient of the rotor was analyzed, non-dimensionalized by the projected area for each case. The results can be seen in Table 5.5. As one can see, instead of the expected improvement, the winglets provide a decrease in the performance for the current case. However, in case of the endplate variant, a small improvement can be seen. The improvement is small, showing that the wingtip losses do not seem to be that significant in this case. Obviously, this statement does not necessarily hold for rotors with more significant wingtip effects.

### 5.4.3 Effect of the wingtip on the distribution of the energy generation

Afterwards, the energy generation of the different sections were analyzed and compared graphically. The results can be seen in Fig. 5.10. Section 0 denotes the first 0.1 m near the symmetry plane (far away from the wingtip), while Section 6 denotes the last section (0.6 m - to the end of the blade). All results are non-dimensionalized by the projected area of the section. As one can see, all configurations resulted in a decreased performance coefficient for the last section compared to the base geometry, except the endplate. This supports the importance of endplates. The decreased performance of the winglets is most probably the result of the decreased camber length and rotor radius. For an efficient improvement, smaller winglet configurations or rotors with more significant wingtip losses would be needed.

A2 proved to be the best configuration among the winglets for Section 6. The B2 and C2 configurations provided an increased performance coefficient far away from the wingtip, as discussed previously.

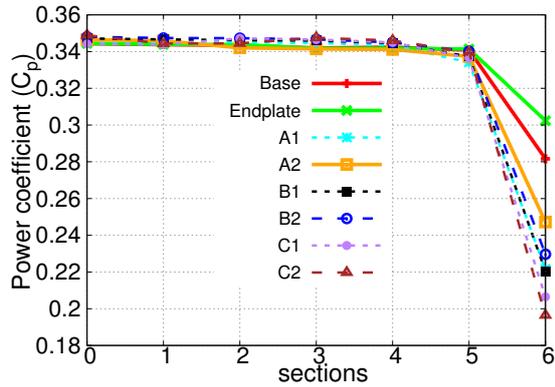


Figure 5.10: Performance coefficients of the different blade sections for the different configurations

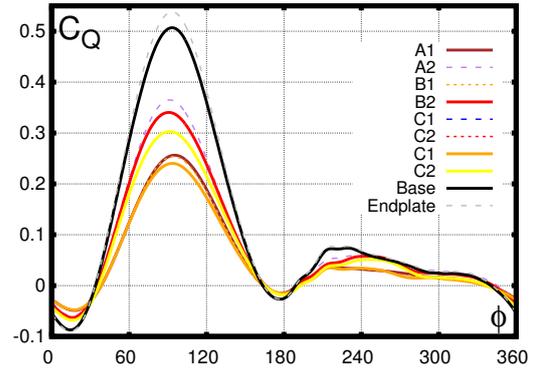


Figure 5.11: Instantaneous torque coefficients for all configuration for a single blade for a single section near the end

The instantaneous torque coefficients for Section 6 (last section of the blade far away from the symmetry plane) of blade 1 are shown in Fig. 5.11. Please note, that  $C_Q$  values are somewhat misleading, as all values were normalized for a blade section of 0.1 m height and nominal radius, while in the reality they have different heights. Unfortunately, no clear tendency can be recognized.

#### 5.4.4 Size of wingtip vortex

Finally, the vortex structures were compared for a single angular position for all the different cases ( $\varphi = 0^\circ$ , starting position), see Fig. 5.12. In this figure the isosurface for  $75 \frac{1}{s}$  vorticity magnitude is presented along with the vorticity magnitude in two cross-sections. As one can see, A1 and B1 configurations efficiently reduced the size of the vortices, while C1 and C2 provided similar sizes to the base and endplate configuration. Additionally, a threshold was created for  $75 \frac{1}{s}$  vorticity magnitude and the volume of this threshold was computed to estimate the size of the vortex structures. The results are shown in Table 5.6. The values further confirm, that A1 and B1 provide the smallest vortex volumes, while C2 shows similar values to the base configuration. Interestingly, the rotor with the endplate did not show a decrease in the size of the vortex.

As noise emission is directly related to the vortex shedding, this effect could be efficiently used to reduce noise emissions from the blades. For this however, additional acoustics computations are necessary.

Table 5.6: Volume of fluid domain with vorticity magnitude larger than  $75 \frac{1}{s}$

	$V_{75}^{vortex} [m^3]$
Base	$4.437 \cdot 10^{-2}$
Endplate	$4.452 \cdot 10^{-2}$
A1	$2.706 \cdot 10^{-2}$
A2	$2.910 \cdot 10^{-2}$
B1	$2.903 \cdot 10^{-2}$
B2	$3.197 \cdot 10^{-2}$
C1	$3.298 \cdot 10^{-2}$
C2	$3.650 \cdot 10^{-2}$

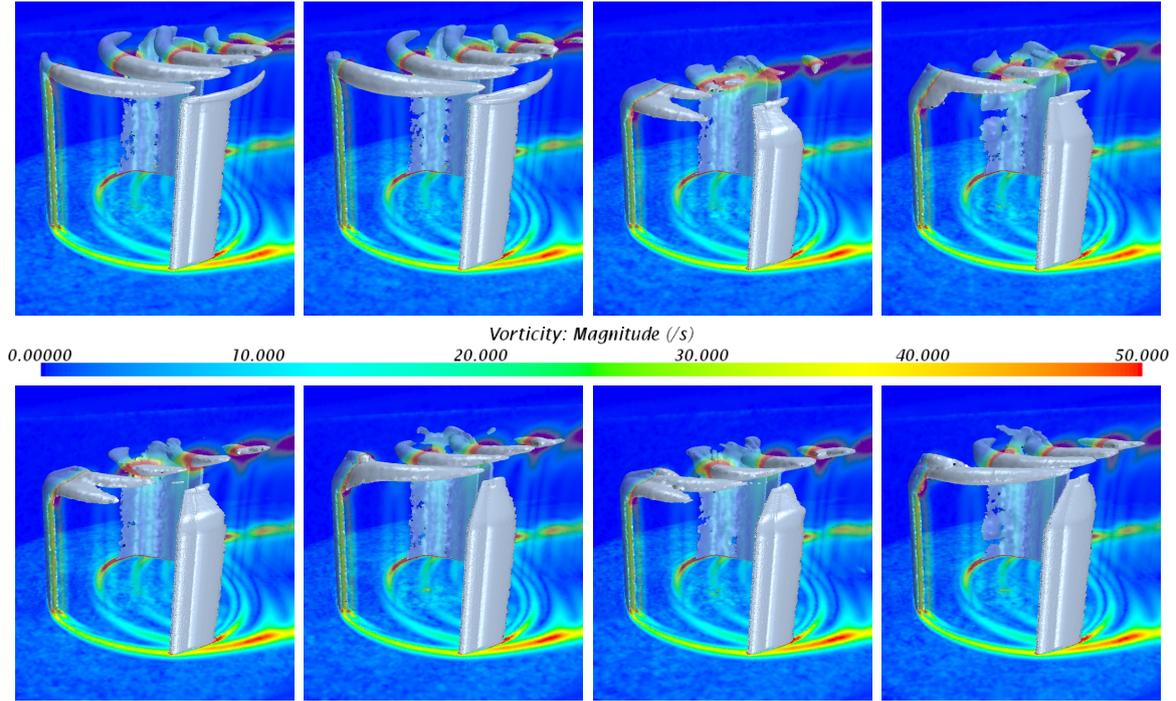


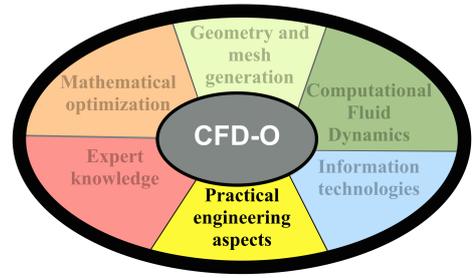
Figure 5.12: Vortex structures near the configurations (first line: base, endplate, A1, A2; second line: B1, B2, C1, C2; isosurface: 75 1/s)

## 5.5 Conclusions and summary

In the current chapter a completely automatic method was developed and presented for the efficient optimization of 3D Darrieus rotor blades and winglets. Parameterization, geometry and mesh generation, CFD simulation are efficiently realized in a fully automatic way.

Following a tentative validation of the model using a rotor from the literature, a Design-Of-Experiment was executed with 8 different winglet configurations to check the validity of the developed approach. Unfortunately, no configuration was found that leads to a performance improvement compared to the rotor with a simple endplate. However, two configurations show an increased performance far away from the wingtips, which could be advantageous for rotors with high aspect ratio. To understand this effect, further studies are needed.

*After providing this example, a completely new field, the question of robustness and reliability follows. The key for robust and reliability-based optimizations is Uncertainty Quantification (UQ), as one has to be able to quantify the effect of unknown (or in a better formulation, uncertain) processes. This field is alone a huge research field; it is out of the scope of the present thesis to present and discuss all details. Therefore, only one of the newest methods, Polynomial Chaos Expansion (PCE) will be discussed. Following a theoretical introduction and validation of the implementation, the method will be applied to the same problem as used in the previous chapters, i.e., to the transient turbulent flow around an H-Darrieus wind turbine.*



## Chapter 6

# Question of robustness - Uncertainty Quantification

“True genius resides in the capacity for evaluation of uncertain, hazardous, and conflicting information.”

---

*Winston Churchill*

*The present chapter will especially focus on a Practical Engineering aspect of optimization, i.e., dealing with stochastic and uncertain processes. For this, Uncertainty Quantification (UQ) of an exemplary H-Darrieus rotor will be presented using CFD computations. The chapter is based on an extended version of:*

DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Analysis of the performance of a H-Darrieus rotor under uncertainty using Polynomial Chaos Expansion. *Energy* 113 (2016), 399–412 [59].

### 6.1 Aim of the analysis

Optimization is a very important field of the industrial engineering practice, but for practical purposes, optimization in itself is not enough. The improved design has to retain its performance under various and uncertain conditions, i.e., the optimization must deliver a robust solution (robustness will be mathematically defined a bit later). Robustness is receiving an increased interest in the last years [190]. In other words, if *a product is superior compared to other products on the market, but 50% of the products are defect, the product still won't succeed*. The reason for such defects is very simple: there are many uncertainties present in the manufacturing process (e.g., dimensions may slightly vary). This can be presented graphically even better, see Fig. 6.1.

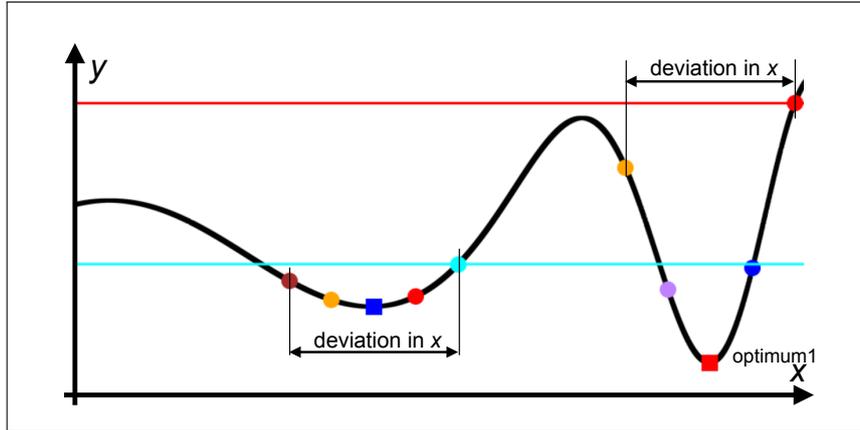


Figure 6.1: Graphical presentation of sensitive and robust optimum

As one can see, there are two local optima (rectangular symbol) for the present case, the right one (red rectangle) being superior. However, if the variable exhibits a deviation, the same deviation can result in locations (circular symbols), which are worse, than for the left local optimum (blue rectangle), i.e., *the optimum on the right is sensitive*.

In the present chapter the Uncertainty Quantification of an H-Darrieus rotor will be performed, using Polynomial Chaos Expansion (PCE), as the question of uncertainties is particularly important for wind turbines due to many sources of uncertainties, e.g., inflow conditions [131] or airfoil geometry [1]. Up to now, only very few studies addressed the question of robustness, mostly considering HAWTs. E.g., Liu et. al. [143] used probabilistic collocation and CFD simulations to study the effect of uncertainty associated to the inlet velocity on the performance coefficient, performance distribution and forces acting on HAWT blades, while Padron et al. used a multi-fidelity UQ method to study the effect of extreme gust uncertainties on the maximum forces acting on the blades of a large (54 m radius) Darrieus rotor by combining 2D Euler simulations and a BEM/vortex lattice solver [174].

In what follows, first, a short introduction will be given to Uncertainty Quantification and Polynomial Chaos Expansion, followed by the description and analysis of the present rotor.

## 6.2 Uncertainties and Polynomial Chaos Expansion

### 6.2.1 Sources of uncertainty

Although in most engineering computations informations (from dimensions to material properties) are considered as having a fixed and perfectly known value, in reality most parameter values are associated to some uncertainty. Dimensions vary throughout the manufacturing process, inflow conditions (temperature, density, concentrations) can change, not to mention unknown material properties or varying operating conditions due to an imprecise control. Thus, uncertainties originate from the most different sources, such as model coefficients, material properties, boundary conditions, operating conditions, dimensions, etc.

Uncertain parameters can be classified as

- 1) *aleatoric* (i.e., being the result of a stochastic process, e.g., varying material properties), which are usually irreducible, i.e., stochastic from their own nature, or
- 2) *epistemic* (result of a lack of knowledge; e.g., wrong model assumptions), which are reducible; they could be quantified with a higher accuracy by collecting more information [24].

In this study, only aleatoric uncertainty will be considered.

## 6.2.2 Quantification of uncertainty

A large area of uncertainty quantification (UQ) is interested in forward uncertainty propagation, i.e., analyzing the effect of uncertain input variables on the outcome. Note that uncertainty can also address mesh dependency or temporal dependency [185]; however, in this case there are no uncertain input variables. Another field is to address inverse uncertainty quantification.

UQ can be applied in optimization as well. It is essential for Robust Design Optimization (RDO, for which the variance is additionally minimized) and Reliability-Based Design Optimization Methods (RBDO, where the failure rate is constrained). Mathematically formulated (if the expectation is denoted as  $E(\cdot)$ , the variance with  $\sigma^2(\cdot)$ , the random input variables with  $\boldsymbol{\xi}$  and the objective function vector with  $\mathbf{y}$ ):

$$\mathbb{O}^{\text{robust}} : \begin{cases} E(\mathbf{y}(\mathbf{x}, \boldsymbol{\xi})) \xrightarrow{\mathbf{x}} \min \\ \sigma^2(\mathbf{y}(\mathbf{x}, \boldsymbol{\xi})) \xrightarrow{\mathbf{x}} \min \\ \text{so that } \mathbf{x} \in \mathbb{X} \end{cases} \quad (6.1) \quad \mathbb{O}^{\text{reliability}} : \begin{cases} E(\mathbf{y}(\mathbf{x}, \boldsymbol{\xi})) \xrightarrow{\mathbf{x}} \min \\ P(\text{failure}) < 5\% \\ \text{so that } \mathbf{x} \in \mathbb{X} \end{cases} \quad (6.2)$$

For example, an uncertainty handling method is used in an Evolutionary Algorithm (EA) in the work of Hansen et al. [108] for the online optimization of gas combustors, while Wang et al. [222] used non-intrusive Polynomial Chaos Expansion (PCE) for the robust optimization of compressor rotor blades.

## 6.2.3 Polynomial Chaos Expansion

Several methods exist for UQ, e.g., Monte-Carlo sampling, or univariate reduced quadrature methods, which have been for instance used for wind turbines in [31]. In the following we will only focus on the Polynomial Chaos Expansion (PCE).

The basic idea of Polynomial Chaos Expansion is the separation of the solution into a stochastic and a deterministic part. Let us consider a probability space  $(\Omega, \sigma, P)$ , consisting of a sample space  $(\Omega)$ , of a  $\sigma$ -algebra on  $\Omega$ , and of  $P$ , probability measure on  $(\Omega, \sigma)$  [164]. Given the independent random input variables  $(\boldsymbol{\xi})$ , any random variable with a finite variance (or output, written  $R$ ) can be decomposed into deterministic  $\alpha_i(\mathbf{x})$  components (or mode strengths) and random basis functions  $(\Psi_i(\boldsymbol{\xi}))$ , multivariate polynomials) [112]:

$$R = \alpha_0(\mathbf{x})\Psi_0 + \sum_{i=0}^{\infty} \alpha_i(\mathbf{x})\Psi_1(\xi_i) + \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \alpha_{i,j}(\mathbf{x})\Psi_2(\xi_i, \xi_j) + \dots = \sum_{i=0}^{\infty} \alpha_i(\mathbf{x})\Psi_i(\boldsymbol{\xi}) . \quad (6.3)$$

In the multi-dimensional case, with  $n$  (independent) variables, the random basis functions can be computed as a product of the corresponding one-dimensional basis functions:

$$\Psi_i(\boldsymbol{\xi}) = \prod_{j=1}^n \Psi_j^{m_j^i}(\xi_j) , \quad (6.4)$$

i.e., the multivariate polynomial basis is the product of the univariate optimal polynomial basis for the distribution of each uncertainty, where

$$\sum_{j=1}^n m_j^i = p_i; \quad \deg(\Psi_j^{m_j^i}(\xi_j)) = m_j^i , \quad (6.5)$$

i.e.,  $p_i$  is the total order of  $\Psi_i(\boldsymbol{\xi})$ .

$\Psi_j^0(\xi_j), \Psi_j^1(\xi_j), \Psi_j^2(\xi_j), \dots$  are the basis functions for variable  $j$  with order  $0, 1, 2, \dots$  and  $m_j^i$  is the order of variable  $j$  in the  $i$ -th multivariate basis function. In practice, evaluation up to infinite polynomial degree is obviously impossible. Thus, the summation is truncated in practice to a total order of  $p$  ( $p_i \leq p \forall i$ ), resulting in

$$P + 1 = \frac{(n + p)!}{n!p!} \quad (6.6)$$

terms, where  $n$  is the number of random variables,  $p$  is the total order of the polynomial and  $P$  is the number of polynomial coefficients. An alternative, not considered further in what follows, is to perform the truncation on a per-dimension basis [75]:

$$m_j^i \leq q \forall i, j \quad (6.7)$$

This computation of the random basis functions is valid for independent random variables, but is only an approximation for uncorrelated, but not independent non-Gaussian variables [75]. Multi-variate weight functions (distributions) can be computed similarly.

In the original formulation of Wiener about the theory of homogeneous chaos [224], only unbounded independent Gaussian input variables ( $\mathcal{N}(0, 1)$ ) were supported using Hermite polynomials. Later, this was extended with the Wiener-Askey scheme to different distributions and basis functions (Generalized Polynomial Chaos), as described e.g., by Xiu and Karniadakis [229]. These polynomials satisfy orthogonality to the weight functions, i.e.,

$$\langle \Psi_i(\boldsymbol{\xi}), \Psi_j(\boldsymbol{\xi}) \rangle = \langle \Psi_i^2(\boldsymbol{\xi}) \rangle \delta_{ij} , \quad (6.8)$$

where  $\delta_{ij}$  is the Kroenecker-delta and the inner product  $\langle \cdot, \cdot \rangle$  is defined as:

$$\langle \Psi_i(\boldsymbol{\xi}), \Psi_j(\boldsymbol{\xi}) \rangle = \int_{S_{\boldsymbol{\xi}}} \Psi_i(\boldsymbol{\xi}) \Psi_j(\boldsymbol{\xi}) w(\boldsymbol{\xi}) d\boldsymbol{\xi} , \quad (6.9)$$

where  $w(\boldsymbol{\xi})$  is the weight function (probability distribution). Based on the properties of Equations (6.8) and (6.9), the coefficients can be evaluated as:

$$\langle \Psi_j(\boldsymbol{\xi}), R \rangle = \sum_{i=0}^P \langle \Psi_j(\boldsymbol{\xi}), \alpha_i(\mathbf{x}) \Psi_i(\boldsymbol{\xi}) \rangle , \quad (6.10)$$

$$\alpha_j(\mathbf{x}) = \frac{\langle \Psi_j(\boldsymbol{\xi}), R \rangle}{\langle \Psi_j^2(\boldsymbol{\xi}) \rangle}. \quad (6.11)$$

This is called Galerkin projection [74]. Mean value  $\mu$  and deviation  $\sigma$  can be determined due to the orthogonality as:

$$\mu_R = \alpha_0, \quad \sigma_R^2 = \sum_{i=1}^P \alpha_i^2 \langle \Psi_i^2(\boldsymbol{\xi}) \rangle. \quad (6.12)$$

However, only classical distributions are supported using the Wiener-Askey scheme. For other distributions, non-linear transformations can be used, but this results in a degradation of the convergence rate [75]. Alternative numerical methods exist to compute orthogonal polynomials for an arbitrary distribution [95, 101]. Witteveen and Bijl showed that using Gram-Schmidt orthogonalization the optimal polynomial basis can be computed with better accuracy and convergence [227].

## 6.2.4 Evaluation of Polynomial Chaos Expansion (PCE)

When handling uncertainties, two different approaches exist.

- For *intrusive* methods the original code has to be modified, i.e., the original system of equations for the model has to be extended to include stochastic effects, resulting in a (larger) deterministic system of equations. This requires usually a large effort, but the problem has to be solved only once (it is thus much faster).
- When relying on *non-intrusive* methods the original code does not have to be modified at all, but the simulation must be executed several times. The implementation of latter methods is usually easier, but leads to higher computational efforts. However, non-intrusive methods can be efficiently applied even to complex, coupled systems without huge implementation effort; they can be also used with proprietary or commercial software.

As a consequence, only non-intrusive techniques will be discussed in the followings, as all simulations are based on a commercial software, CD-Adapco StarCCM+ [37].

The integration for Eq. (6.11) can be for instance performed using a Monte-Carlo approach, quadrature methods, Smolyak sparse grids, or linear regression (also called stochastic collocation). According to Eldred et al. [74], “cubature is the most affordable for the largest dimensionalities and expansion order”. Considering that in the present study only two input uncertainties are analyzed, only tensor-product quadrature (TP) and point collocation (PC) methods will be employed.

### 6.2.4.1 Point collocation

With Point Collocation (also named stochastic response surfaces or linear regression method), the polynomial expansion is essentially used as a response surface built using  $N$  support points, at which the responses are computed. As a result, Equation (6.3) is rewritten for the  $N$  points into a matrix form:

$$\boldsymbol{\Psi}\boldsymbol{\alpha} = \mathbf{R}. \quad (6.13)$$

As  $N$  might be larger than the number of coefficients, the equation must be solved using a least-squares method based on the response vector ( $\mathbf{R}$ ) and on the matrix of basis function values ( $\Psi$ ) at the known points. Finally, the coefficients can be determined following [207] as:

$$\boldsymbol{\alpha} = (\Psi^T \Psi)^{-1} \Psi^T \mathbf{R} . \quad (6.14)$$

According to [112], optimal results are obtained with an oversampling of 2, i.e., for total order expansion

$$N = 2 \frac{(n+p)!}{n!p!} \quad (6.15)$$

points, using Latin Hypercube Sampling (LHS) for smoother convergence. Essentially, UQ creates a response surface for the input variables using Least Squares fitting. This also means, that using the collocation method the transformation can be efficiently re-computed for another input distribution, as long as the probability space remains the same [128]. Also, multi-fidelity approaches can be applied [167].

#### 6.2.4.2 Quadrature method

Equation (6.12) can be also computed based on numerical integration with Gaussian quadratures. Starting from the one-dimensional case and assuming that the domain of the weight function  $w(x)$  is  $[a, b]$ , the integral

$$\int_a^b f(x)w(x)dx = \sum_{i=1}^m w_i f(x_i) , \quad (6.16)$$

will be exact up to polynomial order of  $2m - 1$ . To integrate the product of  $R$  and  $\Psi$ ,  $m = p + 1$  is required for accuracy. Finally,  $w_i$  are the weights and  $x_i$  are the abscissas with respect to the  $w(x)$  weight function. For the different basis (weight) functions, different abscissas and weights are needed (e.g., Gauss-Hermite, Gauss-Legendre, etc.). The abscissas can also be computed numerically. In the present study, the Golub-Welsch algorithm was applied to compute the Gaussian quadratures [101]. In the multi-dimensional case the weights and abscissas can be computed with a single tensorial product of the one-dimensional cases, creating a multi-dimensional grid. For the computation,  $(p + 1)^n$  simulations have to be performed. The number of necessary points for the two methods are compared in Table 6.1. The more efficient method (i.e., smaller number of evaluations) is highlighted with green in each case.

Table 6.1: Number of necessary computations for UQ with PC/TP

PC/TP	n=1	n=2	n=3	n=4
$p = 1$	4/2	6/4	8/8	10/16
$p = 2$	6/3	12/9	20/27	30/81
$p = 3$	8/4	20/16	40/64	70/256
$p = 4$	10/5	30/25	70/125	140/625
$p = 5$	12/6	42/36	112/216	252/1296
$p = 6$	14/7	56/49	168/343	420/2401

As one can see, quadrature based methods are efficient for small problems with limited number of input variables, but suffer from the curse of dimensionality, while Point Collocation is more efficient for larger problems.

## 6.2.5 Polynomial Chaos Expansion in CFD

Application of uncertainty quantification methods in CFD is not new. Therefore, many detailed descriptions are available. Knio and Maître [128], and Najm [164] reviewed both intrusive and non-intrusive polynomial chaos expansion methods for CFD applications, while Xiu and Karniadakis [229] presented the application of the Wiener-Askey polynomial chaos for stochastic differential equations, using an intrusive approach.

A non-intrusive Polynomial Chaos Expansion (NIPCE) method was applied by He et al. [109] for the analysis of the Delft Catamaran 372 with variable geometry, speed, and head waves. Non-Intrusive UQ methods (implemented in PSUADE) were applied to a multiphase CFD problem by Gel et al. [97] using a Response Surface Method to leverage the computational burden of function evaluation. Hosder et al. [113] analyzed using a non-intrusive PC method the laminar boundary layer flow over a flat plate, the 2-D supersonic flow over a wedge and the flow over a three-dimensional wing [112]. NIPCE was applied by Hosder and Maddalena [111] around a three-dimensional supersonic pressure probe using an Euler CFD solver. Zhao et al. [124] used PCE for the robust design of a supercritical airfoil, while Loeven and Bijl applied probabilistic collocation for the flow around a NACA0012 profile with 8 uncertain parameters [144].

## 6.2.6 Workflow of non-intrusive Polynomial Chaos Expansion in CFD

In the present section, the workflow for the selected PCE approach (with Gram-Schmidt orthogonalization) is summarized in Fig. 6.2.

In this workflow, large sections (I, II) are completely the same, only small changes appear depending, if one chooses to use TP or SC method. First (I), the problem has to be analyzed to determine the distribution of the input variables. This is a very difficult step for many studies due to the lack of information. Furthermore, the variables have to be independent. If classical distributions are chosen, the orthogonal polynomials can be chosen from tables, otherwise, Gram-Schmidt orthogonalization has to be executed, which delivers the necessary polynomials. In the second step (II), sampling points have to be chosen, where the output quantities will be evaluated. With stochastic collocation (SC), one has to create simply a Latin Hypercube Sample, with an oversampling rate of 2 (II.1b), according to the recommendation of Hosder et al. [112]. For Tensor Product method (TP), one has to compute the Gauss-quadratures based on the Golub-Welsch method (II.1a); the coordinates are simply the combination of the quadratures for each dimension (II.2a).

After this step, the output quantities have to be computed based on CFD simulations (III). This is obviously the most time-consuming part. There can be more than one quantity of interest.

Finally, the coefficients have to be computed. With SC, one needs a simple Least-Squares Fitting, as defined in Eq. 6.14, while for TP a Galerkin-projection (thus a numeric integration) has to be executed, based on Eq. 6.11. If the coefficients are known, the most important quantities, i.e., expected value and standard deviation can be evaluated based on Eq. 6.12, while Cumulative Distribution Function and Probability Distribution Function can be reconstructed using e.g., Monte-Carlo simulation.

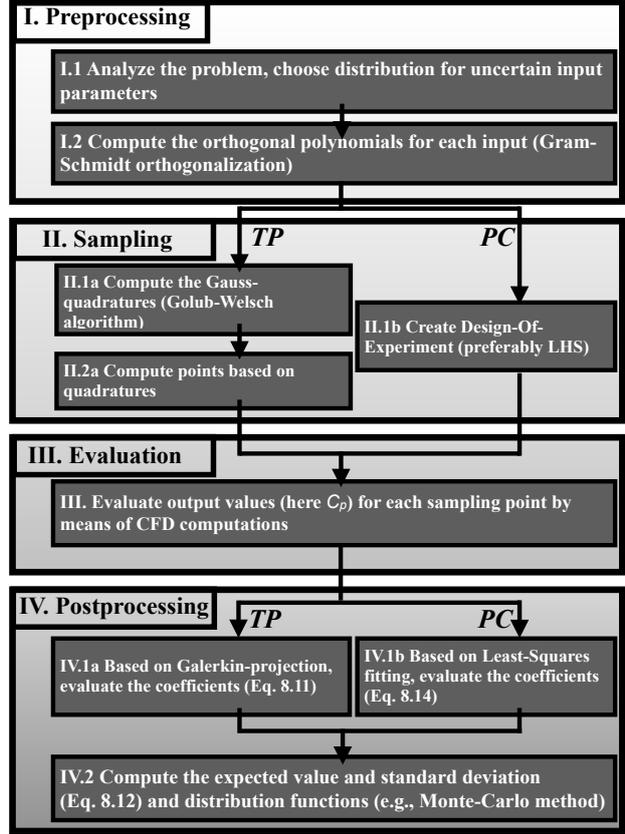


Figure 6.2: Workflow of PCE based on Gram-Schmidt orthogonalization with TP/SC method

## 6.3 Validation of the implementation

### 6.3.1 Validation of the implemented numerical integration method

In order to test the implementation of the numerical integration and of the orthogonalization routines, an analytical example was analyzed in the first step. A single random variable was chosen as input ( $\xi$ ), which was truncated from a standard normal variable to the  $[-2\sigma; 3\sigma]$  domain. The polynomial

$$f(x) = \sum_{i=0}^7 0.5(i+1)x^i \quad (6.17)$$

was retained for testing purposes, the distribution function of  $\xi$  was denoted as  $w(x)$ . The integral

$$S = \int_{-2}^3 f(x)w(x)dx \quad (6.18)$$

was first evaluated by using the Simpson-formula for 100 000 intervals, resulting in  $S = 78.9531$ . In the second step, quadratures and weights were computed for  $p = 4$

order resulting in 5 points (using these quadratures integration up to order of 7 should be exact). Using the numerical integral  $S_{num} = \sum_{i=0}^p w_i f(x_i) = 78.9531$ , exactly the same value was obtained. The Gauss quadratures and weights for the present weight function were computed using the Golub-Welsch method [101].

### 6.3.2 Validation of the implemented generalized PCE method

In order to validate the chosen approach for the rotor, an analytical problem with a similar response was first analyzed. Two input variables were chosen ( $\xi_1, \xi_2$ ). A test function was retained, showing a similar evolution to the characteristic curve of common wind turbines, centered around the maximal performance coefficient. It is not a direct polynomial function, so that the PCE method can be efficiently tested:

$$f(x, y) = 0.6 \cos(0.1x^2 + 0.1y^2)^2 - 0.05(x^2 + y^2) - 0.1 \quad . \quad (6.19)$$

The function is shown in Figure 6.3.

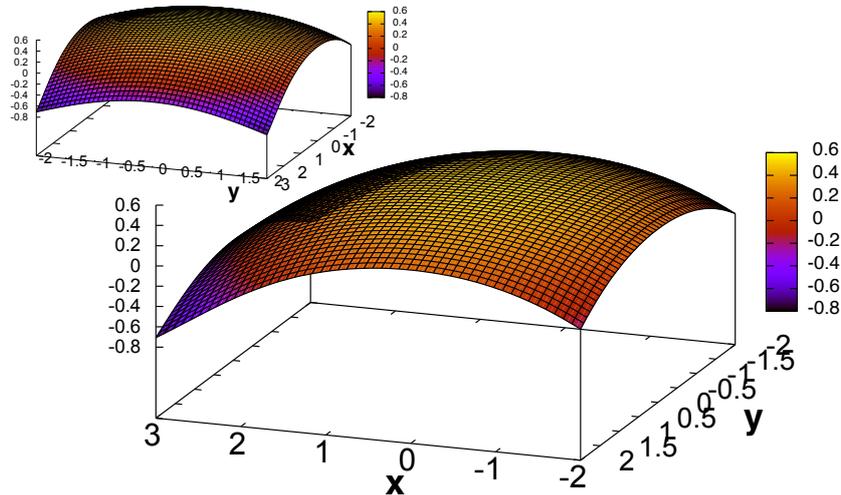


Figure 6.3: Function used for analytical tests

For the Tensor-Product quadrature integration (further denoted simply TP) the two-dimensional Gaussian quadratures and weights were computed with the Golub-Welsch method, and orthogonal polynomials were computed numerically as well. Cumulative distribution functions (CDF) and probability distribution functions (PDF) were evaluated using a Monte-Carlo simulation with 10 million samples in each case (for the exact solution, and for the reconstructed response  $R = \sum_i \alpha_i(\mathbf{x})\Psi_i(\boldsymbol{\xi})$  as well).

For Stochastic Collocation, if not stated otherwise, near-orthogonal Latin Hypercube Sampling (LHS) was used in the C-space, as computed by OPAL++.

In the first step, the input variables were chosen to be truncated from a standard normal variable ( $\mathcal{N}(0, 1)$ ) with:

$$\xi_1 \in [-2\sigma; 3\sigma] ; \quad \xi_2 \in [-2\sigma; 2\sigma]. \quad (6.20)$$

### 6.3.2.1 Exemplary computation with TP

For the present case, for polynomial order  $p = 2$  and  $p = 3$  the method is presented in detail using Tensor Product Quadratures for evaluation, following the workflow as presented in Fig. 6.2. As the input variables (I.1 in Fig. 6.2) were already chosen, the next step (I.2 in Fig. 6.2) is to evaluate the orthogonal polynomials. First, the one-dimensional polynomials are computed using the Gram-Schmidt method in a numerical manner. As a result, the polynomials  $\Psi_1^0(\xi_1)$ ,  $\Psi_1^1(\xi_1)$ ,  $\Psi_1^2(\xi_1)$ ,  $\Psi_1^3(\xi_1)$ , ... and  $\Psi_2^0(\xi_2)$ ,  $\Psi_2^1(\xi_2)$ ,  $\Psi_2^2(\xi_2)$ ,  $\Psi_2^3(\xi_2)$ , ... are obtained as the orthogonal polynomials with order 0, 1, 2, 3, ... for  $\xi_1$  and  $\xi_2$ , respectively. The multivariate polynomials can be computed by multiplying the corresponding one-dimensional polynomials, as already stated in Eq. (6.5). According to Eq. (6.6), 6 ( $\Psi_1(\xi_1, \xi_2) \dots \Psi_6(\xi_1, \xi_2)$ ) and 10 ( $\Psi_1(\xi_1, \xi_2) \dots \Psi_{10}(\xi_1, \xi_2)$ ) polynomials will be needed for  $p = 2$  and  $p = 3$  total order. For  $p = 2$ , one has to use the combinations  $(0 \times 0)$ ,  $(1 \times 0)$ ,  $(0 \times 1)$ ,  $(2 \times 0)$ ,  $(1 \times 1)$ ,  $(0 \times 2)$ , as shown with black color in Table 6.3. The numbering used in the present study is arbitrary, one could use a different one as well. For  $p = 3$ , the combinations  $(3 \times 0)$ ,  $(2 \times 1)$ ,  $(1 \times 2)$ ,  $(3 \times 0)$  have to be added as well, as shown with green color in Table 6.3. Having computed all multivariate polynomials, the preprocessing part is finished.

In the next step, the Gauss-quadratures with the corresponding weights are computed for each order using the Golub-Welsch method (Step II.1a in Fig. 6.2). For variable  $\xi_1$  and  $\xi_2$  and order  $p$ , the weights are denoted as  $w_0^{j,p} \dots w_p^{j,p}$  and the abscissas as  $x_0^p \dots x_p^p$  and  $y_0^p \dots y_p^p$ , respectively ( $p + 1$  values are required always). Please note, that  $x_i^{p_1} = x_i^{p_2} \iff p_1 = p_2$ , i.e., the quadratures and abscissas are different for different orders. Furthermore, as  $\xi_1$  and  $\xi_2$  have different distributions,  $x_i^p \neq y_i^p$ ,  $w_i^{1,p} \neq w_i^{2,p}$ . At last, the coordinates for the sampling points are determined (II.2a in Fig. 6.2).

This is simply done by combining the abscissas of the first variable as the first coordinate, with the abscissas of the second variable as second coordinate, in each possible combination, resulting in 9 combinations for  $p = 2$  and 16 for  $p = 3$ , as presented in Table 6.2. Please note, that as the abscissas are different, different sampling points have to be used.

Table 6.2: Sampling points for  $p = 2$

$(\xi_1, \xi_2)$	$y_0^2$	$y_1^2$	$y_2^2$
$x_0^2$	$(x_0^2; y_0^2)$	$(x_0^2; y_1^2)$	$(x_0^2; y_2^2)$
$x_1^2$	$(x_1^2; y_0^2)$	$(x_1^2; y_1^2)$	$(x_1^2; y_2^2)$
$x_2^2$	$(x_2^2; y_0^2)$	$(x_2^2; y_1^2)$	$(x_2^2; y_2^2)$

In the third step (III in Fig. 6.2), the function has to be evaluated at the sampling points. Although not very demanding in the present case, by CFD a single evaluation could require several days. Thus, for the present case the values  $f(x = x_i^p, y = y_j^p)$  are evaluated for  $p = 2, 3$  and for  $i = 0 \dots p$ ,  $j = 0 \dots p$  resulting in 9 and 16 evaluations, respectively. The values will be denoted from now on as  $f_{i,j}^p$ .

Table 6.3: Multivariate polynomials for  $p = 2$  (black) and  $p = 3$  (black+green)

$\Psi(\xi_1, \xi_2)$	$\Psi_2^0(\xi_2)$	$\Psi_2^1(\xi_2)$	$\Psi_2^2(\xi_2)$	$\Psi_2^3(\xi_2)$
$\Psi_1^0(\xi_1)$	$\Psi_1 = \Psi_1^0(\xi_1)\Psi_2^0(\xi_2)$	$\Psi_3 = \Psi_1^0(\xi_1)\Psi_2^1(\xi_2)$	$\Psi_6 = \Psi_1^0(\xi_1)\Psi_2^2(\xi_2)$	$\Psi_{10} = \Psi_1^0(\xi_1)\Psi_2^3(\xi_2)$
$\Psi_1^1(\xi_1)$	$\Psi_2 = \Psi_1^1(\xi_1)\Psi_2^0(\xi_2)$	$\Psi_5 = \Psi_1^1(\xi_1)\Psi_2^1(\xi_2)$	$\Psi_9 = \Psi_1^1(\xi_1)\Psi_2^2(\xi_2)$	
$\Psi_1^2(\xi_1)$	$\Psi_4 = \Psi_1^2(\xi_1)\Psi_2^0(\xi_2)$	$\Psi_8 = \Psi_1^2(\xi_1)\Psi_2^1(\xi_2)$		
$\Psi_1^3(\xi_1)$	$\Psi_7 = \Psi_1^3(\xi_1)\Psi_2^0(\xi_2)$			

The last step is to evaluate the value of the different coefficients. This can be done by Galerkin-projection (Step IV.1a in Fig. 6.2) using Eq. (6.11). The integration has to be done in a numerical manner, using a simple formula:

$$\alpha_k(\mathbf{x}) = \frac{\langle \Psi_j(\boldsymbol{\xi}), R \rangle}{\langle \Psi_k^2(\boldsymbol{\xi}) \rangle} = \frac{\sum_{i=0}^p \sum_{j=0}^p w_i^{1,p} w_j^{2,p} f_{i,j}^p \Psi_k(x_i^p, y_j^p)}{\langle \Psi_k^2(\boldsymbol{\xi}) \rangle} \quad (6.21)$$

where  $\langle \Psi_k^2(\boldsymbol{\xi}) \rangle$  are constants computed previously by the software, independent from the problem. In this equation, all constants are already computed in Step I (quadratures, abscissas and polynomials) or evaluated in Step III ( $f_{i,j}^p$ ). Thus, using a simple summation the coefficients are determined. After evaluating the equation for  $k = 1 \dots 6$  and  $k = 1 \dots 10$  for  $p = 2$  and  $p = 3$ , respectively, the response is completely reconstructed ( $\sum_k \alpha_k(\mathbf{x}) \Psi_k(\boldsymbol{\xi})$ ). Based on this function many different analyses can be done (Step IV.2 in Fig. 6.2). Furthermore, expected value and standard deviation are evaluated based on Eq. (6.12).

### 6.3.2.2 Results

The results of the PCE expansions are shown in Figure 6.4. As one can see, TP and SC methods show a very good agreement. For a polynomial order  $p \geq 5$ , the PDF function is almost perfectly reconstructed. The convergence can be seen in the upper left corner, showing the logarithm of the average error between the Monte-Carlo simulation of the original function and the reconstructed function. As one can see, the convergence is very fast. Reaching order 8, the error cannot be reduced any more, as the precision has already reached the confidence interval of the Monte-Carlo simulation.

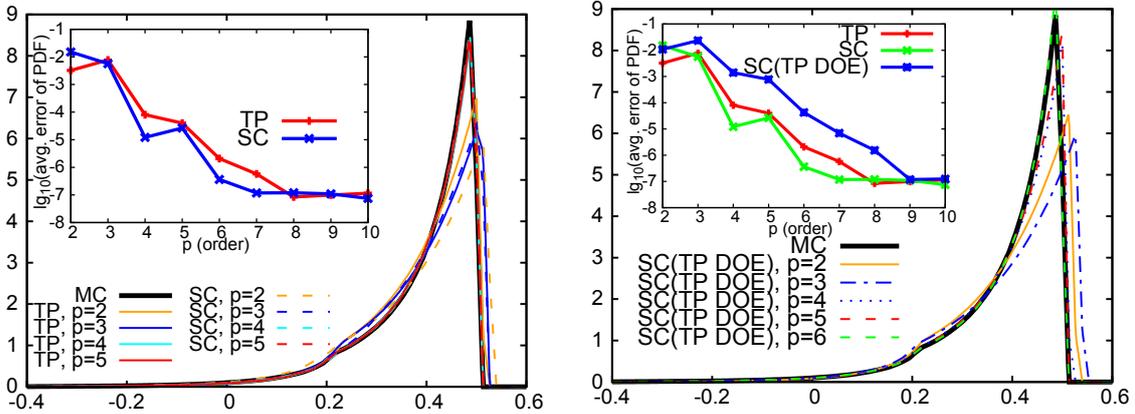


Figure 6.4: (a) Convergence of the Probability Distribution Function (PDF) with truncated normal variables (b) Convergence of the Probability Distribution Function (PDF) with truncated normal variables for SC method with TP samples

Although LHS should be the optimal sampling method for SC according to Hosder et al. [112], there is no restriction in using a different sampling method and oversampling rate. To test the accuracy, Stochastic Collocation was applied to the samples of the

quadrature method, resulting in a slightly lower oversampling rate ( $n_p = 1.333, 1.5, 1.6, 1.667, 1.714, 1.75$ ). The resulting PDF functions can be seen in Figure 6.4. The method ( $SC(TP\ DOE)$ ) is still able to converge to the same PDF function. However, due to the smaller oversampling, the error increases for the same polynomial order.

Then, a second test was performed. The input variables were replaced to show uniform distribution in the same domain. After executing again PCE, the results are presented in Fig. 6.5. Both methods are again able to converge very fast in this case.

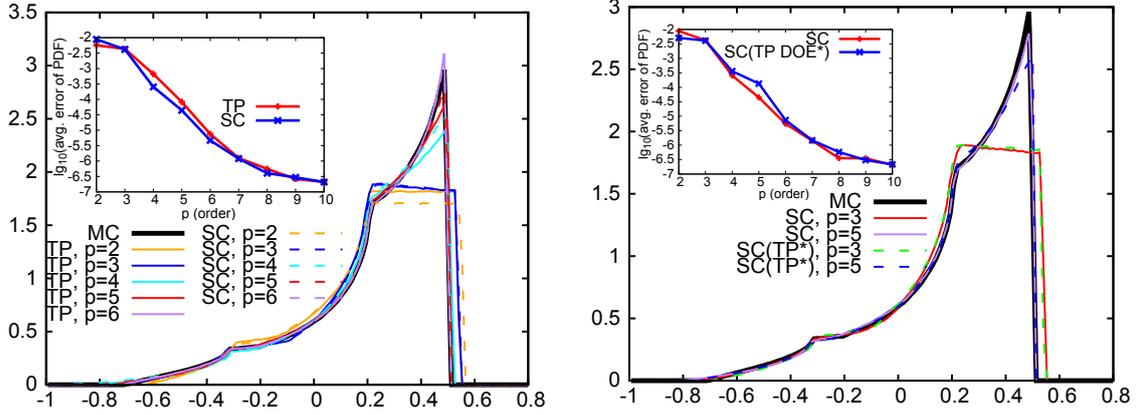


Figure 6.5: (a) Convergence of PDF with uniform variables (b) Convergence of PDF with uniform variables for SC method with optimal sampling and using the DOE of TP with truncated normal variables (starred)

At last, it was investigated if it is possible to simply recompute the PDF functions for a different distribution without computing additional points. The Design-Of-Experiment from the first step for TP was retained (abscissas for the truncated normal variables, denoted with star symbol), and the PCE was computed using SC, assuming uniform variables and compared to the optimal sampling. The PDF functions are compared in Figure 6.5 for  $p = 3$  and  $p = 5$  along with the convergence history. The agreement is very good, supporting the original assumption.

## 6.4 Setup of CFD Model

### 6.4.1 Selected physical models

In the present study the same two-dimensional approach was chosen again, as in Chapter 4. For the physical modeling, the same approach was repeated. The settings are summarized again below:

- The flow around Darrieus rotors was modeled as incompressible ( $M < 0.3$ ).
- Reference density  $\rho = 1.225 \text{ kg/m}^3$  was chosen according to DIN EN 61400 [71], the dynamic viscosity was correspondingly  $\mu = 1.7894 \cdot 10^{-5} \text{ Pa s}$ .
- Second-order implicit temporal discretization is applied with second-order (central or upwind) derivatives.

- Coupled solver is applied.
- In the first step at least 10 revolutions were computed with lower temporal resolution (72 time steps per revolution) to initialize the flow field.
- In the second step at least 5 high-resolution revolutions were computed, with 720 time steps per revolution ( $\Delta\varphi = 0.5^\circ$ ).
- For each time step, 24 inner iterations were computed.
- Only the last quasi-periodic revolution was used for post-processing.
- Considering earlier results of the thesis the  $k$ - $\omega$  SST model was applied.
- The sides of the domain were defined as velocity inlet, pressure outlet and symmetry boundaries. At the inlet,  $I = 0.1\%$  and turbulent viscosity ratio of 10 were retained in agreement with recommendations from the literature [54, 134, 135].

### 6.4.2 Selected rotor geometry

For the UQ study, the same *exemplary* rotor was chosen with NACA0021 blades, as for the blade optimization in Chapter 4. As a result, description of the validation, computation of the characteristic curve, spatial discretization and numerical validation will not be repeated here. Following the conclusions of the optimization, only  $k$ - $\omega$  SST model was retained for the present study.

## 6.5 Setup of Uncertainty Quantification

The completely automated OPAL++ workflow presented in Chapter 4 was applied again and will thus not be presented here again. The PCE expansion and Gauss quadratures were computed with OPAL++ as well.

### 6.5.1 Uncertain parameters

In the followings, uncertainties are introduced originating from two sources. For both, truncated normal distributions were used with an orthogonalization, following [144]:

- *Operating conditions*, here the rotation speed: for rotors with a variable rotation speed it is very important, that the control is able to set the correct angular velocity. Otherwise, the rotor will operate at an incorrect tip-speed-ratio. The actual angular velocity can deviate from the optimal one due to the latency of the controller, incorrect rotation speed sensor, incorrectly reported wind speed or sudden gust accelerating/decelarating the rotor. The angular velocity was chosen to be truncated by  $\pm 2\sigma$ . This means, that for  $\lambda = 2.5$  the distribution  $\mathcal{N}(20, 1)$  was truncated to  $u \in [18 \frac{\text{rad}}{\text{s}}; 22 \frac{\text{rad}}{\text{s}}]$ , while for  $\lambda = 3.5$  the distribution  $\mathcal{N}(28, 1)$  was truncated to  $u \in [26 \frac{\text{rad}}{\text{s}}; 30 \frac{\text{rad}}{\text{s}}]$ .

- *Geometric parameters*: one of the most likely imprecisions during manufacturing of the rotor might be in the pitch angle of the blades. Blades might be mounted in a slightly displaced position (an offset is also equivalent to a pitch angle). A slightly incorrect pitch angle can result from deformations of the mounting structure as well. Under operation, it might be a consequence of the exerted aerodynamic forces. Unfortunately, there is no published information concerning the distribution of this quantity, so  $\mathcal{N}(0, 1)$  was truncated to the domain  $\beta \in [-2^\circ; 2^\circ]$ .

## 6.5.2 Applied PCE expansion

Polynomial Chaos Expansion with order 5 was created using Tensor Product Quadrature method, resulting in 36 simulations for each tip-speed-ratio. In order to evaluate the convergence, a lower-order (order 4) PCE was computed as well with 25 simulations for each tip-speed-ratio. Altogether, 61 simulations were necessary to quantify all necessary values for a single tip-speed-ratio. Thus, a total of 122 CFD-simulations were executed in parallel with OPAL++ on the institute's own Linux cluster, using 25 computers, resulting in several weeks run-time.

## 6.6 Results

### 6.6.1 Performance coefficient

In the first step, PCE was computed for the performance coefficient.  $C_p$  was evaluated in all cases for the last revolution. The PDF function can be seen in Figure 6.6.

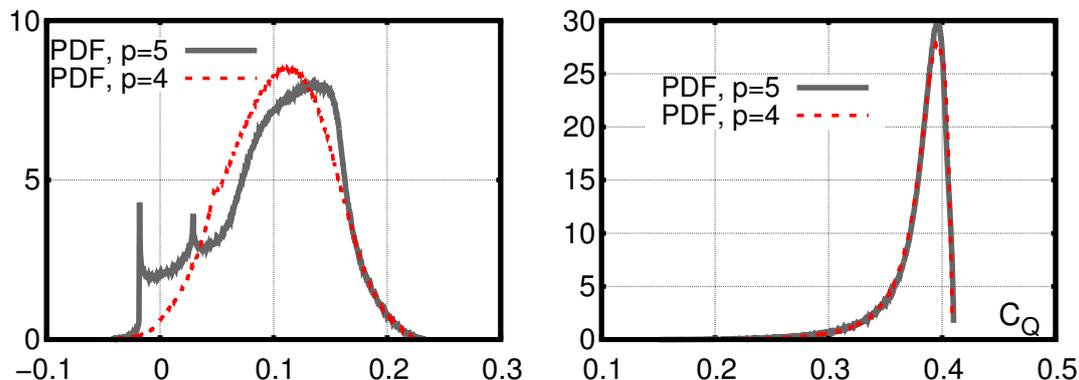


Figure 6.6: Probability Distribution Function of  $C_p$  with  $p = 4$  and  $p = 5$  for  $\lambda = 2.5$  and  $\lambda = 3.5$

As one can see, the agreement between the 4<sup>th</sup> and 5<sup>th</sup> order method is perfect for  $\lambda = 3.5$ , while large discrepancies appear for  $\lambda = 2.5$ . This is not a surprise, since the aerodynamic behavior of the rotor is far more complex for  $\lambda = 2.5$ , so that the obtained response shows a much wider spectra. However, one should not forget, that for practical applications not the Probability Distribution Function, but the Cumulative Distribution Function is mainly of interest; this is the integral of the former, thus inherently smoother.

The convergence is appropriate for practical purposes. Although the full distribution has not converged yet, the most important values already agree, as documented in Table 6.4. The table first presents the performance coefficient obtained without UQ ( $C_p^{NUQ}$ ); then, using UQ, the expected value ( $E(C_p)$ ), the variance ( $\sigma^2(C_p)$ ) and the chosen percentiles ( $C_p^x : P(C_p < C_p^x) = x\%$ ). The error is in each case very small, showing that the expected values and variances can be obtained thanks to PCE with a high confidence level. Note that the expected values have been always found smaller than in computations without UQ techniques.

Table 6.4: UQ analysis for  $C_p$

	$\lambda = 2.5$ ( $p = 5$ )	$\lambda = 2.5$ ( $p = 4$ )	$\lambda = 3.5$ ( $p = 5$ )	$\lambda = 3.5$ ( $p = 4$ )
$C_p^{NUQ}$	0.11693	0.11693	0.38719	0.38719
$E(C_p)$	0.10207	0.10440	0.37985	0.38012
$\sigma(C_p)$	0.05118	0.04414	0.02742	0.02638
$C_p^{1\%}$	-0.01606	0.00354	0.26900	0.27553
$C_p^{99\%}$	0.19916	0.19891	0.40840	0.40742

## 6.6.2 Torque for the quasi-periodic revolution

In a second step, the torque coefficient ( $C_Q$ ; positive values denote regions with energy generation) for a single blade (non-dimensionalized by  $0.5\rho Au^3$ , where  $A = 2R = 2 \text{ m}^2$  and  $u = 8 \text{ m/s}$ ) is presented for a quasi-periodic revolution of the rotor in Figures 6.7 and 6.8. The UQ method was repeated for each  $\varphi$  position (with  $0.5^\circ$  step). The black line represents the computation without UQ ( $\beta = 0^\circ$ ,  $\omega = 20$  and  $28$ , respectively for  $\lambda = 2.5$  and  $\lambda = 3.5$ ), the blue dotted line represents the expected value for  $C_Q(\varphi)$ , the two black dotted lines represent the  $E(C_Q) \pm 2\sigma$  interval. The green-red color scale represents the percentiles defined as  $C_Q^x : P(C_Q < C_Q^x) = x\%$ .

One can see that for  $\lambda = 2.5$  the variance is large, except where the angle-of-attack is close to zero. This is again an indication of the very complex aerodynamic features found in this case. Furthermore the expected  $C_Q(\varphi)$  curve does not overlap completely with the  $C_Q^{NUQ}$  curve; it contains far less oscillations and is smoother than the result obtained without UQ.

For  $\lambda = 3.5$  the variance is only large for the position, for which stall appears in some configurations considered during UQ due to the large angle-of-attack. Otherwise the distributions are mostly symmetric with a small variance. The expected values overlap with the values obtained without uncertainty quantification.

In Figures 6.7 and 6.8, inserts show the obtained PDF functions for  $\varphi = 30^\circ$ ,  $\varphi = 140^\circ$  and  $\varphi = 330^\circ$ . As one can see, the distributions are symmetric and similar to a Gaussian at many points. However, highly asymmetric distributions are also observed. As a consequence, the percentile curves are not symmetric around the expected values and the error bars defined by  $\pm 2\sigma$  do not completely cover the real behavior of the system. This highlights the importance of advanced UQ techniques, since applying simply a safety interval  $\pm 2\sigma$  will lead to incorrect error ranges in many cases.

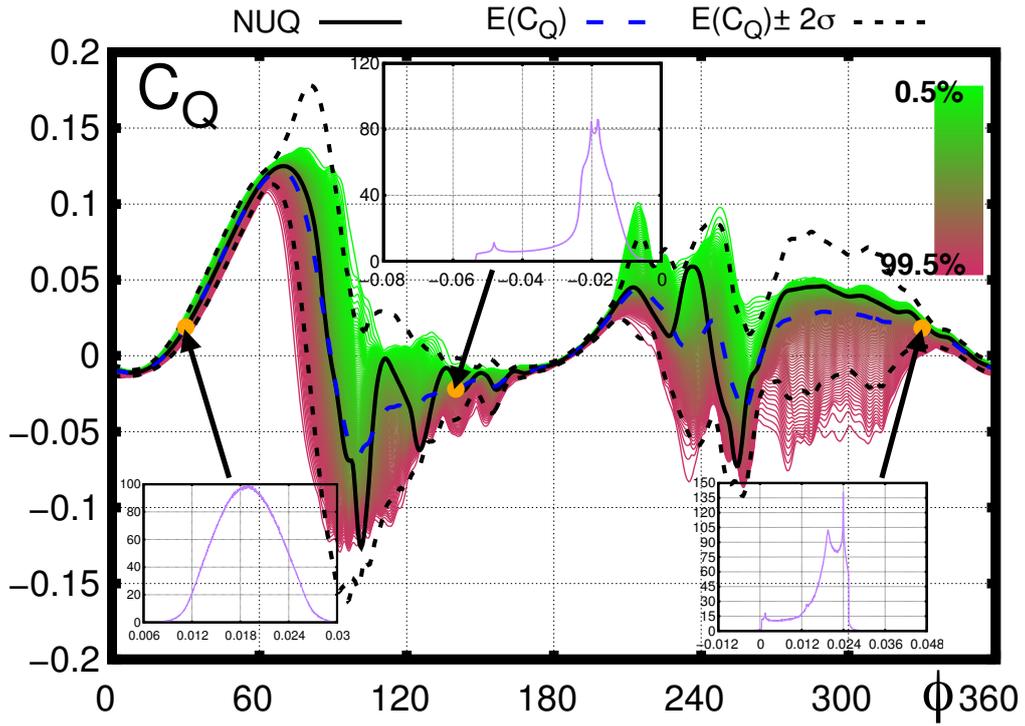


Figure 6.7: Analysis of  $C_Q(\varphi)$  with PCE for  $\lambda = 2.5$

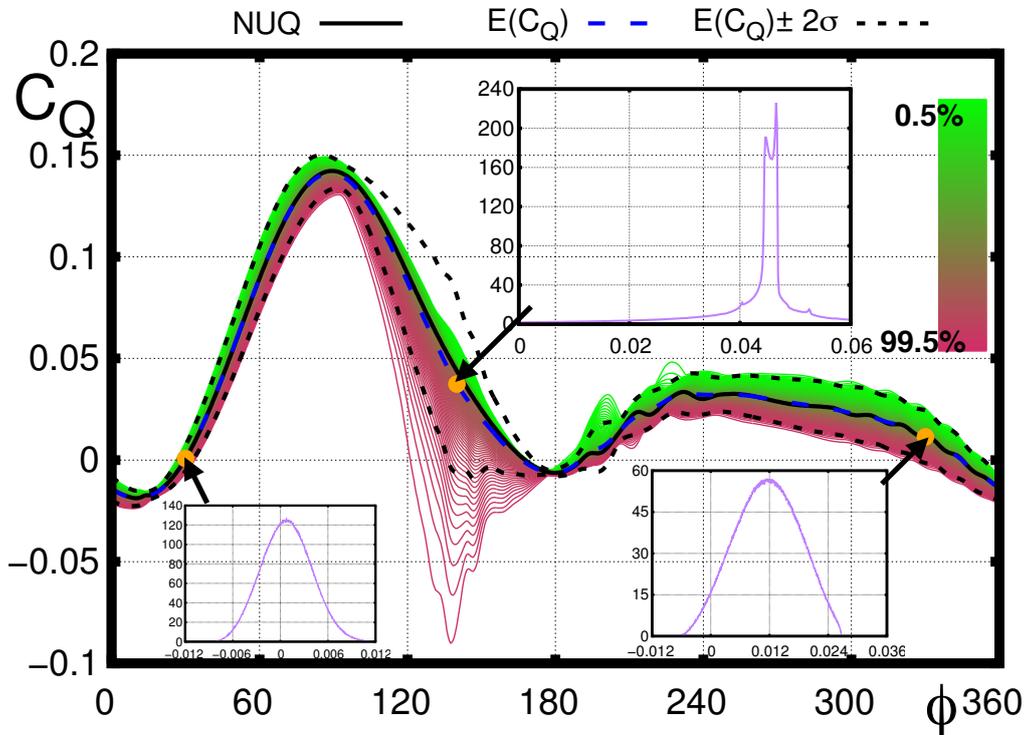


Figure 6.8: Analysis of  $C_Q(\varphi)$  with PCE for  $\lambda = 3.5$

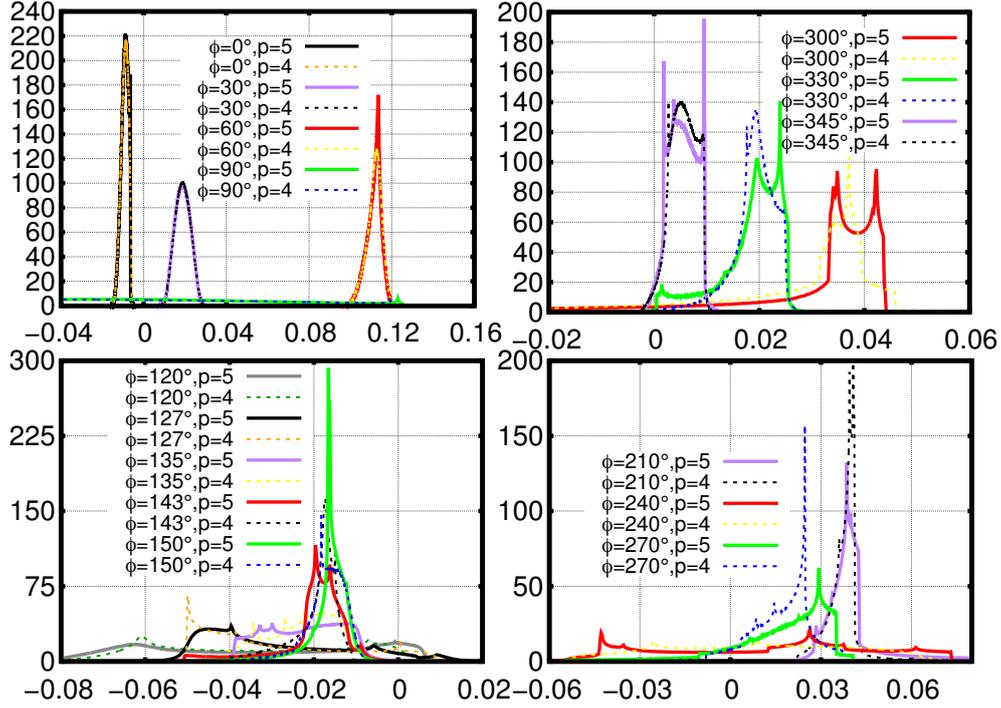


Figure 6.9: Distribution function for  $C_Q(\varphi)$  with  $p = 4$  and  $p = 5$  for different quadrants of the rotor (0-180=upstream, 180-360=downstream) for  $\lambda = 2.5$

### 6.6.3 Convergence

Finally, the convergence between the two approximation orders ( $p = 4$  and  $p = 5$ ) was analyzed. The PDF function was evaluated using  $p = 4$  and  $p = 5$  for several phase angles. The selected points are presented in Figs. 6.9 and 6.10. As one can see, the agreement of the PDF functions is perfect for  $\lambda = 3.5$ , except for  $120^\circ < \varphi < 150^\circ$ , which corresponds to the largest angle of attack, and for which small differences appear. Thus, applying a  $p = 4$  order UQ is appropriate for this operating point. However, this is not the case for  $\lambda = 2.5$ . There, the agreement is still perfect for  $0^\circ < \varphi < 90^\circ$  and acceptable for  $300^\circ < \varphi < 360^\circ$ ; but, for all conditions leading to stall and/or wake-blade interaction the agreement is still not satisfactory. Thus, if the distributions are also of interest, even higher order approximations are recommended ( $p = 6$  or  $7$ ).

However, when comparing the expected values of  $C_Q$  and the variance, the agreement is almost perfect, as illustrated in Figs. 6.11. This highlights the robustness of the applied UQ method. This means that the convergence of the first moments is already good enough for practical applications, despite the fact that the distribution functions are not completely converged.

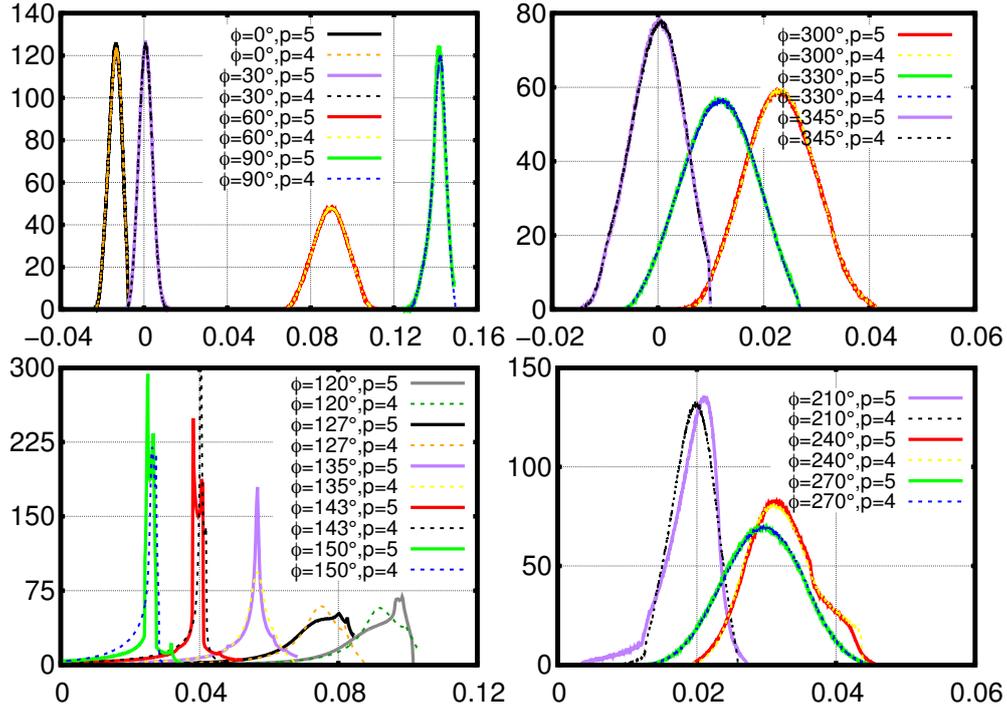


Figure 6.10: Distribution function for  $C_Q(\varphi)$  with  $p = 4$  and  $p = 5$  for different quadrants of the rotor (0-180=upstream, 180-360=downstream) for  $\lambda = 3.5$

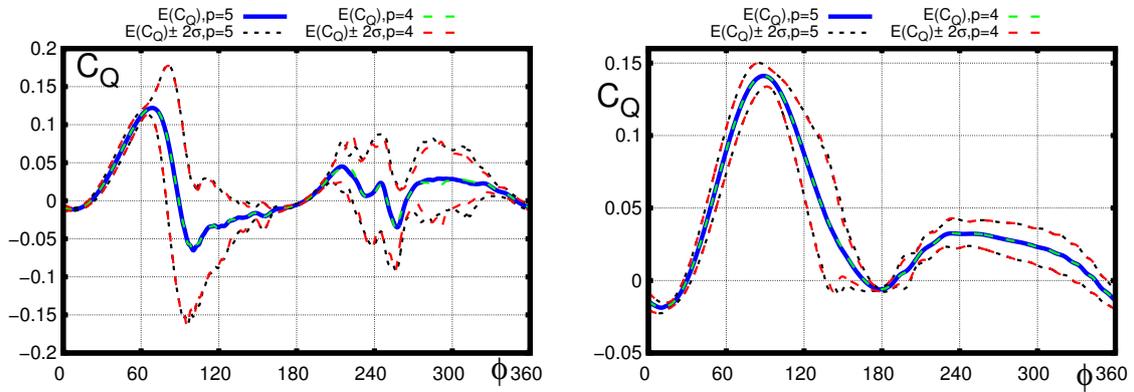


Figure 6.11:  $E(C_Q)$  and  $E(C_Q) \pm 2\sigma$  for a single revolution with  $p = 4$  and  $p = 5$  for  $\lambda = 2.5$  and  $\lambda = 3.5$

## 6.7 Conclusions and summary

In the present chapter Uncertainty Quantification based on Polynomial Chaos Expansion was applied to an H-Darrieus rotor. In a first step, the theoretical background was described, followed by the numerical validation of the implementation for an analytical case.

Afterwards, the PCE method was applied to analyze the operation of an H-Darrieus rotor using CFD simulations with the  $k-\omega$  SST model, considering geometric and operational uncertainties at  $\lambda = 2.5$  and  $\lambda = 3.5$ . Pitch angle and angular velocity were defined as uncertain inputs and modeled as truncated normal variables. The performance coefficient ( $C_p$ ) and torque coefficient ( $C_Q(\varphi)$ ) as function of the phase angle were analyzed as output quantities. These simulations have shown that an order  $p = 4$  is already appropriate to reconstruct the distribution functions at the higher tip-speed-ratio. For lower tip-speed-ratios, higher-order approximations ( $p > 5$ ) would be needed for that purpose.

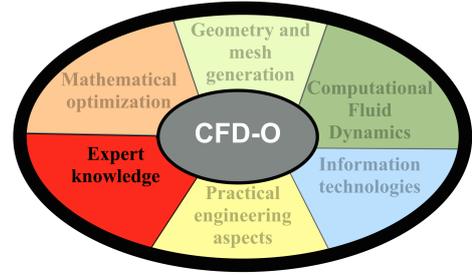
However, the first moments (expected value and variance) were already converged at order 4, the approximations with  $p = 4$  and  $p = 5$  indicating only very small differences. Since these quantities are essential for practical operation, UQ can already be used to obtain robust results. It has been observed that the expected performance value with UQ was systematically lower than the results without uncertainty analysis, which could partly explain systematic discrepancies between numerical and experimental comparisons.

Additionally, it was also shown that applying  $\pm[1\dots 2]\sigma$  might not be a correct indicator for the error bar, since the distributions are found to be asymmetric in many real conditions. This highlights the need for advanced UQ methods.

*This chapter successfully provided one example for using modern Uncertainty Quantification methods on a transient, turbulent complex CFD problem. It has also successfully highlighted that for high-quality robust optimization, advanced mathematical methods are required. After this example, medical optimizations will be covered in the next chapter. In previous studies, every step was completely automated. In principle, the optimization processes were standardized to describe **all** configurations of a specific product line and could be adapted into an industrial development process, without requiring any human intervention. Medical optimizations are very different, as*

- *all cases are different (patient-specific),*
- *expert intervention cannot be completely eliminated (experience of the doctors is very important),*
- *the topology of the geometries can be very different.*

*As a result, an approach will be presented, in which the optimization serves only as a tool to support the work of experts. The optimization works in an automated way, but requires expert intervention, resulting instead of a computer-driven process into an **expert-driven** optimization process.*



## Chapter 7

# Optimization of medical problems - Expert driven optimizations

“An expert is someone who has succeeded in making decisions and judgements simpler through knowing what to pay attention to and what to ignore.”

---

*Edward de Bono*

*In the previous chapters, various aspects of CFD-optimization processes were presented by analyzing Darrieus rotors in 2D, 3D and with Uncertainty Quantification. All these optimizations had a common point: they were completely automated. In principle, after the time-consuming setup all these methods could be used in the production as well, with minimal user intervention.*

*In the followings a completely different problem will be analyzed, where human intervention cannot (and should not) be avoided: a patient-specific medical treatment will be discussed. This necessitates a completely different approach, in which instead of the complete automatization support provided by the experts is desired, to the highest possible level. Hence, experts should be able to perform interventions in the optimization process in a fast and efficient way, while the process should perform all other operations in the fastest possible way. Thus, the present chapter will focus on optimizations, where Expert knowledge is of the utmost importance.*

*The present chapter is based on an extended version of a journal publication and a conference presentation:*

JANIGA, G., DARÓCZY, L., BERG, P., THÉVENIN, D., SKALEJ, M., AND BEUING, O. An automatic CFD-based flow diverter optimization principle for patient-specific intracranial aneurysms. *Journal of Biomechanics* 48, 14 (2015), 3846–3852 [119].

DARÓCZY, L., BERG, P., AND JANIGA, G. Optimization of Flow Diverter Treatment for a Patient-specific Giant Aneurysm Using STAR-CCM+. In *STAR Global Conference* (Prague, Czech Republic, 2016) [53].

## 7.1 Aneurysms and their consequences

Aneurysms are local, permanent dilatations of the arterial vessel walls, which can occur at different places of the body (cerebral or abdominal) and can take different forms (saccular or fusiform). The present chapter will focus on saccular cerebral aneurysms.

Despite the fact, that most people are not even aware of the existence of this medical condition, aneurysms have a very high prevalence (up to 2-5% in western countries [23, 47]). Most of these are detected incidentally during other examinations. Although occurrence of the rupture itself is low, ruptures are in 40% lethal and 66% of the survivors will suffer from permanent neurological deficits [27]. Aneurysm rupture is still responsible for almost 500 000 deaths every year [27].

## 7.2 Treatment of aneurysms

Several options exist for the treatment of intracranial aneurysms. One possibility is the clipping. In this case, a small metallic clip is placed along the neck of the aneurysm, see Fig. 7.1 (a). This blocks effectively the blood flow into the aneurysm sac. However, for clipping dangerous open-skull surgery is necessary, requiring 6-12 months for the skull bone to heal [27].

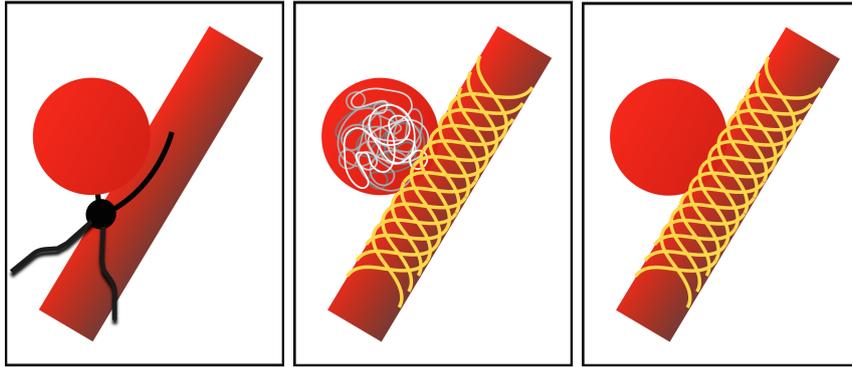


Figure 7.1: Possible treatments for aneurysms schematically: (a) Clipping (b) Coiling (c) Stenting

A less invasive approach is coiling, where platinum coils are placed in the aneurysm using a catheter, see Fig. 7.1 (b). The coils are simply detached through electrical current. To prevent the coils from entering the circulatory system, a stent is then placed in the artery to block them. An increasingly popular method is stenting without coils, where a low porosity (high-density) flow-diverter is used. This is less invasive [219] and reduces the risk of perforation [220] while still reducing the mass flow into the sac [187]. This reduces the load on the vessels and initiates thrombosis and finally occlusion. However, it does not always lead to the desired effect. To avoid complications, simulative methods could be used to assist the treatment planning. “The optimal treatment ... is a fundamental issue for neuroradiologists as well as neurosurgeons. Due to highly irregular manifold aneurysm shapes and locations, the choice of the stent and the patient-specific deployment strategy can be a very difficult decision.” [119].

## 7.3 Towards patient-specific treatment

CFD-based treatment optimizations are not completely unprecedented in the literature, but these studies considered usually only a single device with few variations, without the possibility of a full optimization; or with significant simplifications. E.g., Lee et al. used multi-objective optimization for minimizing velocity and vorticity, but based on a strongly idealized cases [137], while other studies were based on only two-dimensional models [8, 204, 205]. In contrast, Ma et al. used Finite Element method for the simulation of a realistic deployment process, which is very time-consuming, but fluid dynamics aspects were not considered [145, 146]. Computer supported treatments are not unprecedented in other medical fields either and are increasingly considered [193].

For all patient-specific treatments for stenting the same steps have to be performed. Figure 7.2 presents these steps, denoted as “single evaluation”. Red boxes represent steps, which cannot be automated (yet) while green boxes present steps, which can be automated with an appropriate expertise. The small figures represent, if medical (M)/technical (T) experts are needed for the current step, or if they can be executed only by relying on computers (C):

1. *Angiography (M)*: First, the aneurysm has to be identified, based on medical image acquisition (e.g., angiography) and segmentation. At this point, the medical expert has to define, if treatment is necessary.
2. *3D vessel reconstruction (M&T)*: The next step is the reconstruction of the 3D vessel geometry. In this step, both technical and medical experts have to be involved. This is necessary, as the reconstruction is a very complex technical process. Due to technological limitations artifacts have to be removed: this can only be done with the help of a physician in order to obtain realistic geometries.
3. *Parameters for stent deployment (M&T)*: Following the reconstruction, definition of the parameters for the virtual stent implantation follows. For this step, both technical and medical experts are needed. These software usually require some expertise, while only medical experts have an extensive knowledge of the adoptable flow diverters.
4. *Parameters for meshing and CFD (T)*: Finally, for the simulations, the technical experts have to decide the mesh generation and CFD settings. After this point, all steps could be theoretically automated, apart the last one.
5. *Stent deployment (C)*: Stent deployment is executed either based on geometrical [26] or structural mechanical considerations [146], using the defined parameter space.
6. *Mesh generation (C)*: Mesh generation software has to be employed to resolve and discretize the volume between the vessel and flow diverter.
7. *CFD (C)*: Detailed Computational Fluid Dynamics simulations are executed.
8. *Post-processing (C)*: Quantities of interest are extracted by the simulation software itself or through an additional post-processing software.
9. *Evaluation (M)*: The final evaluation still has to be based on the decision of the physician.

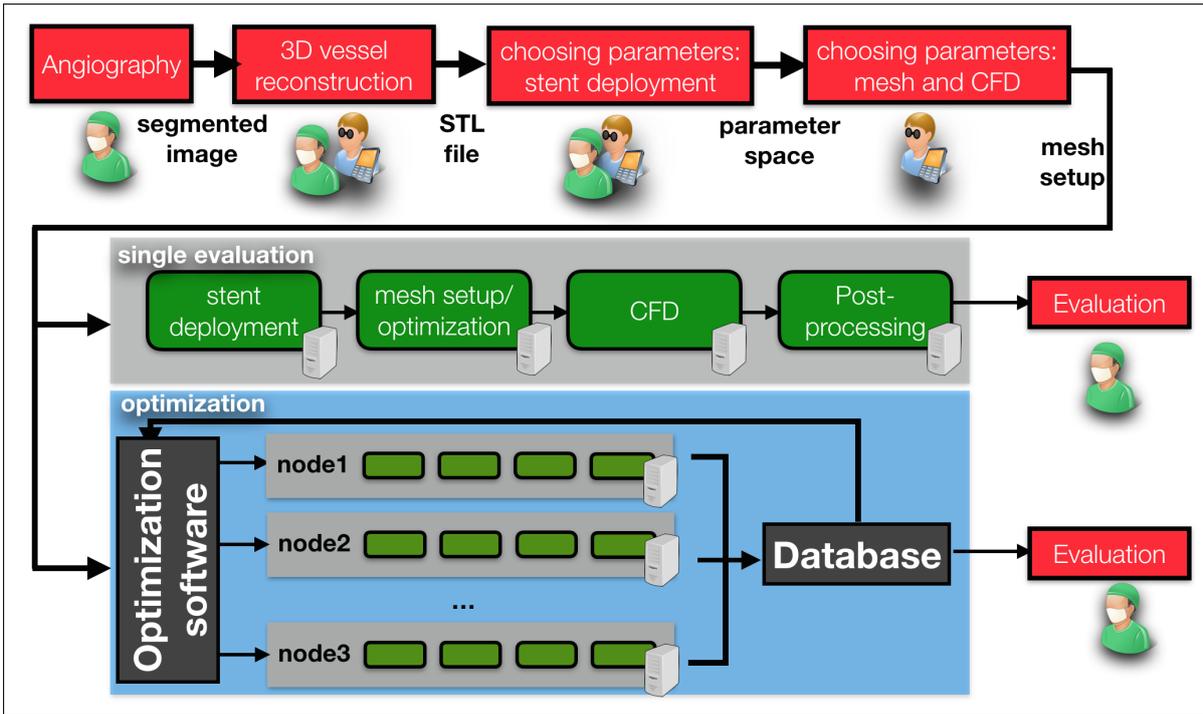


Figure 7.2: Workflow of patient specific treatment

## 7.4 Expert-driven Computer Aided Stent Evaluation (ECASE)

To enable patient-specific treatments based on realistic CFD models, a new approach is proposed, which combines realistic virtual stent deployment with detailed 3D CFD simulations. The presented approach is fully automated and can be applied to an arbitrary patient-specific geometry in a relatively short time, making it compatible with clinical practice. The ultimate goal of the research is to enable patient-specific treatment. The present study is the first of its kind to the best of our knowledge.

The proposed method was named **ECASE**, which stands for **Expert-driven Computer Aided Stent Evaluation**. The idea is very simple: if optimal efficiency is already achieved for a single evaluation (i.e., all steps, where expert intervention is not needed are already automated), there is no reason not to compute and compare many different cases. This way, instead of a single evaluation, many different configurations can be evaluated in parallel, without any increase of the user waiting time. All results can be collected in a database, see Fig. 7.2. This process can be easily coupled with an optimization algorithm. However, the final decision still has to be based on the decision of the physician.

The most important assumptions/requirements behind the idea are:

- Completely automatic stent treatment is not (yet) possible (as many different factors contribute to the efficacy of the treatment) [53].
- Important medical decisions are made by a medical expert [53]; all other operations (e.g., meshing, CFD) are executed by a computer; if needed, the technical expert can have an influence on the process.
- An optimization algorithm is used to find the most efficient solution in the defined parameter space.
- For high-fidelity results, the CFD computations have to completely resolve the geometry (see Fig. 7.3).

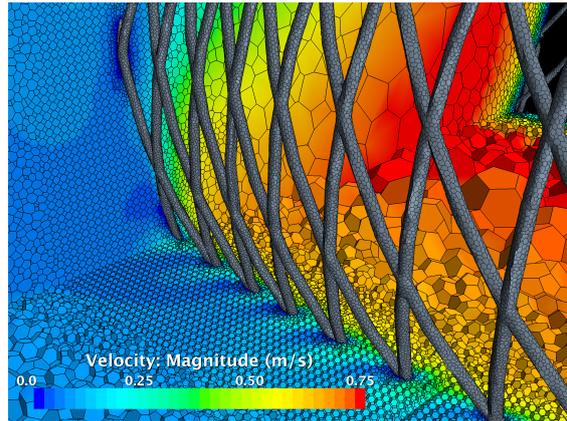


Figure 7.3: Completely resolved struts

However, automation of these steps is still not trivial, as several new challenges appear:

- *Coupling of specialized software:* Coupling in itself is not a problem, as discussed in previous chapters. However, the highly specialized in-house virtual stent deployment software used in the present study cannot be installed on all systems due to compatibility issues.
- *Mesh generation and mesh quality:* for such complex geometries with huge scale differences, automated mesh generation is not easy. Although newer automatic mesh generation methods (e.g., the Polyhedral Mesh generation in StarCCM+) provide a great help for a robust and automatic mesh generation, even such methods will fail to provide the requested quality.

The completely automated workflow was implemented in OPAL++. In the followings, the different steps implemented in the present study will be discussed in detail. Afterwards, a real clinical case will be analyzed with the method.

### 7.4.1 Reconstruction and preparation of the geometry

In the first step, the vessel geometry has to be segmented and prepared (as an STL format; Step 1-2, see Section 7.3), and a parameterized model of the stent has to be created based upon the manufacturing technology (Step 3).

### 7.4.2 Virtual stent deployment

A previously validated [120] in-house stent deployment software was applied for Step 5. The software has a batch mode to support optimizations, but lacks an appropriate maintenance and presents many compatibility problems; it cannot be installed on all systems. This problem was solved by using a Virtual Machine. The software was installed

in this virtual machine and can be accessed simply by an SSH connection. This way, the software can be efficiently used on any system (Windows/MAC/Linux) without problems, avoiding compatibility issues.

### 7.4.3 Mesh and CFD setup

Mesh generation was performed using the commercial CFD software package Star-CCM+. StarCCM+ supports robust and automatic mesh generation based on polyhedral cells for the finite-volume discretization. Polyhedral mesh results usually in better quality compared to, e.g., tetrahedral meshes, as “the presence of multiple neighbors leads to a better approximation of the gradients and faster convergence” [119].

For the mesh generation an intelligent JAVA macro was developed (appr. 3000 lines of code). The macro needs only two input files (vessel and stent), and automatically corrects smaller artifacts, identifies inlets and outlets, creates physical setup and mesh generation process, defines Automated Mesh Operations with several refinements. Based on different recommendations [118] and algorithmic methods most parameters can be defined in an automated matter. Nevertheless, it is still recommended to have the setup checked by the technical expert. The reason behind this recommendation is very simple: aneurysms, flow diverters and vessel geometries may vary widely, resulting in possibly faulty parameter values for some special cases. But to handle real patients, it is crucial to ensure a correct setup.

These points result in very different requirements. First of all, it is necessary that the expert can quickly review an exemplary case of different aneurysm setups and modify the meshing process if necessary. On the other hand, meshing still has to be executed in a completely automated way during the optimization, while ensuring appropriate mesh quality. One possible solution to remedy these problems would be to use macro recording in StarCCM+. However, this is not the most efficient way. When relying on recorded macros, the experts would need to repeat the same steps for each optimization (e.g., define boundaries), when in the reality only a couple of values have to be modified.

The solution was achieved by a combination of innovative methods:

- For mesh generation, one needs a high-quality and correct CAD file describing the volume of the model. However, in the present case the geometry is created by importing vessel and flow-diverter geometry. Using simply a subtraction could fail due to geometrical imprecisions. To avoid a lingering surface-repair process, Surface Wrapper was used as an alternative to extract the volumes. As a result, the flow-diverter geometry can be separately created and effectively imported into the vessel geometry.
- To provide mesh generation both in a manual (i.e., expert-driven) and automated way, a Script-in-Script (SiS) approach was applied. First, the script tries to perform the mesh generation in an automated way and computes the mesh settings. Throughout the process, it presents the recommended values to the expert, who has the option to accept or modify the value (Step 4). Following a successful mesh generation, the values are recorded to a secondary script (with an own syntax). Afterwards, each mesh generation is completely automated (Step 6), as the JAVA

script simply performs all operations based on the secondary script containing the settings.

- Treated geometries are very complex (sharp angles, where flow diverter struts meet walls, small struts, etc.). As a result, when generating the mesh in an automated way, some settings may result in incorrect mesh, i.e., inappropriate mesh quality. However, as the JAVA script already handles the different mesh settings as parameters, it becomes possible to optimize the mesh quality. In the present study, for each considered case many different discretizations were generated with slightly varying settings and the best one (according to skewness angle) was chosen for the CFD computation.

These methods ensure a robust, high-quality and automatic mesh generation.

Following the automated geometry generation and mesh automatization process, blood flow simulations were carried out in StarCCM+, with the same JAVA macro. In principle, choosing laminar or turbulent, steady or transient, Newtonian or non-Newtonian fluid makes no difference regarding the effort, it only impacts the computational time. In order to avoid any divergence due to deformed cells, activating Cell Quality Remediation in StarCCM+ is recommended.

#### **7.4.4 Post-processing**

Post-processing is a very important part of the optimization process: the medical expert will decide based on the results at this point, which flow-diverter geometry is more appropriate. This is not simple in all cases, as the efficacy of the treatment depends on many different factors. Most popular options are to look at the Bundle of Inflow, Wall Shear Stress (WSS), mass inflow through the ostium, maximal velocities, etc. However, none of these provide a complete description of the efficacy of the treatment: although a large mass flow is damaging, no flow at all would lead to cell death. Similarly, increased wall shear stress leads to “mural-cell-mediated destructive remodeling and weakening of the aneurysm wall” [38]. In contrast, low wall shear stress may trigger inflammatory-cell-mediated destructive remodeling, if high oscillatory shear is present [151].

Therefore, one has to decide which quantity should be used as an objective function for the optimization. This value can be simply extracted by the CFD software or by the post-processing software.

### **7.5 Exemplary optimization**

Following the description of the method, a real patient geometry was chosen for the study to present the applicability of the present method for clinical practice.

### 7.5.1 Analyzed clinical case

“A 53-year-old female with an acute headache was examined in the emergency department of the University Hospital Magdeburg. The patient reported no additional symptoms, and a physical examination yielded no further findings. CT revealed a parasellar mass on the right side. Through CT-angiography, the suspected giant aneurysm of the cavernous segment of the right internal carotid artery was confirmed. Treatment was performed electively 13 days after initial presentation. A flow diverter (Silk, Balt, France) sized 3.5 mm × 30 mm was implanted, with massive reduction of blood flow into the aneurysm sac at the end of the procedure ... MRI performed six months after the implantation of the flow diverter showed complete resolution of the aneurysm.” [119].

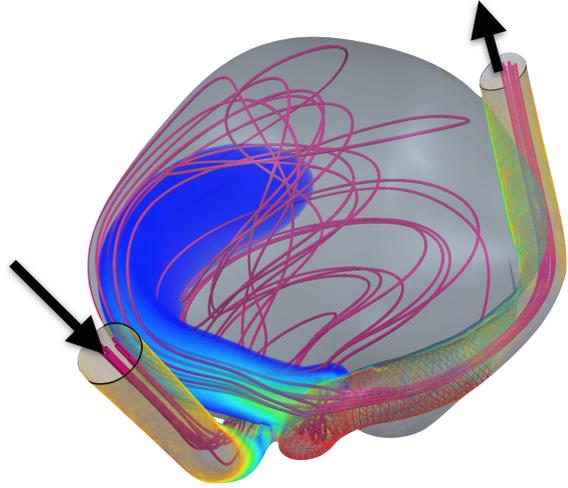


Figure 7.4: Analyzed giant aneurysm with iso-surface (0.04 m/s) and streamlines for a treated case

### 7.5.2 Vascular reconstruction and preprocessing

“A 3D rotational angiography was performed with a flat-panel angiography system (Artis Twin dBA, Siemens Healthcare Sector, Forchheim, Germany). The subsequent segmentation of the data set containing the aneurysm was carried out using MeVisLab 2.3 (MeVis Medical Solutions AG, Bremen, Germany), and the three-dimensional surface model was obtained by applying a seeded region-growth algorithm. Afterwards, visually obvious artifacts that occurred during the imaging process were corrected manually (Blender 2.68a, Amsterdam, The Netherlands), especially in areas where the vessel and the aneurysm were melted. Finally, Taubin-smoothing on the discrete surface mesh ensured a more realistic representation [159,166]. In order to check the plausibility of the reconstructed shape, the treating neuroradiologist reviewed the virtual geometry ... The following geometric properties characterize the three-dimensional aneurysm model: The diameter of the parent vessel ranges from 3.2 to 4.2 mm and is 75.2 mm long. Located at a distance of approximately 40 mm from the inlet cross-section, the aneurysm has a maximal transverse diameter of 31 x 28 mm. The distance between the dome and the ostium accounts for 23.1 mm, whereby the ostium itself spans an elliptical area of 4.1 x 9.7 mm in tangential and axial direction, respectively.” [119]. The 3D reconstruction was carried out by Dr. Mathias Neugebauer (Fraunhofer MEVIS, Bremen, Germany).

### 7.5.3 Stent geometry

“Based on the reconstructed patient-specific aneurysm model, the centerline of the parent vessel was generated with VMTK (Vascular Modeling Toolkit [7]) and used for the virtual stent deployment. A virtual SILK stent was generated with dimensions matching the

real implant using Creo Parametric 2.0 (Parametric Technology Corporation, Needham, USA). For each strut, a mean diameter of  $30\ \mu\text{m}$  was applied, and the stent pores of the uncompressed regions spanned an angle of  $44^\circ$  and  $136^\circ$ , respectively. In areas of local compression, the angle ratio was changed to  $22^\circ$  and  $158^\circ$ .” [119]

#### 7.5.4 Stent deployment

”An in-house software enabled the deformation and placement of the virtual implant. The procedure was previously validated using dyna-CT as well as micro-CT in a patient-specific in-vitro silicone phantom model [120] and compared to in vivo flow diverter deployments in large white swines [115]” [119].

#### 7.5.5 Meshing

For the present study, 0.25 mm was chosen for vessel and aneurysm walls, 0.5 mm resolution for the core region [118]. Additionally, local refinements were defined for the ostium and flow diverter walls (0.01-0.02 mm) to resolve the geometry completely. Between the refinements and core regions, a mesh density growth rate of 0.8 was applied to ensure smooth transition. These settings resulted in mesh resolutions of 1.9 million polyhedral cells without flow diverter and 9-11 million cells for the treated cases.

Mesh generation for a single treated case required approximately 4-6 hours on an Intel(R) Xeon E5-2620 (2.10GHz) processor. However, based on the parallelization of OPAL++, the different cases were meshed in parallel on different workstations. The final spatial discretization is presented in Figure 7.5 at different locations for a treated case.

#### 7.5.6 Parameterization & optimization setup

In this exemplary optimization the effect of compression was analyzed. First, three reference cases were defined for comparison: the untreated case (UN), the “normal” uncompressed stent (NC) and a high compression case (HC) was added as well, see Fig. 7.6.

Afterwards, the actual optimization variable was chosen. A medium compression was chosen and the goal was to minimize the inflow into the aneurysm through the ostium, when the position of the local partial stent compression along the centerline is varied. However, due to the small parameter space, there is no need for a complete optimization. Instead, a Design-Of-Experiment was executed with seven equidistant locations (C1-C7). If needed, an arbitrary parameterization could be obviously performed as well.

#### 7.5.7 Setup of CFD Model

As the ultimate goal of the present chapter was to present a proof-of-concept for expert-driven optimization, several simplifications were made to reduce the runtime:

- Steady flow was analyzed with an average inlet velocity of  $\bar{u} = 0.5\ \text{m/s}$  and plug flow profile. Although real flows are transient in the blood vessels, steady simulations generate almost equivalent flow patterns to the cycle-averaged results [96].



Figure 7.5: Mesh resolution of the analyzed case at different locations

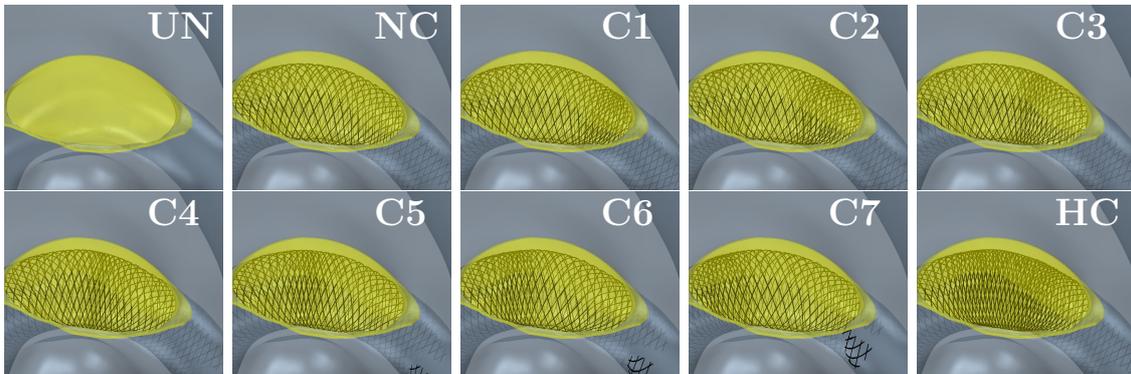


Figure 7.6: 10 different configurations (yellow color highlights ostium)

- Fluid was assumed to be incompressible, laminar and Newtonian with  $\rho = 1055 \frac{\text{kg}}{\text{m}^3}$  density and  $\mu = 0.004 \text{ Pa s}$  dynamic viscosity [36], as in the cerebral vasculature non-Newtonian effects are negligible due to the relatively high shear rate.
- 4000 iterations were performed for each simulation (this ensured normalized residuals below  $10^{-6}$ ).

The CFD simulations were performed on a small computational cluster in parallel. Each simulation was executed on a single Intel (R) Xeon E5-2620 (2.1 GHz) processor, resulting in 10-14 hours of computational time with StarCCM+. As simulations were executed in parallel, all simulations were completed within 14 hours. Together with the mesh generation, this ensured that the complete optimization did not require more than a single day, making it compatible with clinical use.

## 7.5.8 Results

### 7.5.8.1 Qualitative analysis

**Isosurface** Flow structures inside an aneurysm are complex and difficult to analyze. One possible way to visualize the overall flow structures is to use an isosurface of the velocity (0.04 m/s in this case). Figure 7.7 shows the iso-surfaces for the untreated (UN), uncompressed (NC) and C5 cases (which provided the strongest reduction of the jet size). As one can see, blood flow enters the aneurysm sac through the ostium and stays close

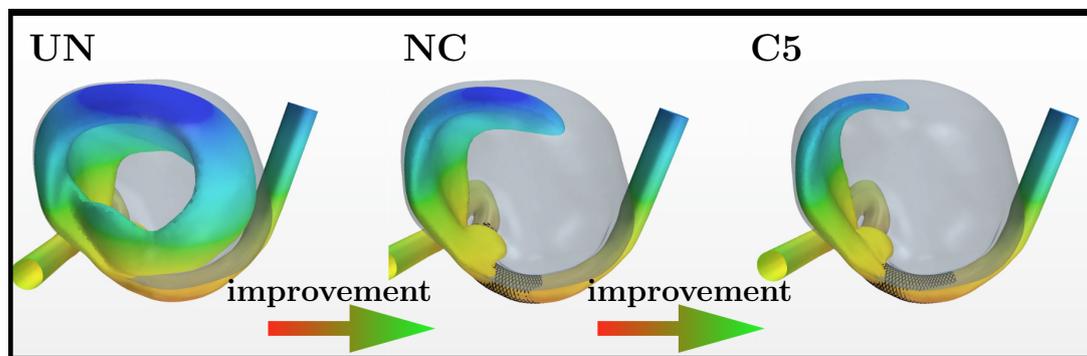


Figure 7.7: Isosurface for 0.04 m/s velocity

to the wall forming a vortex, leading to the development of a stagnation zone. When treating the aneurysm with flow diverter, a clear reduction is seen and the expansion of the isosurface is significantly reduced, for C5 configuration especially. All configurations are compared in Fig. 7.8.

**Wall shear stress** The reduction in jet size and velocities leads to additional changes in the aneurysms: wall shear stresses (WSS) change significantly too. The original untreated case shows elevated values near the neck region and along the impingement zone of the giant aneurysm (see Fig. 7.9). In contrast, all treated cases show a significant reduction, with C5 showing the most significant reduction from NC and C1-C7 (see Fig. 7.10). Of course, HC shows an even stronger reduction, but at the cost of significantly lower porosity (which may lead to complications in the clinical cases).

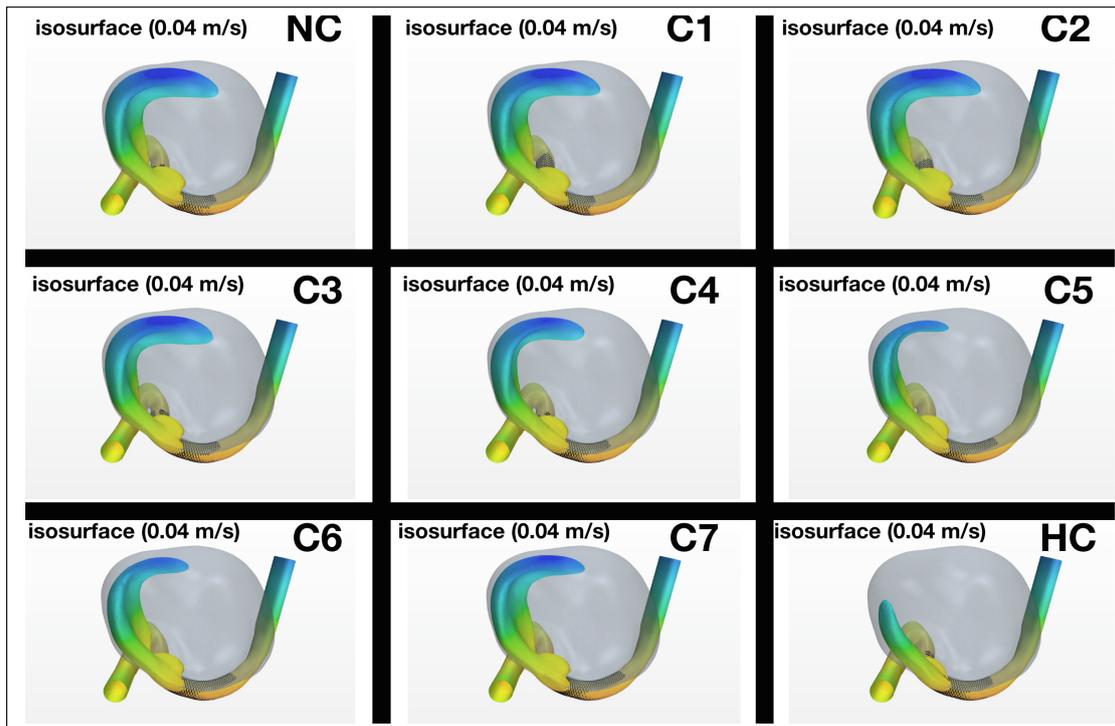


Figure 7.8: Isosurface for 0.04 m/s velocity

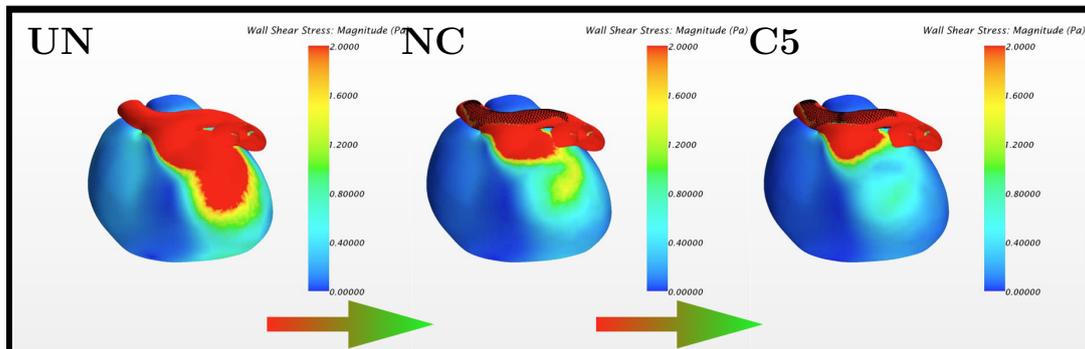


Figure 7.9: Wall Shear Stress

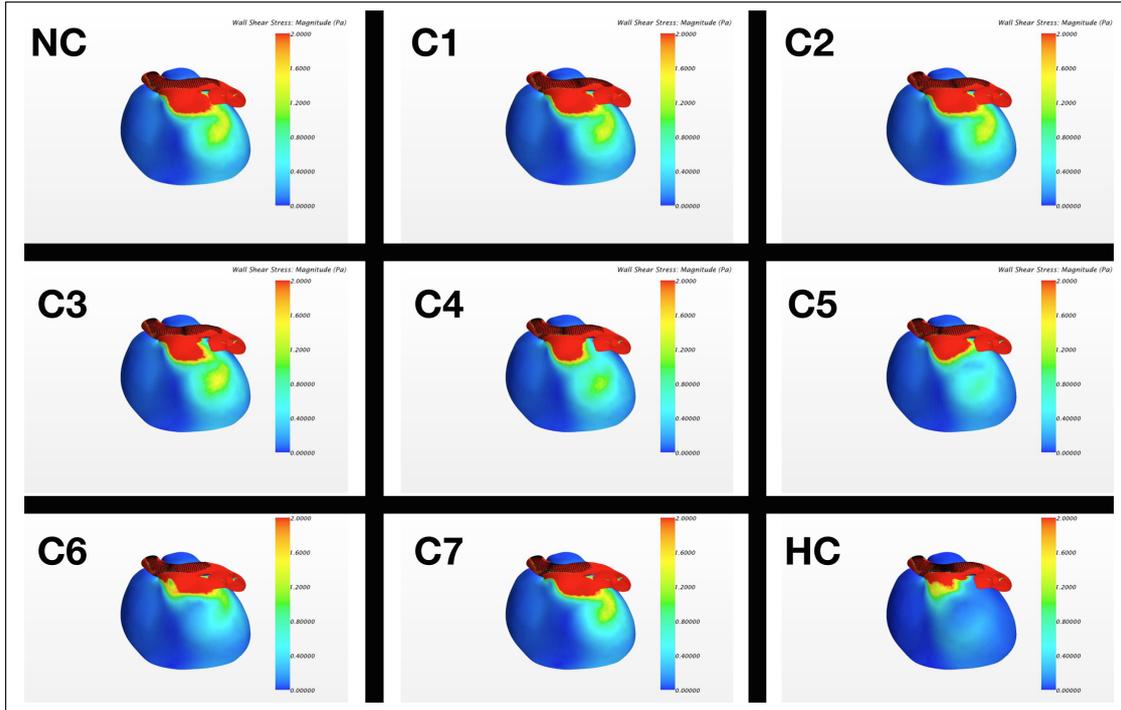


Figure 7.10: Wall Shear Stress

### 7.5.8.2 Quantitative analysis

To enable quantitative analysis, mass inflow through the ostium was integrated for each analyzed case, and compared to the untreated case (UN). The uncompressed flow diverter resulted in 24.4% reduction, indicating that using an arbitrary flow diverter leads to improvements in many cases as well. When looking on C1-C7 configurations, a wide range of improvements are found (see Fig. 7.11). The 'worst' configuration (C1) leads to 27.3% improvement, while the best (C5) leads to 33.3%. This shows, that the optimization can indeed lead to significant improvements, even when keeping the porosity constant (C1-C7).

This confirms the findings of the qualitative analysis, i.e., that C5 is the best configuration. This example concludes the present optimization.

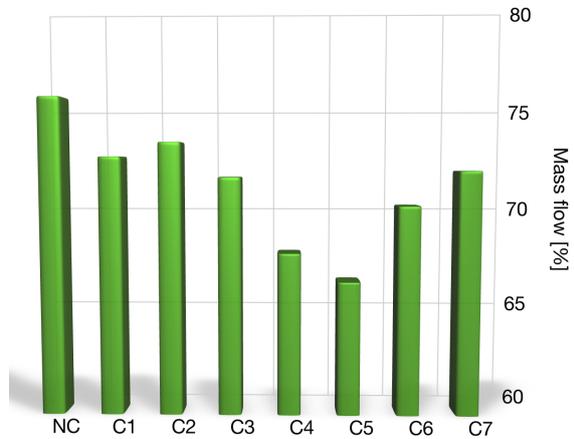


Figure 7.11: Reduction in mass inflow

## 7.6 Future improvements

Although the present case confirmed the applicability of the approach for real clinical cases, there are still many points that could be improved regarding the physical modeling. The possible improvements can be organized into two groups. The first group is relatively easy to implement. It requires almost no effort, but only a small change in the CFD settings:

- In the present study flow was considered to be steady. This is a very strong simplification, as in the reality due to the heart a pulsating flow is encountered in aneurysms. However switching the simulation from steady to transient does not require a huge effort, it is only a question of available computational resources. This way, one could compute several cardiac cycles.
- Blood was modeled as a Newtonian fluid. However, it is actually a non-Newtonian fluid in reality. Again, changing the model for the dynamic viscosity is relatively simple.

The second group contains improvements, which require significant effort and/or require a significantly different approach for the optimization process:

- In the present optimization, walls were assumed to be rigid, i.e., exhibited infinite resistance. In the reality, this is not true. Due to the pulsating flow and forces exerted by the fluid, walls are moving and deforming during the cardiac cycle. However, to model this, not only a Fluid-Structure Interaction (FSI) simulation has to be executed using an additional finite-element software (e.g., ABAQUS), but precise information concerning wall properties would be needed, as wall thickness and mechanical properties vary widely along vessel walls. Accurate measurements of these properties is not feasible yet.
- Virtual stent deployment was computed in the present case based upon geometrical considerations. However, for a completely realistic approach one could execute transient finite-element simulations of the unfolding procedure as well. This would increase computational times significantly [146].
- Parameterization of the stent for the present optimization considered only compression. Diameter of the struts, interwire distance or angle were not considered. This could be important for clinical practice and for the development of better devices.

## 7.7 Second proof-of-concept

In order to show, that the present ECASE method is flexible and can be easily extended, only the stent parameterization and virtual stent deployment methods were replaced. All other parts (mesh generation, CFD setup and post-processing) were performed relying on the same methods and scripts in OPAL++.

### 7.7.1 Alternative stent deployment

To enable a significantly larger parametric freedom, an additional in-house software was developed in C++ by the present author (approximately 9000 lines of code). In this

method, the stent deployment only needs the geometry of the vessel as an STL format. Thus, any other third-party CAD software is eliminated from the process. In this implementation, after importing the geometry, inlets and outlets are marked first (see Fig. 7.12/Step 1). Afterwards, the volume is discretized by voxels (Step 2). Using the voxels one can run a skeleton algorithm to detect the topology (Step 3 [51]). Based on the topology, a toroidal coordinate system is defined (Step 4), which is optimized for maximal cross-section diameter and curvature radius (Step 5). Based on this coordinate system, arbitrary stent geometry can be put in place based on geometrical considerations (Step 6 [26]). With this method, among others, wire radius, interwire distance, number of wires, angle between the wires, stent radius and stent location can be handled as parameters.

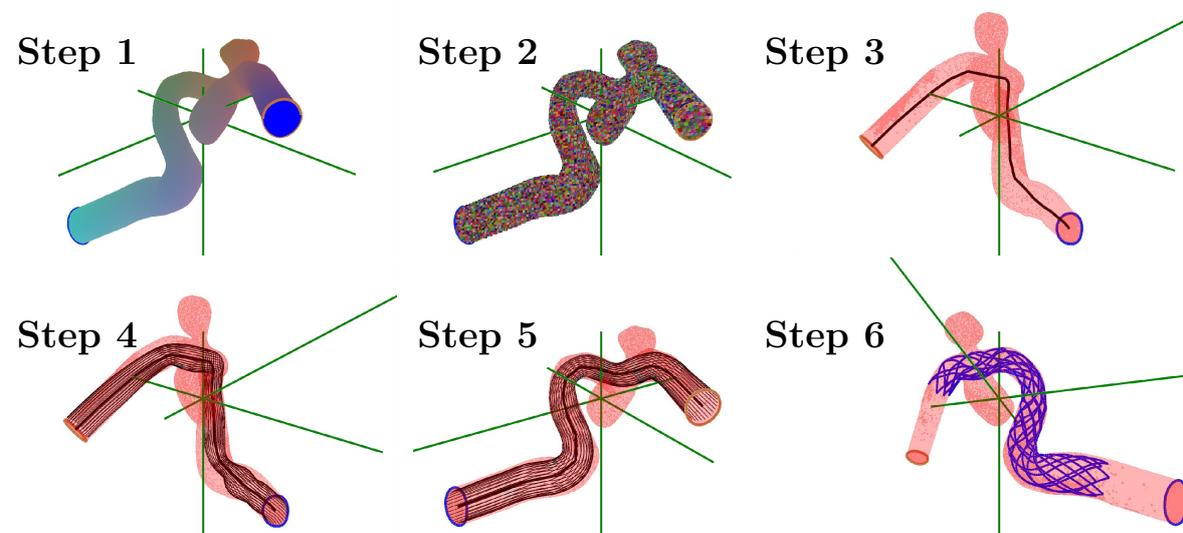


Figure 7.12: Steps of the implemented in-house stent deployment method

### 7.7.2 Setup of the optimization

To present the parametric freedom, a second short optimization was performed. In this optimization, only a few configurations were tested as the present study is only a proof-of-concept. Instead of the compression, the interwire distance and the number of struts were varied. Each strut has  $30\ \mu\text{m}$  diameter, and  $20\times 2$ ,  $24\times 2$  and  $28\times 2$  struts were tested (denoted as A, B and C, respectively), see Fig. 7.13. Interwire distance was varied to be 14 and 19.2 mm/*number of wires* (denoted as 1, 2). The list of configurations were given to OPAL++, and mesh generation, CFD computations and post-processing was executed in a completely automated way, relying on the earlier method.

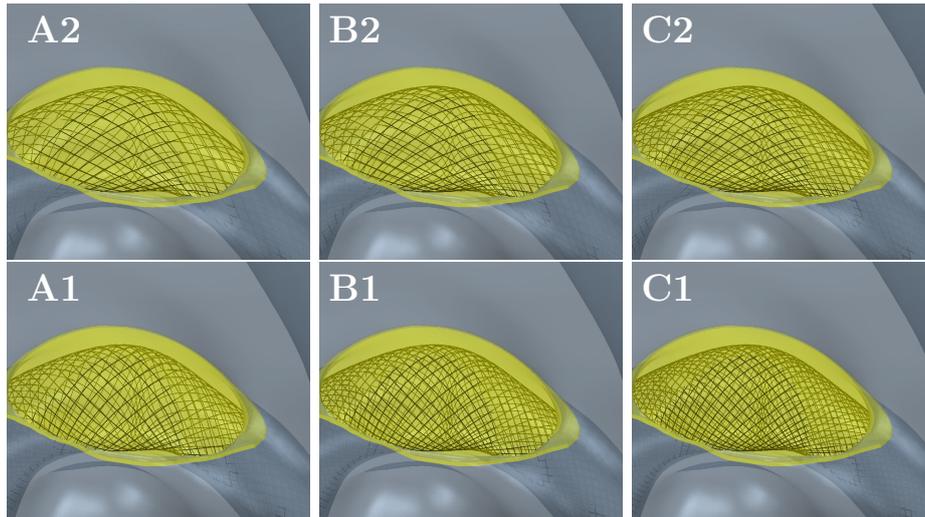


Figure 7.13:  $3 \times 2$  different configurations with varying interwire angle and interwire distance

### 7.7.3 Results

The whole process again did not require more than 24 hours. The same qualitative analysis was performed, as presented earlier. However, due to the larger parametric freedom, much wider differences are obtained. When looking at the iso-surfaces and Wall Shear Stresses (see Figs. 7.14 and 7.15), there is almost no difference between the worst treated case and the untreated (UN) case. In contrast, the best case shows significant reduction regarding both values. This confirms the necessity of a flexible parameterization.

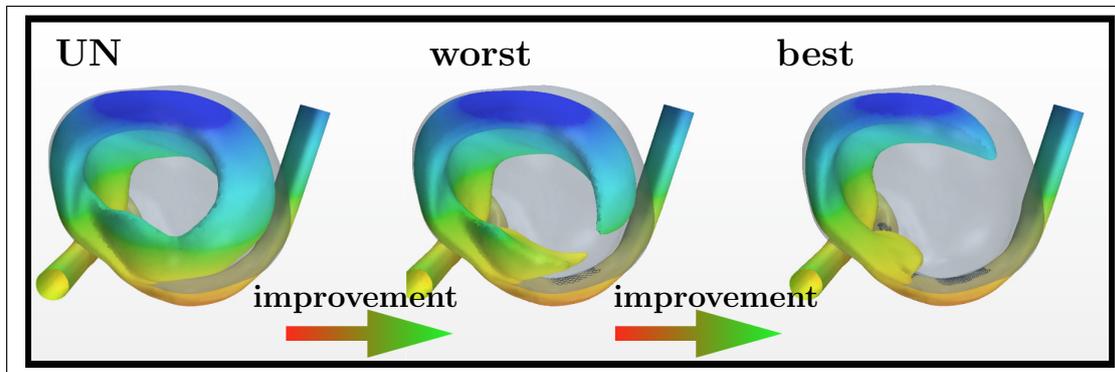


Figure 7.14: Isosurface for 0.04 m/s velocity

This example concludes the present optimization.

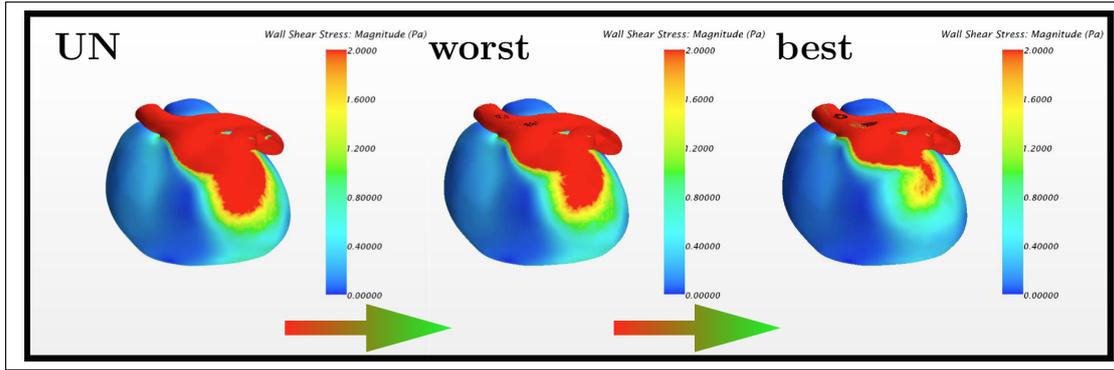


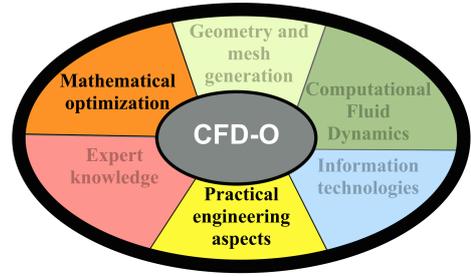
Figure 7.15: Wall Shear Stress

## 7.8 Conclusions and summary

In this chapter a proof-of-concept was successfully presented for the optimization of intracranial aneurysm treatment. First, hemodynamic simulations were performed for 8 different compression scenarios, and an optimal configuration (minimal mass flow) was identified for a fixed porosity. Afterwards, a second optimization was performed by varying number of struts and interwire angle. All simulations and evaluations were embedded into a completely automated framework in OPAL++ to enable a qualitative and quantitative analysis of the analyzed cases. The method proved to be fast and efficient and would be compatible with real clinical treatments to enable patient-specific treatments under the control of a medical expert.

*Thus, the present chapter provided an example for an optimization problem, where the input of the expert cannot be disregarded from. Thus an **expert-driven** optimization process was implemented. As one can see, the present approach enables a fast and efficient optimization of patient-specific medical cases. The research is not yet finished at this point. Hopefully, many other contributions will be made on this field to truly enable modern and affordable patient-specific treatment for everyone. With this, all chapters dealing with CFD problems are concluded.*

*In what follows, based on the experience gained through the previous chapters, a new optimization method (HYENA) will be proposed especially focusing on realistic engineering problems. This will conclude the present thesis.*



## Chapter 8

# HYENA - Practical engineering optimization algorithms

“In theory, there is no difference between theory and practice. But, in practice, there is.”

---

*Jan L. A. van de Snepscheut*

*After concluding all chapters directly focusing on CFD-based problems, the present chapter will focus on incorporating Practical Engineering aspects into Mathematical Optimization, by proposing a new optimization algorithm based on the experiences gained in the previous chapters.*

### 8.1 Introduction

Multi-objective optimization methods have undergone a significant development in the last years, resulting in reduced number of necessary function evaluations, better applicability to industrial tasks and easier access for everyday users. Nevertheless, application of these methods to many industrial problems is still impossible. As previously demonstrated in various chapters, optimization in the engineering practice is not easy and not identical to **mathematical optimization**. The most important differences are, as already pointed out in Chapter 1.2.1:

- The evaluation of the objective function may be very costly, and can require up to several days.
- During the optimization, differences of the variables below manufacturing tolerance is irrelevant (e.g., optimization of a car geometry with nm precision is meaningless).
- Due to numerical noise and model uncertainties, the objective functions are usually noisy.
- The geometry and mesh has to be created/morphed for each configuration in an automated and robust way.

- Different software (including proprietary commercial software) have to be coupled to cooperate for the optimization.

Industrial problems may be examined as well using Response Surface Methods or Surrogate models. However, without additional a priori knowledge, success is not guaranteed. With the advancement of computational power the new methods introduced in practice (e.g., LES) tend to be increasingly sophisticated, detailed and show complex responses. As the number of such new methods has significantly increased in the last years, sufficient understanding of all methods would require an excessive time and effort from the common industrial engineer, often resulting in a poor choice and incorrect conclusions. For industrial projects, only optimization methods are of interest, which are fast, take into account manufacturing precision and are error-tolerant. Furthermore, choosing an appropriate method for optimization is a difficult and complex task requiring expert knowledge from the engineer. As a result, intelligent and adaptive methods should be preferred by the industry to leverage the necessary time of training for engineers.

When the problem is simple (e.g., with SCH1, see later), applying a genetic method would result in a waste of computational resources. In this case a metamodel can be created and the multi-objective optimization can be performed on this model. Such an approach was chosen in the work of Pasquale et al. [178] or Leusink et al. [140]. Similar approaches are quite often found (e.g., [22,126,140,165,178]), but are usually restricted to creating a metamodel and performing an optimization on it. However, one cannot know prior to testing, if the answer is really that simple. If the number of variables or the parameter domain is too large, metamodels might completely fail to work. For example in Chapter 2 a heat exchanger configuration was analyzed with a genetic method [56]. The analysis has revealed that the domain is severely disconnected both in the variable and in the objective space. Due to the large number of permutations, a high number of local and global optima were observed. Approaching this problem with a metamodel would have been completely incorrect and ineffective.

The goal of the current chapter is to introduce a method that is **more** suitable for such problems and will adaptively choose between the methods by introducing competition not only between the individuals but also *between the methods*. It is demonstrated in several analytical cases that the new method overperforms one of the most popular genetic methods, NSGA-II [68], while it falls back to normal genetic methods in complex cases. Finally, a fluid dynamics example is described relying on a broadly used panel method (XFOIL) for function evaluation.

### 8.1.1 Iterative methods

In the followings, different aspects of engineering optimization will be analyzed and solutions will be proposed to the individual questions. Finally, all recommendations are put together to propose a new algorithm. For the discussions, the iterative scheme of optimization methods will be used, as discussed in Section 1.1.3.4. For the sake of understanding, the algorithm is repeated here again. Consider a modified form of an iterative search procedure as defined by Laumanns et al. [136]:

```

0: optimize( $\mathbf{y}(\mathbf{x})$ ):
1:  $c := 0$ 
2:  $A^{(0)} := \emptyset$ 
3: while  $terminate(A^{(c)}, i) = false$  do
4:    $c := c + 1$ 
5:    $F^{(c)} := generate(A^{(c-1)})$ 
6:   compute  $\mathbf{y}(\mathbf{x}) \quad \forall \mathbf{x} \in F^{(c)}$ 
7:    $A^{(c)} := update(A^{(c-1)}, F^{(c)})$ 
8: end while
9: return  $A^{(c)}$ 

```

where  $c$  is the iteration cycle counter,  $A^{(c)}$  is the archive in step  $c$  and  $F^{(c)}$  is the list of new individuals generated by the different operators. Compared to [136] it is emphasized that in Step 5 the new points are generated based on the old archive and Step 6 is added to emphasize that generating the new points and evaluating them is logically different. Finally, the Pareto front is approximated with

$$P(\mathbb{X}) \approx P(A^{(C)}) = \{\mathbf{x} \in A^{(C)} \mid \nexists \hat{\mathbf{x}} \in A^{(C)} : \hat{\mathbf{x}} \prec \mathbf{x}\}, \quad (8.1)$$

if the total number of generations is  $C = \max(c)$ . The quality of this approximation can be evaluated using different metrics, e.g., set coverage ratio, spacing, generational distance, inverted generational distance, generalized spread [65], hyper volume indicator [234], etc.

For Step 5 different operators can be applied. If a genetic operator is used, the approach is a genetic optimization method; if the individuals are based on an interpolation method, it is a metamodel-assisted optimization technique.

### 8.1.1.1 Speed and runtime of the optimization

Let us denote the computational time needed for 'Step 5' with  $t_{gen}^{(c)}$ , 'Step 6' by  $t_{eval}^{(c)}$  and 'Step 7' by  $t_{upd}^{(c)}$ . The time needed for the evaluation of a single configuration is  $t_{comp}$ , the number of individuals in the archive is  $N_{arch}^{(c)} = |A^{(c)}|$  and the number of generated individuals is  $N^{(c)} \equiv N_{new}^{(c)} = |F^{(c)}|$ . Moreover, suppose that one can compute  $p$  individuals in parallel (number of processors, or alternatively number of available licenses needed for the evaluation). The time needed for the optimization with  $C$  cycles is

$$T(C) = \sum_{c=1}^C (t_{gen}^{(c)} + t_{eval}^{(c)} + t_{upd}^{(c)}) = \sum_{c=1}^C (t_{gen}^{(c)} + t_{upd}^{(c)}) + t_{comp} \sum_{c=1}^C \left\lceil \frac{|F^{(c)}|}{p} \right\rceil \quad (8.2)$$

## 8.2 Analysis of optimization process from engineering point of view

For engineering optimization problems relying on simulations, the evaluation of the objectives is done by a dedicated numerical simulation. Then,  $t_{comp}$  becomes significantly

larger, ranging from a couple of minutes up to several days (as compared to a couple of milliseconds in the case of academic analytical problems). This and many other aspects result in modified requirements, which will be discussed in the present chapter.

### 8.2.1 Accuracy

Accuracy is important in the industry, but only up to practical limits (which is not the case in theoretical mathematics). As a result, making conclusions based on the distance from the Pareto-front (e.g., the metrics) with several thousands of function evaluations is completely impractical, as most methods have already converged at this point up to several digits. Such comparisons are still widely done in the mathematical literature. If the difference in a design variable or objective is not measurable in practice or is below manufacturing tolerance, both designs should be considered equal for engineering purposes. This question was already partially answered with the introduction of  $\epsilon$ -dominance [136]. In the present work a slightly modified (submissive) form is defined:

$$\mathbf{x}_1 \prec_{\epsilon} \mathbf{x}_2 \iff \begin{cases} \mathbf{x}_1 \prec \mathbf{x}_2 \\ \exists j : f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2) - \epsilon_j \end{cases} \quad (8.3)$$

where  $\epsilon_j$  is the precision of the  $j$ -th objective. This means that if the difference in the objective values are not large enough (i.e., not “measurable”), the configurations are treated as equal (in the objective space). The first criterion is needed to avoid the removal of the extremes of the Pareto-front. Using this method the approximation of the Pareto-front (8.1) is replaced with the Pareto approximate front:

$$P_{\epsilon}(A^{(c)}) := \{\mathbf{x} \in \mathbb{X} \mid \nexists \hat{\mathbf{x}} \in A^{(c)} : \hat{\mathbf{x}} \prec_{\epsilon} \mathbf{x}\}. \quad (8.4)$$

Moreover, the same approach has to be applied to the variable space:

$$\mathbf{x}_1 \equiv_{\epsilon} \mathbf{x}_2 \iff \forall i \in [1 \dots m] : |x_{1,i} - x_{2,i}| \leq \epsilon_i \quad (8.5)$$

where  $x_{1,i}$  is the  $i$ -th component of  $\mathbf{x}_1$  and  $\epsilon_i$  is the precision of the  $i$ -th design variable. This precision is usually easy to be defined. For geometrical variables it is the manufacturing precision; for process variables it is the precision of the control mechanism. This means that two configurations are equivalent if the difference between any two components of two configurations is not larger than a defined value. This approach remains valid because if small measurable differences result in large differences of the objective values, the sensitivities are too large and robust optimization has to be applied [76, 222].

A common problem with genetic algorithms is the unwanted clustering of configurations. When a superior element appears in the Pareto front, it will be favored during the cross-over operations compared to other configurations. As a result, similar configurations are generated in order to explore the nearby domain. If a large number of generations is used, the number of similar (or even completely identical) configurations will significantly increase. This effect is even more pronounced in single-objective optimization. In order to avoid such an issue a modified non-redundant algorithm is proposed for ‘Step 5’, which will generate new elements until  $N_{new}^{(c)} = |F^{(c)}|$  is satisfied, but accepting only elements

that are not equivalent to already existing ones. Thus, the new population has to fulfill the following two criteria:

$$\forall \mathbf{x}_i \in F^{(c)} \quad \nexists \mathbf{x}_j \in A^{(c-1)} : \mathbf{x}_i \equiv_{\epsilon} \mathbf{x}_j \quad (8.6)$$

$$\forall \mathbf{x}_i \in F^{(c)} \quad \nexists \mathbf{x}_j \in F^{(c)} (i \neq j) : \mathbf{x}_i \equiv_{\epsilon} \mathbf{x}_j \quad (8.7)$$

## 8.2.2 Resource-efficient optimization

Every time a new optimization method has been proposed, it has been almost immediately used for some real-life engineering problem. Most algorithms focus on efficiency, but very few focus on reducing the number of evaluations (although exceptions can be found, e.g., [213]); even fewer are focusing specifically on problems associated to Computational Fluid Dynamics [202], as considered in our group. Three different efficiencies can be defined:

- Total time needed for the optimization:  $T(C)$
- Limit behavior (e.g.  $\mathcal{O}(n^2)$ ,  $\mathcal{O}(n \log(n))$  etc.) of administrative operations of the optimization algorithm (big  $\mathcal{O}$  notation or Bachmann-Landau symbol)
- Number of function evaluations:  $N_{tot} = \sum_{c=1}^C |F^{(c)}|$

Although minimizing the second criteria will result in a decrease of  $T(C)$ , it cannot capture all aspects of the process appropriately. If  $t_{comp} \gg \sum_{c=1}^C (t_{gen}^i + t_{upd}^i)$ , the evaluations require much longer time than all other operations in the optimization algorithm together. As a result, many algorithms focus on the minimization of  $\sum_{c=1}^C |F^{(c)}|$ , which does not cover the whole truth either (for  $|F^{(c)}| \bmod p \neq 0$  some resources will be wasted). If  $p$  number of computations can be computed in parallel,  $\sum_{c=1}^C \left\lceil \frac{|F^{(c)}|}{p} \right\rceil$  has to be minimized for an efficient algorithm, i.e.,  $|F^{(c)}| \bmod p = 0$  has to be ensured. In reality  $t_{comp} = \text{constant}$  is not true either, but it has a distribution, see Fig. 8.1 for an example [56]. Future research will try to analyze, identify and take advantage of this distribution during the run-time as well.

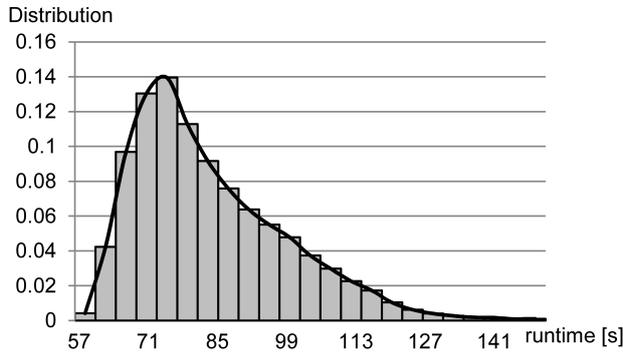


Figure 8.1: Exemplary distribution in CFD-O [56] (run-time of evaluation when varying design)

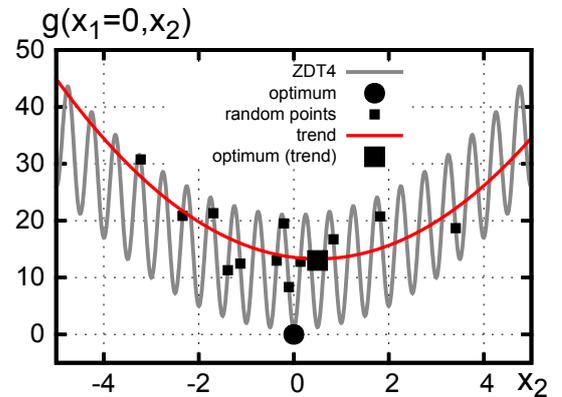


Figure 8.2: Direction of optimum as shown by trend analysis for ZDT4 ( $n = 2$ )

### 8.2.3 Adaptive Operator Selection (AOS)

As already mentioned, different operators can be applied in 'Step 5' but none of them is suited for all different problems. Instead, specific operators should be chosen as function of the considered problem. One may try different operators, and after each step the success of the different operators is evaluated and the number of individuals generated by each method is adjusted. In such a way, not only the configurations compete with each other during the optimization process, but the different operators and methods as well. This increases the adaptivity and applicability of the model. Such an approach is called Adaptive Operator Selection (AOS) [64].

Another adaptive approach is used at the common metamodel-assisted or surrogate-based optimization methods, where the optimization is performed on the metamodel, and if the accuracy of the Response Surface is not accurate enough, more points are added to the database and the optimization is performed again [2].

Here, a combination of the two ideas is used. Several operators compete with each other, but the operators are not necessarily only "simple" genetic operators; some of them are the combinations of metamodels and optimization processes on the metamodels. Although this sounds very time consuming (it is really!), the idea is, that as long as  $t_{comp} \gg 1$ , one may increase  $t_{gen}$  to increase the efficiency. As long as  $t_{gen} \ll t_{comp}$ , even a time-consuming operator will not significantly increase the total runtime.

Another advantage of the proposed approach is the indirect information exchange between the operators. Although in some cases the metamodel might be inaccurate (so that it is impossible to use it directly in an optimization), it might still provide important information concerning the trends, which could be used to accelerate the convergence of a genetic algorithm. Figure 8.2 shows the objective function of ZDT4 ( $n = 2, x_1 = 0$ ). Twelve points were generated randomly and a second-order regression was created. It can be seen that the quadratic metamodel is able to show the direction of the optimum, but not the exact location nor the value. The combination of the local search (by genetic operators) and global search (trend analysis) can lead to drastic speed-ups.

### 8.2.4 Preventing loss of information

Most genetic and iterative algorithms use for Step 7 some kind of filtering mechanism to keep  $A^{(c)}$  moderate in size, thus reducing the burden of the big O notation. However,  $\sum_{i=1}^C |F^{(i)}|$  is usually limited to a couple of hundreds (at the very most thousands) in simulation-based problems. Thus, filtering unnecessarily removes information, as  $P(A^{(C)}) \subseteq P(A^{(0)} \cup \dots \cup A^{(C)})$ , but the equality might not necessarily hold. It is well-known, e.g., that the filtering mechanism might result in deterioration of the NSGA-II algorithm [136]. To avoid this effect, 'Step 7' is simply replaced with  $A^{(c)} := A^{(c-1)} \cup F^{(c)}$ . This also increases the quality of the tested metamodels, as more information is available.

## 8.3 HYENA

In this section, the algorithm of HYENA (**HY**brid **EN**gineering optimization **Al**gorithm) will be explained. The flowchart of HYENA can be seen in Fig. 8.3. In principle, HYENA is a combination of the ideas behind  $\epsilon$ -dominance [136], adaptive population size of Fast-GPA [78], nondominated sorting [68], Adaptive Operator Selection [64] and metamodel-assisted optimization [2].

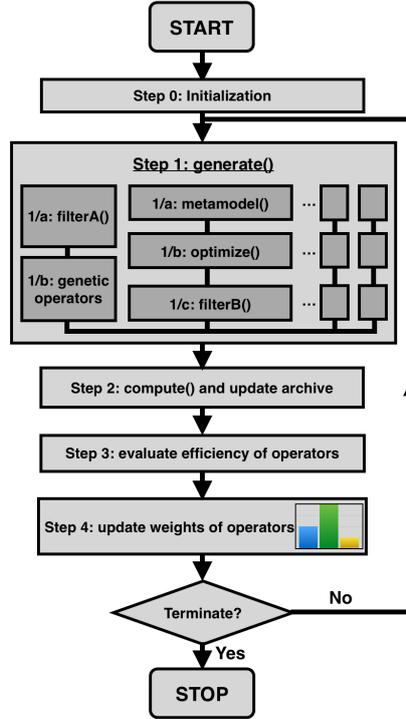


Figure 8.3: Workflow of the optimization method HYENA

### 8.3.1 Step 0: Initialization

In order to avoid unnecessary waste of resources, a reduced number of  $N^{(0)}$  individuals are generated in the first step using a Design-Of-Experiment method: for this purpose, a SOBOL sequence [122] or Near Orthogonal Latin Hypercube [142] method is recommended. Then, the iterative method is started.

### 8.3.2 Step 1: Generating new individuals

This step does not have to be executed in the first cycle. Afterwards, the number of individuals to be generated for the new generation is always dynamically adjusted. This has the advantage that it will waste fewer individuals in the first generations, when not enough information is available concerning the function fields. The number of individuals in a new generation is evaluated as:

$$N_{new}^{(c)} = \left\lfloor \frac{\min(N_0 + |P_\epsilon(A^{(c-1)})|_{c_{inc}, N_{max}})}{p} + 0.5 \right\rfloor p, \quad (8.8)$$

where  $N_0$  is the minimal number of individuals,  $N_{max}$  is the maximal number of individuals and  $c_{inc}$  is the increment factor. This approach was first introduced in the Fast-GPA method [78]. Moreover, this value is rounded to increase the efficiency by ensuring  $N_{new}^{(c)} \bmod p = 0$ .

For each operator  $o$ ,  $N_{op=o}^{(c)} = N_{new}^{(c)} \cdot w_o^{(c)}$  elements are generated (with the necessary rounding), where  $w_o^{(c)}$  is the weight of operator  $o$ . The genetic operator is always denoted with  $o = 0$  in the current study, while all other operators are denoted with  $o = 1, 2, \dots$   $w_o^{(c)}$  is updated dynamically during the optimization.

### 8.3.2.1 Step 1: Genetic operators/a

As in HYENA the archive is simply updated by adding all new elements, its size can become quickly too large to work efficiently with the tournament method. To avoid this problem, the new individuals are generated from a reduced set of parents. The archive  $A^{(c-1)}$  is reduced to  $N_{new}^{(c)}$  elements using the following algorithm:

```

0: filterA(S, N_{new}^{(c)}):
1:  $Q := \emptyset$ 
2:  $r := 0$ 
3: ranking(S)
4: while  $|Q| + |R_r| < N_{new}^{(c)}$  do
5:    $Q := Q \cup R_r$ 
6:    $r := r + 1$ 
7: end while
8: while  $|Q| + |R_r| > N_{new}^{(c)}$  do
9:   crowding(R_r)
10:   $R_r := R_r \setminus p : p_{crowd} \leq q_{crowd} \forall q \in R_r$ 
11: end while
12:  $Q := Q \cup R_r$ 
13: return  $R$ 

```

Hence, the set  $S := A^{(c-1)}$  is reduced by adding all individuals with rank 0,1, ... until the number of total individuals is  $N_{new}^{(c)}$ . Finally, the last rank is reduced by iteratively removing the element with the smallest crowding distance (while recomputing it in each step). This is a modified (more time-consuming) version of the filtering of NSGA-II, since the elements from the last rank are reduced iteratively. However, the modified filtering method still requires significantly smaller time compared to  $t_{comp}$ .

### 8.3.2.2 Step 1: Genetic operators/b

For creating new elements the classical cross-over (SBX [66]) and polynomial mutation operators are applied. Moreover, a non-redundant generation is enforced, as defined by Equations (8.6)-(8.7). For choosing the parents a 2-level tournament was implemented in HYENA.

### 8.3.2.3 Step 1: Metamodel operators/a - Creating the metamodel

For metamodel-based operators the archive ( $A^{(c-1)}$ ) is normalized in the first step to the  $[0; 1]^n$  and  $[0; 1]^m$  domain. Then, a metamodel is created based on the choice of the user. Here  $op = 1$  is a response surface with second-order total expansion (containing all 0, first and second-order effects and all interactions with two variables);  $op = 2$  is a modified moving least squares interpolation (MLSQ, with weight function  $w(r) = ((r/R)^8 + 1)^{-1}$ , where  $R$  depends on the location and density of points) with second-order total expansion;  $op = 0$  is the normal genetic (SBX and mutation) operator.

### 8.3.2.4 Step 1: Metamodel operators/b - Optimization on the metamodel

After the metamodel was created, the original optimization problem is cloned (with all variables and analytical constraints) while replacing the objectives with the virtual metamodel. A virtual optimization is carried out. The number of individuals and generations used in the virtual optimization depends on the cost of the applied operators. For simple Response Surfaces 200x300 (individuals×generations) is appropriate, while for MLSQ 100×100 or 150×150 may be better suited to reduce runtime. The result of this optimization ( $A_v^{(last)}$ ) is sent for filtering to the next step.

### 8.3.2.5 Step 1: Metamodel operators/c - Filtering the results

To improve the exploitation and exploration of the algorithm as well,  $N_{op}^{(c)}$  is further divided into two groups (50-50%). The first group will simply select the best configurations (largest crowding distance) from the Pareto front (rank 0) of the virtual optimization ( $P(A_v^{(last)})$ ). Moreover, redundancy (8.6-8.7) is taken into account to avoid similar configurations. This promotes convergence. The second group takes the remaining individuals and chooses the configurations with the largest minimal distance from the archive, based on the following algorithm. The distance for an existing set ( $S$ ) for a specific configuration  $p$  is defined as  $dist_{var}(p, S) = \min_{q \in S, q \neq p} \|p - q\|_{2, norm, var}$  and  $dist_{obj}(p, S) = \min_{q \in S, q \neq p} \|p - q\|_{2, norm, obj}$ , where  $\|p - q\|_{2, norm, var}$  and  $\|p - q\|_{2, norm, obj}$  is the normalized  $L_2$  norm in the variable and objective space, respectively. Since the objective values from the metamodel might be inaccurate or have a bias, the objective field cannot be taken into account.

```
0: filterB(A, S, N):
1:  r := 0
2:  while r < N do
3:    p := (q : distvar(q, S) ≥ distvar(t, S) ∀t ∈ A)
4:    A := A ∪ p
5:    S := S \ p
6:    r := r + 1
7:  end while
```

where the subroutine has to be called with  $A := A^{(c-1)}$ ,  $S := A_v^{(last)}$ . This ensures that only the best individuals will be considered but only from regions not yet discovered, thus improving the multi-modality and exploration.

### 8.3.3 Step 2: Evaluation and updating archive

In Step 2 the configurations generated by Step 1 (or by the initialization) are evaluated in parallel. Then, the archive is updated ( $A^{(c)} := A^{(c-1)} \cup F^{(c)}$ ) and the rank and crowding distance of the individuals are computed. The ranks are computed similarly to the method developed by Deb [68]:

```

0: ranking( $S$ ):
1:  $r := 0$ 
2: while  $|S| > 0$  do
3:    $R_r := P_\epsilon(S)$ 
4:   for each  $p \in R_r$ 
5:      $p_{rank} := r$ 
6:   end for
7:    $S := S \setminus R_r$ 
8:    $r := r + 1$ 
9: end while

```

This subroutine has to be executed for  $S := A^{(c)}$ . The original form used in the NSGA-II method for the crowding distance is modified by using a weighting mechanism, similar to the one of the Omni-Optimizer [69]:

```

0: crowding( $S$ ):
1: for each  $p \in S$ 
2:    $p_{crowd,var} := dist_{var}(p, S)$ 
3:    $p_{crowd,obj} := dist_{obj}(p, S)$ 
4: end for
5:  $max_{var} := \max_{q \in S}(p_{crowd,var})$ 
6:  $max_{obj} := \max_{q \in S}(p_{crowd,obj})$ 
7: for each  $p \in S$ 
8:    $p_{crowd} := \frac{p_{crowd,var}}{max_{var}} + \frac{p_{crowd,obj}}{max_{obj}}$ 
9: end for

```

The subroutine has to be executed for each  $S := R_r$  separately. Furthermore, for individuals corresponding to the extremes of the objectives and design variables  $p_{crowd} := \infty$  is applied to support exploration. When comparing two configurations, a modified dominance is defined ( $\prec_{combined}$ ), in which first the  $\epsilon$ -dominance is compared; if no configuration is dominating, the configuration with larger crowding distance is chosen.

### 8.3.4 Step 3: Efficiency of the operators

The efficiency of the operators is evaluated based on the ranks using the following method. First, the maximal rank is identified,  $MR_c = \max_{q \in A^{(c)}}(q_{rank})$ . If the set  $S$  contains all individuals generated by a specific operator  $o$ , then its efficiency is defined as

$$success_o = \frac{\sum_{q \in S} (MR_c - q_{rank})}{|S|}. \quad (8.9)$$

Larger values indicate a more successful operator. If an operator is always efficient (generating only individuals with rank 0),  $success_o = MR_c$ , while a completely unsuccessful operator results in  $success_o = 0$ .

### 8.3.5 Step 4: Updating weights

Based on the success of the different operators, the weights have to be updated. First the change in weight is computed:

$$\Delta w_o = \frac{success_o - \overline{success_o}}{\sum_o success_o}. \quad (8.10)$$

If the largest change exceeds an allowed limit, all changes are scaled down with  $\frac{\Delta_{max}}{\max \Delta w_o}$  and the new weights are computed as  $w_o^{(c+1)} = w_o^{(c)} + \Delta w_o$ . The success of the operators can only be tested, if a minimal number of individuals were generated. Thus, each weight is corrected using:

$$w_o^{(c+1)} = \max \left( w_o^{(c+1)}, \frac{N_{min,o}}{N^{(c+1)}} \right), \quad (8.11)$$

where  $N_{min,o}$  is the minimal number of individuals to be generated by each operator. Using this method, it is not sure that  $\sum_o w_o = 1$  will be fulfilled. Thus, the largest weight is corrected to ensure  $\sum_o w_o = 1$ .

In the current study the weights were initialized with  $w_0 = 0.75$ ,  $w_1 = 0.125$ ,  $w_2 = 0.125$  or  $w_0 = 0.75$ ,  $w_1 = 0.25$  and  $N_{min} = 5$ . The algorithm was implemented in OPAL++.

## 8.3.6 Summary

### 8.3.6.1 Summary of the applied ideas

The most important properties of HYENA compared to classical multi-objective genetic optimization are:

- During the optimization method, no individuals are eliminated from the archive to prevent deterioration and loss of information.
- In order to take into account the accuracy of the objective values,  $\epsilon$ -dominance is applied.
- To avoid redundancy in the archive, no individual can be generated that would be equivalent to an existing configuration up to a defined precision.

- The size of the generations is dynamically adjusted to avoid wasting function evaluations. The number of parallel computing units (processor, computers, licences) are taken into account to maximize the efficiency.
- Different operators are used to generate new individuals. Not only the individuals, but the operators are competing with each other, increasing the adaptivity of the method.
- Surrogate models are used parallel to genetic methods, and information is exchanged between the methods to ensure the highest efficiency.

### 8.3.6.2 Summary of parameters

In the followings, the list of parameters necessary for HYENA are summarized, with recommended values:

- $p$ : Number of parallel nodes used in the optimization (depends on the compute cluster)
- $N_0$ : Minimal number of individuals in each generation (recommended value: 10-40)
- $N_{max}$ : Maximal number of individuals in each generation (recommended value: 50-200)
- $c_{inc}$ : Increment factor for the number of individuals in each generation (recommended value: 1)
- $\epsilon$ : value needed for  $\epsilon$ -dominance
- list of operators: the operators to be used during the optimization (recommended: one genetic operator, one global trend analysis, one MLSQ method)
- operator settings: as the user may choose any operator, the necessary settings (e.g., distribution index) have to be specified for each operator
- $w_i$ : starting weight for each operator (recommended value: 0.75 for genetic operator, the rest equally distributed)
- $N_{min}$ : minimal number of individuals generated with each individual (recommended value: 5)

As one can see, for most parameters the recommended values are applied throughout the study, the users should in principle only change  $N_0$ ,  $N_{max}$  and the list of operators (Note: although  $p$  is important as well, e.g., parallel MPI implementations receive this value automatically at startup).

## 8.4 Results

For the initialization of the optimizations, SOBOL method was used, for the genetic operators  $\eta_c = 20$ ,  $\eta_m = 20$  was applied for NSGA-II and HYENA. If not stated otherwise,  $\epsilon = 0.01$  was applied in HYENA with  $c_{inc} = 1$ ,  $p = 10$ . For NSGA-II the

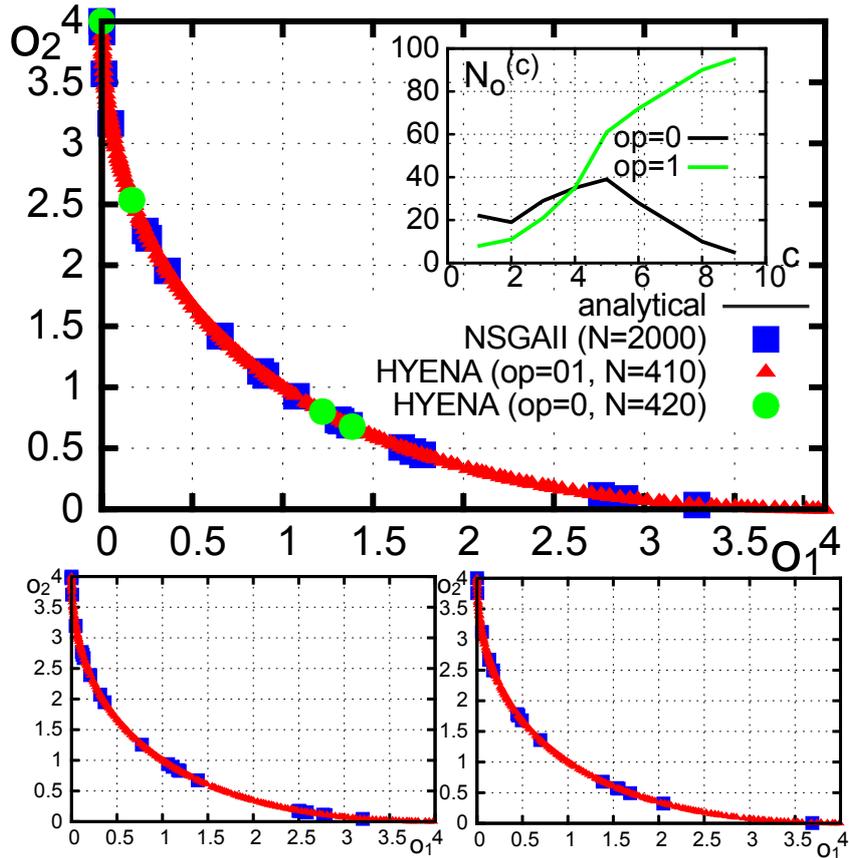


Figure 8.4: Solution of the test-function SCH1

recommended settings were retained following [68] ( $N = 100$ ). In all tests the optimizations were stopped when one of the methods managed to approximate the Pareto front with high precision. Besides the presented cases, additional tests were analyzed as well. For Schaffer's study [191] HYENA overperformed NSGA-II significantly due to the simple response surface and solved the problem with 400 real function evaluations. For ZDT4 neither NSGAI nor HYENA could solve the problem considering 1000 function evaluations.

Please note, that one could compare the method against other adaptive, hybrid methods as well, but most of these algorithms are commercial and proprietary.

### 8.4.1 SCH1

In the first test the problem from Schaffer's study (SCH1 [191], convex) was tested:

$$\textcircled{O} : \begin{cases} f_1(x) = x^2 \\ f_2(x) = (x - 2)^2 \\ -1000 \leq x \leq 1000 \end{cases} . \quad (8.12)$$

SCH1 is a very simple analytical problem. However, due to the large domain, evolutionary algorithms often have problem with it. HYENA was tested with  $op = 0$  and  $op = 01$  operators with  $N_0 = 30$  and  $N_{max} = 100$ . For the virtual optimization 200 individuals

Table 8.1: SCH1 statistics

Method	gen.	$\overline{N_{tot}}$	$Var(N_{tot})$	$\overline{N_P}$	$\overline{\gamma}$	$\overline{d_1}$
NSGA-II	10	2000	0	8.6	0.042	2.47
HYENA ( $op = 01$ )	7	410	$3.5^2$	212	0.0016	0.086

were used with 300 generations; the overhead is not significant. Figure 8.4 shows the result with HYENA ( $op = 01$ ) after 410 evaluations, the result with HYENA ( $op = 0$ ) after 420 evaluations and the results of NSGA-II after 2020 evaluations. The right top side shows the number of individuals generated with operator 0 (genetic), 1 (RSM). The bottom two figures show two independent tests only with HYENA ( $op = 01$ ) and NSGA-II. With  $op = 01$  HYENA almost immediately switches to the metamodel operator as the objective functions can be reconstructed with a second-order trend analysis. This test is somewhat unfair due to the very simple function. However, it indicates that if the model is simple, HYENA can effectively switch to a Response Surface Approach. Within 400 full evaluations, HYENA finds the whole Pareto-front, while NSGA-II is not able to identify the whole front even in 2000 evaluations.

Statistically, the same tendency was observed. The optimization was executed 200 times with the same DOE for the initial generation. The results are presented in Table 8.1 for generations 10 and 7, respectively, showing the average and variance of the total number of evaluations, the average of the number of solutions in the Pareto front ( $N_P$ ), the average convergence metric ( $\overline{\gamma}$ ) and the average of the maximal distance between consecutive elements in the Pareto front (denoted by  $\overline{d_1}$ ). In the present case instead of the spread ( $\Delta$ ) the (first) maximal distance between consecutive elements in the Pareto front (denoted by  $\overline{d_1}$ ) is presented, as it is better in this case to show how large the gaps are within the Pareto front. Although for fully converged Pareto fronts the spread is a better suited metrics, in the present case the convergence is not complete yet in all cases.

## 8.4.2 KUR

In the second test Kursawe’s study (KUR [132]) was considered, which is a non-convex problem. For the definition please refer to [68]:

$$\textcircled{O} : \begin{cases} f_1(x) = \sum_{i=1}^2 \left( -10 \exp \left( -0.2 \sqrt{(x_i^2 + x_{i+1}^2)} \right) \right) \\ f_2(x) = \sum_{i=1}^3 (|x_i|^{0.8} + 5 \sin x_i^3) \\ -5 \leq x_1, x_2, x_3 \leq 5 \end{cases} . \quad (8.13)$$

The results of HYENA with  $op = 012$ ,  $N_0 = 40$  and  $N_{max} = 100$  can be seen in Fig. 8.5. For the virtual optimization 100 generations were used with 100 individuals. Again, HYENA is in all cases superior to NSGA-II in convergence and spread. At the beginning the trend analysis  $op = 1$  is able to provide a small boost, but is turned off shortly afterwards due to the complex objective field. Then,  $op = 2$  becomes effective after the 5th generation, when enough information is present. This results in a superior exploration of the Pareto front as the algorithm is able to detect at which point enough information

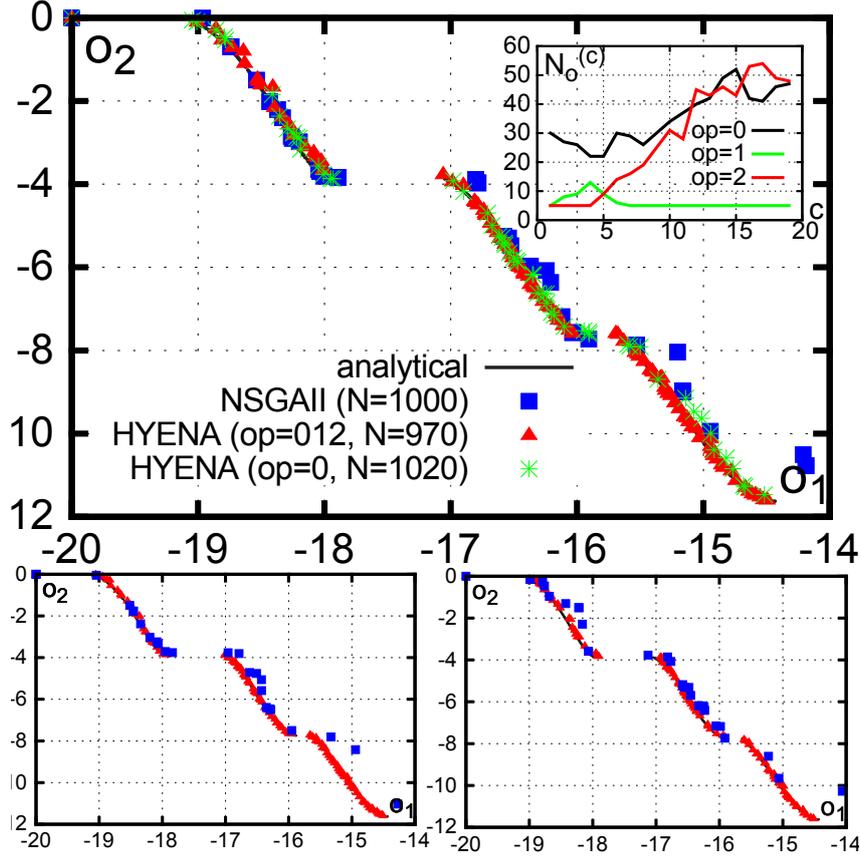


Figure 8.5: Solution of the test-function KUR

Table 8.2: KUR statistics

Method	gen.	$\overline{N_{tot}}$	$Var(N_{tot})$	$\overline{N_P}$	$\overline{\gamma}$	$\overline{d_4}$
NSGA-II	10	1000	0	40	0.145	1
HYENA ( $op = 012$ )	16	892	$92^2$	100	0.047	0.53

is available for the metamodel. Even before this point it focuses on the Pareto front due to the genetic optimization method.

The statistics (64 samples) for the optimization are presented in Table 8.2. As the Pareto-front is discontinuous (3 gaps), the fourth largest distance between consecutive elements of the Pareto front is presented in the statistics ( $d_4$ ), as it is able to show how large the gaps are within the Pareto front. To show that the relation between different metrics is not only valid for a chosen time-step, the evolution of the convergence metric and the number of elements in the Pareto-front ( $N_P$ ) in function of the number of total function evaluations ( $N_{tot}$ ) is presented in Fig. 8.6. As one can see, HYENA converges much faster and is superior in every sense.

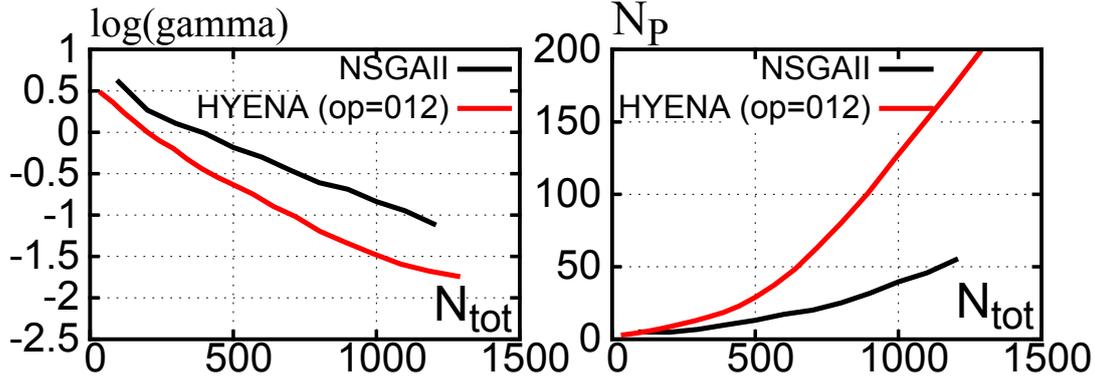


Figure 8.6: Evolution of different metrics for KUR

### 8.4.3 ZDT4 (n=4)

ZDT4 ([232]) is a very complex problem with a large number of local extrema. The case  $n = 4$  case was selected:

$$\textcircled{O} : \begin{cases} f_1(x) = x_1 \\ f_2(x) = g(\mathbf{x})[1 - \sqrt{x_1/g(\mathbf{x})}] \\ g(\mathbf{x}) = 10n - 9 + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i)) \\ 0 \leq x_1 \leq 1 \\ -5 \leq x_i \leq 5, i = 2 \dots n \end{cases} \quad (8.14)$$

As one can see (Fig. 8.7), it is not possible to solve such a complex problem with such a small number of evaluations, neither with NSGA-II nor with HYENA. In this case, HYENA behaves similarly to NSGA-II, since it simply switches back to normal genetic optimization. For the present case no statistics are presented, as neither method could solve the problem with such a small number of evaluations.

### 8.4.4 FON

The problem proposed in Fonseca and Fleming's study [90] is a non-convex problem.

$$\textcircled{O} : \begin{cases} f_1(x) = 1 - \exp\left(-\sum_{i=1}^3 \left(x_i - \frac{1}{\sqrt{3}}\right)^2\right) \\ f_2(x) = 1 - \exp\left(-\sum_{i=1}^3 \left(x_i + \frac{1}{\sqrt{3}}\right)^2\right) \\ -4 \leq x_1, x_2, x_3 \leq 4 \end{cases} \quad (8.15)$$

The results of HYENA with  $op = 012$ ,  $N_0 = 40$  and  $N_{max} = 100$  can be seen in Fig. 8.8. For the virtual optimization 150 generations were used with 150 individuals. Once more, HYENA is in all cases superior to NSGA-II. At the beginning the trend analysis  $op = 1$  is able to provide a larger boost, but is shortly afterwards turned off due to the complex objective field and  $op = 2$  becomes more effective. The  $op = 0$  version is also faster compared to NSGA-II, but leads to a worse convergence compared to  $op = 012$ . Using 64 samples, the statistics of the different metrics are presented in Table 8.3, showing the improved convergence and better coverage of the Pareto front with HYENA.

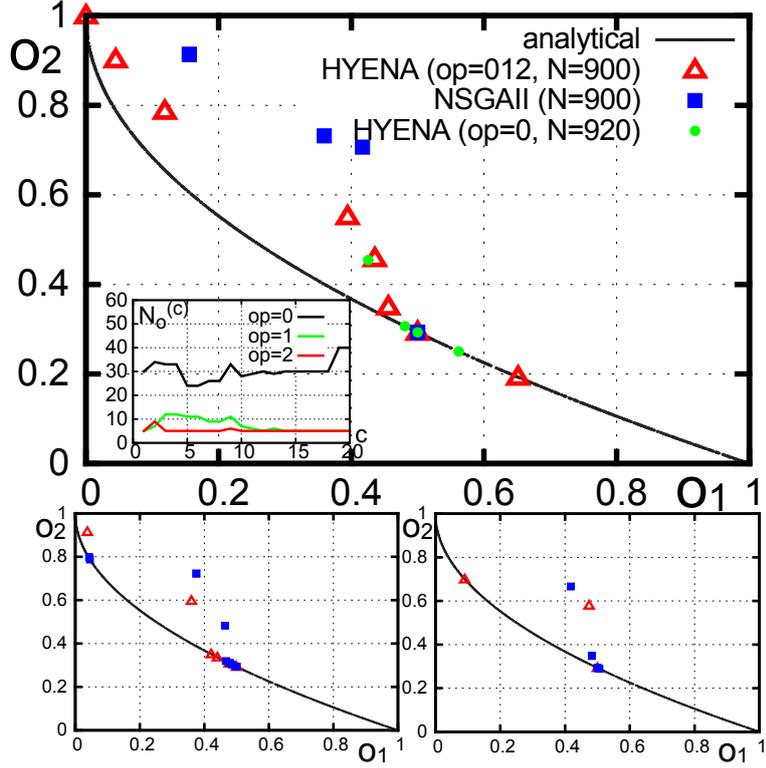


Figure 8.7: Solution of the test-function ZDT4 ( $n=4$ )

Table 8.3: FON statistics

Method	gen.	$\overline{N}_{tot}$	$Var(N_{tot})$	$\overline{N}_P$	$\overline{\gamma}$	$\overline{d}_1$
NSGA-II	6	600	0	29	0.023	0.20
HYENA ( $op = 012$ )	8	494	$22^2$	116	0.003	0.065

### 8.4.5 POL

Poloni's problem (POL [180]) is non-convex and disconnected.

$$\textcircled{O} : \begin{cases} f_1(x) = 1 + (A_1 - B_1)^2 + (A_2 - B_2)^2 \\ f_2(x) = (x_1 + 3)^2 + (x_2 + 1)^2 \\ A_1 = 0.5 \sin(1) - 2 \cos(1) + \sin(2) - 1.5 \cos(2) \\ A_2 = 1.5 \sin(1) - \cos(1) + 2 \sin(2) - 0.5 \cos(2) \\ B_1 = 0.5 \sin(x_1) - 2 \cos(x_1) + \sin(x_2) - 1.5 \cos(x_2) \\ B_2 = 1.5 \sin(x_1) - \cos(x_1) + \sin(x_2) - 0.5 \cos(x_2) \\ -\pi \leq x_1, x_2 \leq \pi \end{cases} . \quad (8.16)$$

The results can be seen in Fig. 8.9, which shows the two disconnected parts of the Pareto front for both cases. HYENA ( $op = 012$ ) outperforms NSGA-II in both convergence and spread in all cases, illustrating the efficiency of the method. It is interesting to see that, even for disconnected problems, the implemented Moving Least Squares method works

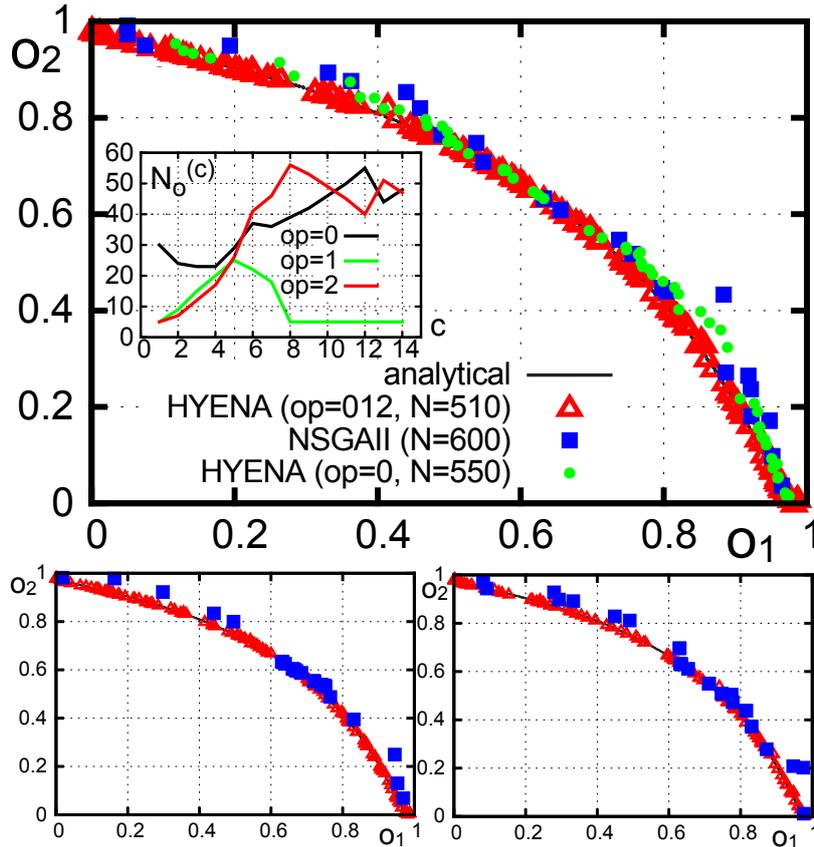


Figure 8.8: Solution of the test-function FON

Table 8.4: POL statistics

Method	gen.	$\overline{N}_{tot}$	$Var(N_{tot})$	$\overline{N}_P$	$\overline{\gamma}$	$\overline{d}_2$
NSGA-II	6	600	0	45	0.17	2.57
HYENA ( $op = 012$ )	8	523	$14^2$	187	0.015	0.46

well.

The statistics (64 samples) are presented in Table 8.4, using the second largest distance ( $d_2$ ) since the Pareto-front is discontinuous. The convergence metric is more than 10 times smaller with HYENA, while the number of points in the Pareto front is more than 4 times larger, with a smaller number of function evaluations.

When the evolution of the convergence metric,  $d_2$  and the number of elements in the Pareto-front ( $N_P$ ) as function of the number of total function evaluations, the same tendency was revealed, i.e., HYENA converges much faster and is superior in every respect.

#### 8.4.6 XFOIL-based optimization

Finally, a realistic engineering optimization problem is considered. For this purpose, a blade-shape optimization was performed using XFOIL [72] for the evaluations. The

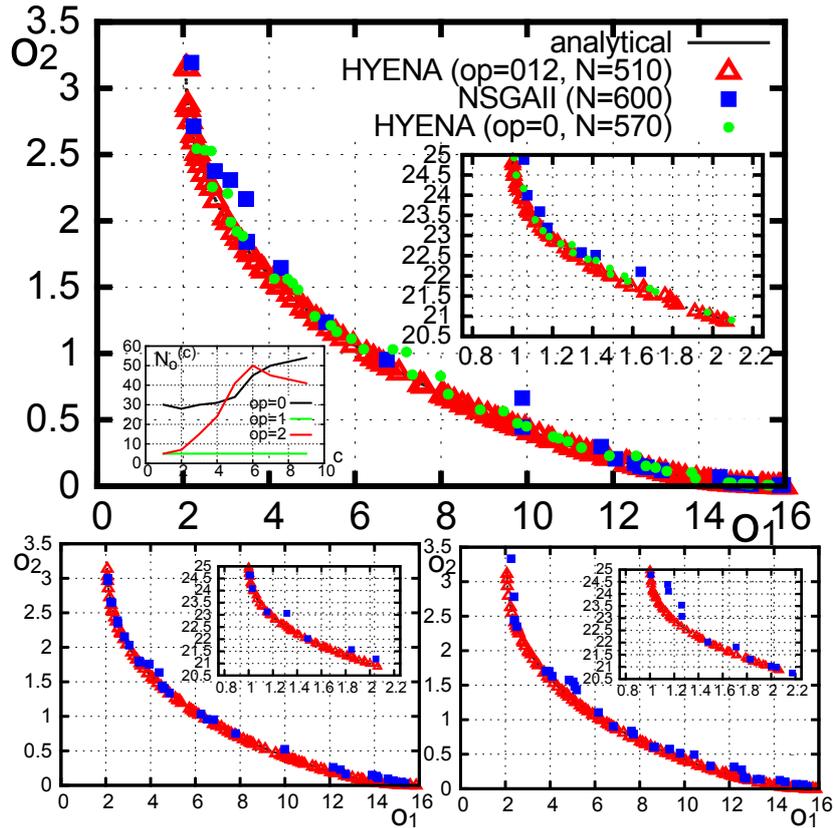


Figure 8.9: Solution of the test-function POL

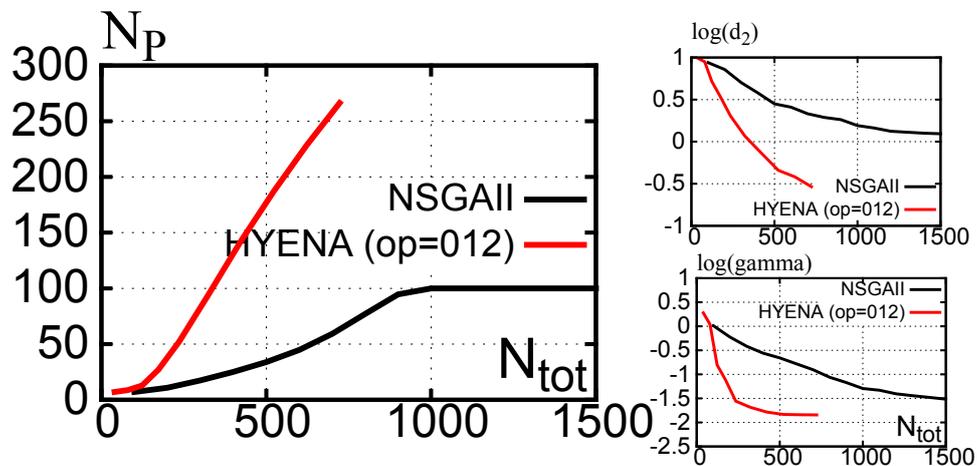


Figure 8.10: Evolution of different metrics for POL

software XFOIL uses a panel method for the evaluation of the aerodynamic coefficients around blades. A Reynolds number  $Re=400\,000$  (typical for wind turbines) was retained in the present case with  $N_{crit} = 4$ . The geometry of the airfoil was generated by OPAL++ based on the NACA4 parameterization, and loaded into XFOIL.

XFOIL computes the different coefficients for every  $\Delta\alpha = 0.025^\circ$  with 200 iterations and the stall angle is approximated as  $\left. \frac{dC_L}{d\alpha} \right|_{\alpha_{stall}} = 0$ . Lift and drag coefficients were determined for the stall angle and the lift to drag ratio was computed for this angle as

well. The efficiency of wind turbines is mostly influenced by  $C_L/C_D$  [176]. For Darrieus-type wind turbines the stall angle is a critical parameter, as this value influences the separation (airfoils with higher stall angle are able to operate under higher angle of attack, thus at lower rotation speed, allowing lightweight components). To address both issues, the following optimization was formulated using  $m$  (relative camber in percent),  $p$  ( $10\times$  relative location of maximal camber) and  $t$  (relative thickness in percent) as parameters for a NACA4 profile:

$$\textcircled{O} : \begin{cases} f_1(x) = \frac{C_L}{C_D} \Big|_{\alpha=\alpha_{stall}} \rightarrow \max \\ f_2(x) = \alpha_{stall} \rightarrow \max \\ 0 \leq m \leq 7.5, 2 \leq p \leq 5, 15 \leq t \leq 25 \end{cases} \quad (8.17)$$

To avoid airfoils that would not be able to withstand high bending moments, the imposed minimal thickness was set to 15%.

For the optimization, the tolerance of the variables is chosen as  $\epsilon = 0.01$ . For the objectives,  $\epsilon = 0$  was applied with  $N_0 = 40$ ,  $N_{max} = 100$ ; for NSGA-II  $N = 80$  was retained. Only the  $op = 2$  operator was enabled, as the response would be too complicated for quadratic Response Surfaces. A statistical evaluation with several hundreds of optimizations was not possible in this case, as a single evaluation already requires 10-46 minutes (average: 18 minutes CPU). Thus, the whole optimization requires several hours even in parallel. For the virtual optimization, 200 individuals were used with 300 generations. Although this results in a long runtime, the corresponding overhead (1-3 hours) is equivalent to less than 10 evaluations, which is deemed acceptable. Although in the present study the virtual optimization was run in serial, there is no practical limitation to take advantage of the parallel environment used for the simulations. When using several computers or cores, a common 8-16 core workstation reduces this overhead to a couple of minutes.

The results are shown in Fig. 8.11, for three different independent optimization runs. As one can see, the MLSQ method ( $op = 2$ ) is able to successfully generate new elements, which results in a better covered and better converged Pareto front. Some elements were also found in the  $o_1 < 10$  domain as well, but are not shown, since they are irrelevant for practical purposes. The highest regions were only discovered by HYENA. The NSGA-II method did not converge completely to the Pareto front and is less well-spread in all domains. Certainly, NSGA-II would converge to the same front in later generations, but with a considerably larger number of function evaluations.

In a second test the optimization was performed in parallel on 16 cores. In this case, both NSGA-II (400 evaluations) and HYENA (350 evaluations) needed 7 hours of CPU to finish the test, but the results of HYENA were superior. This signifies, that HYENA with MLSQ becomes more efficient and faster, when a single function evaluation requires more than 20 minutes.

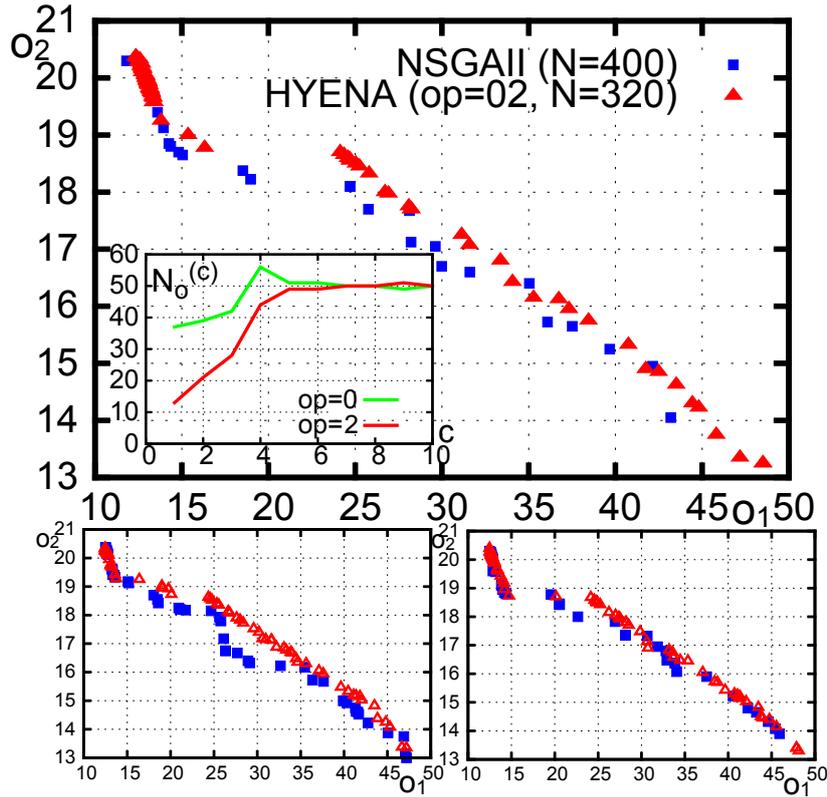


Figure 8.11: Solution of the XFOIL-based optimization

## 8.5 Conclusions and summary

In the present study a new hybrid multi-objective optimization algorithm was proposed for simulation-based engineering problems. In this approach, called HYENA, metamodels and genetic operators are combined using Adaptive Operator Selection (AOS). Hence, not only the individuals but also the different methods are competing with each other. In this manner, the adaptivity of the method increases significantly and the obtained efficiency is higher.

The proposed method was tested considering a variety of analytical problems. At the end, a real simulation-based engineering problem was investigated, using XFOIL to perform the evaluations. HYENA was able to perform significantly better in all cases but one (ZDT4). Only in this most complex case, HYENA switches back automatically to the standard genetic optimization method, and thus performs comparably to NSGA-II. As a whole, the method proved to be very promising. Future research will focus on further benchmarking HYENA by comparisons with multiploid methods [173], concentrating on complex multi-objective applications.

*This concludes the present thesis.*

# Summary

“Optimization saturates what we do and drives almost every aspect of engineering.”

*Dennis Bernstein*

In the present thesis, the importance and multidisciplinary aspects of CFD-O have been presented by using a few selected engineering problems. Most importantly, an own software was developed to carry out all studies in an efficient manner. The examples were selected so, that all aspects of the multidisciplinary CFD-based optimization process could be analyzed (see Fig. 8.12).

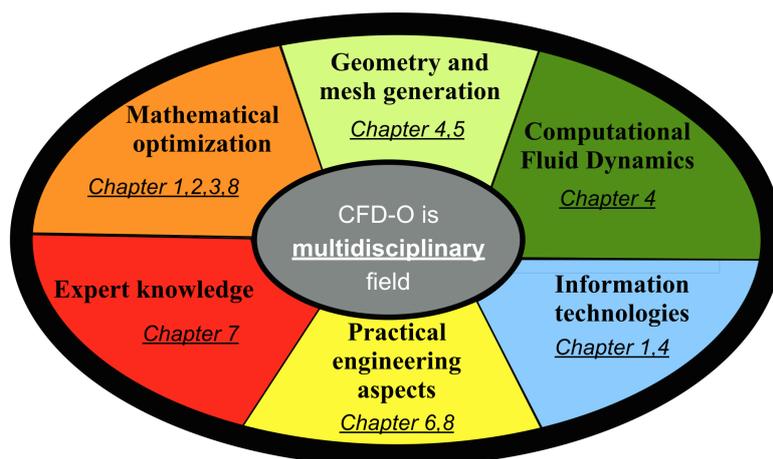


Figure 8.12: Multidisciplinary aspects of CFD-O

Using these selected examples it was presented,

- that mathematical optimization is very different from practical engineering optimization processes, though they share many aspects (see Chapter 1). Thus, specialized methods are required, as proposed in the present thesis (HYENA, see Chapter 8);
- that developing an appropriate in-house or well-suited commercial software can increase the efficiency of engineers significantly (see Chapter 1);
- that even the simplest problem can result in a topologically complex Pareto front. Thus, appropriate care should be given for the post-processing and analysis phase of multi-objective optimizations (see Chapter 2);

- that using an unnecessarily high number of constraints will hinder convergence (see Chapter 2);
- that choosing an appropriate parameterization can speed-up convergence and increase exploration of the problem, even when using the same number of variables (see Chapter 3 and Chapter 4);
- that uncertainties can be quantified by using an appropriate UQ method even for physically complex cases; one should be careful when choosing too simple models or assuming symmetric distributions (see Chapter 6);
- that mesh generation can be efficiently and robustly implemented for 2D and even for 3D cases, by using a systematic approach and using appropriate mesh generation tools (see Chapter 4, 5, 7);
- that even problems, where interaction with experts cannot be eliminated, can be efficiently supported by optimization to speed-up the process (see Chapter 7);
- that metamodel-based optimization can be efficient for many problems, but direct CFD-based optimization with small number of iterations ( $< 1000$ ) suffices in many cases, when choosing an appropriate parameterization, automation and optimization method (see Chapter 4).

Research is being continued regarding many aspects. Methods such as robust optimization, multi-fidelity optimization processes, hybrid global-local search methods have been only rarely applied in the corresponding scientific literature. To introduce these promising methods in everyday engineering processes, there is still a long way to go.

# Bibliography

- [1] ABDALLAH, I., NATARAJAN, A., AND SØRENSEN, J. Predicting extreme loads effects on wind turbines considering uncertainty in airfoil data. In *Safety, Reliability, Risk and Life-Cycle Performance of Structures and Infrastructures - Proceedings of the 11th International Conference on Structural Safety and Reliability, ICOSSAR 2013* (2013), pp. 215–222.
- [2] AHMED, M. AND QIN, N. Surrogate-based aerodynamic design optimization: Use of surrogates in aerodynamic design optimization. In *13th Conference on Aerospace Sciences & Aviation Technology* (2009).
- [3] ALMOHAMMADI, K., INGHAM, D., MA, L., AND POURKASHAN, M. Computational fluid dynamics (CFD) mesh independency techniques for a straight blade vertical axis wind turbine. *Energy* 58 (2013), 483–493.
- [4] AMATO, F., BEDON, G., CASTELLI, M. R., AND BENINI, E. Numerical Analysis of the Influence of Tip Devices on the Power Coefficient of a VAWT. *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering* 7, 6 (2013), 1053–1060.
- [5] AMET, E. *Simulation numérique d'une hydrolienne à axe vertical de type Darrieus*. PhD thesis, Polytechnical National Institute, France, Grenoble, 2009.
- [6] ANSYS INC. *Ansys Fluent 14.0: Users Guide*. Canonsburg, PA, 2011.
- [7] ANTIGA, L., PICCINELLI, M., BOTTI, L., ENE-IORDACHE, B., REMUZZI, A., AND STEINMAN, D. A. An image-based modeling framework for patient-specific computational hemodynamics. *Medical & Biological Engineering & Computing* 46, 11 (2008), 1097–1112.
- [8] ANZAI, H., OHTA, M., FALCONE, J.-L., AND CHOPARD, B. Optimization of flow diverters for cerebral aneurysms. *Journal of Computational Science* 3, 1-2 (2012), 1–7.
- [9] BALDOCCHI, D. Lecture 18, Wind and Turbulence, Part 3, Surface Boundary Layer: Theory and Principles. *Biometeorology, ESPM 129, University of California* (2012).
- [10] BALDUZZI, F., BIANCHINI, A., MALECI, R., FERRARA, G., AND FERRARI, L. Blade design criteria to compensate the flow curvature effects in H-Darrieus wind turbines (GT2014-26389). In *Proceedings of the ASME Turbo Expo* (2014).

- [11] BALDUZZI, F., BIANCHINI, A., MALECI, R., FERRARA, G., AND FERRARI, L. Critical issues in the CFD simulation of Darrieus wind turbines. *Renewable Energy* 85 (2016), 419–435.
- [12] BAODONG, S., LIFENG, W., JIANYUN, L., AND HEMING, C. Multi-objective optimization design of a micro-channel heat sink using adaptive genetic algorithm. *International Journal of Numerical Methods for Heat and Fluid Flow* 21, 3 (2011), 353–364.
- [13] BARANYI, L. Computation of unsteady momentum and heat transfer from a fixed circular cylinder in laminar flow. *Journal of Computational and Applied Mechanics* 4, 1 (2003), 13–25.
- [14] BARANYI, L., AND DARÓCZY, L. Mechanical energy transfer and flow parameter effects for a cylinder in transverse oscillation. *International Journal of Heat and Fluid Flow* 43 (2013), 251–258.
- [15] BARKLEY, D., AND HENDERSON, R. D. Three-dimensional Floquet stability analysis of the wake of a circular cylinder. *Journal of Fluid Mechanics* 322 (1996), 215–241.
- [16] BEDON, G., CASTELLI, M. R., AND BENINI, E. Optimization of a Darrieus vertical-axis wind turbine using blade element – momentum theory and evolutionary algorithm. *Renewable Energy* 59 (2013), 184–192.
- [17] BELFERHAT, S., MEFTAH, S.M.A., YAHIAOUI, T., AND IMINE, B. Aerodynamic Optimization of a Winglet Design. *EPJ Web of Conferences* 45 (2013), 01010.
- [18] BELLO-OCHEDE, T., OLAKOYEJO, O., MEYER, J., BEJAN, A., AND LORENTE, S. Constructal flow orientation in conjugate cooling channels with internal heat generation. *International Journal of Heat and Mass Transfer* 57, 1 (2013), 241–249.
- [19] BIANCHI, F. D., DE BATTISTA, H., AND MANTZ, R. J. *Wind Turbine Control Systems*. Springer London, 2007.
- [20] BIANCHINI, A., FERRARA, G., AND FERRARI, L. Design guidelines for H-Darrieus wind turbines: Optimization of the annual energy yield. *Energy Conversion and Management* 89 (2015), 690–707.
- [21] BIANCHINI, A., FERRARA, G., FERRARI, L., AND MAGNANI, S. An improved model for the performance estimation of an H-Darrieus wind turbine in skewed flow. *Wind Engineering* 36, 6 (2012), 667–686.
- [22] BIDAROUNI, A., AND DJAVARESHKIAN, M. An optimization of wind turbine airfoil possessing good stall characteristics by genetic algorithm utilizing CFD and neural network. *International Journal of Renewable Energy Research* 3, 4 (2013), 993–1003.

- [23] BONNEVILLE, F., SOUROUR, N., AND BIONDI, A. Intracranial aneurysms: an overview. *Neuroimaging Clinics of North America* 16, 3 (2006), 371–82, vii.
- [24] BOOKER, J., AND ROSS, T. An evolution of uncertainty assessment and quantification. *Scientia Iranica* 18, 3 (2011), 669–676.
- [25] BOTTASSO, C., CAMPAGNOLO, F., CROCE, A., DILLI, S., GUALDONI, F., AND NIELSEN, M. Structural optimization of wind turbine rotor blades by multilevel sectional/multibody/3D-FEM analysis. *Multibody System Dynamics* (2013), 1–30.
- [26] BOUILLOT, P., BRINA, O., OUARED, R., YILMAZ, H., FARHAT, M., ERCEG, G., LOVBLAD, K.-O., VARGAS, M. I., KULCSAR, Z., AND PEREIRA, V. M. Geometrical deployment for braided stent. *Medical Image Analysis* 30 (2016), 85–94.
- [27] BRAIN ANEURYSM FOUNDATION. [www.bafound.org](http://www.bafound.org) (date of last access: 21.04.2016).
- [28] BRAVO, R., TULLIS, S., AND ZIADA, S. Performance Testing of a Small Vertical-Axis Wind Turbine. In *Proceedings of the 21st Canadian Congress of Applied Mechanics (CANCAM07)* (Toronto, Canada, 2007), pp. 470–471.
- [29] BUNDESMINISTERIUM FÜR WIRTSCHAFT UND TECHNOLOGIE. Zahlen und Fakten - Energiedaten (Date of last access: 15.04.2014), 2012.
- [30] CAMPOBASSO, M. S., GIGANTE, F., AND DROFELNIK, J. Turbulent Unsteady Flow Analysis of Horizontal Axis Wind Turbine Airfoil Aerodynamics Based on the Harmonic Balance Reynolds-Averaged Navier-Stokes Equations (GT2014-25559). In *ASME Turbo Expo Conference* (Düsseldorf, 2014).
- [31] CAMPOBASSO, M. S., MINISCI, E., AND CABONI, M. Aerodynamic design optimization of wind turbine rotors under geometric uncertainty. *Wind Energy* 19, 1 (2014), 51–65.
- [32] CAPUZZI, M., PIRRERA, A., AND WEAVER, P. A novel adaptive blade concept for large-scale wind turbines. Part I: Aeroelastic behaviour. *Energy* 73 (2014), 15–24.
- [33] CAPUZZI, M., PIRRERA, A., AND WEAVER, P. A novel adaptive blade concept for large-scale wind turbines. Part II: Structural design and power performance. *Energy* 73 (2014), 25–32.
- [34] CASTELLI, M. R., ARDIZZON, G., BATTISTI, L., BENINI, E., AND PAVESI, G. Modeling Strategy and Numerical Validation for a Darrieus Vertical Axis Micro-Wind Turbine (IMECE2010-39548). *ASME 2010 International Mechanical Engineering Congress and Exposition* 7 (2010), 409–418.
- [35] CASTELLI, M. R., ENGLARO, A., AND BENINI, E. The Darrieus wind turbine: Proposal for a new performance prediction model based on CFD. *Energy* 36, 8 (2011), 4919–4934.

- [36] CASTRO, M. A., OLIVARES, M. C. A., PUTMAN, C. M., AND CEBRAL, J. R. Unsteady wall shear stress analysis from image-based computational fluid dynamic aneurysm models under Newtonian and Casson rheological models. *Medical & Biological Engineering & Computing* 52, 10 (2014), 827–839.
- [37] CD-ADAPCO. *USER Guide Star-CCM+ Version 9.06*. CD-adapco, 2014.
- [38] CEBRAL, J. R., MUT, F., WEIR, J., AND PUTMAN, C. Quantitative characterization of the hemodynamic environment in ruptured and unruptured brain aneurysms. *American Journal of Neuroradiology* 32, 1 (2011), 145–151.
- [39] CHANDRASHEKARAPPA, P., AND DUVIGNEAU, R. Radial basis functions and kriging metamodels for aerodynamic optimization. Research Report RR-6151, INRIA, 2007.
- [40] CHAUDHARI, A. *Large-eddy simulation of wind flows over complex terrains for wind energy applications*. PhD thesis, Lappeenranta University of Technology, 2014.
- [41] CHAUDHARI, A. C., VUORINEN, V., AGAFONOVA, O., HELLSTEN, A., AND HÄMÄLÄINEN, J. Large-Eddy simulations for atmospheric boundary layer flows over complex terrains with applications in wind energy. In *6th European Conference on Computational Fluid Dynamics (ECFD VI)* (Barcelona, Spain, 2014), E. Oñate, X. Oliver, and A. Huerta, Eds.
- [42] CHENG, X. Entropy resistance minimization: An alternative method for heat exchanger analyses. *Energy* 58 (2013), 672–678.
- [43] CLIFTON-SMITH, M. Aerodynamic noise reduction for small wind turbine rotors. *Wind Engineering* 34, 4 (2010), 403–420.
- [44] COELLO, C. A. C. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering* 191, 11–12 (2002), 1245–1287.
- [45] COELLO, C. C., LAMONT, G., AND VAN VELDHUIZEN, D. *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed. Genetic and Evolutionary Computation. Springer, Berlin, Heidelberg, 2007.
- [46] COPIELLO, D., AND FABBRI, G. Multi-objective genetic optimization of the heat transfer from longitudinal wavy fins. *International Journal of Heat and Mass Transfer* 52, 5-6 (2009), 1167–1176.
- [47] CURTIS, S. L., BRADLEY, M., WILDE, P., AW, J., CHAKRABARTI, S., HAMILTON, M., MARTIN, R., TURNER, M., AND STUART, A. G. Results of screening for intracranial aneurysms in patients with coarctation of the aorta. *American Journal of Neuroradiology* 33, 6 (2012), 1182–1186.
- [48] DA SILVA, A., LORENTE, S., AND BEJAN, A. Optimal distribution of discrete heat sources on a plate with laminar forced convection. *International Journal of Heat and Mass Transfer* 47, 10-11 (2004), 2139–2148.

- [49] DA SILVA, A., LORENTE, S., AND BEJAN, A. Optimal distribution of discrete heat sources on a wall with natural convection. *International Journal of Heat and Mass Transfer* 47, 2 (2004), 203–214.
- [50] D’ALESSANDRO, V., MONTELPARE, S., RICCI, R., AND SECCHIAROLI, A. Unsteady Aerodynamics of a Savonius wind rotor: a new computational approach for the simulation of energy performance. *Energy* 35, 8 (2010), 3349–3363.
- [51] DARÓCZY, L. Topology optimization and sizing of truss structures (Tartószerkezetek topológiai optimalálása es méretezése). Master’s thesis, University of Miskolc, 2012.
- [52] DARÓCZY, L. *OPAL++ Command Line Interface Manual*. University of Magdeburg ”Otto von Guericke”, Magdeburg, Germany, 274 pages, 2015.
- [53] DARÓCZY, L., BERG, P., AND JANIGA, G. Optimization of Flow Diverter Treatment for a Patient-specific Giant Aneurysm Using STAR-CCM+. In *STAR Global Conference* (Prague, Czech Republic, 2016).
- [54] DARÓCZY, L., JANIGA, G., PETRASCH, K., WEBNER, M., AND THÉVENIN, D. Comparative analysis of turbulence models for the aerodynamic simulation of H-Darrieus rotors. *Energy* 90, 1 (2015), 680–690.
- [55] DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Genetic optimization of heat-exchanger arrangement problem. In *ProcessNet Jahrestreffen der Fachgruppen Computational Fluid dynamics, Mischvorgänge und Rheologie* (Würzburg, Germany, 2014).
- [56] DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Systematic analysis of the heat exchanger arrangement problem using multi-objective genetic optimization. *Energy* 65 (2014), 364–373.
- [57] DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Correlation of the power coefficients of H-Darrieus wind turbines with different turbulence models in CFD computations. In *Conference on Modelling Fluid Flow (the 16th International Conference on Fluid Flow Technologies)* (Budapest, Hungary, 2015), J. Vad, Ed.
- [58] DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Workshop on Turbomachine Optimization based on Computational Fluid Dynamics. In *Conference on Modelling Fluid Flow (the 16th International Conference on Fluid Flow Technologies)* (Budapest, Hungary, 2015), J. Vad, Ed.
- [59] DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Analysis of the performance of a H-Darrieus rotor under uncertainty using Polynomial Chaos Expansion. *Energy* 113 (2016), 399–412.
- [60] DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Towards the optimization of winglets for H-Darrieus rotors: Parameterization and automatization for performance evaluation based on 3D-URANS. In *European Congress on Computational Methods in Applied Sciences and Engineering* (Crete Island, Greece, 2016).

- [61] DARÓCZY, L., AND JÁRMAI, K. From a quasi-static fluid-based evolutionary topology optimization to a generalization of BESO. *Engineering Optimization* 47, 5 (2015), 689–705.
- [62] DARÓCZY, L., MOHAMED, M., JANIGA, G., AND THÉVENIN, D. Analysis of the effect of a slotted flap mechanism on the performance of an H-Darrieus turbine using CFD (GT2014-25250). In *ASME Turbo Expo Conference* (Düsseldorf, 2014).
- [63] DARÓCZY, L., MOHAMED, M., AND THÉVENIN, D. Numerical aero-acoustics assessment of double-airfoil vertical axis wind turbine. In *Jahrestagung für Akustik DAGA* (Nürnberg, 2015).
- [64] DAVIS, L. Adapting operator probabilities in genetic algorithms. In *Proceedings of the Third International Conference on Genetic Algorithms* (San Francisco, CA, USA, 1989), Morgan Kaufmann Publishers Inc., pp. 61–69.
- [65] DEB, K. *Multi-Objective optimization using Evolutionary Algorithms*. Wiley, London, 2001.
- [66] DEB, K., AND AGRAWAL, R. B. Simulated binary crossover for continuous search space. *Complex Systems* 9, 2 (1995), 115–148.
- [67] DEB, K., AND GOYAL, M. A Combined Genetic Adaptive Search (GeneAS) for Engineering Design. *Computer Science and Informatics* 26 (1996), 30–45.
- [68] DEB, K., PRATAP, A., AGARWAL, S., AND MEYARIVAN, T. A fast elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6 (2000), 182–197.
- [69] DEB, K., AND TIWARI, S. Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization. *European Journal of Operational Research* 185, 3 (2008), 1062–1087.
- [70] DIAS, M. A. G. Quasi-Monte Carlo Simulation (Date of last access: 2016-05-03). [http://marcoagd.usuarios.rdc.puc-rio.br/quasi\\_mc.html](http://marcoagd.usuarios.rdc.puc-rio.br/quasi_mc.html).
- [71] DIN DEUTSCHES INSTITUT FÜR NORMUNG E.V. UND VDE VERBAND DER ELEKTROTECHNIK ELEKTRONIK INFORMATIONSTECHNIK E. V. Design requirements for small wind turbines, DIN EN 61400-2 (VDE 0127-2):2007-2.
- [72] DRELA, M., AND YOUNGREN, H. *XFOIL 6.94 User Guide*. Massachusetts Institute of Technology, Cambridge, MA, 2001.
- [73] EASTMAN, N., JACOBS, K., WARD, E., AND PINKERTON, R. M. The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel. Tech. rep., NACA, 1933.
- [74] ELDRED, M., WEBSTER, C., AND CONSTANTINE, P. Evaluation of Non-Intrusive Approaches for Wiener-Askey Generalized Polynomial Chaos (Paper No. AIAA

- 2008-1892). In *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference* (2008), American Institute of Aeronautics and Astronautics.
- [75] ELDRED, M. S., AND BURKHARDT, J. Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification (paper no. AIAA 2009-0976). In *AIAA proceedings* (2009).
- [76] ELSAYED, K., AND LACOR, C. Robust parameter design optimization using Kriging, RBF and RBFNN with gradient-based and evolutionary optimization techniques. *Applied Mathematics and Computation* 236 (2014), 325–344.
- [77] ENERCON GMBH. ENERCON Wind energy converters - Product overview (Date of last access: 21.04.2016). <http://docs.wind-watch.org/Enercon.pdf>, 2010.
- [78] ESKANDARI, H., GEIGER, C. D., AND LAMONT, G. B. FastPGA: A Dynamic Population Sizing Approach for Solving Expensive Multiobjective Optimization Problems. In *Evolutionary Multi-Criterion Optimization, 4th International Conference* (March 2007), vol. 4403, pp. 141–155.
- [79] EUROPEAN COMMISSION. ENERGY - Country Factsheets V.1.3 (Date of last access: 15.01.2014). <http://ec.europa.eu/energy/observatory/countries/doc/2012-country-factsheets.pdf>, 2012.
- [80] EUROPEAN COMMISSION - DIRECTORATE-GENERAL FOR ENERGY. EU energy trends to 2030 (Date of last access: 15.01.2014). [https://ec.europa.eu/energy/sites/ener/files/documents/trends\\_to\\_2030\\_update\\_2009.pdf](https://ec.europa.eu/energy/sites/ener/files/documents/trends_to_2030_update_2009.pdf), 2009.
- [81] EUROPEAN COMMISSION - DIRECTORATE-GENERAL FOR ENERGY. EU energy trends to 2050 (Date of last access: 03.12.2015). [http://ec.europa.eu/energy/sites/ener/files/documents/trends\\_to\\_2050\\_update\\_2013.pdf](http://ec.europa.eu/energy/sites/ener/files/documents/trends_to_2050_update_2013.pdf), 2013.
- [82] FABBRI, G. Heat transfer optimization in internally finned tubes under laminar flow conditions. *International Journal of Heat and Mass Transfer* 41, 10 (1998), 1243–1253.
- [83] FABBRI, G. Effect of viscous dissipation on the optimization of the heat transfer in internally finned tubes. *International Journal of Heat and Mass Transfer* 47, 14-16 (2004), 3003–3015.
- [84] FERREIRA, C. J. S., BIJL, H., VAN BUSSEL, G., AND VAN KUIK, G. Simulating Dynamic Stall in a 2D VAWT: Modeling strategy, verification and validation with Particle Image Velocimetry data. *Journal of Physics: Conference Series* 75, 1 (2007), 012023.
- [85] FERREIRA, S., VAN BUSSEL, G., SCARANO, F., AND VAN KUIK, G. 2D PIV visualization of dynamic stall on a vertical axis wind turbine. In *Collection of Technical Papers - 45th AIAA Aerospace Sciences Meeting* (2007), vol. 23.

- [86] FIEDLER, A. J., AND TULLIS, S. Blade Offset and Pitch Effects on a High Solidity Vertical Axis Wind Turbine. *Wind Engineering* 33, 3 (2009), 237–246.
- [87] FISCHER, G., KIPOUROS, T., AND SAVILL, A. Multi-objective optimisation of horizontal axis wind turbine structure and energy production using aerofoil and blade properties as design variables. *Renewable Energy* 62 (2014), 506–515.
- [88] FLUENT INC. *GAMBIT 2.4 User's Guide*. Lebanon, New Hampshire, 2007.
- [89] FOLI, K., OKABE, T., OLHOFFER, M., JIN, Y., AND SENDHOFF, B. Optimization of micro heat exchanger: CFD, analytical approach and multi-objective evolutionary algorithms. *International Journal of Heat and Mass Transfer* 49, 5-6 (2006), 1090–1099.
- [90] FONSECA, C. M., AND FLEMING, P. J. Multiobjective optimization and multiple constraint handling with evolutionary algorithms—Part II: Application example. *IEEE Transactions on Systems, Man and Cybernetics Part A* 28 (1998), 38–47.
- [91] GALANTI, S., AND JUNG, A. R. Low-Discrepancy Sequences: Monte Carlo Simulation of Option Prices. *The Journal of Derivatives* 5, 1 (1997), 63–83.
- [92] GAO, X., YANG, H., AND LU, L. Investigation into the optimal wind turbine layout patterns for a Hong Kong offshore wind farm. *Energy* 73 (2014), 430–442.
- [93] GASPAR-CUNHA, A., ANTUNES, C. H., AND COELLO, C. C., Eds. *Evolutionary Multi-Criterion Optimization - 8th International Conference, EMO 2015, Guimarães, Portugal, March 29 –April 1, 2015. Proceedings*. Springer International Publishing, 2015.
- [94] GAUNAA, M., AND JOHANSEN, J. Determination of the maximum aerodynamic efficiency of wind turbine rotors with winglets. *Journal of Physics: Conference Series* 75, 1 (2007), 012006.
- [95] GAUTSCHI, W. *Orthogonal Polynomials: Computation and Approximation*. Oxford University Press, 2004.
- [96] GEERS, A. J., LARRABIDE, I., MORALES, H. G., AND FRANGI, A. F. Approximating hemodynamics of cerebral aneurysms with steady flow simulations. *Journal of Biomechanics* 47, 1 (2014), 178–185.
- [97] GEL, A., GARG, R., TONG, C., SHAHNAME, M., AND GUENTHER, C. Applying uncertainty quantification to multiphase flow computational fluid dynamics. *Powder Technology* 242 (2013), 27–39.
- [98] GENÇ, M. S., KARASU, I., AND AÇKEL, H. H. An experimental study on aerodynamics of NACA2415 aerofoil at low Re numbers. *Experimental Thermal and Fluid Science* 39 (2012), 252–264.
- [99] GLAUERT, H. Airplane Propellers. In *Aerodynamic Theory*. Springer Berlin Heidelberg, 1935, pp. 169–360.

- [100] GÖÇMEN, T., AND ÖZERDEM, B. Airfoil optimization for noise emission problem and aerodynamic performance criterion on small scale wind turbines. *Energy* 46, 1 (2012), 62–71.
- [101] GOLUB, G. H., AND WELSCH, J. H. Calculation of Gauss Quadrature Rules. *Mathematics of Computation* 23, 106 (1969), 221–230.
- [102] GORLE, J., BARDWELL, S., CHATELLIER, L., PONS, F., BA, M., AND PINEAU, G. PIV Investigation of the Flow Across a Darrius Water Turbine. In *17th International Symposium on Application of Laser Techniques to Fluid Mechanics* (July 2014).
- [103] GOSSELIN, R., DUMAS, G., AND BOUDREAU, M. Parametric study of H-Darrius vertical-axis turbines using uRANS simulations (Paper CFDSC-2013 178). In *21st Annual Conference of the CFD Society of Canada* (Sherbrooke, Canada, May 2013).
- [104] GOUDE, A., LUNDIN, S., AND LEIJON, M. A parameter study of the influence of struts on the performance of a vertical-axis marine current turbine. In *Proceedings of the 8th European wave and tidal energy conference, EWTEC2009* (Uppsala, Sweden, 2009), pp. 477–483.
- [105] GROSAN, C., ABRAHAM, A., AND ISHIBUCHI, H. *Hybrid evolutionary algorithms*. Springer-Verlag, Berlin, Heidelberg, 2007.
- [106] GUERRERO, J. E., MAESTRO, D., AND BOTTARO, A. Biomimetic spiroid winglets for lift and drag control. *Comptes Rendus Mécanique* 340, 1–2 (2012), 67–80.
- [107] HÄMÄLÄINEN, J., MADETOJA, E., AND RUOTSALAINEN, H. Simulation-based Optimization and Decision Support for Conflicting Objectives in Papermaking. *Nordic Pulp & Paper Research Journal* 25, 3 (2010), 405–410.
- [108] HANSEN, N., NIEDERBERGER, A., GUZZELLA, L., AND KOUMOUTSAKOS, P. A Method for Handling Uncertainty in Evolutionary Optimization With an Application to Feedback Control of Combustion. *IEEE Transactions on Evolutionary Computation* 13, 1 (2009), 189–197.
- [109] HE, W., DIEZ, M., CAMPANA, E. F., STERN, F., AND ZOU, Z. A one-dimensional polynomial chaos method in CFD-based uncertainty quantification for ship hydrodynamic performance. *Journal of Hydrodynamics, Ser. B* 25, 5 (2013), 655–662.
- [110] HILBERT, R., JANIGA, G., BARON, R., AND THÉVENIN, D. Multi-objective shape optimization of a heat exchanger using parallel genetic algorithms. *International Journal of Heat and Mass Transfer* 49, 15-16 (2006), 2567–2577.
- [111] HOSDER, S., AND MADDALENA, L. Non-Intrusive Polynomials Chaos for the Stochastic CFD study of a Supersonic Pressure Probe. In *47th AIAA Aerospace Sciences Meeting and Exhibit* (2009).

- [112] HOSDER, S., WALTERS, R., AND BALCH, M. Efficient Sampling for Non-Intrusive Polynomial Chaos Applications with Multiple Uncertain Input Variables (Paper No. AIAA 2007-1939). In *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference* (2007), American Institute of Aeronautics and Astronautics.
- [113] HOSDER, S., WALTERS, R., AND PEREZ, R. A non-intrusive polynomial chaos method for uncertainty propagation in CFD simulations. In *44th AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2006-891* (Reno, 2006).
- [114] IMAMURA, H., HASEGUWA, Y., AND KIKUYAMA, K. Numerical analysis of the horizontal axis wind turbine with winglets. *JSME International Journal Series B* 41, 1 (1998), 170–176.
- [115] IOSIF, C., BERG, P., PONSONNARD, S., CARLES, P., SALEME, S., PEDROLO-SILVEIRA, E., MENDES, G., WAHRICH, E., TROLLIARD, G., COUQUET, C.-Y., YARDIN, C., AND MOUNAYER, C. The role of terminal and anastomotic circulation in the patency of arteries jailed by flow diverting stents: Animal flow model evaluation and preliminary results. *Journal of Neurosurgery* 15 (2016), 1–11.
- [116] IQBAL, Z., SYED, K., AND ISHAQ, M. Optimal convective heat transfer in double pipe with parabolic fins. *International Journal of Heat and Mass Transfer* 54, 25-26 (2011), 5415–5426.
- [117] JANIGA, G. Flow optimization using Computational Fluid Dynamics (Habilitation Thesis). *University of Magdeburg "Otto von Guericke"* (2011).
- [118] JANIGA, G., BERG, P., BEUNG, O., NEUGEBAUER, M., GASTEIGER, R., PREIM, B., ROSE, G., SKALEJ, M., AND THÉVENIN, D. Recommendations for accurate numerical blood flow simulations of stented intracranial aneurysms. *Biomedical Engineering* 58, 3 (2013), 303–314.
- [119] JANIGA, G., DARÓCZY, L., BERG, P., THÉVENIN, D., SKALEJ, M., AND BEUNG, O. An automatic CFD-based flow diverter optimization principle for patient-specific intracranial aneurysms. *Journal of Biomechanics* 48, 14 (2015), 3846–3852.
- [120] JANIGA, G., RÖSSL, C., SKALEJ, M., AND THÉVENIN, D. Realistic virtual intracranial stenting and computational fluid dynamics for treatment analysis. *Journal of Biomechanics* 46, 1 (2013), 7–12.
- [121] JERICHA, H., GÖTTLICH, E., SELIC, T., AND SANZ, W. Novel Vertical-Axis Wind Turbine with Articulated Blading (GT2012-68969). In *ASME Turbo Expo Conference* (Copenhagen, 2012).
- [122] JOE, S., AND KUO, F. Y. Remark on algorithm 659: Implementing Sobol’s quasirandom sequence generator. *ACM Transactions on Mathematical Software* 29, 1 (Mar. 2003), 49–57.
- [123] KAIMAL, J. C., AND FINNIGAN, J. J. *Atmospheric Boundary Layer Flows: Their Structure and Measurement*. Oxford University Press, 1994.

- [124] KE, Z., ZHENGHONG, G., JIANGTAO, H., AND JING, L. Uncertainty quantification and robust design of airfoil based on polynomial chaos technique. *Chinese Journal of Theoretical and Applied Mechanics* 46, 1 (2014), 10.
- [125] KELECY, F. J. Coupling Momentum and Continuity Increases CFD Robustness. *ANSYS Advantage II*, 2 (2008), 49–51.
- [126] KHATIR, Z., THOMPSON, H., KAPUR, N., TOROPOV, V., AND PATON, J. Multi-objective Computational Fluid Dynamics (CFD) design optimisation in commercial bread-baking. *Applied Thermal Engineering* 60, 1–2 (2012), 480–486.
- [127] KJELLIN, J., BÜLOW, F., ERIKSSON, S., DEGLAIRE, P., LEIJON, M., AND BERNHOFF, H. Power coefficient measurement on a 12 kW straight bladed vertical axis wind turbine. *Renewable Energy* 36, 11 (2011), 3050–3053.
- [128] KNIO, O., AND LE MAÎTRE, O. Uncertainty propagation in CFD using polynomial chaos decomposition. *Fluid Dynamics Research* 38, 9 (2006), 616–640.
- [129] KOOIMAN, S., AND TULLIS, S. Response of a Vertical Axis Wind Turbine to Time-Varying Wind Conditions found within the Urban Environment. *Wind Engineering* 34, 4 (2010), 389–401.
- [130] KROO, I. Drag due to lift: Concepts for Prediction and Reduction. *Annual Review of Fluid Mechanics* 33, 587–617 (2001).
- [131] KUMAR, K., PIRAGALATHALWAR, S., BIJLANI, J., AND MADSEN, J. Effect of inflow conditions on wind turbine blade performance. In *ASME 2013 Gas Turbine India Conference, GTINDIA 2013* (2013).
- [132] KURSAWE, F. A variant of evolution strategies for vector optimization. In *Parallel Problem Solving from Nature* (1990), H.-P. Schwefel and R. Männer, Eds., pp. 193–197.
- [133] LAIN, S., AND OSORIO, C. Simulation and evaluation of a straight-bladed Darrieus-type cross flow marine turbine. *Journal of Scientific & Industrial Research* 69, 12 (2010), 906–912.
- [134] LANGTRY, R. B. *A Correlation-Based Transition Model using Local Variables for Unstructured Parallelized CFD codes*. PhD thesis, Universitaet Stuttgart, 2006.
- [135] LANZAFAME, R., MAURO, S., AND MESSINA, M. 2D CFD modeling of H-Darrieus wind turbines using a transition turbulence model. *Energy Procedia* 45 (2014), 131–140.
- [136] LAUMANN, M., THIELE, L., DEB, K., AND ZITZLER, E. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation* 10, 3 (Sept. 2002), 263–282.
- [137] LEE, C., TOWNSEND, S., AND SRINIVAS, K. Optimisation of stents for cerebral aneurysm. In *Computational Fluid Dynamics 2010*, A. Kuzmin, Ed. Springer, 2011, pp. 377–382.

- [138] LEE, H., SALEH, K., HWANG, Y., AND RADERMACHER, R. Optimization of novel heat exchanger design for the application to low temperature lift heat pump. *Energy* 42, 1 (2012), 204–212.
- [139] LEMOUEDDA, A., BREUER, M., FRANZ, E., BOTSCH, T., AND DELGADO, A. Optimization of the angle of attack of delta-winglet vortex generators in a plate-fin-and-tube heat exchanger. *International Journal of Heat and Mass Transfer* 53, 23-24 (2010), 5386–5399.
- [140] LEUSINK, D., ALFANO, D., CINNELLA, P., AND ROBINET, J.-C. Aerodynamic rotor blade optimization at Eurocopter - a new way of industrial rotor blade design. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition* (2013).
- [141] LI, B., LI, J., TANG, K., AND YAO, X. Many-objective evolutionary algorithms: A survey. *ACM Computing Surveys* 48, 1 (2015), 13:1–13:35.
- [142] LIN, C., MUKERJEE, R., AND TANG, B. Construction of orthogonal and nearly orthogonal latin hypercubes. *Biometrika* 96, 1 (2009), 243–247.
- [143] LIU, Z., WANG, X., AND KANG, S. Stochastic performance evaluation of horizontal axis wind turbine blades using non-deterministic CFD simulations. *Energy* 73 (2014), 126–136.
- [144] LOEVEN, G. J. A., AND BIJL, H. An Efficient Framework for Uncertainty Quantification in CFD Using Probabilistic Collocation. In *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference* (2009).
- [145] MA, D., DARGUSH, G. F., NATARAJAN, S. K., LEVY, E. I., SIDDIQUI, A. H., AND MENG, H. Computer modeling of deployment and mechanical expansion of neurovascular flow diverter in patient-specific intracranial aneurysms. *Journal of Biomechanics* 45, 13 (2012), 2256–2263.
- [146] MA, D., DUMONT, T. M., KOSUKEGAWA, H., OHTA, M., YANG, X., SIDDIQUI, A. H., AND MENG, H. High Fidelity Virtual Stenting (HiFiVS) for Intracranial Aneurysm Flow Diversion: In Vitro and In Silico. *Annals of Biomedical Engineering* 41, 10 (2013), 2143–2156.
- [147] MAÎTRE, T., AMET, E., AND PELLONE, C. Modeling of the flow in a Darrieus water turbine: Wall grid refinement analysis and comparison with experiments. *Renewable Energy* 51 (2013), 497–512.
- [148] MĂLĂEL, I., AND DUMITRESCU, H. Numerical Simulation of VAWT Flow using Fluent. *U.P.B. Scientific Bulletin* 76, 1 (2014).
- [149] MARNETT, M. *Multiobjective Numerical Design of Vertical Axis Wind Turbine Components*. PhD thesis, RWTH Aachen, 2012.

- [150] MATSUMOTO, M., AND NISHIMURA, T. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation* 8, 1 (Jan. 1998), 3–30.
- [151] MENG, H., TUTINO, V. M., XIANG, J., AND SIDDIQUI, A. High WSS or low WSS? Complex interactions of hemodynamics with intracranial aneurysm initiation, growth, and rupture: toward a unifying hypothesis. *American Journal of Neuroradiology* 35, 7 (2014), 1254–1262.
- [152] MET OFFICE. Beaufort wind force scale. <http://www.metoffice.gov.uk/weather/marine/guide/beaufortscale.html>. Date of last access: 2014-05-24.
- [153] MEYER, J., DARÓCZY, L., AND THÉVENIN, D. New design approach for Pitot-tube jet pump (GT2014-25310). In *ASME Turbo Expo Conference* (Düsseldorf, 2014).
- [154] MEYER, J., DARÓCZY, L., AND THÉVENIN, D. Optimizing the external geometry of a pick-up tube for a Pitot-tube-jet pump. In *4th International Conference on Engineering Optimization* (Lisbon, 2014).
- [155] MOHAMED, M. Performance investigation of H-rotor Darrieus turbine with new airfoil shapes. *Energy* 47, 1 (2012), 522–530.
- [156] MOHAMED, M. Impacts of solidity and hybrid system in small wind turbines performance. *Energy* 57 (2013), 495–504.
- [157] MOHAMED, M. Aero-acoustics noise evaluation of H-rotor Darrieus wind turbines. *Energy* 65 (2014), 596–604.
- [158] MOLNÁR, B. Comparison of structured and unstructured meshes for the computations of an H-type Darrieus wind turbine. Master’s thesis, University of Miskolc, 2014.
- [159] MÖNCH, T., GASTEIGER, R., JANIGA, G., THEISEL, H., AND PREIM, B. Context-aware mesh smoothing for biomedical applications. *Computers & Graphics* 35 (2011), 755–767.
- [160] MONTEIRO, D. B., AND DE MELLO, P. E. B. Thermal performance and pressure drop in a ceramic heat exchanger evaluated using CFD simulations. *Energy* 45, 1 (2012), 489–496.
- [161] MUKINOVIĆ, M., BRENNER, G., AND RAHIMI, A. Analysis of vertical axis wind turbines. In *New Results in Numerical and Experimental Fluid Mechanics VII - Contributions to the 16th STAB/DGLR Symposium Aachen, Germany* (2010), pp. 587–594.
- [162] MYERS, R. *Response surface methodology : process and product optimization using designed experiments*. J. Wiley, New York, 2002.
- [163] NAJAFI, H., AND NAJAFI, B. Multi-objective optimization of a plate and frame heat exchanger via genetic algorithm. *Heat and Mass Transfer* 46 (2010), 639–647.

- [164] NAJM, H. N. Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annual Review of Fluid Mechanics* 41, 1 (2009), 35–52.
- [165] NAMURA, N., OBAYASHI, S., AND JEONG, S. Efficient global optimization of vortex generators on a super critical infinite-wing using kriging-based surrogate models. In *52nd AIAA Aerospace Sciences Meeting - AIAA Science and Technology Forum and Exposition, SciTech 2014* (2014).
- [166] NEUGEBAUER, M., LAWONN, K., BEUING, O., AND PREIM, B. Automatic generation of anatomic characteristics from cerebral aneurysm surface models. *International Journal of Computer Assisted Radiology and Surgery* 8 (2013), 279–289.
- [167] NG, L., AND ELDRED, M. Multifidelity uncertainty quantification using non-intrusive polynomial chaos and stochastic collocation. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference* (2012).
- [168] NOAA. Beaufort wind scale. <http://www.spc.noaa.gov/faq/tornado/beaufort.html>. Date of last access: 2014-05-24.
- [169] NOBILE, E., PINTO, F., AND RIZZETTO, G. Geometric parameterization and multiobjective shape optimization of convective periodic channels. *Numerical Heat Transfer, Part B: Fundamentals* 50, 5 (2006), 425–453.
- [170] NOBILE, R., BARLOW, M. V. J., AND MEWBURN-CROOK, A. Dynamic stall for a vertical axis wind turbine in a two-dimensional study. In *World Renewable Energy Congress* (Linköping, Sweden, May 2011).
- [171] NORBERG, C. An experimental investigation of the flow around a circular cylinder: influence of aspect ratio. *Journal of Fluid Mechanics* 258 (1996), 287–316.
- [172] OKABE, T., FOLI, K., OLHOFFER, M., JIN, Y., AND SENDHOFF, B. Comparative studies on micro heat exchanger optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation* (2003), pp. 647–654.
- [173] ÖKSÜZ, Ö., AND AKMANDOR, I. Multi-objective aerodynamic optimization of axial turbine blades using a novel multilevel genetic algorithm. *Journal of Turbomachinery* 132, 4 (2010), 1–14.
- [174] PADRON, A., ALONSO, J., PALACIOS, F., BARONE, M., AND ELDRED, M. Multi-fidelity uncertainty quantification: Application to a vertical axis wind turbine under an extreme gust. In *AIAA AVIATION 2014 -15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference* (2014).
- [175] PAPOUTSIS-KIACHAGIAS, E., AND GIANNAKOGLU, K. Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: Industrial applications. *Archives of Computational Methods in Engineering* (2014), 1–45.
- [176] PARASCHIVOIU, I. *Wind Turbine Design: With Emphasis on Darrieus Concept*. École Polytechnique de Montréal, 2002.

- [177] PARASCHIVOIU, I., AND DELCLAUX, F. Double multiple streamtube model with recent improvements (for predicting aerodynamic loads and performance of Darrieus vertical axis wind turbines). *Journal of Energy* 7, 3 (1983), 250–255.
- [178] PASQUALE, D., PERSICO, G., AND REBAY, S. Optimization of turbomachinery flow surfaces applying a CFD-based throughflow method. In *Proceedings of the ASME Turbo Expo* (2012), vol. 8, pp. 2383–2393.
- [179] PERFILIEV, D., HÄMÄLÄINEN, J., AND BACKMAN, J. Robust Analyzing Tools for Wind Turbine Blades Coupled with Multiobjective Optimization. *Journal of Energy and Power Engineering* 7 (2013), 1831–1836.
- [180] POLONI, C. Hybrid GA for multiobjective aerodynamic shape optimization. In *Genetic Algorithms in Engineering and Computer Science* (1997), J. Periaux, M. Galan, and P. Cuesta, Eds., Wiley, pp. 397–414.
- [181] PUSSOLI, B., BARBOSA JR., J., DA SILVA, L., AND KAVIANY, M. Optimization of peripheral finned-tube evaporators using entropy generation minimization. *International Journal of Heat and Mass Transfer* 55, 25-26 (2012), 7838–7846.
- [182] QIN, N., HOWELL, R., DURRANI, N., HAMADA, K., AND SMITH, T. Unsteady Flow Simulation and Dynamic Stall Behaviour of Vertical Axis Wind Turbine Blades. *Wind Engineering* 35, 4 (2011), 511–528.
- [183] RANUT, P., JANIGA, G., NOBILE, E., AND THÉVENIN, D. Multi-objective shape optimization of a tube bundle in cross-flow. *International Journal of Heat and Mass Transfer* 68 (2014), 585–598.
- [184] RECHENBERG, I. *Evolutionstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Fommann-Holzboog, Stuttgart, 1973.
- [185] ROACHE, P. J. Quantification of uncertainty in computational fluid dynamics. *Annual Review of Fluid Mechanics* 29 (1997), 123–160.
- [186] ROBBE, M., AND SCIUBBA, E. Application of a genetic optimization method to a 2-D constructal-based cooling configuration. *ASME Conference Proceedings 2008*, 48715 (2008), 613–626.
- [187] ROSZELLE, B. N., GONZALEZ, L. F., BABIKER, M. H., RYAN, J., ALBUQUERQUE, F. C., AND FRAKES, D. H. Flow diverter effect on cerebral aneurysm hemodynamics: an in vitro comparison of telescoping stents and the pipeline. *Neuroradiology* 55, 6 (2013), 751–758.
- [188] ŞAHIN, A. D. Progress and recent trends in wind energy. *Progress in Energy and Combustion Science* 30, 5 (2004), 501–543.
- [189] ŞAHIN, B., AKKOCA, A., ÖZTÜRJ, N., AND AKILLI, H. Investigations of flow characteristics in a plate fin and tube heat exchanger model composed of single cylinder. *International Journal of Heat and Fluid Flow* 27, 3 (2006), 522–530.

- [190] SARKAR, S., AND DESAI, A. Uncertainty quantification in the aeroelastic analysis of wind turbines: A review and future trend. In *Wind Turbines: Types, Economics and Development*, G. Martin and J. Roux, Eds., vol. Wind Turbines: Types, Economics and Development. Nova Science Publishers, 2011, pp. 257–276.
- [191] SCHAFFER, J. D. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the First International Conference on Genetic Algorithms* (1987), J. J. Grefenstette, Ed., pp. 93–100.
- [192] SCHLICHTING, H. *Boundary Layer Theory*. McGraw-Hill College, 1979.
- [193] SCHOLZ, E. P., KEHRLE, F., VOSSEL, S., HESS, A., ZITRON, E., KATUS, H. A., AND SAGER, S. Discriminating atrial flutter from atrial fibrillation using a multilevel model of atrioventricular conduction. *Heart Rhythm* 11, 5 (2014), 877–884.
- [194] SCHWEFEL, H.-P. *Numerische Optimierung von Computer-Modellen*. Birkhäuser Verlag, Basel, 1977.
- [195] SHELDAHL, R. E., KLIMAS, P., AND FELTZ, L. Aerodynamic Performance of a 5-Metre-Diameter Darrieus Turbine With Extruded Aluminium NACA-0015 Blades, SAND80-0179. Tech. rep., Sandia National Laboratories, March 1980.
- [196] SIERRA, M. R., AND COELLO COELLO, C. A. Improving PSO-Based multi-objective optimization using crowding, mutation and  $\epsilon$ -dominance. In *Proceedings of the Third international conference on Evolutionary Multi-Criterion Optimization* (Berlin, Heidelberg, 2005), EMO'05, Springer-Verlag, pp. 505–519.
- [197] SIMON, D. *Evolutionary Optimization Algorithms*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2013.
- [198] SIMPSON, T. W., MAUERY, T. M., KORTE, J. J., AND MISTREE, F. Kriging models for global approximation in simulation-based multidisciplinary design optimization. *AIAA Journal* 39, 12 (2001), 2233–2241.
- [199] SLEESONGSOM, S., AND BUREERAT, S. New conceptual design of aeroelastic wing structures by multi-objective optimization. *Engineering Optimization* 45, 1 (2013), 107–122.
- [200] SOBIECZKY, H. Parametric airfoils and wings. In *Notes on Numerical Fluid Mechanics*, K. Fuji and G. S. Dulikravich, Eds., vol. 68. Springer Vieweg, 1998.
- [201] SOBOL, I. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics* 7, 4 (1967), 86–112.
- [202] SOULAT, L., FERRAND, P., MOREAU, S., AUBERT, S., AND BUISSON, M. Efficient optimisation procedure for design problems in fluid mechanics. *Computers & Fluids* 82 (2013), 73–86.

- [203] SPALART, P. R., AND RUMSEY, C. L. Effective Inflow Conditions for Turbulence Models in Aerodynamic Calculations. *AIAA Journal* 45, 10 (2007), 2544–2553.
- [204] SRINIVAS, K., NAKAYAMA, T., OHTA, M., OBAYASHI, S., AND YAMAGUCHI, T. Studies on design optimization of coronary stents. *Journal of Medical Devices* 2, 1 (2008), 011004.
- [205] SRINIVAS, K., TOWNSEND, S., LEE, C. J., NAKAYAMA, T., OHTA, M., OBAYASHI, S., AND YAMAGUCHI, T. Two-dimensional optimization of a stent for an aneurysm. *Journal of Medical Devices* 4, 2 (2010), 021003.
- [206] STORN, R., AND PRICE, K. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *J. of Global Optimization* 11, 4 (1997), 341–359.
- [207] STRANG, G. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 2009.
- [208] STRICKLAND, J. H. Darrieus turbine: a performance prediction model using multiple streamtubes. Tech. rep., Sandia National Laboratories, October 1975.
- [209] SUDHAKAR, T., BALAJI, C., AND VENKATESHAN, S. A heuristic approach to optimal arrangement of multiple heat sources under conjugate natural convection. *International Journal of Heat and Mass Transfer* 53, 1-3 (2010), 431–444.
- [210] TABORDA, M., LAÓN, S., LÓPEZ, O., AND LÓPEZ, Y. Design Improvement of Cross Flow Water Turbines using CFD. In *International Conference on Renewable Energies and Power Quality (ICREPEQ'15)* (2015).
- [211] TEMPLIN, R. J. Aerodynamic Performance Theory for the NRC Vertical-Axis Wind Turbine. LTR-LA-160. Tech. rep., National Research Council of Canada, June 1974.
- [212] THÉVENIN, D., AND JANIGA, G., Eds. *Optimization and Computational Fluid Dynamics*. Springer, Berlin, 2008.
- [213] TOSCANO-PULIDO, G., COELLO COELLO, C. A., AND SANTANA-QUINTERO, L. V. Emopso: A multi-objective particle swarm optimizer with emphasis on efficiency. In *Proceedings of the 4th International Conference on Evolutionary Multi-criterion Optimization* (Berlin, Heidelberg, 2007), EMO'07, Springer-Verlag, pp. 272–285.
- [214] TRIVELLATO, F., AND CASTELLI, M. R. On the Courant–Friedrichs–Lewy criterion of rotating grids in 2D vertical-axis wind turbine analysis. *Renewable Energy* 62 (2014), 53–62.
- [215] UYS, P., FARKAS, J., JÁRMAI, K., AND VAN TONDER, F. Optimisation of a steel tower for a wind turbine structure. *Engineering Structures* 29, 7 (2007), 1337–1342.
- [216] VERSTRAETE, T., COLETTI, F., BULLE, J., VANDERWIELEN, T., AND ARTS, T. Optimization of a U-Bend for Minimal Pressure Loss in Internal Cooling Channels—Part I: Numerical Method. *Journal of Turbomachinery* 135, 5 (2013).

- [217] VICENTE, D., GAMBOA, P., AND SILVESTRE, M. *Engineering Optimization 2014*. CRC Press, 2014, ch. Parameterization formulations for aerofoil shape optimization.
- [218] WALL, A., AND PENNA, P. The role of NRC wind tunnels in the study of the aerodynamic performance of small wind turbines. In *European Wind Energy Conference & Exhibition* (2010).
- [219] WALLACE, R. C., KARIS, J. P., PARTOVI, S., AND FIORELLA, D. Noninvasive imaging of treated cerebral aneurysms, Part I: MR angiographic follow-up of coiled aneurysms. *American Journal of Neuroradiology* 28, 6 (2007), 1001–1008.
- [220] WALLACE, R. C., KARIS, J. P., PARTOVI, S., AND FIORELLA, D. Noninvasive imaging of treated cerebral aneurysms, Part II: CT angiographic follow-up of surgically clipped aneurysms. *American Journal of Neuroradiology* 28, 7 (2007), 1207–1212.
- [221] WANG, S., INGHAM, D. B., MA, L., POURKASHANIAN, M., AND TAO, Z. Numerical investigations on dynamic stall of low Reynolds number flow around oscillating airfoils. *Computers & Fluids* 39, 9 (2010), 1529–1541.
- [222] WANG, X., HIRSCH, C., LIU, Z., KANG, S., AND LACOR, C. Uncertainty-based robust aerodynamic optimization of rotor blades. *International Journal for Numerical Methods in Engineering* 94, 2 (2013), 111–127.
- [223] WHITCOMB, R. T. A design approach and selected wind-tunnel results at high subsonic speeds for wing-tip mounted winglets (NASA TN D-8260). Tech. rep., NACA, 1976.
- [224] WIENER, N. The homogeneous chaos. *American Journal of Mathematics* 60, 4 (1938), 897–936.
- [225] WIERINGA, J. Updating the Davenport roughness classification. *Journal of Wind Engineering and Industrial Aerodynamics* 41, 357–368 (1992).
- [226] WINCHESTER, J., AND QUAYLE, S. Torque ripple and variable blade force: A comparison of Darrieus and Gorlov-type turbines for tidal stream energy conversion. In *Proceedings of the 8th European wave and tidal energy conference, EWTEC2009* (Uppsala, Sweden, 2009), pp. 668–676.
- [227] WITTEVEEN, J. A., AND BIJL, H. Modeling Arbitrary Uncertainties Using Gram-Schmidt Polynomial Chaos. In *44th AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2006-896* (Reno, 2006).
- [228] WOLPERT, D., AND MACREADY, W. No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on* 1, 1 (Apr 1997), 67–82.
- [229] XIU, D., AND KARNIADAKIS, G. E. The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations. *SIAM Journal on Scientific Computing* 24, 2 (2002), 619–644.

- [230] YANG, X.-S. *Nature-Inspired Optimization Algorithms*, 1st ed. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, 2014.
- [231] YU, S.-H., LEE, K.-S., AND YOOK, S.-J. Optimum design of a radial heat sink under natural convection. *International Journal of Heat and Mass Transfer* 54, 11-12 (2011), 2499–2505.
- [232] ZITZLER, E., DEB, K., AND THIELE, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation* 8, 2 (2000), 173–195.
- [233] ZITZLER, E., LAUMANN, M., AND THIELE, L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Tech. Rep. 103, Computer Engineering and Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zurich, May 2001.
- [234] ZITZLER, E., AND THIELE, L. Multiobjective optimization using evolutionary algorithms - a comparative study. In *Parallel Problem Solving from Nature V*, A. E. Eiben, Ed. Springer-Verlag, Amsterdam, 1998, pp. 292–301.

# List of relevant publications

- BERG, P., DARÓCZY, L., BEUING, O., AND JANIGA, G. Patient-specific treatment of intracranial aneurysms: an automatic CFD-based flow-diverter optimization principle. In *Summer Biomechanics, Bioengineering and Biotransport Conference SB3C* (Snowbird, Utah, 2015), pp. 356–357.
- BERG, P., DARÓCZY, L., BEUING, O., SKALEJ, M., THÉVENIN, D., AND JANIGA, G. CFD-based investigation of various flow diverter stent compressions in a patient-specific giant aneurysm. In *Interdisciplinary Cerebrovascular Symposium* (Zurich, Switzerland, 2014).
- DARÓCZY, L., BERG, P., BEUING, O., AND JANIGA, G. How can physicians benefit from hemodynamic simulations? An individualized therapy planning approach for intracranial aneurysms. In *Interdisciplinary Cerebrovascular Symposium/Intracranial Stent Meeting* (Gold Coast, Australia, 2015).
- DARÓCZY, L., BERG, P., AND JANIGA, G. Optimization of Flow Diverter Treatment for a Patient-specific Giant Aneurysm Using STAR-CCM+. In *STAR Global Conference* (Prague, Czech Republic, 2016).
- DARÓCZY, L., BERG, P., AND JANIGA, G. Treatment Improvement for Intracranial Aneurysms: Stent Individualization using Computational Fluid Dynamics Coupled with Optimization Techniques. In *5th MICCAI-Workshop on Computing and Visualisation for Intravascular Imaging and Computer Assisted Stenting (MICCAI CVII-STENT)* (Athens, Greece, 2016).
- DARÓCZY, L., BERG, P., JANIGA, G., AND THÉVENIN, D. Multi-Objective Optimization of Flow Diverter Geometry Towards Patient-Specific Treatment of Intracranial Aneurysms. In *Optimize This! 2016 International HEEDS User Conference* (Detroit, USA, 2016).
- DARÓCZY, L., JANIGA, G., PETRASCH, K., WEBNER, M., AND THÉVENIN, D. Comparative analysis of turbulence models for the aerodynamic simulation of H-Darrieus rotors. *Energy* 90, 1 (2015), 680–690.
- DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Computational Fluid Dynamics (CFD) simulations of an H-Darrieus rotor with different turbulence models (Paper MS215A-3). In *6th European Conference on Computational Fluid Dynamics (ECFD VI)* (Barcelona, Spain, 2014), E. Oñate, X. Oliver, and A. Huerta, Eds.

- DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Genetic optimization of heat-exchanger arrangement problem. In *ProcessNet Jahrestreffen der Fachgruppen Computational Fluid dynamics, Mischvorgänge und Rheologie* (Würzburg, Germany, 2014).
- DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Systematic analysis of the heat exchanger arrangement problem using multi-objective genetic optimization. *Energy* 65 (2014), 364–373.
- DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Correlation of the power coefficients of H-Darrieus wind turbines with different turbulence models in CFD computations. In *Conference on Modelling Fluid Flow (the 16th International Conference on Fluid Flow Technologies)* (Budapest, Hungary, 2015), J. Vad, Ed.
- DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Workshop on Turbomachine Optimization based on Computational Fluid Dynamics. In *Conference on Modelling Fluid Flow (the 16th International Conference on Fluid Flow Technologies)* (Budapest, Hungary, 2015), J. Vad, Ed.
- DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Analysis of the performance of a H-Darrieus rotor under uncertainty using Polynomial Chaos Expansion. *Energy* 113 (2016), 399–412.
- DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Towards the optimization of winglets for H-Darrieus rotors: Parameterization and automatization for performance evaluation based on 3D-URANS. In *European Congress on Computational Methods in Applied Sciences and Engineering* (Crete Island, Greece, 2016).
- DARÓCZY, L., MOHAMED, M., JANIGA, G., AND THÉVENIN, D. Analysis of the effect of a slotted flap mechanism on the performance of an H-Darrieus turbine using CFD (GT2014-25250). In *ASME Turbo Expo Conference* (Düsseldorf, 2014).
- DARÓCZY, L., MOHAMED, M., AND THÉVENIN, D. Numerical aero-acoustics assessment of double-airfoil vertical axis wind turbine. In *Jahrestagung für Akustik DAGA* (Nürnberg, 2015).
- JANIGA, G., DARÓCZY, L., BERG, P., THÉVENIN, D., SKALEJ, M., AND BEUING, O. An automatic CFD-based flow diverter optimization principle for patient-specific intracranial aneurysms. *Journal of Biomechanics* 48, 14 (2015), 3846–3852.
- MEYER, J., DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Using 3D shape optimization to reduce turbulent mixing losses inside the rotor cavity of a Pitot-Tube-Jet-Pump for fluid-fluid separation. In *Conference on Modelling Fluid Flow (the 16th International Conference on Fluid Flow Technologies)* (Budapest, Hungary, 2015), J. Vad, Ed.
- MEYER, J., DARÓCZY, L., AND THÉVENIN, D. Validierung einer mittels CFD optimierten Savonius Turbine im Feldversuch. In *Magdeburger Maschinenbau-Tage* (2013).

- MEYER, J., DARÓCZY, L., AND THÉVENIN, D. New design approach for Pitot-tube jet pump (GT2014-25310). In *ASME Turbo Expo Conference* (Düsseldorf, 2014).
- MEYER, J., DARÓCZY, L., AND THÉVENIN, D. Optimizing the external geometry of a pick-up tube for a Pitot-tube-jet pump. In *4th International Conference on Engineering Optimization* (Lisbon, 2014).

# List of all publications

- BARANYI, L., BOLLÓ, B., AND DARÓCZY, L. Simulation of low-Reynolds number flow around an oscillated cylinder using two computational methods. In *ASME 2011 Pressure Vessels & Piping Conference (PVP 2011)* (Baltimore, USA, 2011).
- BARANYI, L., AND DARÓCZY, L. Effect of oscillation amplitude on force coefficients of a cylinder oscillated in transverse or in-line directions. In *Conference on Modelling Fluid Flow (the 14th International Conference on Fluid Flow Technologies)* (Budapest, Hungary, 2009), J. Vad, Ed.
- BARANYI, L., AND DARÓCZY, L. Áramlási paraméterek hatása rezgő henger és folyadék közötti energiaátadásra (Flow parameter effects on energy transfer between an oscillating cylinder and fluid). *GÉP 63*, 9 (2012), 13–16.
- BARANYI, L., AND DARÓCZY, L. Első- és másodrendű időbeli diszkretizáció körhenger körüli áramlás esetén (First- and second-order temporal discretization for computation of flow around a circular cylinder). *GÉP 63*, 9 (2012), 17–20.
- BARANYI, L., AND DARÓCZY, L. Floquet stability analysis of the wake flow of a circular cylinder of a low-Reynolds number flow. In *26th MicroCAD International Scientific Conference* (Miskolc, Hungary, 2012).
- BARANYI, L., AND DARÓCZY, L. Numerical investigation of mechanical energy transfer between the fluid and the cylinder oscillating transverse to the main stream. In *Conference on Modelling Fluid Flow (the 15th International Conference on Fluid Flow Technologies)* (Budapest, Hungary, 2012), J. Vad, Ed.
- BARANYI, L., AND DARÓCZY, L. Mechanical energy transfer and flow parameter effects for a cylinder in transverse oscillation. *International Journal of Heat and Fluid Flow 43* (2013), 251–258.
- BERG, P., DARÓCZY, L., BEUNG, O., AND JANIGA, G. Patient-specific treatment of intracranial aneurysms: an automatic CFD-based flow-diverter optimization principle. In *Summer Biomechanics, Bioengineering and Biotransport Conference SB3C* (Snowbird, Utah, 2015), pp. 356–357.
- BERG, P., DARÓCZY, L., BEUNG, O., SKALEJ, M., THÉVENIN, D., AND JANIGA, G. CFD-based investigation of various flow diverter stent compressions in a patient-specific giant aneurysm. In *Interdisciplinary Cerebrovascular Symposium* (Zurich, Switzerland, 2014).

- BOLLÓ, B., BARANYI, L., SZABÓ, S., BORDÁS, R., BENCs, P., AND DARÓCZY, L. Numerical and experimental investigation of momentum and heat transfer from a heated circular cylinder. In *MicroCAD 2008* (Miskolc, Hungary, 2008).
- DARÓCZY, L., AND BARANYI, L. Euler and second-order Runge-Kutta methods for computation of flow around a cylinder. In *26th MicroCAD International Scientific Conference* (Miskolc, Hungary, 2012).
- DARÓCZY, L., BERG, P., BEUNG, O., AND JANIGA, G. How can physicians benefit from hemodynamic simulations? An individualized therapy planning approach for intracranial aneurysms. In *Interdisciplinary Cerebrovascular Symposium/Intracranial Stent Meeting* (Gold Coast, Australia, 2015).
- DARÓCZY, L., BERG, P., AND JANIGA, G. Optimization of Flow Diverter Treatment for a Patient-specific Giant Aneurysm Using STAR-CCM+. In *STAR Global Conference* (Prague, Czech Republic, 2016).
- DARÓCZY, L., BERG, P., AND JANIGA, G. Treatment Improvement for Intracranial Aneurysms: Stent Individualization using Computational Fluid Dynamics Coupled with Optimization Techniques. In *5th MICCAI-Workshop on Computing and Visualisation for Intravascular Imaging and Computer Assisted Stenting (MICCAI CVII-STENT)* (Athens, Greece, 2016).
- DARÓCZY, L., BERG, P., JANIGA, G., AND THÉVENIN, D. Multi-Objective Optimization of Flow Diverter Geometry Towards Patient-Specific Treatment of Intracranial Aneurysms. In *Optimize This! 2016 International HEEDS User Conference* (Detroit, USA, 2016).
- DARÓCZY, L., JANIGA, G., PETRASCH, K., WEBNER, M., AND THÉVENIN, D. Comparative analysis of turbulence models for the aerodynamic simulation of H-Darrieus rotors. *Energy* 90, 1 (2015), 680–690.
- DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Computational Fluid Dynamics (CFD) simulations of an H-Darrieus rotor with different turbulence models (Paper MS215A-3). In *6th European Conference on Computational Fluid Dynamics (ECFD VI)* (Barcelona, Spain, 2014), E. Oñate, X. Oliver, and A. Huerta, Eds.
- DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Genetic optimization of heat-exchanger arrangement problem. In *ProcessNet Jahrestreffen der Fachgruppen Computational Fluid dynamics, Mischvorgänge und Rheologie* (Würzburg, Germany, 2014).
- DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Systematic analysis of the heat exchanger arrangement problem using multi-objective genetic optimization. *Energy* 65 (2014), 364–373.
- DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Correlation of the power coefficients of H-Darrieus wind turbines with different turbulence models in CFD computations. In *Conference on Modelling Fluid Flow (the 16th International Conference on Fluid Flow Technologies)* (Budapest, Hungary, 2015), J. Vad, Ed.

- DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Workshop on Turbomachine Optimization based on Computational Fluid Dynamics. In *Conference on Modelling Fluid Flow (the 16th International Conference on Fluid Flow Technologies)* (Budapest, Hungary, 2015), J. Vad, Ed.
- DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Analysis of the performance of a H-Darrieus rotor under uncertainty using Polynomial Chaos Expansion. *Energy* 113 (2016), 399–412.
- DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Towards the optimization of winglets for H-Darrieus rotors: Parameterization and automatization for performance evaluation based on 3D-URANS. In *European Congress on Computational Methods in Applied Sciences and Engineering* (Crete Island, Greece, 2016).
- DARÓCZY, L., AND JÁRMAI, K. Új eljárás rácsos tartók topológiai optimalására (New method for topology optimization of truss structures). *GÉP* 62, 7–8 (2011), 29–34.
- DARÓCZY, L., AND JÁRMAI, K. Acélvázás szerkezetek tervezésének automatizálása topológiai optimalással (Automatization of design of frame and truss structures using topology optimization). *GÉP* 63, 2 (2012), 13–18.
- DARÓCZY, L., AND JÁRMAI, K. Topology optimization by a quasi-static fluid-based evolutionary method. In *Proceedings of the 5th International Conference "Computational Mechanics and Virtual Engineering" COMEC 2013* (Brasov, Romania, 2013), pp. 191–196.
- DARÓCZY, L., AND JÁRMAI, K. From a quasi-static fluid-based evolutionary topology optimization to a generalization of BESO. *Engineering Optimization* 47, 5 (2015), 689–705.
- DARÓCZY, L., LINDELL, G., AND BARANYI, L. Safety analysis of cold Helium releases in the CERN LHC underground tunnel using CFD simulations. In *26th MicroCAD International Scientific Conference* (Miskolc, Hungary, 2012).
- DARÓCZY, L., MOHAMED, M., JANIGA, G., AND THÉVENIN, D. Analysis of the effect of a slotted flap mechanism on the performance of an H-Darrieus turbine using CFD (GT2014-25250). In *ASME Turbo Expo Conference* (Düsseldorf, 2014).
- DARÓCZY, L., MOHAMED, M., AND THÉVENIN, D. Numerical aero-acoustics assessment of double-airfoil vertical axis wind turbine. In *Jahrestagung für Akustik DAGA* (Nürnberg, 2015).
- ESHGHINEJADFARD, A., DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Calculation of the permeability in porous media using the Lattice Boltzmann method. *Int. J. Heat Fluid Flow* (2016).
- JANIGA, G., DARÓCZY, L., BERG, P., THÉVENIN, D., SKALEJ, M., AND BEUING, O. An automatic CFD-based flow diverter optimization principle for patient-specific intracranial aneurysms. *Journal of Biomechanics* 48, 14 (2015), 3846–3852.

- MEYER, J., DARÓCZY, L., JANIGA, G., AND THÉVENIN, D. Using 3D shape optimization to reduce turbulent mixing losses inside the rotor cavity of a Pitot-Tube-Jet-Pump for fluid-fluid separation. In *Conference on Modelling Fluid Flow (the 16th International Conference on Fluid Flow Technologies)* (Budapest, Hungary, 2015), J. Vad, Ed.
- MEYER, J., DARÓCZY, L., AND THÉVENIN, D. Validierung einer mittels CFD optimierten Savonius Turbine im Feldversuch . In *Magdeburger Maschinenbau-Tage* (2013).
- MEYER, J., DARÓCZY, L., AND THÉVENIN, D. New design approach for Pitot-tube jet pump (GT2014-25310). In *ASME Turbo Expo Conference* (Düsseldorf, 2014).
- MEYER, J., DARÓCZY, L., AND THÉVENIN, D. Optimizing the external geometry of a pick-up tube for a Pitot-tube-jet pump. In *4th International Conference on Engineering Optimization* (Lisbon, 2014).