Drofelnik, Jernej (2017) *Massively parallel time- and frequency-domain Navier-Stokes Computational Fluid Dynamics analysis of wind turbine and oscillating wing unsteady flows.* PhD thesis.

http://theses.gla.ac.uk/8284/

Massively parallel time– and frequency–domain Navier–Stokes Computational Fluid Dynamics analysis of wind turbine and oscillating wing unsteady flows

Jernej Drofelnik

Submitted in fulfilment of the requirements for the Degree of
Doctor of Philosophy

School of Engineering
College of Science and Engineering
University of Glasgow

June 2017

Abstract

Increasing interest in renewable energy sources for electricity production complying with stricter environmental policies has greatly contributed to further optimisation of existing devices and the development of novel renewable energy generation systems. The research and development of these advanced systems is tightly bound to the use of reliable design methods, which enable accurate and efficient design. Reynolds–averaged Navier–Stokes Computational Fluid Dynamics is one of the design methods that may be used to accurately analyse complex flows past current and forthcoming renewable energy fluid machinery such as wind turbines and oscillating wings for marine power generation. The use of this simulation technology offers a deeper insight into the complex flow physics of renewable energy machines than the lower–fidelity methods widely used in industry. The complex flows past these devices, which are characterised by highly unsteady and, often, predominantly periodic behaviour, can significantly affect power production and structural loads. Therefore, such flows need to be accurately predicted.

The research work presented in this thesis deals with the development of a novel, accurate, scalable, massively parallel CFD research code COSA for general fluid–based renewable energy applications. The research work also demonstrates the capabilities of newly developed solvers of COSA by investigating complex three–dimensional unsteady periodic flows past oscillating wings and horizontal–axis wind turbines.

Oscillating wings for the extraction of energy from an oncoming water or air stream, feature highly unsteady hydrodynamics. The flow past oscillating wings may feature dynamic stall and leading edge vortex shedding, and is significantly three–dimensional due to finite–wing effects. Detailed understanding of these phenomena is essential for maximising the power generation efficiency. Most of the knowledge on oscillating wing hydrodynamics is based on two–dimensional low–Reynolds number computational fluid dynamics studies and experimental testing. However, real installations are expected to feature Reynolds numbers of the order of 1 million and strong finite–wing–induced losses. This research investigates the impact of finite wing effects on the hydrodynamics of a realistic aspect ratio 10 oscillating wing device in a stream with Reynolds number of 1.5 million, for two high–energy extraction operating regimes. The benefits of using endplates in order to reduce finite–wing–induced losses are also analyzed. Three–dimensional time–accurate Reynolds–averaged Navier–Stokes simulations using Menter's shear stress transport turbulence model and a 30–million–cell grid are performed. Detailed comparative hydrodynamic analyses of the finite and infinite wings highlight that the power generation efficiency of the finite wing with sharp tips for the considered high energy–extraction regimes decreases by up to 20 %, whereas the maximum power drop is 15 % at most when using the endplates.

Horizontal–axis wind turbines may experience strong unsteady periodic flow regimes, such as those associated with the yawed wind condition. Reynolds–averaged Navier–Stokes CFD has been demonstrated to predict horizontal–axis wind turbine unsteady flows with accuracy suitable for reliable turbine design. The major drawback of conventional Reynolds–averaged Navier–Stokes CFD is its high computational cost. A time–step–independent time–domain simulation of horizontal–axis wind turbine periodic flows requires long runtimes, as several rotor revolutions have to be simulated before the periodic state is achieved. Runtimes can be significantly reduced by using the frequency–domain harmonic balance method for solving the unsteady Reynolds–averaged Navier–Stokes equations. This research has demonstrated that this promising technology can be efficiently used for the analyses of complex three–dimensional horizontal–axis wind turbine periodic flows, and has a vast potential for rapid wind turbine design. The three–dimensional simulations of the periodic flow past the blade of the NREL 5–MW baseline horizontal–axis wind turbine in yawed wind have been selected for the demonstration of the effectiveness of the developed technology. The comparative assessment is based on thorough parametric time–domain and harmonic balance analyses. Presented results highlight that horizontal–axis wind turbine periodic flows can be computed by the harmonic balance solver about fifty times more rapidly than by the conventional time–domain analysis, with accuracy comparable to that of the time–domain solver.

Keywords: Compressible multigrid Navier–Stokes solvers, Computational fluid dynamics, Dynamic stall, Energy–extracting oscillating wing, Finite wing effects, Harmonic balance Navier–Stokes equations, Horizontal–axis wind turbine periodic aerodynamics, Fully coupled multigrid integration, Leading edge vortex shedding, Point–implicit Runge–Kutta smoother, Shear stress transport turbulence model

Contents

List of Figures

List of Tables

Acknowledgements

First and foremost I would like to express my sincere gratitude to my PhD supervisor Dr M. Sergio Campobasso, for the continuous support of my research, his guidance and encouragement over the past three and a half years. His immense knowledge of the field, passion and enthusiasm for the research, were extremely motivational for me during this invaluable experience. I would also like to acknowledge Mr. Adrian Jackson for his constant support and assistance on the parallelization part of this PhD project.

I would like to extend my gratitude to all my office colleagues and friends.

My warmest and deepest thanks goes to my loving, encouraging and supportive wife Teja, whose faithful support has helped me through some of the most challenging times of my PhD.

Lastly, I would like to thank my entire extended family and closest friends, for all their love and encouragement. Special thanks goes to my wonderful parents Silva and Janko who always supported me in all my decisions.

Author's Declaration

I hereby declare that this thesis has been written entirely by myself and is a record of the work performed by myself. This thesis has not been previously submitted for a higher degree at this or any other university.

The research was carried out in the Systems, Power and Energy Research Division of School of Engineering, College of Science and Engineering at the University of Glasgow under the supervision of Dr M. Sergio Campobasso.

Part of the work presented in this thesis has been published in the following journal and conference publications, respectively:

J. Drofelnik, M.S. Campobasso, 2016, *Comparative turbulent three–dimensional Navier–Stokes hydrodynamic analysis and performance assessment of oscillating wings for renewable energy applications.* International Journal of Marine Energy, Vol. 16, pp. 100–115

M.S. Campobasso, J. Drofelnik, F. Gigante, 2016, *Comparative Assessment of the Harmonic Balance Navier Stokes Technology for Horizontal and Vertical Axis Wind turbine Aerodynamics.* Computers and Fluids, Vol. 136, pp. 354–370

M.S. Campobasso, A. Piskopakis, J. Drofelnik, A. Jackson, 2013, *Turbulent Navier–Stokes analysis of an oscillating wing in a power–extraction regime using the shear stress transport turbulence model.* Computers and Fluids, Vol. 88, pp. 136–155

J. Drofelnik and M.S. Campobasso, *Three–dimensional turbulent Navier–Stokes hydrodynamic analysis and performance assessment of oscillating wings for power generation.* Conference on Modelling Fluid Flow (CMFF'15), September 1–4, 2015 Budapest, Hungary

M.S. Campobasso, F. Gigante, J. Drofelnik, *Turbulent Unsteady Flow Analysis of Horizontal Axis Wind Turbine Airfoil Aerodynamics based on the Harmonic Balance Reynolds–Averaged Navier–Stokes Equations.* ASME paper GT2014–25559, ASME Turbo Expo Technical Conference, June 16–20, 2014, Düsseldorf, Germany

M.S. Campobasso, M.Yan, J. Drofelnik, A. Piskopakis, M. Caboni, *Compressible Reynolds–Averaged Navier–Stokes Analysis of Wind Turbine Turbulent Flows using a Fully Coupled Low–Speed Preconditioned Multigrid Solver.* ASME paper GT2014–25562, ASME Turbo Expo Technical Conference, June 16–20, 2014, Düsseldorf, Germany

Nomenclature

Acronyms

2D	Two–dimensional
3D	Three–dimensional
ALE	Arbitrary Lagrangian–Eulerian
API	Application programming interface
AR	Aspect ratio
BEMT	Blade element momentum theory
CFD	Computational fluid dynamics
CGNS	CFD General Notation System
COSA	CFD Optimised Structured multi–block Algorithm
CPU	Central processing unit
GPU	Graphics processing units
HAWT	Horizontal–axis wind turbine
HB	Harmonic balance
HDHB	High–dimensional harmonic balance
HPC	High performance computing
I/O	Input/output
LE	Leading edge
LEVS	Leading edge vortex shedding
MERWind	Multidisciplinary design and analysis framework for wind turbines
MIMD	Multiple instruction multiple data
MPI	Message Passing Interface
NLFD	Non–linear frequency–domain
NREL	National Renewable Energy Laboratory
NS	Navier–Stokes
NUMA	Nonuniform memory access
NWTC	National Wind Technology Center
OC3	Offshore Code Comparison Collaboration
PDE	Partial differential equation
PGAS	Partitioned global address space
RANS	Reynolds–Averaged Navier–Stokes
SIMD	Single instruction multiple data
SST	Shear stress transport
TD	Time–domain
UMA	Uniform Memory Access
URANS	Unsteady Reynolds–Averaged Navier–Stokes
WEO	World Energy Outlook

Greek symbols

Δl_r	Logarithm in base 10 of normalized residual RMS of RANS equations
Ω	Angular velocity
$\underline{\mathbf{\Phi}}_c$	Convective flux vector

$\boldsymbol{\Phi}_{c,f}$	Convective flux
$\underline{\boldsymbol{\Phi}}_d$	Diffusive flux vector
α	Angle of attack
α_m	m^{th} Runge–Kutta coefficient
δ	Yaw angle
δ_{ij}	Kronecker delta function
η	Efficiency
γ	Specific heat ratio / twist angle
μ	Dynamic viscosity
μ_{ref}	Reference dynamic viscosity
μ_T	Eddy viscosity
ω	Angular frequency / specific dissipation rate
ϕ	Angle of the relative wind/Phase angle between heaving and pitching motions
ρ	Fluid density
τ	Pseudo time–step
$\underline{\underline{\tau}}$	Stress tensor
$\underline{\underline{\tau}}^M$	Molecular viscous stress tensor
$\underline{\underline{\tau}}^R$	Reynolds stress tensor
θ	Pitching motion
θ_0	Pitching amplitude

Latin symbols

$'$	Transpose operator
C	Control volume
CD_ω	Cross–diffusion term of ω equation
C_M	Instantaneous pitching moment coefficient
C_{P_θ}	Instantaneous pitching power coefficient
C_{P_y}	Instantaneous heaving power coefficient
C_{P_z}	Instantaneous power coefficient per unit wing length
$C_{P_{z\theta}}$	Instantaneous pitching power coefficient per unit wing length
$C_{P_{zy}}$	Instantaneous heaving power coefficient per unit wing length
C_Y	Instantaneous heaving force coefficient
D	HB antisymmetric matrix
D_ω	Destruction term of ω rate
D_k	Destruction term of k
E	Total energy per unit mass
E_H	Fourier transformation matrix
E_P	Parallel efficiency
F_x	Tangential force
F_{x_y}	Tangential force per unit blade length
F_z	Axial force / thrust
F_{z_y}	Axial force per unit blade length
H	Total enthalpy
$I_{N_{PDE}}$	Identity matrix of size $(N_{PDE})^2$
M	Hydrodynamic torque
M_x	Out–of–plane bending moment
M_y	Torsional moment
M_z	In–plane bending moment / torque
M_{z_y}	Torque per unit blade length
N_H	Retained number of harmonics
N_{PDE}	Number of PDEs
P	Overall power

P_a	Total available power
P_d	Turbulent production term
Pr	Prandtl number
Pr_T	Turbulent Prandtl number
P_θ	Instantaneous pitching power
P_y	Instantaneous heaving power
P_z	Sum of the instantaneous pitching and heaving power
$\mathbf{Q}$	Array of unknowns
$\mathbf{R}$	Cell residuals
$\underline{\mathrm{R}}$	Resultant force
R	Rotor radius
$\mathbf{S}$	Source term
S	Control surface
$\tilde{S}_{ij}$	Reynolds strain rate tensor
S_{ij}	Instantaneous strain rate tensor
S_k	Source term of k equation
S_P	Speedup
S_ω	Source term of ω equation
T	Absolute temperature
T_1	Time taken to run a program on a single core
T_n	Time taken to run a program on n number of cores
T_{ni}	Ideal time taken on n number of cores
T_{ref}	Reference temperature
T_S	Sutherland temperature
TSR	Tip–speed ratio
$\mathbf{U}$	Conservative variables
$\mathbf{U}_H$	Conservative variables in HB representation
V	Cell volume
X	Horizontal force component
Y	Vertical force component
μ_T	Turbulent or eddy viscosity
b	Semi–span
c	Airfoil/foil chord
c_p	Specific heat at constant pressure
c_v	Specific heat at constant volume
d	Swept area of the oscillating wing device
dC	Control volume element
dS	Control surface element
e	Internal energy per unit mass
f	Vibration frequency
$\mathbf{h}$	HB solution surface integral
$\hat{\mathbf{h}}$	HB solution Fourier coefficient variables of surface integral
$\tilde{\mathbf{h}}$	Time–domain representation of HB Fourier coefficients of surface integral
h	Heaving motion
h_0	Heaving amplitude
i, j, k	Unit vectors
k	Turbulent kinetic energy
k_T	Thermal conductivity
l	Wing semispan
$\underline{\mathrm{n}}$	Unit normal vector
$\underline{\mathrm{q}}$	Thermal heat flux

r	Distance from the rotational axis
$\underline{\mathrm{r}}_a$	Position vector
t	Physical time–step
$\mathbf{u}$	HB solution volume integral
$\hat{\mathbf{u}}$	HB solution Fourier coefficient variables of volume integral
$\tilde{\mathbf{u}}$	Time–domain representation of HB Fourier coefficients of volume integral
$\underline{\mathrm{u}}_b$	Instantaneous velocity of the boundary S in vector notation
u_i	Instantaneous flow velocity in componential notation
u_∞	Freestream velocity
v_e	Effective velocity
v_y	Heaving velocity
$\underline{\mathrm{w}}$	Absolute freestream velocity vector
$\underline{\mathrm{x}}$	Position vector in vector notation
x_i	Position vector in componential notation
x_p	Location of the pitching axis
x, y, z	Cartesian components of position vector $\underline{\mathrm{x}}$

Chapter 1

Introduction

1.1 Renewable energy

Renewable energy can generally be regarded as energy that comes from any source that is not depleted when used. In the distant past mankind has only been able to convert few of the natural resources, such as the power of wind and water mostly into mechanical work, and biofuels to produce thermal energy. The advancement of technology and scientific discoveries during the industrial revolution in 18^{th} century, driven by the desire for improved standard of living, enabled mankind to utilise energy from many more natural resources. During the industrial revolution we have become dependent on large amount of energy, and therefore, renewable energy sources no longer sufficed our needs. Since then, the mankind has become increasingly dependent on fossil fuels. Eventhough the formation of fossil fuels is a natural process, they are not considered to be renewable energy source, due to the extremely long accumulation process. Increased usage of the fossil fuels since the beginning of the industrial revolution has resulted in the production of significant amounts of carbon dioxide into the atmosphere, which interferes with the natural carbon cycle, which has been in near equilibrium for thousands of years [10]. The main concern with large amounts of carbon dioxide in Earth's atmosphere is that carbon dioxide acts as one of the greenhouse gases. Usually, short–wave solar radiation entering Earth's atmosphere is absorbed by the Earth's surface, and is radiated back into the space as infrared energy, featuring much longer wavelength. In the atmosphere, greenhouse gases redirect part of the long–wave infrared radiation, which would without their presence escape into space, back towards the Earth's surface. Thereby, higher concentration of carbon dioxide in the atmosphere contributes to global warming. The usage of renewable energy sources has, therefore, a vast potential of reducing global greenhouse gas emissions and to limit global warming.

According to the World Energy Outlook (WEO) 2015 New Policies Scenario, world's electricity generation will increase from about $23.3\,TWh$ produced in 2013 to about $39.4\,TWh$ in 2040. To cover world's electricity demands in the near future, renewable energy sources are expected to increase significantly. Wind energy generation capacity is expected to increase by nearly 300 % in 2040 with respect to 2013, and is expected to cover 9 % of electricity generation. An important increase in marine energy sources is also anticipated, however, the share of these technologies is still expected to be relatively small. In the light of current predictions, there is a need to further increase energy efficiency of currently available renewable energy technologies, and to develop novel engineering technologies to harvest more renewable energy sources cheaply and efficiently. This project aims to develop novel, efficient and accurate Computational Fluid Dynamics (CFD) tools for the design and optimisation of efficient renewable energy technology, and examines in great detail the aerodynamics of the rotor of the oscillating wing device, typically used in marine energy sector, and the conventional horizontal–axis wind turbine (HAWT). To design efficient state–of–the–art renewable energy devices, the aerodynamic design can no longer rely only on the usage of low fidelity tools and/or semi–empirical models, such as the blade element momentum theory (BEMT) and dynamic stall models, in the case of wind turbines [11, 12, 13]. These techniques feature extremely high computational speed, however, their reliance on the existence and availability of high–quality airfoil data hinders their applicability to the design of radically new configurations of renewable energy devices. A promising high–fidelity method, which may to a limited extent reduce uncertainty associated with the flow predictions of low–fidelity models, is the Navier–Stokes (NS) CFD. This method has recently gained a wide interest by the wind energy community, since it enables the detailed flow analysis of the renewable energy devices, which cannot be predicted by the simpler methods. The NS CFD simulation may also simulate the flowfield of the wind/marine farms, complex terrains and atmospheric/wind conditions. Generally, the NS CFD simulation data are in reasonable agreement with the experimental data, and on many occasions in much better agreement than the results obtained by the low–fidelity models. Nonetheless, the method still has its limitations and faces several difficulties, that are currently still under the investigation to be fully understood.

1.2 Computational fluid dynamics for oscillating wing devices

Oscillating wing device is a promising new technology for renewable energy production in the fields of wind and marine energy systems. It relies on the use of oscillating wings simultaneously heaving and pitching to extract energy from an oncoming water or air stream. The device was pioneered by McKinney and DeLaurier [1] in 1981 with their 90 W prototype called "wingmill". The pitching and heaving motions of the wing were translated to a rotating shaft through a mechanism called "scotch–yoke". The experimental model of the wingmill is depicted in Fig. 1.1.

Oscillating wing device was further investigated by Jones *et al.* [14, 15], by conducting several numerical investigations on oscillating wings for propulsion and power extraction applications. For the unsteady analyses of the flow around an airfoil they have used an unsteady panel code with a non–linear wake model. Authors have shown that the power was extracted for heaving and pitching motions when the geometric pitch amplitude exceeded the maximum induced angle–of–attack due to the heaving motion for phase angles about 90 degrees. Later Jones *et al.* [2] analysed this problem using panel and NS codes, as well as with the experiment in a water tunnel using the two oscillating foils in a tandem configuration. Comparison the experimental data of the two oscillating foils with the 2D NS simulation of one oscillating foil, revealed a large discrepancy between the two data sets. Their experimental model employing two tandem wings is depicted in Fig. 1.2.

Several other numerical, experimental and prototype–based studies of the oscillating wing device for power generation followed these pioneering studies. The exact design for the commercial use is still under ongoing development by many industrial and scientific communities.

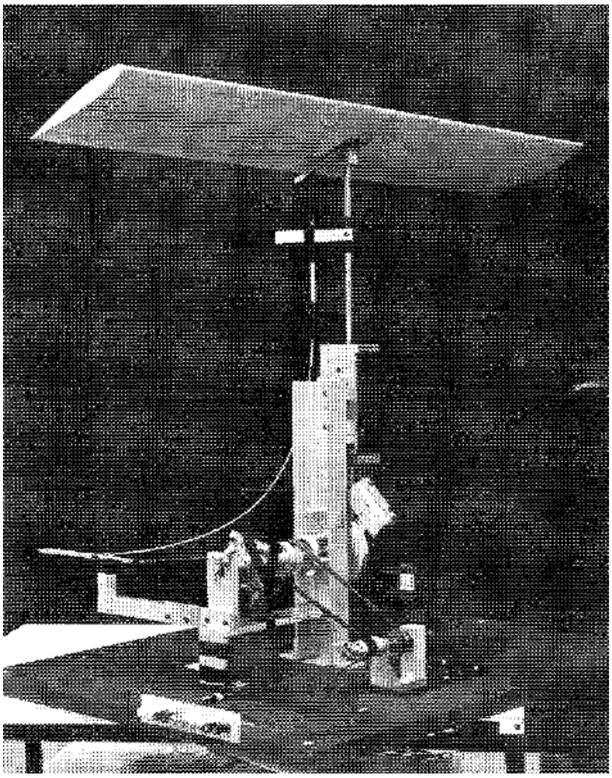


Figure 1.1: Wingmill experimental setup. Taken from [1].

One of the prototypes of the oscillating wing device called "Stingray", depicted in Fig. 1.3, was built by the Engineering Business Ltd. in England. A 150 kW prototype includes a single oscillating hydrofoil deployed in the Shetland Islands, Scotland. After several years of testing, the Stingray has unfortunately been tagged as a non–economically viable device, due to its poor performance. Reference [16] also mentions authors have performed a numerical analysis, which confirmed that the Stingray configuration was not optimal, inferring a higher pitch amplitude in addition to a higher

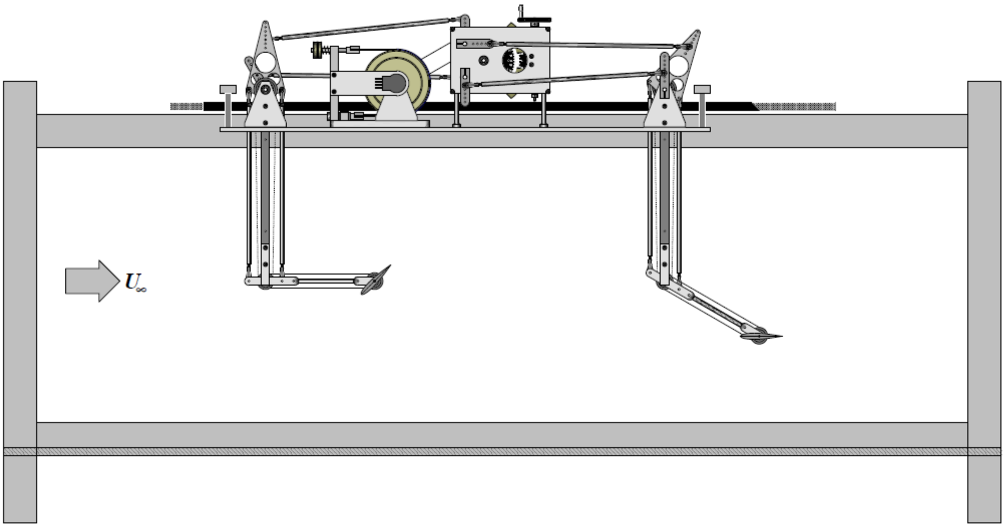


Figure 1.2: Side view of the oscillating–wing hydropower generator. Taken from [2].

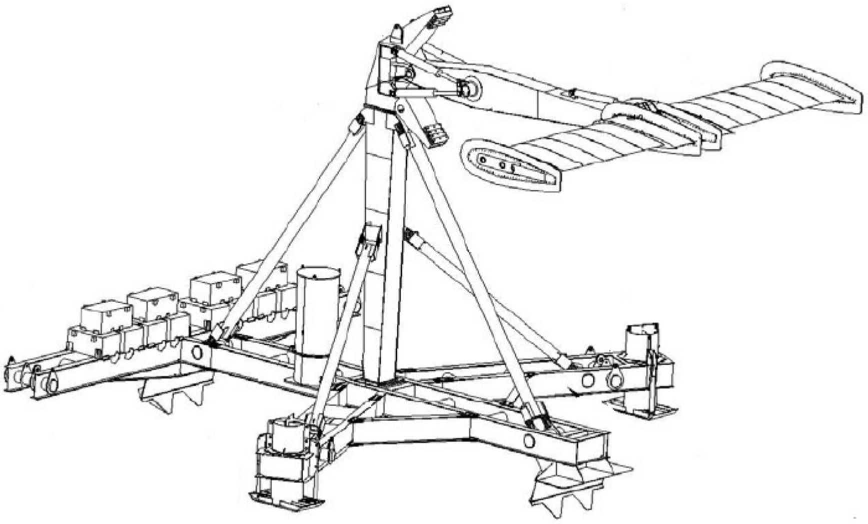


Figure 1.3: Stingray Assembly. Taken from [3].

frequency could lead to better performances.

Figure 1.4: The 100 kW Pulse tidal oscillating wing device. Taken from [4].

The Pulse Tidal Ltd., England has developed two prototypes of oscillating wing turbine which feature two hydrofoils in a tandem configuration. The first prototype, depicted in Fig. 1.4, featured 100 kW of power and was deployed in the Humber estuary in 2009. The oscillating wing device has 12 m–long hydrofoils mounted on a marine platform. The second prototype, which is the largest known prototype so far, was installed in the Bristol Channel in 2014 and features rated power of 1.2 MW [17]. Figure 1.5 depicts the artistic impression of the Pulse Tidal oscillating wing device.

Another experimental prototype of the oscillating wing for power generation was designed, built and tested by Laval University in water at Lac–Beauport near Quebec

Figure 1.5: An artistic impression of the 1.2 MW Pulse Tidal oscillating wing device. Taken from [5].

City, Canada [6]. A 2 *kW* prototype included two aspect ratio (*AR*) 7 rectangular hydrofoils in a tandem configuration. Their tips featured endplates and the Reynolds number was 0.5 million. The coupling of the pitching and heaving motion of each hydrofoil is coupled through four–link mechanism. The turbine has been mounted on a pontoon boat and dragged on a lake, in order to be easily tested at various operating conditions. Measured data reported fairly high values of the energy conversion efficiency. The experimental setup is depicted in Fig. 1.6.

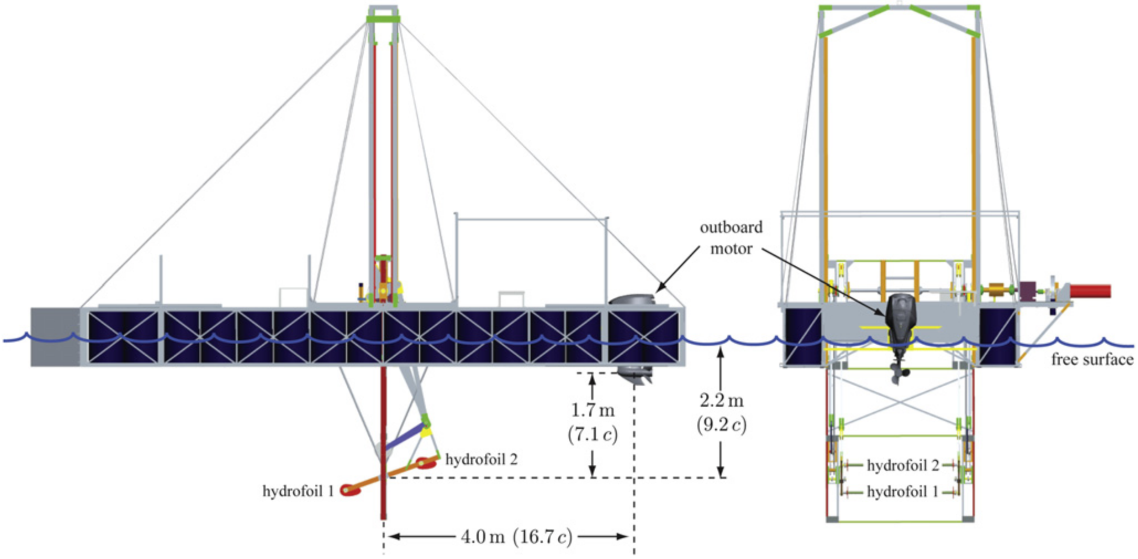


Figure 1.6: Side and rear views of the experimental setup. Taken from [6].

Most recently Young *et al.* [17] published a comprehensive review of the analytical, numerical and experimental research work carried out in this field. The review also focuses on the influence of flapping kinematics and foil geometry parameter choice on the characteristics of the LEVS observed in certain operating conditions. This feature has been initially thought to have a beneficial effect on the efficiency of the energy generation of oscillating wings, and its analysis in realistic installations is one of the underlying threads of the present study. The authors of [17] also highlight outstanding questions on the fluid mechanics of the oscillating wing in real installations, characterised by relatively high values of the Reynolds number based on the foil chord and the freestream velocity, and complex 3D flow features. The kinematic set–ups of oscillating wings for power generation can be subdivided in three classes [17, 18]: fully active, semi–passive and fully passive. In the fully active set–up all parameters of the heaving and pitching motions are prescribed; in the semi–passive set–up only the pitching motion is prescribed and the heaving motion parameters are determined by the hydrodynamic forces acting on the wing; in the fully passive arrangement, both the pitching and heaving motion parameters are determined by the forces acting on the wing. To date, it is still unclear which of the three set–ups provides the best performance [17], but progress made on improving the understanding of the hydrodynamic characteristics of any one of the three set–ups is likely to contribute to progress in the study and application of the other two [18]. The wing oscillation considered in most analyses is harmonic, but it has been shown that performance benefits can also be achieved by considering non–harmonic wing trajectories [19, 20]. The remainder of the literature survey in this subsection and the analyses in this work focus on the baseline configuration of the oscillating wing, namely that using a fully active kinematic set–up and harmonic wing motion.

Kinsey and Dumas [21] performed a thorough parametric computational fluid dynamics (CFD) investigation into the dependence of the energy conversion efficiency of a foil oscillating in a laminar Reynolds $1,100$–stream on the choice of motion parameters (heaving and pitching amplitude and motion frequency) and foil characteristic parameters (foil thickness and location of pitching axis). Their study used the commercial CFD code FLUENT and concluded that, by suitably choosing motion frequency and pitching amplitude, efficiencies as high as 34 % could be obtained. They also reported that the main factor enabling this efficiency level is the achievement of an optimal synchronisation (or phase) of wing motion and unsteady LEVS associated with the dynamic stall observed for certain choices of foil trajectory parameters. The study also considers the detailed aerodynamic analysis of this device to better understand the inside of the complex unsteady aerodynamics mechanism, which controls the energy extraction process. The detailed flow analyses highlight the importance of the use of NS CFD for this application. Similar findings were also reported in a later independent study using the NS research code COSA [22].

Thereafter, the hydrodynamics of the devices tested at Lac–Beauport [6] was investigated numerically by Kinsey and Dumas [23]. Both two–dimensional (2D) and three–dimensional (3D) turbulent FLUENT simulations using the Spalart–Allmaras turbulence model [24] were performed. The study highlighted that the loss of power generation efficiency of a single AR 7 wing with endplates in a water stream with $Re = 0.5 \times 10^6$ is about 15 % of the efficiency of the infinite wing. In a follow–up study, the same authors extended their numerical analyses to wings of AR 5, 7 and 10 with and without endplates to assess the dependence of the losses induced by finite–wing effects on aspect ratio and wing tip type setting again using $Re = 0.5 \times 10^6$. Making use of FLUENT simulations based on 3D grids with up to 3.5 million cells and using the Spalart–Allmaras model for the turbulence closure, their investigations concluded that, for a finite wing of $AR \geq 10$ with endplates such a loss could be limited to about 10 % of the efficiency of the infinite wing [25]. For the AR 10 case, however, no simulation of the wing without endplates was performed, and therefore, it was not possible to assess separately the efficiency improvement due to the use of endplates and that due to the use of a fairly large and more realistic AR 10.

The dependence of the oscillating wing hydrodynamics on the Reynolds number is another crucial factor essential to maximising the energy extraction efficiency of future real installations. Cross–comparison of laminar low–Reynolds number and turbulent high–Reynolds number CFD simulations using the same wing motion parameters reveals that such efficiency is significantly higher in the latter regime [21, 23, 26]. This was reported in [22], where COSA has been used to carry out a 2D fully laminar Reynolds 1,100–simulation and later on a 2D fully turbulent Reynolds 1.5 million–simulation [26] of the oscillating wing using the same wing motion parameters for both regimes. The comparative analysis reported in [26] used a wing trajectory that had been previously optimised for maximum energy extraction efficiency in the considered laminar regime, and provided two important observations. Firstly, the wing power generation efficiency increased at the turbulent high Reynolds number regime due primarily to thinner boundary layers, resulting in thinner effective foil and thus larger lift forces. Secondly, LEVS was delayed in the turbulent high Reynolds number regime with respect to the laminar low Reynolds number regime due to higher stability of the turbulent boundary layers. Thus the optimal synchronisation of wing motion and LEVS of the laminar regime was reduced in the high–Reynolds number case. However, the beneficial effect of thinner turbulent boundary layers outweighed the detrimental effect of abovesaid reduction of optimal synchronisation, resulting in higher efficiency of the foil in the turbulent stream. It was assumed that, for high Reynolds number regimes, resetting an optimal synchronisation of wing motion and LEVS by suitably varying the trajectory parameters could lead to an efficiency level even higher than that of 40 % obtained for the considered turbulent regime. However, Kinsey and Dumas later showed that high power generation efficiency at high Reynolds numbers does not necessarily rely on the occurrence of LEVS [27]. The detailed flow analyses of the 3D flow effects at realistic Reynolds numbers are still missing for this promising device, and will be addressed in this thesis.

1.3 Computational fluid dynamics for horizontal–axis wind turbines

Wind turbine is defined as a device which harnesses the kinetic energy of the wind and converts it into mechanical work, which can then be used for electricity production. HAWT represent the most common design of the wind turbines. The production of the HAWT power mainly depends on the interaction between the wind turbine rotor and the wind. The aerodynamic blade forces, generated by the wind, determine the main characteristics of the wind turbine performance, such as the power output and loads [28].

In this work the aerodynamics of two HAWT rotors have been considered. Firstly, NREL Phase VI rotor is used for the validation of newly developed 3D predictive capabilities of CFD Optimised Structured multi–block Algorithm (COSA), in straight and yawed wind conditions, and also to explore how well the physics of this flow problem can be captured with a NS technology. NREL Phase VI experiment [9] features many accurate and reliable measurements, and therefore represents a very suitable test case for CFD code validation. Secondly, the NREL 5–MW baseline turbine is herein used for the aerodynamic analyses in straight and yawed wind conditions. NREL 5–MW baseline fluid flow problem also serves the purpose of highlighting the high computational efficiency of the newly developed 3D HB solver in this research work.

Several other valuable experiments of the HAWT rotor aerodynamics have been conducted. One of the well known configurations is the MEXICO (Model Experiments in Controlled Conditions) experiment [29], which was designed to compliment the NREL Phase VI experiment [9]. However, for the purpose of this thesis only one validation test campaign was selected, which was the NREL Phase VI experiment.

1.3.1 NREL Phase VI wind turbine

NREL Phase VI wind turbine [9] has been developed in order to quantify the aerodynamic behaviour and the flow three–dimensionality of the full scale HAWTs. The turbine was designed using low–fidelity design codes, which rely on aerodynamic forces based on the steady 2D wind tunnel airfoil test results. The experiment has highlighted that very strong 3D effects exist in the wind turbine field operation. The experiment has been conducted by the National Renewable Energy Laboratory (NREL) at the National Wind Technology Center (NWTC) near Golden, Colorado, USA. The wind tunnel used was that located at the NASA Ames Research Center at Moffett Field, California, and has the size of $24.4m \times 36.6m$. The experiments included both upwind and downwind configurations, as well as rigid and teetered. The test matrix also included various cone, blade tip pitch and yaw angles. The turbine was studied at rotating and parked conditions and for the different blade tip configurations. Many accurate and reliable quantitative aerodynamic and structural measurements on a Phase VI wind turbine have been acquired, therefore, these data are excellent for validation of CFD models for novel designs and analyses of advanced wind energy devices. For this thesis several measurements where the turbine was yawed to various angles are of particular importance.

The set of Phase VI experimental data has been used by many CFD studies, particularly for the validation purposes. The test cases with the rotating blade at $72\,RPM$ are usually considered at various wind velocities. One of the first numerical studies on Phase VI HAWT was a blind code study organised by the NREL [30], which included BEM models, prescribed wake models, free wake models and NS codes. The study has highlighted that the majority of the predicted data deviated quite a lot from the experimental data. The results suggested that a lack of high–fidelity models for accurately predicting the aerodynamic behaviour of the flow on the HAWT blades exists. According to the NREL blind code comparison, Sørensen *et al.*have obtained one of the best overall agreement with the experimental data [31]. They have considered five operating conditions in straight wind, with the tip pitch angle 3°, where only the rotor has been modelled. For their work, they have used EllipSys3D CFD code, which is a multiblock finite volume discretization of the incompressible Reynolds–Averaged Navier–Stokes (RANS) equations and the $k-\omega$ shear stress transport (SST) turbulence model. The agreement between the computed results and experimental data is quite good for most of the operating conditions, except for the operating point at the $10m/s$, where the flow behaviour was very sensitive to the employed turbulence model.

Later on, Duque *et al.* [32] obtained a good agreement between the numerical and experimental data for the straight wind flow. They used compressible overset RANS code OVERFLOW–D2, and the 1–equation Baldwin–Barth turbulence model. They also compared RANS CFD calculations with a vortex lattice code CAMRAD II, and came to the conclusion that overall, the CFD calculations show much better agreement with the experimental data. The OVERFLOW–D2 is capable of predicting the stalled flow conditions, whereas the CAMRAD II code fails to accurately predict the stalled flow conditions.

Le Pape *et al.* [33] have conducted their work on Phase VI, as a continuation of previous numerical studies of Sørensen *et al.* and Duque *et al.* [31, 32]. They have used the compressible RANS solver ELSA with the $k-\omega$ SST turbulence model. The detailed flow analysis of the 3D and unsteady flow effects have been conducted for the zero–yaw configuration, for various numbers of operating conditions. The obtained results were in a reasonable agreement with the experimental data, however, it was suggested that the usage of the low speed preconditioning will most likely improve the results for a compressible NS solver. Later on Le Pape *et al.* [33] conducted the calculations for both the straight and 10° yawed wind flow, with the usage of the low speed preconditioning [34]. This study has shown that a low speed preconditioner indeed improves the accuracy of the results at low wind speeds and allows better prediction of the stall point.

Due to the popularity of this test case for code validation, and further aerodynamic analyses, more recent studies have followed. Gómez–Iradi *et al.* [35] performed analyses using the compressible NS solver with the $k-\omega$ SST turbulence model. Overall good agreement between the numerical and experimental data has been obtained. This study has also taken into the account the effect of the wind–tunnel wall effects on the blade aerodynamics, and investigated the blade/tower interaction. Yu *et al.* [36] conducted a study on the overpredicted lift and stall delay of straight wind calculation, using the incompressible RANS solver with the $k-\omega$ SST turbulence model with transition correction. Stall delay is a condition, where the AoA at which stall occurs is greater for a rotating blade compared to static airfoil. The study has confirmed there is a stall delay phenomenon in the inboard part of the turbine blade. The NS results were compared with the experimental data and the lifting surface method with and without Du–Selig stall delay model. The NS results for low–speeds are in good agreement with the experimental data, however, at relatively high wind speed ($10m/s$), when the massive flow separation occurs, some discrepancy between the numerical and experimental data exists. Lifting surface method with Du–Selig stall delay model agrees well with the experimental data, whereas, without Du–Selig model it fails to accurately predict wind turbine performance. Another study, performed by Moshfeghi *et al.* [37], investigated the effects of near–wall grid spacing and aerodynamic behaviour of Phase VI blade using Ansys–CFX11 and the $k-\omega$ SST turbulence model. The study reports that in order to obtain grid–independent solution, presently used 5 million cell grids, would require more refinement in both chordwise and spanwise directions. Furthermore, the authors show there is a significant dependence of computed forces on near–wall spacing. Moreover, the results indicate that the $k-\omega$ SST turbulence model mispredicts separation point, and over predicts separation. Yu *et al.* [38] examined the rotor under the yawed flow conditions for two yaw angles of 30° and 60° for three wind speeds of 7, 10 and 15 m/s. The unstructured incompressible RANS solver with the $k-\omega$ SST and correlation–based transition turbulence model has been used. The study reports that the blade aerodynamic loading is significantly reduced under the yawed wind flow. An overall good agreement between the numerical and experimental data for all wind speeds has been observed.

1.3.2 NREL 5–MW baseline wind turbine

NREL offshore 5–MW baseline wind turbine, is a virtual model of a modern commercial three–bladed upwind variable–speed variable blade–pitch–to–feather–controlled turbine, developed at NREL [39]. The initial purpose of the turbine has been to establish the reference specifications for a number of research projects supported by the U.S. DOE's Wind & Hydropower Technologies Program [39]. However, since the NREL offshore 5–MW baseline wind turbine has been adopted as a reference model by Union UpWind research program and the IEA Wind Annex XXIII Subtask 2 Offshore Code Comparison Collaboration (OC3), the model has been used as a reference by many research teams worldwide, for various different purposes. The aim of this model is to standardise onshore and offshore multimegawatt turbine specifications, and to quantify the benefits of advanced onshore and offshore wind energy technologies.

Sørensen and Johansen [40] have performed the NREL 5–MW baseline rotor calculations in zero yaw wind for the uniform inflow and atmospheric boundary layer cases. The purpose of the study was to investigate the unsteady effects due to the severe vertical shear, and to compare the agreement with the BEM code. The calculations were performed using an incompressible NS CFD code EllipSys3D using the $k-\omega$ SST turbulence model. Later on, Chow and van Dam [41] have investigated the aerodynamic characteristics of the NREL 5–MW baseline rotor, using the compressible NS code OVERFLOW2 featuring the $k-\omega$ SST turbulence model. The study reports that the significant radial flow and inboard blade separation exist, and can be limited using various passive geometric modifications, resulting in improved power prediction. Additionally, it was concluded that the increasing inboard blade twist does not have a beneficial effect on power production. This study also compared the CFD results obtained by the incompressible CFD code EllipSys3D, reported by Sørensen and Johansen in [40], and BEM predictions. The two CFD results at low wind velocities are in excellent agreement. However, once the rated power is reached, OVERFLOW2 predicts slightly higher power with respect to the EllipSys3D solutions. Before the rated speed, BEM predictions match well with both CFD codes. However, after the rated speed is achieved, the BEM analyses significantly overpredict the result of both CFD codes. Chow and van Dam's study was further extended in [42], where a comprehensive near–body grid independence study has been performed. It was found that the solution strongly depends on the near–body wake grid refinement, as the rapid diffusion of the wake in case of insufficient refinement contributes to the overpredicted torque and thrust. Chow and van Dam have performed another study on the NREL 5–MW baseline rotor [43], where they investigated the impact of the blade twist and blunt trailing–edge in the inboard region of the blade on the aerodynamic performance. They concluded that decreasing the sectional twist angle, which increases the sectional AoA, causes deeper stall of the inboard region of the blade, keeps the rotor torque constant and increases the thrust. Whereas, when increasing the sectional twist angle, the rotor thrust drops at faster pace than the torque, which could potentially lead to a more optimal designs in terms of the torque versus thrust ratio. Furthermore, it is also reported that the twist distribution of the NREL 5–MW baseline rotor is already nearly optimal. Moreover, the study pointed out that the blunt trailing edge modifications in the inboard part of the blade have a positive effect on the turbine performance, and completely change the flow behaviour. Chow and van Dam also pointed out that only high–fidelity 3D simulations are able to accurately predict the flow field of the modern rotor designs, due to their geometric complexity.

Troldborg *et al.* [44] investigated the wake of the NREL 5–MW baseline rotor, using the incompressible EllipSys3D NS code, either using the $k-\omega$ SST or the DES version of $k-\omega$ SST turbulence model by Strelets. The analyses included fully resolved rotor simulations, the actuator disc model and the actuator line model, and featured uniform or turbulent inflow. RANS simulations in uniform inflow found good resemblance between the actuator disc and line methods, whereas, fully resolved simulation predicted stronger vorticity generated by the tip vortices. Switching from RANS to DES in uniform inflow strongly affected the wake dynamics predicted by the fully resolved simulation, but had limited effect on the actuator disc and line methods. The differences between the actuator line model and fully resolved simulation became much smaller when using turbulent inflow, therefore, an actuator disc or line representation of the rotor may be sufficient when simulating wind turbines in the atmospheric boundary layer. Imiela *et al.* [45] also investigated the NREL 5–MW baseline HAWT within the multidisciplinary design and analysis framework for wind turbines (MERWind) project. They performed various steady and unsteady aerodynamic analyses of the NREL 5–MW baseline turbine at rated wind speed for both rotor calculations and the complete turbine calculations. For all aerodynamic analyses the compressible NS code TAU has been used, which uses the $k-\omega$ SST turbulence model. The study reports that for rotor only configuration, the average torque of logarithmic boundary layer profile inflow was about 15 % lower than in uniform inflow conditions. It was also found that the mean torque is only slightly lower in the complete wind turbine simulation with respect to the rotor only configuration, for both inflow conditions. This project involved also the fluid–structure interaction simulations using the multi–body simulation software SIMPACK and TAU CFD code.

As per above it may be concluded that none of the studies on the NREL 5–MW baseline HAWT are based on yawed aerodynamics. Hence, this will be addressed herein.

1.4 Harmonic balance method in computational fluid dynamics

The harmonic balance (HB) NS technology for the solution of unsteady periodic flows [46] is one of the most promising technologies of this type. This method has been successfully applied to the prediction of the periodic flow associated with the flutter and forced response of turbomachinery blades [46, 47, 48], and various vibratory motion modes of aircraft configurations [49, 50, 51]. For this type of application, the use of the HB NS approach for the calculation of periodic flows can lead to runtime reductions varying between one and two orders of magnitude with respect to conventional time–domain (TD) NS analyses.

The 2D investigations into the use of the HB NS technology [46] for reducing the analysis runtime of the periodic flow field past HAWT rotor blade sections have already reported significant improvements in terms of computational speedup. The aerodynamic analysis study on HAWT airfoils, reported in [52], was based on the compressible laminar NS equations and used low–speed preconditioning to handle the numerical difficulties resulting from the typically low speeds of wind turbine flows. More realistic turbulent flow demonstrations of this method for HAWT turbulent aerodynamics have followed, including the study on the unsteady aerodynamics of a pitching S809 airfoil [53], making use of the one–equation Spalart–Allmaras turbulence model, that in [54], which is also studying the unsteady aerodynamics of a pitching S809 airfoil and using a zonal transition model with the Spalart–Allmaras model. The studies [55] and [56] are both using airfoils which construct NREL 5–MW wind turbine and making use of Menter's two–equation SST turbulence model. These studies indicate a growth in the use of this high–fidelity approach for the analysis of HAWT periodic aerodynamics.

The author of the following studies [53] and [54] has also investigated the flutter phenomena for the 1.5 MW WindPACT rotor blade [57]. The aeroelastic model used includes the HB method and a fully turbulent aerodynamic model. The study concluded that HB method can be successfully used for the prediction of wind turbine blade flutter.

In addition to the HB method for the solution of unsteady periodic flows [46], several other frequency–domain methods exist, which exhibit resemblance to the HB approach of [46]. One of the recent studies on the unsteady aerodynamics of HAWT has been performed on the rotor–tower interactions of the DTU 10 MW reference wind turbine [58], making use of the Non–Linear Harmonic (NLH) method [59]. The important difference between the HB [46] and NLH [59] methods is that the calculation of the zeroth harmonic (mean flow) is decoupled from the first harmonic (unsteady flow component) in NLH case. To date it is still unclear whether the HB and NLH have the same predictive capabilities. The authors of [58] concluded that the NLH method is capable of capturing the complex unsteady aerodynamics of the rotor–tower interaction, and is about 10 times faster than the time–dependent RANS approach for comparable accuracy. Several other examples of the application of the NLH and other promising frequency–domain technologies to periodic flows of engineering interest exist.

As per above it may be concluded that most of the studies on HAWTs aerodynamics using the HB approach of [46] so far were based mainly on 2D analyses, with the exception of [57] which investigated the flutter phenomena of the wind turbine rotor blade. More studies on realistic 3D turbulent analyses of HAWTs unsteady aerodynamics are still missing in the literature and will be addressed herein.

1.5 Motivation, aims and objectives

The aerodynamic design of renewable energy devices, such as oscillating wings and HAWTs represents a very challenging, complex and multidisciplinary task. The consideration of a large number of operating regimes due to the extreme variability of the environmental conditions needs to be taken into the account. The oscillating wing devices are presently installed either on the shallow river– or sea–beds and are used as a hydro or as a tidal power generating machines. The environmental variability of water flow or tides can vary on the time scales ranging from hours (e.g. tide changes), months to years (e.g. seasonal tide or river flow variations) due to a number of factors. The variability of the wind is even more complex, as it can range on the time scales from seconds (e.g. wind gusts) to months (e.g. seasonal wind variations). Several operating unsteady regimes for both devices, however, are predominantly periodic. In the case of utility–scale HAWTs, periodic fluid–induced excitations of the rotor blades and drivetrain may result from the blades rotating *a*) through wind stratifications associated with the atmospheric boundary layer, *b*) through the variable pressure field due to the presence of the tower (multimegawatt turbines typically feature upwind rotors), *c*) through portions of the wake shed by an upstream turbine in the wind farm environment, *d*) in yawed wind, a condition occurring when the freestream wind velocity is not orthogonal to the turbine rotor [60], and *e*) in a region of nonuniform wind resulting from the combination of two or more of the kind of phenomena mentioned above. With regard to yaw misalignments, utility–scale HAWTs typically feature yaw control systems that monitor the direction of the wind and rotate the entire nacelle towards the wind [28]. However, yaw actuators adjust the nacelle position only after a yaw error has been detected for a relatively long time–interval, usually 10 minutes. Therefore, at sites with frequent variations of the wind direction, blade and drivetrain fatigue due to yawed wind can be significant. HAWT rotors experience constant periodic excitations when the turbines are placed at inclined sites, such as mountainous terrains. Here wind speeds are often higher than on flat terrain due to the acceleration induced by the surface geometry, however the entire wind stream is inclined on the ground, and this yields periodic rotor flows similar to those induced by yaw errors [61]. In all these cases, the fundamental frequency of the periodic excitation is a multiple of the rotor speed.

The comments above highlight the necessity of accurately predicting periodic flows when designing both oscillating wing devices and HAWTs. This is of crucial importance for reliably predicting the actual amount of harvested energy and the fatigue–inducing loads which may reduce machine life and/or increase its operation and maintenance costs.

The concept of oscillating wing device is relatively new, and due to the extremely complex flowfield of such device, most of the studies in this field are already carried out using high–fidelity numerical methods such as NS CFD, or directly the experimental studies. However, as previously stated in the subsection 1.2, most of the studies so far are based solely on 2D fully laminar or fully turbulent analyses, and do not take into the account the impact the flow three–dimensionality. The significant uncertainty on the impact of 3D flow features on the power generation efficiency of future real installations still exists, particularly at the expectedly high Reynolds numbers. The detailed study of the hydrodynamics of AR 10 oscillating wings with and without endplates at a realistic Reynolds number of 1.5 million, is a challenge that has not been previously addressed. The investigation considers two operating regimes both close–to–optimal wing motion parameters, based on reported 2D CFD analyses [27]. The study aims at assessing the level and mechanisms of the power generation efficiency loss, estimating the efficiency improvements due to the use of endplates for the considered aspect ratio, and highlighting a new route to further efficiency improvement of a wing with endplates. The investigation is based on 3D time–dependent RANS simulations based on a 30.7 million cell–grid and using the Menter's SST turbulence model. To the best of the authors' knowledge, a detailed study of the comparative analyses of the infinite and the two AR 10 finite span oscillating wings for high–extraction efficiency regimes, has not yet been published. This work presents important observations and guidelines for further design and development of such devices. It uses much finer grids than other state–of–the–art computational works and it also includes careful analysis of grid–refinement–induced uncertainty.

HAWT design methods still rely on low–fidelity and/or semi–empirical models, such as the BEMT and dynamic stall models [11, 12, 13]. The main advantage of these techniques is their extremely high computational speed, whereas, the main drawback is their reliance on the existence and availability of high–quality airfoil data, hindering their applicability to the design of radically new turbine configurations. Moreover, these low–fidelity methods model strongly unsteady 3D flow features, such as HAWT yawed flows and the radial pumping effect, occurring in the presence of stalled flow [62] with a high degree of uncertainty, even when detailed airfoil data are available. A wider discussion on the predictive reliability of low–fidelity tools for the wind turbine design can be found in [55]. The use of high–fidelity computational aerodynamics tools such as NS CFD codes has the potential of greatly reducing the uncertainty associated with the flow predictions of low–fidelity models. Several remarkable examples of the predictive capabilities of NS CFD for HAWT yawed flows have been published, including the articles [34, 63, 64, 65]. The article [63] also includes comparisons of CFD NS results, experimental data and results obtained with low–fidelity codes, including a BEMT code. The report shows that the agreement between NS CFD analysis and measured data is substantially better than that between low–fidelity analyses and measured data, as expected. The prediction of the yawed wind–induced periodic loads acting on the NREL 5–MW baseline turbine blade has previously not been addressed. The investigation, which is based on comparative TD and HB RANS simulations based on a 2.7 million cell–grid per blade sector and using the Menter's SST turbulence model, provides and insight into the flow patterns caused by the yaw misalignment, and more importantly, it highlights the potential of the HB method for the solution of the HAWT periodic flows [46]. One of the main drawbacks of TD NS simulations is their high computational cost. A fully time–resolved TD NS simulation of HAWT periodic flows requires a long runtime as several rotor revolutions have to be simulated before the periodic state of interest is achieved. This runtime could be significantly reduced by using a frequency–domain HB formulation and solution of the governing unsteady equations. This work assesses quantitative measures of the actual benefits of using realistic turbulent 3D HB NS solvers for HAWT design. More precisely, the reduction in analysis runtime for HAWT periodic flows using a 3D turbulent HB NS code, while still maintaining a prediction accuracy comparable to that of the corresponding TD code, were assessed. To the best of the author's knowledge, this work is the first reported computational study of the 3D yawed flow unsteady HAWT aerodynamics, by means of the HB method [46].

Starting point of the numerical tools to achieve the aims of presented thesis, was the 2D version of COSA, and corresponding 2D pre– and post–processing tools, developed at School of Engineering of Glasgow University and Engineering Department of Lancaster University. As a necessary step for this PhD research, the following algorithmic developments had to be performed:

- Optimisation and further validation of the existing 2D COSA solvers.
- Extension from 2D to 3D of all existing serial and parallel COSA solvers.
- Extension from 2D to 3D of all existing pre– and post–processing tools.
- Implementation RANS and SST equations in non–inertial frame for the external aerodynamics rotational applications.
- Implementation of steady and multi–frequency periodicity boundary conditions.
- Parallelization of newly developed 3D capabilities.
- Standardisation of the I/O of COSA by the means of standard portable CFD formats such as CGNS based on HDF5 and Tecplot 360's binary file format and the newer subzone load format, which was crucial for the compatibility reasons with the state–of–the–art commercial grid generators and post–processing tools.

The aims of this research work are summarised below:

- Detailed numerical investigations into the impact of flow three–dimensionality on the power generation efficiency for the two realistic oscillating wing configurations.
- Investigation of the yawed wind–induced periodic loads acting on the NREL 5–MW baseline turbine blade.

- Application of the HB method to complex 3D yawed flow aerodynamics of HAWTs.
- Thorough assessment of the computational efficiency of the 3D turbulent RANS SST HB solver, by comparing the runtimes of TD and HB turbulent analyses of a NREL 5–MW baseline turbine yawed wind flows.
- Maintenance of the highly efficient parallel environment of extended and newly developed parts of the 3D solvers.

1.6 Thesis outline

The thesis has the following structure:

Chapter 1 summarises past studies on CFD analyses of wind turbines and oscillating wing devices, as well as harmonic balance method for Navier Stokes CFD. The aims and objectives of the thesis are outlined.

Chapter 2 presents the governing equations in inertial and non–inertial frame of reference. It explains the turbulence closure problem, the Reynolds–Favre averaging approach of the turbulence time–scales, and the Boussinesq approximation. Boundary conditions used in this work are also presented.

Chapter 3 describes the spatial and temporal discretization, as well as the integration approach of the steady, time–domain and harmonic balance solvers. A broad description of a multi–block approach and the steady and multi–frequency periodicity cut boundary conditions is given. Various convergence acceleration techniques used in this work are also explained.

Chapter 4 details massive parallelism, based on MPI standard, used in this work. The definitions of parallelism, high performance computing (HPC), a brief description of different types of HPC machines, and parallel programming approaches are all reported.

Chapter 5 presents numerical validation of newly developed 3D solvers. Firstly, preliminary validation cases, delta wing, ONERA M6 wing, and S809 airfoil are reported. Secondly, the main validation test cases, the H–Darrieus vertical–axis wind turbine, oscillating wing, and the upwind configuration of the NREL Phase VI wind turbine are provided. Numerical results, have either been compared with the experimental data, if those existed, or several other state–of–the–art CFD codes.

Chapter 6 presents main results of the numerical analyses undertaken in this research work. It is organised into two separate sections, one is focusing on the analyses of the oscillating wing devices, and the other reports the analyses of the NREL 5–MW baseline HAWT. In both sections the fundamentals of each device are first explained.

The analyses of oscillating wing devices are based on two considered operating regimes: case A, which is characterised by a high efficiency of energy extraction due to the occurrence of LEVS; and case B, characterised by high efficiency of the energy extraction in absence of LEVS. First the 2D grid–independency study was performed for both regimes. Second, a detailed numerical flow investigation, based on the comparative performance assessment of an infinite wing and two aspect ratio 10 wings featuring different wing end geometry, either sharp tips or endplates was conducted. Comparative analysis between the two considered operating regimes are also reported.

The section on HAWTs, contains the yawed flow analysis of the 3D turbulent periodic flow past the rotating blade of the NREL 5–MW baseline HAWT. A detailed aerodynamic discussion of this flow problem is provided. Furthermore, the time refinement analyses with the TD solver and spectral refinement analyses with the HB solver were performed to determine the speed–up of the HB simulation, yielding a solution accuracy comparable to that of the fully resolved TD simulation.

Chapter 7 summarises the main conclusions and observations of the presented research, alongside some perspectives for the future work.

Chapter 2

Governing equations

This chapter outlines the derivation of the Arbitrary Lagrangian–Eulerian (ALE) form of the steady, TD and frequency–domain HB RANS equations and the $k-\omega$ SST turbulence model, in a Cartesian coordinate system, for both inertial and non–inertial frames of reference. The presented equations are the basis of the finite volume algorithm of the newly developed solvers of COSA. The derivation is a multi–stage process, in which the first step is the derivation of the NS equations from general conservation laws of fluid dynamics. Furthermore, the turbulence closure problem, Reynolds–Favre averaging approach are explained. Boussinesq approximation is also introduced, which is required to define the Reynolds stress tensor in the context of the adopted linear eddy viscosity approach. To complete the set of RANS equations, Menter's $k-\omega$ SST turbulence model is described. Once all approximations used to define the complete set of governing equations have been introduced, the unsteady RANS and SST equations are expressed in the differential conservative form. The following sections introduce the derivation of the ALE form of steady and TD RANS and SST equations for inertial frame. The same derivation is also provided for a non–inertial frame of reference. Next, the HB formulation of the RANS and SST equations for both frames of reference is introduced. The chapter is concluded by the description of the boundary conditions, which are the fundamental part of any numerical simulation.

2.1 Conservation laws

A conservation law of a particular quantity of an isolated physical system states that its variation within an arbitrary volume can be expressed as a balance between the various processes tending to increase of decrease it.

The derivation of the equations governing the fluid motion is based on the fact that the dynamic behaviour of a fluid is defined by the conservation of the following three fundamental physical quantities [66]:

- the conservation of mass (continuity equation),
- the conservation of momentum (Newton's 2nd Law),
- the conservation of energy (1st Law of Thermodynamics).

Collecting all equations obtained by enforcing these conservational laws yields a coupled system of equations, the so–called NS equations. Strictly speaking, the expression NS equations refers only to the momentum conservation law equations; however, in the recent years, the expression has been frequently used to indicate the complete system of conservational laws, including continuity and energy equation. As a result, the expression "NS equations" is now used to denote all three conservation laws in most cases.

In fluid flows the most convenient and frequently used method is to consider the flow within an arbitrary spatial region, called the control volume C . This model enables one to focus solely on the fluid in the finite region enclosed by the volume itself [66, 67]. By applying the fundamental physical principles to a finite control volume, the integral form of the NS equations is directly obtained. This formulation can then be manipulated and transformed into differential form. As sketched in Fig. 2.1, a non–deforming control volume C may be fixed in space and time with the fluid moving trough it, called flux, bounded by the closed surface boundary called control surface S . We also introduce dC which is a control volume element, dS representing a control surface element and its associated outward pointing unit normal vector $\underline{n}$. The streamlines in Fig. 2.1 represent the general flow field.

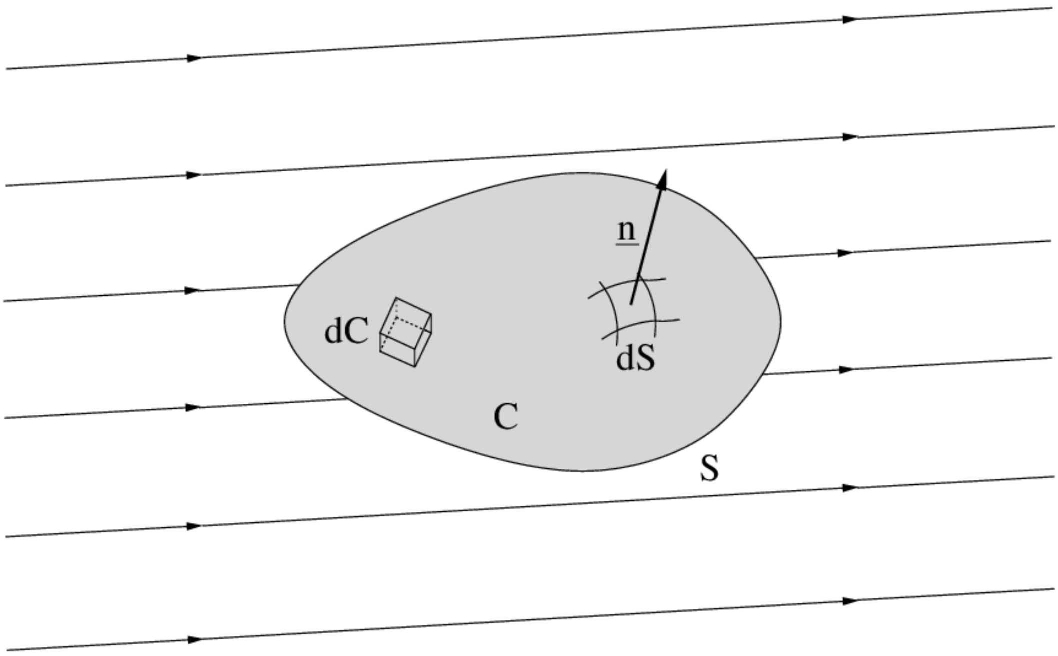


Figure 2.1: Finite control volume.

The flux can generally be decomposed into convective and diffusive part. The first part is due to the convective transport of the fluid, while the second part is due to the molecular motion, which can be present even when the fluid is at rest. The second contribution is of a diffusive nature, due to differences in the intensity of molecular motion fluid tends toward equilibrium and uniformity. This part of contribution to the total flux is proportional to the gradient of the considered quantity as it has to vanish as equilibrium is established [66, 68].

Conservation of mass is one of the fundamental concepts of classical physics and it states that mass can be neither created nor destroyed but can only be moved in space. For the finite control volume which is fixed in space and time C, as sketched in Fig. 2.1, conservation of mass states that the time rate of change of the total mass in the control volume equals the net mass flux through the control surface S of the control volume:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0. \tag{2.1}$$

The symbol ρ denotes the fluid density and u_i, $i = 1, 3$ denotes the $i-th$ component of the flow velocity vector $\underline{u}$ and x_i $i = 1, 3$ denotes the $i-th$ component of the position vector $\underline{x}$. The position vector $\underline{x}$ is a standard Cartesian vector and its components are the coordinates x, y and z. In Eqn. (2.1) and throughout the thesis the Einstein summation convention is used, which implies that when the same index in a term appears twice, the summation with respect to that index over its range is performed.

Another fundamental principle of classical physics is Newton's second law. The law states that the time rate change of momentum of a fluid particle equals the sum of the forces acting on that particle. In Cartesian coordinates it can be written as:

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_j u_i) = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ji}}{\partial x_j}, \tag{2.2}$$

where p represents the pressure, and $\underline{\underline{\tau}}$ is the viscous stress tensor, which is defined as:

$$\underline{\underline{\tau}} = \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix}. \tag{2.3}$$

Normal stresses are defined by the components τ_{xx}, τ_{yy} and τ_{zz}, whereas the rest of the components represent shear stresses. Laminar flows only include molecular viscous stress tensor $\underline{\underline{\tau}} = \underline{\underline{\tau}}^M$, whereas, turbulent flows also include Reynolds stress tensor, which will be defined later on. The components of the molecular viscous stress tensor for a Newtonian fluid are defined by the relation:

$$\tau_{ij}^M = 2\mu \left[S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij} \right], \tag{2.4}$$

$$S_{ij} = \frac{1}{2}\left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right], \tag{2.5}$$

where μ is the dynamic viscosity, S_{ij} stands for the strain rate tensor, and δ_{ij} is the Kronecker delta function.

The dynamic viscosity μ is a function of the absolute temperature T of an ideal gas and is strongly influenced by the temperature. For gases, it is frequently computed by the Sutherland's Law:

$$\mu = \mu_{ref} \left(\frac{T}{T_{ref}} \right)^{3/2} \frac{T_{ref} + T_S}{T + T_S}, \tag{2.6}$$

where T_{ref} and μ_{ref} represent the reference temperature and the reference dynamic viscosity, respectively and T_S is the Sutherland temperature. Sutherland's Law coefficients for air in SI units are the following:

$$\mu_{ref} = 1.716 \times 10^{-5}\ kgm^{-1}s^{-1},\ T_{ref} = 273.15\ K,\ T_S = 110.4\ K.$$

Conservation of energy is another fundamental concept of classical physics and it states that energy can be neither created nor destroyed, it can only be transformed from one form to another. The energy conservation equation in Cartesian coordinates reads:

$$\frac{\partial}{\partial t}\rho E + \frac{\partial}{\partial x_j}(\rho H u_j) = -\frac{\partial q_j}{\partial x_j} + \frac{\partial}{\partial x_j}(\tau_{ij} u_i). \tag{2.7}$$

The total energy E per unit mass is defined as:

$$E = e + \frac{u_i u_i}{2}, \tag{2.8}$$

where e denotes the internal energy per unit mass, and the term $\frac{u_i u_i}{2}$ is the kinetic energy per unit mass. The thermal heat flux $\underline{\mathrm{q}}$, can be expressed by the generalised Fick's gradient law, $\underline{\mathrm{q}} = -k_T \nabla T$, where k_T represents the thermal conductivity. Generally, in CFD most flows can be considered as an isentropic perfect gas, therefore, the equation of state can be written in the following form:

$$p = \rho R T, \tag{2.9}$$

where R denotes the specific gas constant. A calorically perfect gas, has the internal energy e and the static enthalpy h that can be related to the temperature by:

$$e = c_v T; h = c_p T. \tag{2.10}$$

c_v and c_p represent specific heat under constant volume and at constant pressure, respectively. Introducing Mayer's relation for the calorically perfect gas:

$$R = c_p - c_v, \tag{2.11}$$

and combining perfect gas relations, the expression for the static enthalpy becomes:

$$h = c_p T = e + \frac{P}{\rho}. \tag{2.12}$$

Hence, the total energy E can be expressed as the function of total enthalpy H :

$$H = h + \frac{\mathrm{u}^2}{2} = E + \frac{p}{\rho}. \tag{2.13}$$

2.2 Turbulence closure

Turbulence is the state of fluid motion characterised by random three dimensional and time dependent fluctuations of the various flow properties around their mean values. It is a fundamental property of fluid mechanics, and still represents one of the most challenging problems in this field. Majority of flows occurring in nature do not exhibit laminar behaviour, but are rather transitional or turbulent. This is also true for the majority of engineering applications.

To fully describe the nature of the turbulent flow, an enormous amount of information is needed. To reduce the complexity of the system of governing equations, different approximations are available. This work uses RANS approximation, where only the largest eddies are directly resolved, and the rest of the eddies are modelled. In RANS modelling approach the time–dependent NS equations are averaged on the time– and length–scales of turbulence. The approach differs from NS equations mostly because of the presence of the Reynolds stresses, accounting in an averaged manner for the effects of turbulence. The RANS equations are open, because additional equations are needed to determine Reynolds stresses. To close the RANS equations, a variety of turbulence models of various complexities have been developed, and the research in this area is still ongoing. In this work, the Menter's $k - \omega$ SST model is used [69, 70], which features two additional transport equations, one for the turbulent kinetic energy k , and one for the specific dissipation rate ω.

2.2.1 Reynolds–Favre averaging

An important concept to separate turbulent fluctuations from the mean flow solution was presented by Reynolds in 1895 [71] and is called Reynolds averaging [66, 72]. The main idea of this approach is to solve governing equations only for the mean values of the turbulent flow field, and to avoid the costly temporal– and space–solution of its fluctuating components.

In compressible flows, where the density is not constant, it is advisable to use an alternative time averaging procedure called Reynolds–Favre–averaging. Reynolds–Favre averaging approach, which has been used in this work, uses Reynolds averaging for density and pressure, and Favre averaging for all other variables. Favre–averaging is a mass weighted time averaging procedure, by which the additional correlations involving density fluctuations can be avoided.

2.2.2 Boussinesq approximation

Bossinesq introduced an important approximation to turbulence modelling [66, 72], which is based on the assumption that the momentum transfer in turbulent flow caused by turbulent eddies, can be modelled making use of an eddy viscosity. His approximation assumes that, as in the laminar flow, also the turbulent shear stress depends linearly on the mean strain rate. According to Bossinesq approximation the Reynolds stress tensor $\underline{\underline{\tau}}^R$ can be written as:

$$\tau_{ij}^R = 2\mu_T(\tilde{S}_{ij} - \frac{1}{3}\frac{\partial \tilde{u}_k}{\partial x_k}\delta_{ij}) - \frac{2}{3}\bar{\rho}\tilde{k}\delta_{ij}, \tag{2.14}$$

$$\tilde{S}_{ij} = \frac{1}{2}[\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}], \tag{2.15}$$

where $\tilde{S}_{ij}$ denotes the Reynolds strain rate tensor, μ_T is the turbulent or eddy viscosity and the symbols $\bar{\ }$ and $\tilde{\ }$ refer to the Reynolds and Favre averaging operators respectively. This tensor depends primarily on the product of the strain rate tensor and the eddy viscosity.

The viscous stress tensor $\underline{\underline{\tau}}$ is thus given by the sum of the molecular stress tensor defined by Eqn. (2.4) and the Reynolds stress tensor. This can be expressed as:

$$\underline{\underline{\tau}} = \underline{\underline{\tau}}^M + \underline{\underline{\tau}}^R. \tag{2.16}$$

2.2.3 $k - \omega$ SST turbulence closure

Menter's $k - \omega$ SST turbulence model [69, 70] is a combination of the standard $k - \epsilon$ model [73] and Wilcox's $k - \omega$ model [72, 74]. More specifically, the $k - \omega$ model is employed to accurately resolve the laminar sublayer of the turbulent boundary layer, as it does not involve damping functions, and for similar accuracy leads to significantly higher numerical stability in comparison with $k - \epsilon$ model. The $k - \omega$ is also used in the logarithmic part of the boundary layer as it performs significantly better under adverse pressure–gradient conditions than the $k - \epsilon$ model. The $k - \epsilon$ model is instead employed in the wake and farfield regions, as the results of this model do not have a strong sensitivity to the freestream value of the second turbulent variable (ϵ or ω, when the $k - \epsilon$ model is rewritten in terms of ω to obtain the SST formulation). This is a remarkable advantage of the $k - \epsilon$ model over the $k - \omega$ model, the solution of which instead depends significantly on the value of ω outside the shear layer. Given the difficulty of properly choosing freestream values of ω, a fairly low sensitivity of the turbulent characteristics on such freestream value is a desirable feature of two–equation models. Furthermore, $k-\epsilon$ is also used in free shear layers, as it performs well when pressure gradients are small [75].

The transport equations for the turbulent kinetic energy k and the specific dissipation rate ω are respectively:

$$\frac{\partial}{\partial t}(\rho k)+\frac{\partial}{\partial x_j}(\rho u_j k)=\tau_{ij}^R\frac{\partial u_i}{\partial x_j}-\beta^*\rho\omega k+\frac{\partial}{\partial x_j}\left[(\mu+\sigma_k\mu_T)\frac{\partial k}{\partial x_j}\right], \tag{2.17}$$

$$\begin{aligned}\frac{\partial}{\partial t}(\rho\omega)+\frac{\partial}{\partial x_j}(\rho u_j\omega)=&\frac{\gamma\rho}{\mu_T}\tau_{ij}^R\frac{\partial u_i}{\partial x_j}-\beta\rho\omega^2+\frac{\partial}{\partial x_j}\left[(\mu+\sigma_\omega\mu_T)\frac{\partial\omega}{\partial x_j}\right]\\&+2\rho(1-F_1)\sigma_{\omega_2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial\omega}{\partial x_j}.\end{aligned} \tag{2.18}$$

The terms $\tau_{ij}^R\frac{\partial u_i}{\partial x_j}$ in Eqn. (2.17) and $\frac{\gamma\rho}{\mu_T}\tau_{ij}^R\frac{\partial u_i}{\partial x_j}$ in Eqn. (2.18) lead to the eddy viscosity production. The terms $\beta^*\rho\omega k$ in Eqn. (2.17) and $\beta\rho\omega^2$ in Eqn. (2.18) represent the dissipation of the same variable. The transport equation of ω contains also the term $2\rho(1-F_1)\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial\omega}{\partial x_j}$ that describes the conservative diffusion. These terms are so called turbulent source terms. The turbulent source terms can be rewritten based on their representation. Production terms of k and ω can be written as:

$$P_k=\tau_{ij}^R\frac{\partial u_i}{\partial x_j},\ P_\omega=\frac{\gamma\rho}{\mu_T}\tau_{ij}^R\frac{\partial u_i}{\partial x_j}, \tag{2.19}$$

or in the alternative form:

$$P_k=\mu_T P_d-\frac{2}{3}\frac{\partial u_i}{\partial x_j}\rho k,\ P_\omega=\gamma\rho P_d-\frac{\gamma\rho}{\mu_T}\frac{2}{3}\frac{\partial u_i}{\partial x_j}\rho k,\ P_d=2(S_{ij}-\frac{1}{3}\frac{\partial u_i}{\partial x_j})\frac{\partial u_i}{\partial x_j}. \tag{2.20}$$

Destruction terms of k and ω read:

$$D_k=\beta^*\rho\omega k,\ D_\omega=\beta\rho\omega^2. \tag{2.21}$$

The cross–diffusion term CD_ω of the transport equation of ω reads:

$$CD_\omega=2\rho(1-F_1)\sigma_{\omega_2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial\omega}{\partial x_j}. \tag{2.22}$$

It can be shown that the P_d term is always positive. Thus the k transport equation has a term which is always positive (production term P_k proportional to P_d), a term which is always negative (destruction term D_k) and a term which is either positive or negative depending on the sign of $\frac{\partial u_i}{\partial x_j}$. Similarly to the case of the k transport equation, also the ω transport equation has a term which is always positive (production term P_ω proportional to P_d), a term which is always negative (destruction term D_ω), and a term which is either positive or negative depending on the sign of $\frac{\partial u_i}{\partial x_j}$. The ω transport equation, however, features the additional cross–diffusion term CD_ω which can be positive or negative.

The terms $\frac{\partial}{\partial x_j}\left[(\mu+\sigma_k\mu_T)\frac{\partial k}{\partial x_j}\right]$ in Eqn. (2.17) and $\frac{\partial}{\partial x_j}\left[(\mu+\sigma_\omega\mu_T)\frac{\partial\omega}{\partial x_j}\right]$ in Eqn. (2.18) leads to the conservative diffusion.

The turbulent eddy viscosity μ_T at any position in the computational domain and each simulation time of the time–dependent simulations is determined using the expression:

$$\mu_T=\frac{a_1\rho k}{max(a_1\omega,|\underline{\Omega}|F_2)}, \tag{2.23}$$

where $|\underline{\Omega}|$ is the flow vorticity. F_1 and F_2 are blending functions of the $k-\epsilon$ and $k-\omega$ turbulence models and are respectively given by:

$$F_1 = tanh(arg_1^4), \tag{2.24}$$

$$F_2 = tanh(arg_2^2), \tag{2.25}$$

where $arg1$ and $arg2$ are defined as:

$$arg_1 = min[max(\frac{\sqrt{k}}{\beta^* \omega d}, \frac{500\mu}{\rho\omega d^2}), \frac{4\rho\sigma_{\omega 2} k}{CD_{k\omega} d^2}], \tag{2.26}$$

$$arg_2 = max(\frac{2\sqrt{k}}{\beta^* \omega d}, \frac{500\mu}{\rho\omega d^2}), \tag{2.27}$$

The term $CD_{k\omega}$ equals the cross–diffusion term CD_ω when this term is positive, and is set to a small positive constant when CD_ω is negative. The expression of $CD_{k\omega}$ is:

$$CD_{k\omega} = max(2\rho\sigma_{\omega_2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j}, 10^{-20}). \tag{2.28}$$

The terms $arg1$ and $arg2$ are both functions of distance from the closest wall d. The SST model definition of the turbulent eddy viscosity assures that in part of the boundary layer where production of k is bigger than its dissipation ω, Bradshaw's assumption, which states that the shear stress is proportional to turbulent kinetic energy, is satisfied. The coefficients defining the $k-\omega$ SST turbulence model (β,C_ω,σ_k,σ_ω) are obtained by blending the coefficients of $k-\omega$ model (ϕ_1), and the coefficients of $k-\epsilon$ model (ϕ_2). The blended function ϕ reads:

$$\phi = F_1\phi_1 + (1-F_1)\phi_2. \tag{2.29}$$

The coefficients of $k-\omega$ model are:

$$\begin{aligned} &\beta_1 = 0.075,\ \sigma_{k1} = 0.85,\ \sigma_{omega1} = 0.5, \\ &C_{\omega 1} = \frac{\beta_1}{\beta^*} - \frac{\sigma_{\omega 1}\kappa^2}{\sqrt{\beta^*}} = 0.533. \end{aligned} \tag{2.30}$$

The coefficients of $k-\epsilon$ model are instead:

$$\begin{aligned} &\beta_2 = 0.0828,\ \sigma_{k2} = 1.0,\ \sigma_{omega2} = 0.856, \\ &C_{\omega 1} = \frac{\beta_2}{\beta^*} - \frac{\sigma_{\omega 2}\kappa^2}{\sqrt{\beta^*}} = 0.440. \end{aligned} \tag{2.31}$$

Other coefficients used by the SST model are the following:

$$\beta^* = 0.09,\ \kappa = 0.41,\ \alpha_1 = 0.31. \tag{2.32}$$

2.3 Time–domain RANS and SST

In previous sections, all approximations used to simplify NS equations have been introduced, and Menter's $k-\omega$ SST turbulence closure has been described. Therefore, we can now write the RANS and SST turbulence model equations as a closed set of equations for compressible flows. By doing so, we obtain a system of N_{PDE} nonlinear partial differential equations (PDEs). For 3D turbulent flows $N_{PDE} = 7$, five RANS PDEs, and two turbulent PDEs.

Time–dependent problems can be solved due to the inclusion of time–derivatives in the RANS and $k - \omega$ SST equations. Such time–derivatives refer to time–variations taking place on the characteristic time–scales associated with the engineering problem at end. One of the advantages of differential turbulence models over simpler algebraic models, is that the former methods allow inertial (time–dependent) and global (through the spatial derivatives of the turbulence model) effects to be taken into account when modelling the effects of turbulence. The divergence form of the 3D unsteady Reynolds–Favre averaged Navier Stokes equations (URANS), coupled to the two transport equations of the $k - \omega$ SST turbulence model in dimensional form are:
mass conservation:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_i}(\bar{\rho}\tilde{u}_i) = 0, \tag{2.33}$$

momentum conservation:

$$\frac{\partial}{\partial t}(\bar{\rho}\tilde{u}_i) + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}_j\tilde{u}_i) = -\frac{\partial p}{\partial x_i} + \frac{\partial \tilde{\tau}_{ji}}{\partial x_j}, \tag{2.34}$$

energy conservation:

$$\frac{\partial}{\partial t}(\bar{\rho}\tilde{E}) + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{H}\tilde{u}_j) = \frac{\partial}{\partial x_j}\left[\tilde{u}_i\tilde{\tau}_{ij} + (\mu + \sigma_k\mu_T)\frac{\partial \tilde{k}}{\partial x_j} - \tilde{q}_j\right], \tag{2.35}$$

turbulent kinetic energy:

$$\frac{\partial}{\partial t}(\bar{\rho}\tilde{k}) + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}_j\tilde{k}) = \tilde{\tau}^R_{ij}\frac{\partial \tilde{u}_i}{\partial x_j} - \beta^*\bar{\rho}\tilde{\omega}\tilde{k} + \frac{\partial}{\partial x_j}\left[(\mu + \sigma_k\mu_T)\frac{\partial \tilde{k}}{\partial x_j}\right], \tag{2.36}$$

specific dissipation rate:

$$\begin{aligned}\frac{\partial}{\partial t}(\bar{\rho}\tilde{\omega}) + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}_j\tilde{\omega}) =& \frac{\gamma\bar{\rho}}{\mu_T}\tilde{\tau}^R_{ij}\frac{\partial \tilde{u}_i}{\partial x_j} - \beta\bar{\rho}\tilde{\omega}^2 + \frac{\partial}{\partial x_j}\left[(\mu + \sigma_\omega\mu_T)\frac{\partial \tilde{\omega}}{\partial x_j}\right] \\ &+ 2\bar{\rho}(1 - F_1)\sigma_{\omega_2}\frac{1}{\tilde{\omega}}\frac{\partial \tilde{k}}{\partial x_j}\frac{\partial \tilde{\omega}}{\partial x_j}.\end{aligned} \tag{2.37}$$

The system of equations (2.33), (2.34), (2.35) is formally identical to the system of equations (2.1), (2.2), (2.7), however, there are additional features that needs to be introduced.

The variables appearing in the URANS equations denoted by the symbol ¯ represent Reynolds time–averaged variables over the turbulence time–scales and the variables denoted by the symbol ˜ represent Favre averaged variables. The two symbols will be omitted in any further representation of URANS or RANS equations from here on. There is an additional diffusive term in the energy equation (2.35) depending on the eddy viscosity μ_T and the components of the gradient of the turbulent kinetic energy k. As previously discussed when commenting Eqn. (2.16), the stress tensor τ_{ij} is the sum of the molecular stress tensor τ^M_{ij} defined by Eqn. (2.4) and the Reynolds stress tensor τ^R_{ij} defined by Eqn. (2.14). The thermal heat flux vector q_j results from the sum of a laminar and turbulent contribution, where molecular viscosity, constant pressure, specific heat and thermal diffusivity are linked by the Prandtl number Pr :

$$Pr = \frac{\mu c_p}{k_T}. \tag{2.38}$$

The thermal heat flux vector can be written as:

$$q_j = -\left[\frac{\mu}{Pr} + \frac{\mu_T}{Pr_T}\right]\frac{\partial h}{\partial x_j}, \tag{2.39}$$

where Pr_T , is the turbulent Prandtl number, which relates the turbulent eddy viscosity and the turbulent counterpart of the molecular thermal diffusivity.

It should also be noted that coupling of the URANS and SST equations is due to two factors. Firstly, the turbulent kinetic energy k is included in the definition of the total energy, defined by Eqn. (2.8), as well as the total enthalpy, defined by Eqn. (2.13). When using the $k - \omega$ SST model, the definition of these two variables are re–written respectively as:

$$E = e + \frac{(u_i u_i)}{2} + k,\ H = h + \frac{(u_i u_i)}{2} + k. \tag{2.40}$$

Based on the perfect gas relations, the definition of the total energy E implies that the static pressure p is defined as:

$$p = (\gamma - 1)\left[\rho E - \frac{1}{2}\rho(u_i u_i) - \rho k\right], \tag{2.41}$$

where γ defines the specific heat ratio, which is the ratio between c_p and c_v (at standard conditions $\gamma = 1.4$ for air):

$$\gamma = \frac{c_p}{c_v}. \tag{2.42}$$

Secondly, coupling of the URANS and $k - \omega$ SST equations is also caused by the fact that the eddy viscosity μ_T defined by Eqn. (2.23) depends on k and ω, and thus on the SST equations, and it also appears in the momentum and energy equations of the RANS system.

2.4 Arbitrary Lagrangian–Eulerian formulation of URANS and SST equations in inertial frame

Generally, in fluid dynamics there are two basic approaches to describe the motion of fluid and derive the governing equations. The first is called Lagrangian approach, where the computational mesh is embedded in the fluid and moves with it. The second approach is called Eulerian, which treats the computational mesh as a fixed frame trough which the fluid moves. In time–dependent problems characterised by the presence of moving bodies and deforming grids (e.g. rotating and/or deforming wind turbine blades, oscillating wings, vibrating turbomachinery blades), it is necessary to solve the governing equations using moving grids and in some cases also deforming grids. For such problems, it is convenient to use the Arbitrary Lagrangian–Eulerian technique (ALE), that is both Lagrangian and Eulerian. The method is very flexible, as the mesh may move with the fluid, be held fixed or be moved in any other prescribed way with respect to the fluid [76, 77].

Given a moving control volume C with time–dependent boundary $S(t)$, the ALE integral form of the system of the URANS and SST equations in an inertial frame of reference can be written as:

$$\frac{\partial}{\partial t}\left(\int_{C(t)} \mathbf{U}\, dC\right) + \oint_{S(t)} (\underline{\mathbf{\Phi}}_c - \underline{\mathbf{\Phi}}_d) \cdot d\underline{S} - \int_{C(t)} \mathbf{S}\, dC = 0, \tag{2.43}$$

where

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho\underline{\mathrm{u}}' \\ \rho E \\ \rho k \\ \rho\omega \end{bmatrix}, \tag{2.44}$$

represents the array of conservative variables and the symbol $'$ denotes the transpose operator. The generalised convective flux vector $\underline{\mathbf{\Phi}}_c$ is expressed as:

$$\underline{\mathbf{\Phi}}_c = \begin{bmatrix} \rho(\underline{\mathrm{u}} - \underline{\mathrm{u}}_b) \\ \rho(\underline{\mathrm{u}} - \underline{\mathrm{u}}_b)\underline{\mathrm{u}}' + pI_{PD} \\ \rho E(\underline{\mathrm{u}} - \underline{\mathrm{u}}_b) + p\underline{\mathrm{u}} \\ \rho k(\underline{\mathrm{u}} - \underline{\mathrm{u}}_b) \\ \rho\omega(\underline{\mathrm{u}} - \underline{\mathrm{u}}_b) \end{bmatrix}, \tag{2.45}$$

where $\underline{\mathrm{u}}_b$ is the velocity of the boundary S. I_{PD} is the identity matrix of dimension PD, for 3D case $PD = 3$. The generalised diffusive flux vector $\underline{\mathbf{\Phi}}_d$ reads:

$$\underline{\mathbf{\Phi}}_d = \begin{bmatrix} 0 \\ \underline{\underline{\tau}} \\ \underline{\underline{\tau}} \cdot \underline{\mathrm{u}} + (\mu + \sigma_k\mu_T)\nabla k - \underline{\mathrm{q}} \\ (\mu + \sigma_k\mu_T)\nabla k \\ (\mu + \sigma_\omega\mu_T)\nabla\omega \end{bmatrix}. \tag{2.46}$$

The definition of the source term $\mathbf{S}$ in Eqn. (2.43) is given by:

$$\mathbf{S} = \begin{bmatrix} 0 \\ \underline{0}' \\ 0 \\ S_k \\ S_\omega \end{bmatrix}. \tag{2.47}$$

Tke symbols S_k and S_ω in Eqn. (2.47) represent source terms of the k and ω equations respectively, and can be written as:

$$S_k = \mu_T P_d - \frac{2}{3}(\nabla \cdot \underline{\mathrm{u}})\rho k - D_k, \tag{2.48}$$

$$S_\omega = \gamma\rho P_d - \frac{2}{3}(\nabla \cdot \underline{\mathrm{u}})\frac{\gamma\rho k}{\nu_T} - D_\omega + CD_\omega. \tag{2.49}$$

The term P_d, the k destruction term D_k, the ω destruction term D_ω, and the cross–diffusion term CD_ω have already been defined by equations (2.20), (2.21) and (2.22), and are rewritten here in divergence form:

$$P_d = 2\left[\underline{\mathrm{u}} - \tfrac{1}{3}\nabla \cdot \underline{\mathrm{u}}\right]\nabla\underline{\mathrm{u}}, \tag{2.50}$$

$$D_k = \beta^*\rho k\omega \quad D_\omega = \beta\rho\omega^2, \tag{2.51}$$

$$CD_\omega = 2(1 - F_1)\rho\sigma_{\omega 2}\tfrac{1}{\omega}\nabla k \cdot \nabla\omega. \tag{2.52}$$

The steady state or RANS equations are obtained by setting to zero the time derivative appearing in Eqn. (2.43), i.e. $\frac{\partial\mathbf{U}}{\partial t} = 0$, and by considering a fixed control volume. Velocity of boundary appearing in the generalised convective flux vector $\underline{\mathbf{\Phi}}_c$ (2.45) is set to zero, $\underline{\mathrm{u}}_b = 0$. The definition of generalised diffusive flux vector $\underline{\mathbf{\Phi}}_d$ (2.46) and the source term $\mathbf{S}$ (2.47) remains the same. By doing so, one obtains the steady equations for a fixed control volume in an inertial frame of reference, which can be symbolically written as:

$$\oint_S (\underline{\mathbf{\Phi}}_c - \underline{\mathbf{\Phi}}_d) \cdot d\underline{S} - \int_C \mathbf{S} dC = 0, \tag{2.53}$$

Steady equations in stationary frame are most commonly used for numerical simulation of flows past fixed–objects, such as wings, aircraft, vehicles.

2.5 Arbitrary Lagrangian–Eulerian formulation of URANS and SST equations in non–inertial frame

The governing equations of fluid flow, can also be written in a non–inertial frame of reference. In many applications involving rotating motion, such as turbomachinery problems, propellers, and flows around horizontal–axis wind turbine blades, it is advantageous to describe the governing equations in a rotating (non–inertial) frame of reference. Expressing the governing equations in a rotating frame often enables to solve a problem that is unsteady in the stationary (inertial) frame as a steady–state problem in the rotating (non–inertial) frame. When using this approach, the grid motion viewed in rotating frame is frozen and the body is at rest. Many flow problems in engineering practise can be viewed as steady in rotating frame, therefore, such approach offers much lower computational cost in comparison to an unsteady simulation in stationary frame of reference. It should also be noted that for certain unsteady flow problems an unsteady simulation in a rotating frame can also be performed, solving these flow problems in inertial or non–inertial frame does not change computational cost excessively. When the governing equations are written in a rotating frame, there are two choices of how to express the relative velocity vector components. They can be expressed either in the relative or absolute frame. The relative velocity components can be expressed in absolute frame by considering a velocity transformation. At a particular time, when the rotating frame matches the fixed frame, one can choose the same Cartesian coordinates to specify the unit vectors of both frames. In this case the velocity vector is exactly the same as one would see with respect to the fixed frame. This is the reason why the relative velocity components can be thought as the absolute velocity components, even though they are viewed from the rotating frame. For internal turbomachinery problems the relative velocity components are usually expressed in the relative frame [78, 79], whereas for external flow problems, such as propellers and flows around horizontal–axis wind turbine blades, they are expressed in the absolute frame of reference [80, 81, 82, 83]. The two formulations for the velocity vector components should lead to the same flow solution. However, many studies report that for the open rotor aerodynamics applications, the use of the absolute velocity components in the relative frame allows more accurate calculation of the fluxes. The relative velocity components can become extremely large in the farfield, due to large entrainment velocities due to grid motion, and this may result in large numerical errors (e.g. large round–off of flux differences).

The formulation of the governing equations in a stationary Cartesian coordinate system, which was introduced in section (2.4) will now be extended to include rotating frame effects with the component of the relative velocity vector being expressed in an absolute frame. Such rotating frame formulation was chosen, as it allows to impose steady farfield boundary conditions for steady rotor flows, ensures more precise calculation of fluxes, and requires only a few modifications to the existing expression of governing equations.

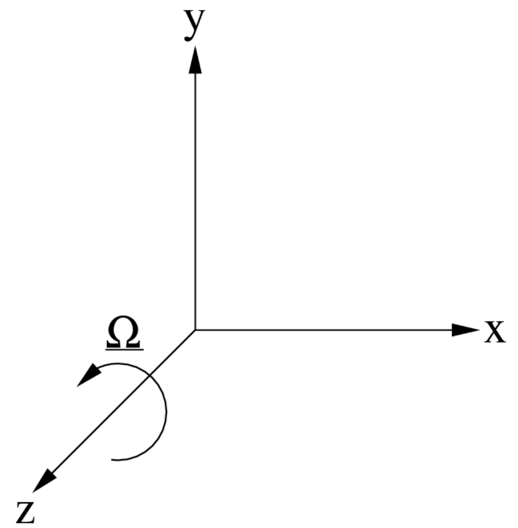


Figure 2.2: Cartesian rotating coordinate system.

The ALE URANS and SST equations are written in a Cartesian rotating coordinate system, where the x and y axes rotate around the z axis with constant angular velocity $\underline{\Omega}$. The rotating Cartesian frame is depicted in Fig. 2.2. Since the z axis was selected to be the rotational axis, $\underline{\Omega}$ has the following components:

$$\underline{\Omega} = \begin{bmatrix} 0 \\ 0 \\ \Omega_z \end{bmatrix}. \tag{2.54}$$

Given a control volume C_r with boundary S_r rotating in the considered system, the ALE integral form of the system of URANS and SST equations in rotating frame can be written as:

$$\frac{\partial}{\partial t}\left(\int_{C_r} \mathbf{U}_r dC_r\right) + \oint_{S_r} (\underline{\mathbf{\Phi}}_{cr} - \underline{\mathbf{\Phi}}_{dr}) \cdot d\underline{S}_r - \int_{C_r} \mathbf{S}_r dC_r = 0. \tag{2.55}$$

Subscript r denotes rotating frame, $\mathbf{U}_r$ is the array of conservative variables:

$$\mathbf{U}_r = \begin{bmatrix} \rho \\ \rho\underline{u}'_r \\ \rho E_r \\ \rho k_r \\ \rho\omega_r \end{bmatrix}, \tag{2.56}$$

where $\underline{u}_r$ represent the absolute velocity vector in the rotating frame, and is thus:

$$\underline{u}_r = \begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} = \begin{bmatrix} cos\Omega_z t & -sin\Omega_z t & 0 \\ sin\Omega_z t & cos\Omega_z t & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}. \tag{2.57}$$

The generalised convective flux vector $\underline{\mathbf{\Phi}}_{cr}$ is expressed as:

$$\underline{\mathbf{\Phi}}_{cr} = \begin{bmatrix} \rho(\underline{u}_r - \underline{u}_b) \\ \rho(\underline{u}'_r - \underline{u}_b)\underline{u}'_r + pI_{PD} \\ \rho E(\underline{u}_r - \underline{u}_b) + p\underline{u}_r \\ \rho k(\underline{u}_r - \underline{u}_b) \\ \rho\omega(\underline{u}_r - \underline{u}_b) \end{bmatrix}, \tag{2.58}$$

where $\underline{\mathrm{u}}_b$ represents the velocity of the cell faces, which is due to the rotation of the coordinate system, and can be expressed as:

$$\underline{\mathrm{u}}_b = \underline{\Omega} \times \underline{\mathrm{r}}_a. \tag{2.59}$$

$\underline{\mathrm{r}}_a$ denotes coordinate vector and is given by:

$$\underline{\mathrm{r}}_a = \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix}. \tag{2.60}$$

The generalised diffusive flux vector is expressed:

$$\underline{\mathbf{\Phi}}_{dr} = \begin{bmatrix} 0 \\ \underline{\underline{\tau}} \\ \underline{\underline{\tau}} \cdot \underline{\mathrm{u}}_r + (\mu + \sigma_k \mu_T)\nabla k - \underline{\mathrm{q}} \\ (\mu + \sigma_k \mu_T)\nabla k \\ (\mu + \sigma_\omega \mu_T)\nabla \omega \end{bmatrix}. \tag{2.61}$$

The definition of the source term $\mathbf{S}_r$ is the following:

$$\mathbf{S}_r = \begin{bmatrix} 0 \\ -\rho \Omega_z v_r \\ \rho \Omega_z u_r \\ 0 \\ 0 \\ S_{kr} \\ S_{\omega r} \end{bmatrix}. \tag{2.62}$$

The definitions of S_{kr} and $S_{\omega r}$ are the same as by equations 2.48 and 2.49 in inertial frame, except for the fact that the absolute velocity vector in the stationary frame $\underline{\mathrm{u}}$ is now replaced by absolute velocity vector in the rotating frame $\underline{\mathrm{u}}_r$, thus $\underline{\mathrm{u}} = \underline{\mathrm{u}}_r$.

Setting to zero the time derivative appearing in Eqn. (2.55), $\frac{\partial \mathbf{U}}{\partial t} = 0$, the RANS and SST equations in rotating frame are obtained:

$$\oint_{S_r} (\underline{\mathbf{\Phi}}_{cr} - \underline{\mathbf{\Phi}}_{dr}) \cdot d\underline{S}_r - \int_{C_r} \mathbf{S}_r dC_r = 0. \tag{2.63}$$

As previously mentioned, the steady equations in the relative frame are used to solve rotor problems that are unsteady in the stationary frame, and can instead be viewed as steady–state problems in the rotating frame. These applications include all internal turbomachinery problems, propellers, fans, helicopter rotors, HAWT rotors, etc.

The source terms that appear in Eqns. (2.55) and (2.63) due to relative frame, make the URANS and RANS equations non conservative. For transonic problems and coarse grids, this may introduce some numerical errors [84]. In order to avoid such errors, the velocity of cell faces $\underline{\mathrm{u}}_b$ are evaluated as in [85], which assures the conservation of URANS and RANS equations. This will be explained in more detail in the next chapter.

2.6 Harmonic balance formulation of Navier–Stokes Equations

Harmonic balance is a frequency–domain analysis technique which can be used for modelling unsteady nonlinear periodic flows. Many flows, such as turbomachinery and wind turbine flows, feature temporal periodicity. Therefore, unsteady flow equations may be written as a Fourier series in time with spatially varying coefficients. Due to the inclusion of a pseudo–time term, they can be solved as a system of steady problem using conventional time–marching techniques. The only addition to the update step of the HB method is the inclusion of a volumometric source term, which couples all snapshots of sought periodic solution. This modern computational approach offers at least an order of magnitude reduction in computational cost and represents computationally efficient alternative method to time marching methods. When using the HB method, periodic steady state can be obtained more quickly as the resolution of long initial transients that needs to be resolved by URANS can be avoided. For rotor flows the solution of HB equations may be obtained even more rapidly, making use of multifrequency periodicity boundary conditions [46, 52, 86].

The HB formulation of the RANS equations assumes that the fundamental frequency ω of the sought periodic flow field is known. Using the truncated Fourier series, a series of sine and cosine functions, one can approximate the ALE integral form of the system of the URANS and SST equations defined by Eqn. (2.43). Generally, two types of HB formulation exist in the literature, classical and so called high–dimensional HB (HDHB) formulation [46, 52, 86]. The detailed comparison of both methods is described in [86]. As pointed out by the authors of [46] and [86], it is very difficult or nearly impossible to use the classical harmonic balance method for high dimensional nonlinear dynamical systems such as RANS equations, as the expression of the nonlinear terms can become massively complex. To overcome such problem the HDHB method can be used. Main aspect of the HDHB method is that rather than working in terms of Fourier coefficient variables as in the classical HB approach, the dependent variables are instead cast in the time–domain and stored at $2NH + 1$ equally spaced sub–time levels over the period of one cycle of motion, so called pseudo–time approach. For this work, HDHB formulation of the RANS and SST equations was used.

It should be mentioned that also other frequency–domain analysis techniques exist. One alternative to HDHB is called non–linear frequency–domain (NLFD) method, introduced in [87]. The main difference between HDHB and NLFD arises in the representation of the non–linear residuals. HDHB approach utilises a pseudo–time approach to represent the non–linear residual in the time–domain, whereas NLFD represents a form of the residual in the frequency–domain. Due to such reason, the HDHB method is easier to implement in an existing TD CFD codes. The detailed comparison of HDHB and NLFD methods is reported in [87]. Another alternative which is following the same direction as HDHB is called the time spectral method [47]. Similarly to HDHB, the Fourier transformation is used for temporal discretization, leading to spectral accuracy. After the transformation back to the physical domain, the time derivative appears as a high–order central difference formula coupling all the time snapshots in the period together. Authors of [47] also pointed out that such method can be implemented relatively easily in an existing CFD code, as the equations are solved in time–domain, whereas one must develop brand new code if the equations are solved in frequency–domain. Additionally, the transformation back and forth from time–domain to the frequency–domain is avoided, which also benefits the computational cost. The HBHD, NLFD and time spectral methods have all successfully demonstrated the ability to represent challenging non–linear flow solutions using a minimum number of complex harmonics. An alternative, which exhibit resemblance to the HDHB approach, is also the Non–Linear Harmonic (NLH) method [59]. However, in NLH method the zeroth harmonic (mean flow) and the first harmonic (unsteady flow component) are decoupled, which represents an important difference.

Denoting $\mathbf{u}$ as the HB solution volume integral and $\mathbf{h}$ as the HB solution surface integral, both variables can be expressed in terms of truncated Fourier series:

$$\mathbf{u}(t) \approx \hat{\mathbf{u}}_0 + \sum_{n=1}^{N_H} \left(\hat{\mathbf{u}}_{2n-1} cos(n\omega t) + \hat{\mathbf{u}}_{2n} sin(n\omega t)\right), \tag{2.64}$$

$$\mathbf{h}(t) \approx \hat{\mathbf{h}}_0 + \sum_{n=1}^{N_H} \left(\hat{\mathbf{h}}_{2n-1} cos(n\omega t) + \hat{\mathbf{h}}_{2n} sin(n\omega t)\right). \tag{2.65}$$

N_H represents the retained number of harmonics and is a user–given parameter. The terms $\hat{\mathbf{u}}$ and $\hat{\mathbf{h}}$, where $(n = 0, 1, \dots, N_H)$ are HB solution Fourier coefficient variables. More precisely, the variables $\hat{\mathbf{u}}_0$ and $\hat{\mathbf{h}}_0$ represent the mean value of the sought periodic solution, whereas the terms $\hat{\mathbf{u}}_{2n-1}$ and $\hat{\mathbf{h}}_{2n-1}$ represent the real part of the N_H complex harmonics, and the terms $\hat{\mathbf{u}}_{2n}$ and $\hat{\mathbf{h}}_{2n}$ their imaginary counterparts. The arrays $\hat{\mathbf{u}}_n$ and $\hat{\mathbf{h}}_n$ have length $N_{PDE} = 7$ for 3D case, where N_{PDE} is the number of considered conservation laws.

For deriving the HB approximation to Eqn. (2.43) it is also necessary to write the time–derivative of $\mathbf{u}$ as a truncated Fourier series:

$$\frac{\partial \mathbf{u}(t)}{\partial t} \approx \sum_{n=1}^{N_H} n\omega \left(-\hat{\mathbf{u}}_{2n-1} sin(n\omega t) + \hat{\mathbf{u}}_{2n} cos(n\omega t).\right) \tag{2.66}$$

Inserting the expressions of Eqns. (2.64), (2.65) and (2.66) into the the ALE integral form of the system of the URANS and SST equations given by Eqn. (2.43), and matching harmonics of the same order, results in a system of $[N_{PDE} \times (2N_H + 1)]$ PDE's, which can be expressed in matrix–vector form as:

$$\omega A \hat{\mathbf{u}} + \hat{\mathbf{h}} = 0. \tag{2.67}$$

The definitions of $\hat{\mathbf{u}}$ and $\hat{\mathbf{h}}$ are respectively:

$$\hat{\mathbf{u}} = \begin{bmatrix} \hat{\mathbf{u}}'_0 \\ \hat{\mathbf{u}}'_1 \\ \cdot \\ \cdot \\ \cdot \\ \hat{\mathbf{u}}'_{2N_H} \end{bmatrix}, \quad \hat{\mathbf{h}} = \begin{bmatrix} \hat{\mathbf{h}}'_0 \\ \hat{\mathbf{h}}'_1 \\ \cdot \\ \cdot \\ \cdot \\ \hat{\mathbf{h}}'_{2N_H}. \end{bmatrix} \tag{2.68}$$

and matrix A can be expressed as:

$$A = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & J_1 & 0 & \dots & 0 \\ 0 & 0 & J_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & J_{N_H} \end{bmatrix}, \tag{2.69}$$

where

$$J_n = I_{N_{PDE}} \otimes n \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \qquad n = 1, 2, \dots, N_H. \tag{2.70}$$

The symbol $\otimes$ denotes the Kronecker tensor product, $I_{N_{PDE}}$ is the identity matrix of the size $(N_{PDE})^2$, and therefore the size of J_n blocks is $(2N_{PDE})^2$. Matrix A has a block diagonal structure, meaning that only diagonal blocks contain non–zero entries. Each J_n block of matrix A has a block–diagonal structure as well. It is also important to stress that the first block of matrix A is 0. Due to described structure, one finds that when writing the equations of the system defined by Eqn. (2.67) explicitly, the unknown harmonic components $\hat{\mathbf{u}}$ are coupled by the harmonic residuals $\hat{\mathbf{h}}$, whereas no coupling occurs through the first term of the equation. As previously mentioned, the analytical derivation of such equations can become extremely complex when dealing with the RANS equations and turbulence models, therefore, reconstructing the $2N_H+1$ harmonic balance Fourier coefficient solution variables $\hat{\mathbf{u}}$ and $\hat{\mathbf{h}}$ to become $2N_H+1$ equally spaced points over one period, enormously simplifies the derivation of the HB equations. Equally spaced points over one period can be defined as:

$$t_n = \frac{n}{2N_H+1}\frac{2\pi}{\omega}, n = 0, 1, \ldots, 2N_H. \tag{2.71}$$

They are related by the Kronecker product of a constant Fourier transformation matrix E_H and identity matrix $I_{N_{PDE}}$:

$$F_H = E_H \otimes I_{N_{PDE}}. \tag{2.72}$$

The expression of the Fourier matrix E_H is given by:

$$E_H = \frac{2}{2N_H+1}\begin{bmatrix} 1/2 & 1/2 & \ldots & 1/2 \\ cos(\omega t_0) & cos(\omega t_1) & \ldots & cos(\omega t_{2N_H}) \\ sin(\omega t_0) & sin(\omega t_1) & \ldots & sin(\omega t_{2N_H}) \\ cos(2\omega t_0) & cos(2\omega t_1) & \ldots & cos(2\omega t_{2N_H}) \\ sin(2\omega t_0) & sin(2\omega t_1) & \ldots & sin(2\omega t_{2N_H}) \\ \vdots & \vdots & \ddots & \vdots \\ cos(N_H\omega t_0) & cos(N_H\omega t_1) & \ldots & cos(N_H\omega t_{2N_H}) \\ sin(N_H\omega t_0) & sin(N_H\omega t_1) & \ldots & sin(N_H\omega t_{2N_H}) \end{bmatrix}. \tag{2.73}$$

The time–domain solutions at the $2NH+1$ equally spaced sub–time levels expressed in terms of the HB Fourier coefficients $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{h}}$, are given respectively:

$$\tilde{\mathbf{u}} = F_H^{-1}\hat{\mathbf{u}}, \tag{2.74}$$

$$\tilde{\mathbf{h}} = F_H^{-1}\hat{\mathbf{h}}. \tag{2.75}$$

The inverse of the matrix F_H is a Kronecker product of an inverse of a constant Fourier transformation matrix E_H and identity matrix $I_{N_{PDE}}$:

$$F_H^{-1} = E_H^{-1} \otimes I_{N_{PDE}}. \tag{2.76}$$

The inverse of the Fourier matrix E_H is given by:

$$E_H^{-1} = \begin{bmatrix} 1 & cos(\omega t_0) & sin(\omega t_0) & \ldots & cos(N_H\omega t_0) & sin(N_H\omega t_0) \\ 1 & cos(\omega t_1) & sin(\omega t_1) & \ldots & cos(N_H\omega t_1) & sin(N_H\omega t_1) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & cos(\omega t_{2N_H}) & sin(\omega t_{2N_H}) & \ldots & cos(N_H\omega t_{2N_H}) & sin(N_H\omega t_{2N_H}) \end{bmatrix}, \tag{2.77}$$

Inserting the relationships from Eqns. (2.74) and (2.75) into Eqn. (2.67), yields the system:

$$\omega D\tilde{\mathbf{u}} + \tilde{\mathbf{h}} = 0, \tag{2.78}$$

where D is the $(N_{PDE} \times N_{PDE})$ matrix, given by:

$$D = F_H^{-1}AF_H. \tag{2.79}$$

The expression of matrix D may be written as follows:

$$D = \frac{2}{2N_H + 1} \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ a_{21} & 0 & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & 0 \end{bmatrix} \otimes I_{N_{PDE}}, \tag{2.80}$$

where the entries of the matrix are defined as:

$$a_{mn} = \sum_{k=1}^{N_H} k \, sin \left(\frac{2\pi k(n-m)}{2N_H + 1} \right).$$

Once matrix D is constructed, it structure becomes block antisymmetric, thus $a_{mn} = -a_{nm}$. When writing the equations of the system defined by Eqn. (2.78) explicitly, the unknown TD harmonic components $\tilde{\mathbf{u}}$ are all coupled by the TD harmonic residuals $\tilde{\mathbf{h}}$. The problem of the decoupled first term no longer exist.

Inserting the integral definitions of $\mathbf{u}$ and $\mathbf{h}$ into Eqn. (2.78) leads to the ALE integral form of HDHB RANS and SST equations in inertial frame of reference:

$$\omega D \left(\int_{C_H} \mathbf{U}_H \, dC_H \right) + \oint_{S_H} (\underline{\boldsymbol{\Phi}}_{cH} - \underline{\boldsymbol{\Phi}}_{dH}) \cdot d\underline{S}_H - \int_{C_H} \mathbf{S}_H \, dC_H = 0, \tag{2.81}$$

where the unknown array of conservative variables $\mathbf{U}_H$ is made up of $2N_H + 1$ flow field snapshots, referring to the equally spaced points of one period:

$$\mathbf{U}_H = \begin{bmatrix} \mathbf{U}(t_0)' \\ \mathbf{U}(t_1)' \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{U}(t_{N_H})' \end{bmatrix}. \tag{2.82}$$

The subarray structure of the following variables in HB representation, all denoted by subscript H, $\underline{\boldsymbol{\Phi}}_{cH}$, $\underline{\boldsymbol{\Phi}}_{dH}$, $\mathbf{S}_H$, dC_H and $d\underline{S}_H$ is similar to that of $\mathbf{U}_H$.

Eqn. (2.81) can also be rewritten for relative frame, hence the ALE integral form of HDHB RANS and SST equations in non–inertial frame of reference reads:

$$\omega D \left(\int_{C_{Hr}} \mathbf{U}_{Hr} \, dC_{Hr} \right) + \oint_{S_{Hr}} (\underline{\boldsymbol{\Phi}}_{cHr} - \underline{\boldsymbol{\Phi}}_{dHr}) \cdot d\underline{S}_{Hr} - \int_{C_{H}r} \mathbf{S}_{Hr} dC_{Hr} = 0, \tag{2.83}$$

where the unknown array of conservative variables in relative frame $\mathbf{U}_{Hr}$ is made up of $2N_H + 1$ flow field snapshots, referring to the equally spaced points of one period:

$$\mathbf{U}_{Hr} = \begin{bmatrix} \mathbf{U}_r(t_0)' \\ \mathbf{U}_r(t_1)' \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{U}_r(t_{N_H})' \end{bmatrix}. \tag{2.84}$$

In a similar way as the subarray $\mathbf{U}_{Hr}$ is constructed, the following subarrays, $\underline{\boldsymbol{\Phi}}_{cHr}$, $\underline{\boldsymbol{\Phi}}_{dHr}$, $\mathbf{S}_{Hr}$, dC_{Hr} and $d\underline{S}_r$ given in relative frame, can be obtained.

The HDHB method solves the frequency–domain governing equations in the time–domain. The representation of the frequency–domain equations in the time–domain yields a set of coupled steady problems or snapshots. As previously pointed out, the number of PDEs increases from N_{PDE} in TD approach to $[N_{PDE} \times (2N_H + 1)]$. Despite this, the HB approach allows turbulent periodic flows to be computed at a significantly lower computational cost than with the TD approach in many problems of engineering interest. In essence, if all snapshots of the solution converge as quickly as a similar steady–state calculation, then the cost of the HDHB calculation is the product of steady–state calculation cost, the number of used harmonics, and the computational overhead due to the calculation of the volumometric source term D, which couples all the snapshots.

2.7 Boundary conditions

Any numerical simulation of the governing equations require specification of boundary conditions. A set of governing equations has an infinite number of solutions, therefore, appropriate boundary conditions must be imposed on the boundaries, as the solution of the governing equations is strongly dependent on such boundary conditions. It is crucially important, that boundary conditions are compatible with both numerical and physical properties of the problem to be solved. The results of turbulent flow simulations are strongly dependent on the models used to define the boundary flow state. The stability of numerical simulation also depends on the specification of initial boundary flow state [66, 88, 89].

2.7.1 Farfield

Farfield boundary conditions play an essential role in the case of external flow problems. They must be located sufficiently far from the body, so as to minimise the negative effects of spurious reflections from such boundaries. It is extremely important that outgoing waves are not reflected back into the flow from the outer boundaries, as in such case the convergence rate of the numerical simulation could also be impaired, and the reflections could spoil the solution accuracy. There are several ways of imposing boundary conditions on the external boundaries. In this research work, they are either based on the one–dimensional Riemann invariants [90, 91] or the multi–dimensional compatibility equations and differential form of the characteristic variables. The former formulation is well established and widely used in the CFD practice and was also used for the majority of the work done in this research. However, when using low–speed preconditioning to enhance accuracy and performance of compressible CFD solvers, characteristic–based far field boundary conditions need to be also preconditioned. Such preconditioning can not be accomplished using one–dimensional Riemann invariants. Since the CFD system developed within this research also features low–speed preconditioning, far field boundary conditions based on the multi–dimensional compatibility equations and differential form of the characteristic variables have also been implemented, as a part of different research project, and used for the simulations reported in the thesis. The formulation of these boundary conditions is described in [92].

Based on the characteristic theory of one–dimensional Euler equations, the idea is to enforce flow data associated with the incoming Riemann invariants and extrapolate flow data associated with the outgoing Riemann invariants from the interior of the domain. The velocity component normal to the outer boundary ($\underline{n}$ pointing outward from the flow domain), and the speed of sound are obtained from one–dimensional Riemann invariants. Assuming that the flow at the outer boundary is subsonic, the prescribed (R_∞) and extrapolated (R_{ext}) Riemann invariants are given respectively by:

$$R_\infty = \underline{\mathrm{u}}_\infty \cdot \underline{\mathrm{n}} + \frac{2c_\infty}{\gamma - 1}, \tag{2.85}$$

$$R_{ext} = \underline{\mathrm{u}}_{ext} \cdot \underline{\mathrm{n}} + \frac{2c_{ext}}{\gamma - 1}, \tag{2.86}$$

where $\underline{\mathrm{u}}_\infty$ and c_∞ represent the freestream velocity and the freestream speed of sound, respectively. The subscript *ext* refers to the extrapolated values from the interior of the computational domain with zero order extrapolation. The normal velocity $\underline{\mathrm{u}}_{bnd} \cdot \underline{\mathrm{n}}$ and the local speed of sound c_{bnd} on the boundary can now be obtained by adding and subtracting both Riemann variables:

$$\underline{\mathrm{u}}_{bnd} \cdot \underline{\mathrm{n}} = \frac{1}{2}(R_{ext} + R_\infty), \tag{2.87}$$

$$c_{bnd} = \frac{\gamma - 1}{4}(R_{ext} - R_\infty). \tag{2.88}$$

Corresponding equations hold for the inflow and outflow cases, and the sign of the computed value of normal velocity can be used to determine if an inflow or outflow condition occurs. A negative characteristic speed corresponds to an inflow boundary, and positive speed to an outflow boundary. At an inflow boundary the tangential velocity component is specified from the freestream flow conditions, whereas at an outflow boundary this variable is extrapolated from the interior. To complete the set of equations required to completely define the flow state on far field boundaries, also the entropy s is introduced:

$$s = \frac{p}{\rho^\gamma}. \tag{2.89}$$

At inflow boundaries, the entropy on the boundary is set to its free stream value $s_{bnd} = s_\infty$, whereas at outflow boundaries it is extrapolated from inside the domain: $s_{bnd} = s_{ext}$. The density, energy and pressure at the boundary can then be calculated from c_{bnd} and s_{bnd}:

$$\rho_{bnd} = (\frac{c_{bnd}^2}{\gamma s_{bnd}})^{1/(\gamma-1)}, \tag{2.90}$$

$$p_{bnd} = \rho_{bnd}(\frac{c_{bnd}^2}{\gamma}). \tag{2.91}$$

This Riemann invariant–based procedure is used to implement the far field boundary conditions for the RANS equations.

For the two equations of the $k-\omega$ SST model, far field boundary values of turbulent kinetic energy and specific dissipation rate must also be prescribed. At inflow boundaries, the former variable is set equal to a freestream value k_∞ determined using the user–given free stream velocity u_∞ and the turbulence intensity I, whereas the latter variable is set equal to a freestream value ω_∞ depending on k_∞ and the user–given value of the turbulent freestream viscosity μ_{T_∞}. The expressions of k_∞ and ω_∞ are respectively:

$$k_\infty = (\underline{\mathrm{u}}_\infty I)^2, \tag{2.92}$$

$$\omega_\infty = \gamma^* \frac{\rho_\infty k_\infty}{\mu_{T_\infty}}, \tag{2.93}$$

where γ^* is a turbulence model constant which has been set to $\gamma^* = 1$ in this work. At outflow boundaries, both turbulence kinetic energy and specific dissipation rate are extrapolated from the interior.

For unsteady problems solved in the relative frame using an absolute–frame representation of the components of the relative velocity, the freestream velocity u_∞ and boundary velocity u_{bnd} must be transformed to the absolute velocity in relative frame, given by Eqn. (2.57).

It is also important to stress that the HB solver uses exactly the same implementation of the farfield boundary condition as the steady and TD solvers. The only difference with respect to steady and TD solvers is that farfield boundary condition needs to be applied to each complex harmonics.

2.7.2 Solid wall

Generally, for all solid wall boundaries, the assumption that the fluid can not pass through a wall is made. For viscous flows governed by the NS equations, the no–slip condition must be satisfied at the surface, therefore, all components of the velocity at a wall are set equal to the wall velocity. In case of stationary wall $u = v = w = 0$, whereas for moving grid problems, the flow velocity at the wall is set equal to the solid surface velocity. We also make the assumption that the wall is adiabatic, so there is no heat flux trough the wall. Static pressure on the wall p_w is extrapolated from the interior solution using first order extrapolation. The fluid density at the wall ρ_w is calculated using the following equation:

$$\rho_w = \left(\frac{p_w}{p_{ext}}\right)^{1/\gamma} \cdot \rho_{ext}, \tag{2.94}$$

where the subscript *ext* denotes the extrapolated value from the interior solution using zero order extrapolation.

For the $k - \omega$ SST turbulence model two additional wall boundary conditions must be prescribed. Turbulent kinetic energy at the wall is set to zero ($k_w = 0$). For the assignment of the specific dissipation rate at the wall ω_w, two different models have been used. One is that proposed by Wilcox [72, 74] and the other is that proposed by Menter [69, 70].

Wilcox wall boundary condition has the advantage that the expression for ω_w can account for the effects of surface roughness. The value of ω_w is given by:

$$\omega_w = \frac{\underline{\mathrm{u}}_\tau^2}{\nu_w} S_R, \tag{2.95}$$

where $\underline{\mathrm{u}}_\tau$ represents the friction velocity, ν_w is the kinematic viscosity at the wall and S_R is a dimensionless surface–roughness function. The friction velocity is a function of the wall shear stress τ_w and the fluid density at the wall:

$$\underline{\mathrm{u}}_\tau = \sqrt{\frac{\tau_w}{\rho_w}}. \tag{2.96}$$

The parameter S_R can vary in the interval $0 \leq S_R \leq \infty$, where $S_R \longrightarrow \infty$ indicates that the wall is smooth. S_R is a function of a nondimensional average sand grain height k_S^+, and was determined based on the correlation. Various definitions of S_R exist in the literature, herein one of the recent definitions has been used [93]:

$$S_R = \begin{cases} \left(\dfrac{50}{k_S^+}\right)^2, & k_S^+ < 25 \\ \dfrac{100}{k_S^+}, & k_S^+ \geq 25 \end{cases}. \tag{2.97}$$

The nondimensional sand grain height is defined as:

$$k_S^+ = \frac{\mathrm{u}_\tau^2 k_S}{\nu_w}. \tag{2.98}$$

where k_S denotes the dimensional average sand grain height. For a smooth surface, the value of k_S must be small, to insure the condition $k_S^+ < 5$. On the other hand, for numerical reasons, the minimum value to be assigned to k_S^+ must be limited. The limitation proposed in [93] is adopted:

$$k_S^+ = max(1; \frac{\mathrm{u}_\tau^2 k_S}{\nu_w}). \tag{2.99}$$

Menter wall boundary condition is valid only for smooth surfaces, and is given by the following expression:

$$\omega_w = \frac{\nu_w}{\beta_1 (d_w)^2}, \tag{2.100}$$

where d_w represents the distance to the next point away from the wall, and β_1 is one of the coefficients of $k - \omega$ model, defined by Eqn. (2.30).

It was observed that both wall boundary conditions give accurate results, but some differences were observed in the results when solving complex physical problems. Menter wall was also numerically slightly more stable than Wilcox wall for some problems. For both wall boundary conditions, it must be assured that $y^+ < 3$, where y^+ is the dimensionless wall distance and is defined as:

$$y^+ = \frac{u_\tau d_w}{\nu_w}. \tag{2.101}$$

More detail of the implementation and the comparative analyses of these two wall boundary conditions for the specific dissipation rate can be found in references [26, 94], which compare the two models for both steady and moving grid problems.

In HB solver the implementation of the wall boundary condition is done in the same way as for the steady and TD solvers, with the difference that wall boundary condition needs to be applied to each complex harmonics.

Chapter 3

Numerical method

In the previous chapter the complete system of the RANS and $k-\omega$ SST PDEs has been obtained for the steady, TD and HB formulations. As the analytical solution of PDEs may be applicable only to few simplified flow problems, these equations need to be solved numerically in most cases. Numerous numerical methods exist, and they all require to represent the continuous nature of PDEs in a discrete form, leading to large system of algebraic equations. This nonlinear algebraic system is then solved using an iterative method. This chapter outlines the detailed description of the spatial and temporal discretization approaches, used to solve various forms of PDEs derived for this research. It also focuses on the description of the interface between blocks in multi–block approach. Furthermore, the description of the steady periodicity cut boundary condition and the multi–frequency periodicity boundary conditions is provided. The chapter is concluded with the integration approach of steady, TD and HB RANS equations.

As it has been mentioned in the introduction, the starting point of the numerical implementation were the 2D solvers of COSA. In this work, all existing 2D solvers have been extended to 3D and new routines have been implemented in both serial and parallel environments. Parallel implementation of the 2D version of COSA has been primarily done in EPCC at the University of Edinburgh, therefore, some support was provided by them. However, all parallel routines have been extended and added by the author.

3.1 Space discretization

Spatial discretization comprises of the numerical approximation of the convective and diffusive fluxes, as well as of source terms. Roe's approximate Riemann solver is one of the flux–difference splitting schemes, which allows very precise calculation of boundary layers. It is used in the discretization of convective fluxes of the developed flow solvers for this work, and will be explained in next section.

3.1.1 Convective fluxes

The discretization of the convective fluxes is based on Roe's approximate Riemann solver [95] and Van Leer's Monotonic Upstream–Centered Scheme for Conservation Laws (MUSCL) extrapolations [96].

Roe's approximate Riemann solver is based on a characteristic decomposition of flux differences over a cell face, ensuring the conservation of governing equations. The convective flux vector is written as the sum of a left and right state, and a flux difference across the cell face. It is a Godunov type method [97], and was developed in order to reduce the computational expense of Godunov's scheme for the exact solution of the Riemann problem. Several other approximate Riemann solvers for the Euler equations exist (e.g. Osher's approximate Riemann solver [98, 99]), however, the approach of Roe is most widely used.

Let us first express the generalised convective flux vector $\underline{\mathbf{\Phi}}_c$ given by Eqn. (2.45), in terms of Cartesian components:

$$\underline{\mathbf{\Phi}}_c = \mathbf{E}_c i + \mathbf{F}_c j + \mathbf{G}_c k, \tag{3.1}$$

where $\mathbf{E}_c$, $\mathbf{F}_c$ and $\mathbf{G}_c$ represent the x, y and z components of the generalised convective flux vector, and i, j and k are the unit vectors. The continuous convective flux component through the face can therefore be represented by the expression:

$$\mathbf{\Phi}_{c,f} = (\underline{\mathbf{\Phi}}_c \cdot \underline{\mathrm{n}})dS = (\mathbf{E}_c n_x + \mathbf{F}_c n_y + \mathbf{G}_c n_z)\, dS, \tag{3.2}$$

where n_x, n_y and n_z are the x, y and z components of the outward normal $\underline{\mathrm{n}}$. The symbol dS represents the cell–face area, across which the flux is being computed, and subscript $_f$ indicates a face value. The interface flux is calculated in all three coordinate directions x, y and z, as the solution of a locally one–dimensional Riemann problem normal to the cell interface.

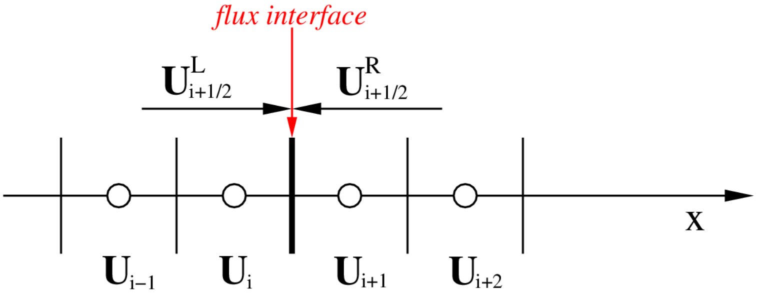


Figure 3.1: One–sided extrapolation of flux interface.

In order to calculate the convective fluxes $\mathbf{\Phi}_{c,f}$, the conservative variables of the flow state $\mathbf{U}$, given by Eqn. (2.44), must be extrapolated from the flowfield. In this work, this is done with the MUSCL approach using a fully one–sided interpolation of the flow variables, which results in second–order accurate upwind approximation on uniformly spaced grid. Figure 3.1 depicts a one–dimensional representation of the computational grid in the x–direction, where each point represents the cell–centered value of $\mathbf{U}$. The flow state at the interface can be approximated as:

$$\mathbf{U}^L_{i+\frac{1}{2}} = \mathbf{U}_i + \frac{1}{2}(\mathbf{U}_i - \mathbf{U}_{i-1}), \tag{3.3}$$

$$\mathbf{U}^R_{i+\frac{1}{2}} = \mathbf{U}_{i+1} - \frac{1}{2}(\mathbf{U}_{i+2} - \mathbf{U}_{i+1}), \tag{3.4}$$

where $\mathbf{U}^L$ and $\mathbf{U}^R$ represent left and right flow state. The turbulent convective flux components can be computed using either a first order or a second order upwind scheme. However, it was observed that for some cases the choice of either approach yields non–negligible differences between the two solutions [26]. For this reason, all the results reported in this work have been obtained using the second order discretization for turbulent convective flux components. To avoid any unwanted oscillations around shocks, and other discontinuities or strong gradients in the flow field, non–linear limiter function must also be employed when using second–order upwind schemes. This is done in order to construct a monotonicity preserving discretization when necessary. For this work, Van Albada limiter [100] was used. Using such a limiter, the left and right flow states at the interface given by Eqns. (3.3) and (3.4) may be rewritten respectively as:

$$\mathbf{U}^{L}_{i+\frac{1}{2}} = \mathbf{U}_i + \frac{1}{2}\delta_l, \tag{3.5}$$

$$\mathbf{U}^{R}_{i+\frac{1}{2}} = \mathbf{U}_{i+1} - \frac{1}{2}\delta_l, \tag{3.6}$$

where δ_l denotes Van Albada limiter function. If a and b are the subsequent finite differences of a smooth solution (e.g. $\mathbf{U}_{i+2}$ and $\mathbf{U}_{i+1}$), then δ_l tends to $\frac{1}{2}(a+b)$. If the solution is not smooth, δ_l tends to the smallest value. The limiter function δ_l may be written as:

$$\delta_l = \frac{(b^2+\epsilon^2)a + (a^2+\epsilon^2)b}{a^2+b^2+2\epsilon^2}, \tag{3.7}$$

where ϵ^2 represents a small non–vanishing bias. For the calculations made in this work, selected ϵ value has usually been of the same order as a non–dimensional minimum distance from the wall. It was also noted that the results are not very sensitive to the selected value of ϵ^2. The usage of the small non–vanishing bias prevents clipping of a smooth extrema, but does not have any negative influence on the smooth solution.

Having defined the left and right states $\mathbf{U}^L$ and $\mathbf{U}^R$, let us write the expression for the numerical approximation of the convective fluxes $\mathbf{\Phi}^*_{c,f}$:

$$\mathbf{\Phi}^*_{c,f} = \frac{1}{2}\left[\mathbf{\Phi}_{c,f}(\mathbf{U}_L) + \mathbf{\Phi}_{c,f}(\mathbf{U}_R) - \delta\mathbf{\Phi}\right], \tag{3.8}$$

where the superscript * denotes numerical approximation, and subscripts $_L$ and $_R$ stand for a value extrapolated from the left and from the right, respectively. The last term of Eqn. (3.8), $\delta\mathbf{\Phi}$, denotes the numerical dissipation which can be written in the following way:

$$\delta\mathbf{\Phi} = |K_U|\delta\mathbf{U}, \tag{3.9}$$

where K_U represents the generalised flux Jacobian or so called Roe's matrix, defined as:

$$K_U = \frac{\partial\mathbf{\Phi}_{c,f}}{\partial\mathbf{U}} = \frac{\partial\mathbf{E}_c}{\partial\mathbf{U}}n_x + \frac{\partial\mathbf{F}_c}{\partial\mathbf{U}}n_y + \frac{\partial\mathbf{G}_c}{\partial\mathbf{U}}n_z = An_x + Bn_y + Cn_z. \tag{3.10}$$

The symbols A, B and C denote the flux Jacobians of the convective fluxes K_U in the following three directions, x, y and z. The second term of Eqn. (3.9), describes the flow state discontinuity across the cell face:

$$\delta\mathbf{U} = (\mathbf{U}_R - \mathbf{U}_L). \tag{3.11}$$

The numerical dissipation expressed by Eqn. (3.9) can be rewritten in an alternative and computationally more efficient way as:

$$\delta\mathbf{\Phi} = |K_U|\,\delta\mathbf{U} = P|\Lambda|P^{-1}\delta\mathbf{U} = P|\Lambda|\delta\mathbf{W}. \tag{3.12}$$

The matrix K_U in Eqn. (3.12) is not symmetric, and therefore it has a different set of left and right eigenvectors. Let us denote by P the matrix of the right eigenvectors, and by Λ the diagonal matrix containing the corresponding eigenvalues. The eigenvalues are the associated wave speeds of the solution of approximate Riemann problem, and the right eigenvectors are associated with the waves. The symbol P^{-1} denotes the matrix of left eigenvectors of K_U, which is the inverse of the matrix P of the right eigenvectors. The symbol $\delta\mathbf{W}$ denotes instead the characteristic variables, which are defined by the following expression:

$$\delta\mathbf{W} = P^{-1}\delta\mathbf{U}. \tag{3.13}$$

In order to calculate the numerical dissipation, it is much easier to work with the governing equations written in non–conservative form as a function of the primitive variables. The relations between the conservative and the non–conservative Jacobians can be expressed through a similarity transformation matrix. The whole process of derivation of the numerical dissipation is described in Appendix A.

The formulation of the convective flux component through the face represented by Eqn. (3.2) will produce an unphysical solution due to the addition of the dissipation term for improved nonlinear instability. When there is a presence of expansion shocks and contact discontinuities in the solutions, the scheme might converge to a nonphysical solution. The difficulty is that the original scheme does not recognise the sonic point. To overcome such difficulty, Harten's entropy correction [101] is introduced to assure physically relevant solutions. The correction is used to modify the modulus of all eigenvalues Λ, and can be written as:

$$|\lambda| = \begin{cases} |\lambda|, & if\ |\lambda| > \delta \\ \dfrac{\lambda^2 + \delta^2}{2\delta}, & if\ |\lambda| \leq \delta \end{cases}, \tag{3.14}$$

where δ represents a tuning parameter. When δ is zero, the entropy correction is not used. The larger the value of δ, the stronger is the effect of the entropy fix in the numerical scheme. Many published studies suggest which value of δ to use. In instances when the entropy correction was used in this work, either a small constant value $\delta < 1$ was selected or the following definition proposed in [102, 103] was used to calculate the δ for each cell of the computational domain:

$$\delta = \tilde{\delta}(|u| + |v| + |w| + c), \tag{3.15}$$

where $0.05 \leq \tilde{\delta} \leq 0.25$.

Moving grid problems

In many application areas we encounter moving–grid problems. If such problems are solved in absolute frame, the grid has to move to accommodate changing boundary. The velocity at which the grid boundary S moves is denoted by $\underline{u}_b$. Governing equations for moving problems are easily derived by replacing the velocity vector $\underline{u}$ in the convective fluxes with the relative velocity $\underline{u} - \underline{u}_b$. The generalised convective flux vector $\underline{\boldsymbol{\Phi}}_c$ for moving grid problems in terms of Cartesian components, may then be written:

$$\underline{\boldsymbol{\Phi}}_c = \mathbf{E}_c i + \mathbf{F}_c j + \mathbf{G}_c k - \underline{u}_b\mathbf{U}, \tag{3.16}$$

where $\mathbf{E}_c$, $\mathbf{F}_c$ and $\mathbf{G}_c$ represent the x, y and z components of generalised convective flux vector, and i, j and k are the unit vectors. The term $-\underline{u}_b\mathbf{U}$ represents the contribution of the cell boundary velocity to the overall flux balance. The expression of the convective fluxes at the boundary of each cell reads:

$$\boldsymbol{\Phi}_{c,f} = (\underline{\boldsymbol{\Phi}}_c \cdot \underline{n})dS = (\mathbf{E}_c n_x + \mathbf{F}_c n_y + \mathbf{G}_c n_z - \mathbf{U}u_{bn})\,dS, \tag{3.17}$$

where

$$u_{bn} = \underline{\mathrm{u}}_b \cdot \underline{n}.$$

n_x, n_y and n_z are the x, y and z components of the outward normal $\underline{\mathrm{n}}$. The numerical representation of the convective fluxes at each cell boundary remains the same as in Eqn. (3.8). However, in the expression of fluxes $\mathbf{\Phi}_{c,f}(\mathbf{U}_L)$ and $\mathbf{\Phi}_{c,f}(\mathbf{U}_R)$ there is now also the flux contribution associated with the boundary velocity. Furthermore, the expression of the numerical dissipation term $\delta\mathbf{\Phi}$ given by Eqn. (3.12), includes the generalised flux Jacobian K_U, which becomes:

$$(K'_U)_f = \frac{\partial \mathbf{\Phi}_{c,f}}{\partial \mathbf{U}} = \left(\frac{\partial \mathbf{E}_c}{\partial \mathbf{U}} n_x + \frac{\partial \mathbf{F}_c}{\partial \mathbf{U}} n_y + \frac{\partial \mathbf{G}_c}{\partial \mathbf{U}} n_z \right)_f - I u_{bn} = (K_U)_f - I u_{bn}. \tag{3.18}$$

The Jacobians $(K_U)_f$ and $(K'_U)_f$ differ only by a diagonal term; thus, the eigenvectors of both matrices are the same. The eigenvalues of $(K_U)_f$ which are provided in Appendix A by Eqn. (A.16) and $(K'_U)_f$ differ only by the constant offset due to u_{bn}. Therefore the eigenvalues of $(K'_U)_f$ are the following:

$$\begin{aligned} &\lambda_1 = \lambda_2 = \lambda_3 = \lambda_6 = \lambda_7 = U_n - u_{bn}, \\ &\lambda_4 = U_n - u_{bn} + c, \\ &\lambda_5 = U_n - u_{bn} - c. \end{aligned} \tag{3.19}$$

Hence, the expression of the numerical dissipation $\delta\mathbf{\Phi}$ for problems with moving grids is then formally identical to the expression (A.24). The only difference with respect to problems with motionless grid is the appearance of the boundary velocity term in the eigenvalues of matrix $(K'_U)_f$.

3.1.2 Diffusive fluxes

The discretization of the diffusive fluxes of governing equations has been done using second order finite–differencing. More precisely, the components of the velocity vector $\underline{\mathrm{u}}$, the dynamic viscosity μ, the eddy viscosity μ_T, which are required for the computation of the viscous fluxes (2.46), are averaged at cell face. The values at the face (i + 1/2) of the control volume are evaluated using simple expression:

$$a_{i+1/2} = \frac{1}{2}(a_{i-1} + a_i), \tag{3.20}$$

where a can represent any of the mentioned variables.

The first derivatives of the velocity ($\nabla \underline{\mathrm{u}}$), temperature ($\nabla T$), turbulent kinetic energy (∇k), specific dissipation rate ($\nabla \omega$) are computed by considering the local transformation from Cartesian coordinates (x,y,z) to the generalised curvilinear coordinates (ξ, η, ζ) associated with the grid lines. The use of generalised curvilinear coordinates implies that the arbitrary geometry is transformed into a rectangular region in the generalised coordinate space, as depicted in Fig. 3.2 [104]. In order to relate the Cartesian derivatives of the velocity components to their derivatives in the (ξ, η, ζ) system and the grid metrics, the chain rule is used. To approximate all required derivatives, second order finite–differences are used. Using the chain rule, the Cartesian derivatives of the velocity vector $\underline{\mathrm{u}}$ can be expressed as follows:

$$\frac{du}{dx} = \frac{du}{d\xi}\frac{d\xi}{dz} + \frac{du}{d\eta}\frac{d\eta}{dz} + \frac{du}{d\zeta}\frac{d\zeta}{dz}, \tag{3.21}$$

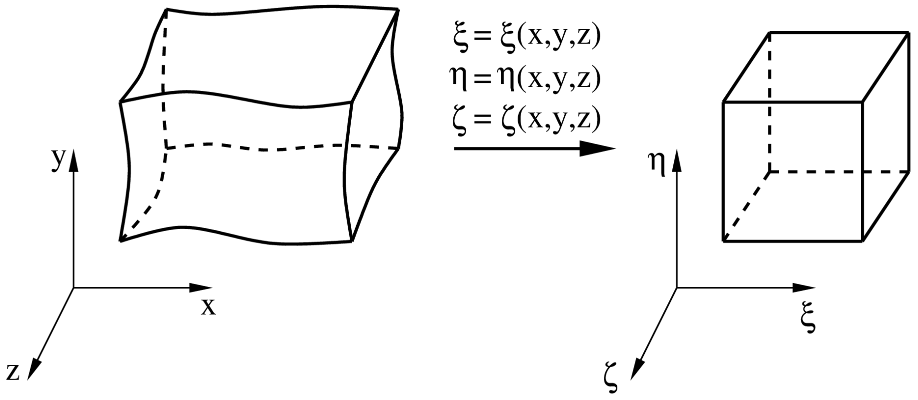


Figure 3.2: Cartesian to generalised curvilinear coordinate transformation.

$$\frac{du}{dy} = \frac{du}{d\xi}\frac{d\xi}{dz} + \frac{du}{d\eta}\frac{d\eta}{dz} + \frac{du}{d\zeta}\frac{d\zeta}{dz}, \tag{3.22}$$

$$\frac{du}{dz} = \frac{du}{d\xi}\frac{d\xi}{dz} + \frac{du}{d\eta}\frac{d\eta}{dz} + \frac{du}{d\zeta}\frac{d\zeta}{dz}, \tag{3.23}$$

$$\frac{dv}{dx} = \frac{dv}{d\xi}\frac{d\xi}{dz} + \frac{dv}{d\eta}\frac{d\eta}{dz} + \frac{dv}{d\zeta}\frac{d\zeta}{dz}, \tag{3.24}$$

$$\frac{dv}{dy} = \frac{dv}{d\xi}\frac{d\xi}{dz} + \frac{dv}{d\eta}\frac{d\eta}{dz} + \frac{dv}{d\zeta}\frac{d\zeta}{dz}, \tag{3.25}$$

$$\frac{dv}{dz} = \frac{dv}{d\xi}\frac{d\xi}{dz} + \frac{dv}{d\eta}\frac{d\eta}{dz} + \frac{dv}{d\zeta}\frac{d\zeta}{dz}, \tag{3.26}$$

$$\frac{dw}{dx} = \frac{dw}{d\xi}\frac{d\xi}{dz} + \frac{dw}{d\eta}\frac{d\eta}{dz} + \frac{dw}{d\zeta}\frac{d\zeta}{dz}, \tag{3.27}$$

$$\frac{dw}{dy} = \frac{dw}{d\xi}\frac{d\xi}{dz} + \frac{dw}{d\eta}\frac{d\eta}{dz} + \frac{dw}{d\zeta}\frac{d\zeta}{dz}, \tag{3.28}$$

$$\frac{dw}{dz} = \frac{dw}{d\xi}\frac{d\xi}{dz} + \frac{dw}{d\eta}\frac{d\eta}{dz} + \frac{dw}{d\zeta}\frac{d\zeta}{dz}. \tag{3.29}$$

u, v and w are the components of the velocity vector $\underline{u}$. The $\xi-$ $\eta-$ and $\zeta-$ derivatives of u, v and w are computed using second order centered finite–differences. The $\xi-$ $\eta-$ and $\zeta-$ derivatives of u on face $(i+1/2, j, k)$ are as follows:

$$\frac{du}{d\xi} = \frac{u(i+1,j,k) - u(i,j,k)}{2\delta\xi/2} = u(i+1,j,k) - u(i,j,k), \tag{3.30}$$

$$\frac{du}{d\eta} = \frac{u_B - u_A}{2\delta\eta/2} = u_B - u_A, \tag{3.31}$$

$$\frac{du}{d\zeta} = \frac{u_D - u_C}{2\delta\zeta/2} = u_D - u_C. \tag{3.32}$$

The variables u_A, u_B, u_C and u_D are not values at cell centres, therefore, they need to be evaluated using the following expressions:

$$u_A = \frac{u(i,j-1,k) + u(i+1,j-1,k) + u(i+1,j,k) + u(i,j,k)}{4}, \tag{3.33}$$

$$u_B = \frac{u(i,j,k) + u(i+1,j,k) + u(i+1,j+1,k) + u(i,j+1,k)}{4}, \tag{3.34}$$

$$u_C = \frac{u(i,j,k-1) + u(i+1,j,k-1) + u(i+1,j,k) + u(i,j,k)}{4}, \tag{3.35}$$

$$u_D = \frac{u(i,j,k) + u(i+1,j,k) + u(i+1,j,k+1) + u(i,j,k+1)}{4}. \tag{3.36}$$

When calculating the $\xi-$ $\eta-$ and $\zeta-$ derivatives of u along the j family of grid lines (e.g. on face $(i, j+1/2, k)$), the u derivatives in the η direction are easily computed using cell centre data, however, the ξ and ζ derivatives must be determined as seen above, expressing the values as function of values at surrounding cell centres. The same holds for estimation of the $\xi-$ $\eta-$ and $\zeta-$ derivatives of u along the k family of grid lines (e.g. on face $(i, j, k+1/2)$), where the u derivatives in the ζ direction are computed using directly cell centre data, and the ξ and η derivatives are determined expressing the values as function of values at surrounding cell centres.

The $\xi-$ $\eta-$ and $\zeta-$ derivatives of v and w velocity components along the i, j and k family of grid lines can be estimated using the same approach as shown above.

The metrics $d\xi/dx$, $d\xi/dy$, $d\xi/dz$, $d\eta/dx$, $d\eta/dy$, $d\eta/dz$, $d\zeta/dx$, $d\zeta/dy$ and $d\zeta/dz$ at the face centers are written as functions of the derivatives of the inverse coordinate transformation, and then discretized using second order finite–differences. Interested reader may refer to [104], where the detailed derivation is provided.

Similarly as for the first derivatives of the velocity vector $(\nabla \underline{u})$, the first derivatives of the temperature (∇T), turbulent kinetic energy (∇k), and specific dissipation rate $(\nabla \omega)$ can be written as:

$$\frac{dT}{dx} = \frac{dT}{d\xi}\frac{d\xi}{dx} + \frac{dT}{d\eta}\frac{d\eta}{dx} + \frac{dT}{d\zeta}\frac{d\zeta}{dx}, \tag{3.37}$$

$$\frac{dT}{dy} = \frac{dT}{d\xi}\frac{d\xi}{dy} + \frac{dT}{d\eta}\frac{d\eta}{dy} + \frac{dT}{d\zeta}\frac{d\zeta}{dy}, \tag{3.38}$$

$$\frac{dT}{dz} = \frac{dT}{d\xi}\frac{d\xi}{dz} + \frac{dT}{d\eta}\frac{d\eta}{dz} + \frac{dT}{d\zeta}\frac{d\zeta}{dz}, \tag{3.39}$$

$$\frac{dk}{dx} = \frac{dk}{d\xi}\frac{d\xi}{dx} + \frac{dk}{d\eta}\frac{d\eta}{dx} + \frac{dk}{d\zeta}\frac{d\zeta}{dx}, \tag{3.40}$$

$$\frac{dk}{dy} = \frac{dk}{d\xi}\frac{d\xi}{dy} + \frac{dk}{d\eta}\frac{d\eta}{dy} + \frac{dk}{d\zeta}\frac{d\zeta}{dy}, \tag{3.41}$$

$$\frac{dk}{dz} = \frac{dk}{d\xi}\frac{d\xi}{dz} + \frac{dk}{d\eta}\frac{d\eta}{dz} + \frac{dk}{d\zeta}\frac{d\zeta}{dz}, \tag{3.42}$$

$$\frac{d\omega}{dx} = \frac{d\omega}{d\xi}\frac{d\xi}{dx} + \frac{d\omega}{d\eta}\frac{d\eta}{dx} + \frac{d\omega}{d\zeta}\frac{d\zeta}{dx}, \tag{3.43}$$

$$\frac{d\omega}{dy} = \frac{d\omega}{d\xi}\frac{d\xi}{dy} + \frac{d\omega}{d\eta}\frac{d\eta}{dy} + \frac{d\omega}{d\zeta}\frac{d\zeta}{dy}, \tag{3.44}$$

$$\frac{d\omega}{dz} = \frac{d\omega}{d\xi}\frac{d\xi}{dz} + \frac{d\omega}{d\eta}\frac{d\eta}{dz} + \frac{d\omega}{d\zeta}\frac{d\zeta}{dz}. \tag{3.45}$$

Let us now consider the discretization of the viscous stresses which appear in momentum equation given by Eqn. (2.2). Viscous stresses need to be discretized on cell faces of each control volume. The net flux of the viscous fluxes on the cell i,j,k in the x–direction is the following:

$$\begin{gathered}
[(\tau_{xx}n_x + \tau_{xy}n_y + \tau_{xz}n_z)\Delta S]_{i+1/2,j,k} + [(\tau_{xx}n_x + \tau_{xy}n_y + \tau_{xz}n_z)\Delta S]_{i-1/2,j,k} + \\
[(\tau_{xx}n_x + \tau_{xy}n_y + \tau_{xz}n_z)\Delta S]_{i,j+1/2,k} + [(\tau_{xx}n_x + \tau_{xy}n_y + \tau_{xz}n_z)\Delta S]_{i,j-1/2,k} + \\
[(\tau_{xx}n_x + \tau_{xy}n_y + \tau_{xz}n_z)\Delta S]_{i,j,k+1/2} + [(\tau_{xx}n_x + \tau_{xy}n_y + \tau_{xz}n_z)\Delta S]_{i,j,k-1/2}.
\end{gathered} \tag{3.46}$$

In y–direction we need to compute:

$$\begin{aligned}&[(\tau_{xy}n_x+\tau_{yy}n_y+\tau_{yz}n_z)\Delta S]_{i+1/2,j,k}+[(\tau_{xy}n_x+\tau_{yy}n_y+\tau_{yz}n_z)\Delta S]_{i-1/2,j,k}+\\&[(\tau_{xy}n_x+\tau_{yy}n_y+\tau_{yz}n_z)\Delta S]_{i,j+1/2,k}+[(\tau_{xy}n_x+\tau_{yy}n_y+\tau_{yz}n_z)\Delta S]_{i,j-1/2,k}+\\&[(\tau_{xy}n_x+\tau_{yy}n_y+\tau_{yz}n_z)\Delta S]_{i,j,k+1/2}+[(\tau_{xy}n_x+\tau_{yy}n_y+\tau_{yz}n_z)\Delta S]_{i,j,k-1/2}.\end{aligned}\tag{3.47}$$

And finally the expression for z–direction is the following:

$$\begin{aligned}&[(\tau_{xz}n_x+\tau_{yz}n_y+\tau_{zz}n_z)\Delta S]_{i+1/2,j,k}+[(\tau_{xz}n_x+\tau_{yz}n_y+\tau_{zz}n_z)\Delta S]_{i-1/2,j,k}+\\&[(\tau_{xz}n_x+\tau_{yz}n_y+\tau_{zz}n_z)\Delta S]_{i,j+1/2,k}+[(\tau_{xz}n_x+\tau_{yz}n_y+\tau_{zz}n_z)\Delta S]_{i,j-1/2,k}+\\&[(\tau_{xz}n_x+\tau_{yz}n_y+\tau_{zz}n_z)\Delta S]_{i,j,k+1/2}+[(\tau_{xz}n_x+\tau_{yz}n_y+\tau_{zz}n_z)\Delta S]_{i,j,k-1/2}.\end{aligned}\tag{3.48}$$

The same discretization procedure applies to other entries of viscous flux vector, given by Eqn. (2.46).

Discretization of the turbulent source terms has been done following the same procedure as for the diffusive fluxes. However, there is an important difference, flow variables required to evaluate the source terms of k and ω are averaged at cell face, whereas the Cartesian derivatives of the flow velocity are determined at the cell centers, using the same approach as for the diffusive fluxes.

3.2 Calculation of cell face velocities

The velocity of the cell faces $\underline{u}_b$ can be calculated in many different ways. Various formulations based on the grid motion types are presented in [105]. If the location of the grid in inertial frame is known as a function of time, the usual approach is to calculate $\underline{u}_b$ explicitly at the cell–face center. However, while using this approach, the governing equations in non–inertial frame may no longer maintain global conservation. The $\underline{u}_b$, which appears in the source terms of governing equations in non–inertial frame (2.55), makes the URANS and RANS equations non–conservative if the geometric terms as not evaluated properly [84].

The continuity equation means that for the steady flow the following condition needs to be satisfied:

$$\oint_S \rho\underline{u}\cdot\underline{n}dS=0.\tag{3.49}$$

By considering a uniform flow and to avoid introducing numerical errors, the following condition must be satisfied for every cell:

$$\sum_{k=1}^{6}(\underline{n}A)_k=\underline{0}',\tag{3.50}$$

where the symbol k represents the six cell faces of the control volume. The condition is satisfied by evaluating the cell face normals and their area with the cross product of the two diagonal vectors, where each diagonal is perpendicular to the surface normal. Due to the rotation of the coordinate system, the following condition must be satisfied:

$$\oint_{S_r}\rho(\underline{u}_r-\underline{\Omega}\times\underline{r}_a)\cdot\underline{n}_r dS_r=0.\tag{3.51}$$

By considering a rotating grid through a uniform flow and to avoid introducing numerical errors due to the rotation of the grid, the following equation must be valid for every cell in the computational domain:

$$\sum_{k=1}^{6}((\underline{\Omega}\times\underline{r}_a)\cdot\underline{n}A)_k=\underline{0}'.\tag{3.52}$$

If this condition is not satisfied, the governing equations are no longer conservative, and this may introduce numerical errors into all equations. To satisfy Eqn. (3.52) we may define the moment of the area vector for each cell–face as in [105]:

$$\underline{\mathrm{M}} = \oint_{S_r} \underline{\mathrm{r}}_a \times \underline{\mathrm{n}} dS_r. \tag{3.53}$$

Once the area moment vector is normalized by the cell face area:

$$\underline{\mathrm{M}}' = \frac{\underline{\mathrm{M}}}{A}, \tag{3.54}$$

it can be used for the calculation of the cell faces $\underline{\mathrm{u}}_b$:

$$\underline{\mathrm{u}}_b = \underline{\Omega} \cdot \underline{\mathrm{M}}'. \tag{3.55}$$

The conservation of governing equations is now assured.

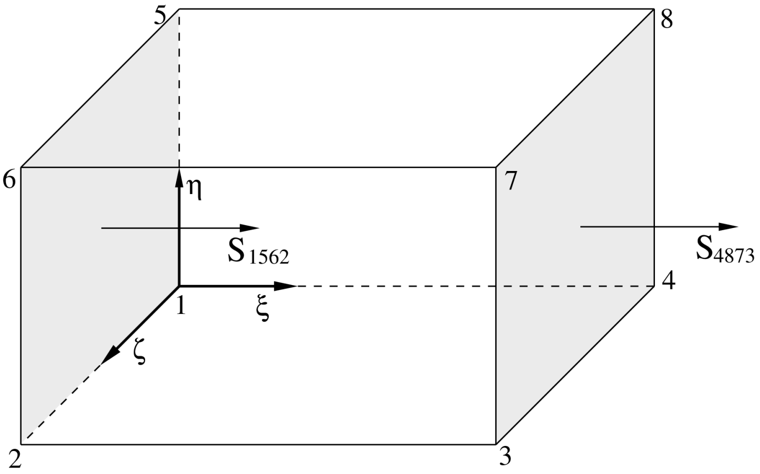


Figure 3.3: Sketch of a hexahedral cell.

For evaluation of $\underline{\mathrm{M}}$ the approach as in [85, 105] has been adopted. Figure 3.3 depicts a hexahedron defined by eight vertices with edges denoting ξ, η and ζ directions. The ξ–face surface vectors are $\underline{\mathrm{S}}_{1562}$ and $\underline{\mathrm{S}}_{4873}$. Surface area can be calculated as the vector product of the two diagonals, let us write the expression for the former surface vector:

$$\underline{\mathrm{S}}_{1562} = \frac{1}{2}(\underline{\mathrm{r}}_6 - \underline{\mathrm{r}}_1) \times (\underline{\mathrm{r}}_5 - \underline{\mathrm{r}}_2), \tag{3.56}$$

where $\underline{\mathrm{r}}_i, i = 1, 8$ is the position vector of a point in space. Inserting the expression above into Eqn. (3.53) we obtain:

$$\underline{\mathrm{M}}_{1562} = \int_{1562} \underline{\mathrm{r}}_{1562} \times \underline{\mathrm{S}}_{1562}, \tag{3.57}$$

where:

$$\underline{\mathrm{r}}_{1652} = 1/4(\underline{\mathrm{r}}_1 + \underline{\mathrm{r}}_6 + \underline{\mathrm{r}}_5 + \underline{\mathrm{r}}_2). \tag{3.58}$$

Such expression of area moment results in non–zero sum over a cell and is therefore not well defined for computing area moment. For the sum over a cell to be zero, the area moment must be obtained by dividing the cell–face along one of the diagonals. In this case Eqn. (3.57) is modified in such way:

$$\underline{\mathrm{M}}_{1562} = \int_{1562} \underline{\mathrm{r}}_{165} \times \underline{\mathrm{S}}_{165} + \underline{\mathrm{r}}_{126} \times \underline{\mathrm{S}}_{126}, \tag{3.59}$$

where,

$$\begin{aligned}\underline{\mathrm{S}}_{165} &= \frac{1}{2}(\underline{\mathrm{r}}_6 - \underline{\mathrm{r}}_1) \times (\underline{\mathrm{r}}_5 - \underline{\mathrm{r}}_1),\\ \underline{\mathrm{S}}_{126} &= \frac{1}{2}(\underline{\mathrm{r}}_2 - \underline{\mathrm{r}}_1) \times (\underline{\mathrm{r}}_6 - \underline{\mathrm{r}}_1),\end{aligned} \tag{3.60}$$

and

$$\begin{aligned}\underline{\mathrm{r}}_{165} &= 1/3(\underline{\mathrm{r}}_1 + \underline{\mathrm{r}}_6 + \underline{\mathrm{r}}_5),\\ \underline{\mathrm{r}}_{126} &= 1/3(\underline{\mathrm{r}}_1 + \underline{\mathrm{r}}_2 + \underline{\mathrm{r}}_6).\end{aligned} \tag{3.61}$$

The expression given by Eqn. (3.59) is well suited for the computation of the area moment and it satisfies the discretized geometric identity of Eqn. (3.52).

3.3 Cut boundary condition

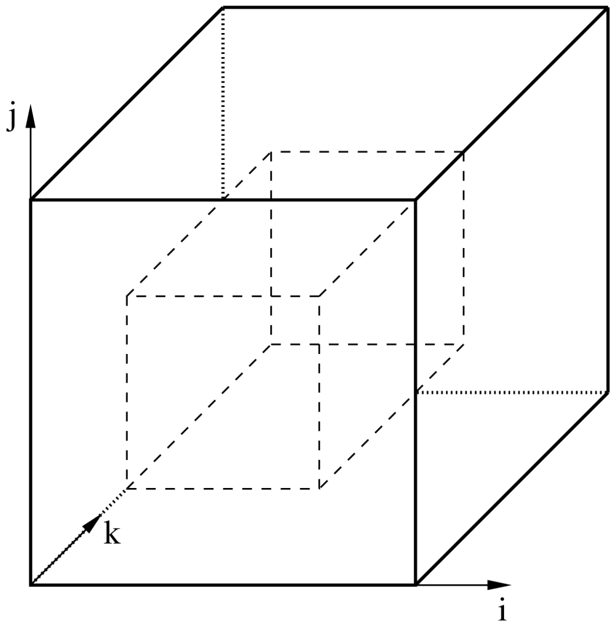


Figure 3.4: Block, surrounded by the two rows of halo cells representing boundary conditions.

COSA is using structured grids with multi–block topology. The computational domain is decomposed into smaller structured subdomains called blocks. Generally, the physical solution inside each block will depend on one or multiple neighbouring blocks, therefore, data structure which allows an efficient exchange of information between blocks is provided. Such exchange is done through the interface between the blocks, called grid cut [66, 106].

The structure of cut boundary condition is quite complex, as the different blocks require all relevant information at their boundaries at all times. In order to exchange these information efficiently, two rows of halo cells are added to the computational domain, as depicted in Fig. 3.4. Two rows are needed to assure second order spatial accuracy. As the grid points of the structured grids are uniquely identified by the indices $(i,\, j,\, k)$, the boundary of each block face can be divided into various non–overlapping patches, as depicted in Fig. 3.5. This approach is very flexible, as it enables declaration of multiple different boundary conditions per block face. It also simplifies balancing

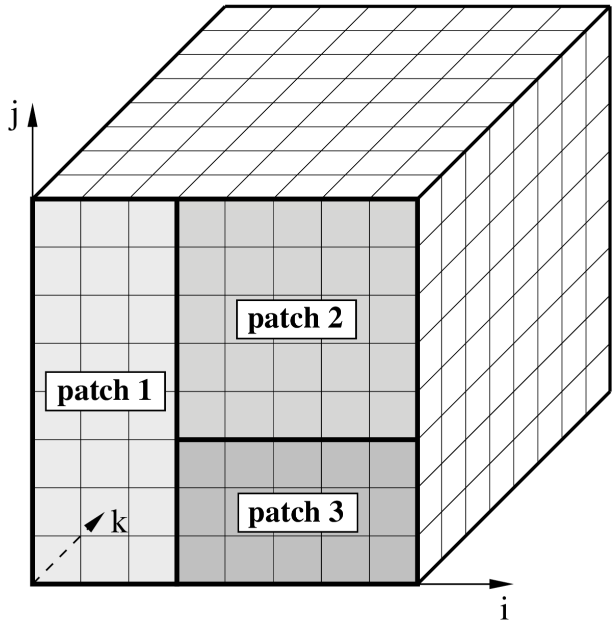


Figure 3.5: Representation of three different boundary patches on boundary face.

the grid blocks in the computational domain, which is a very important aspect for parallel computing.

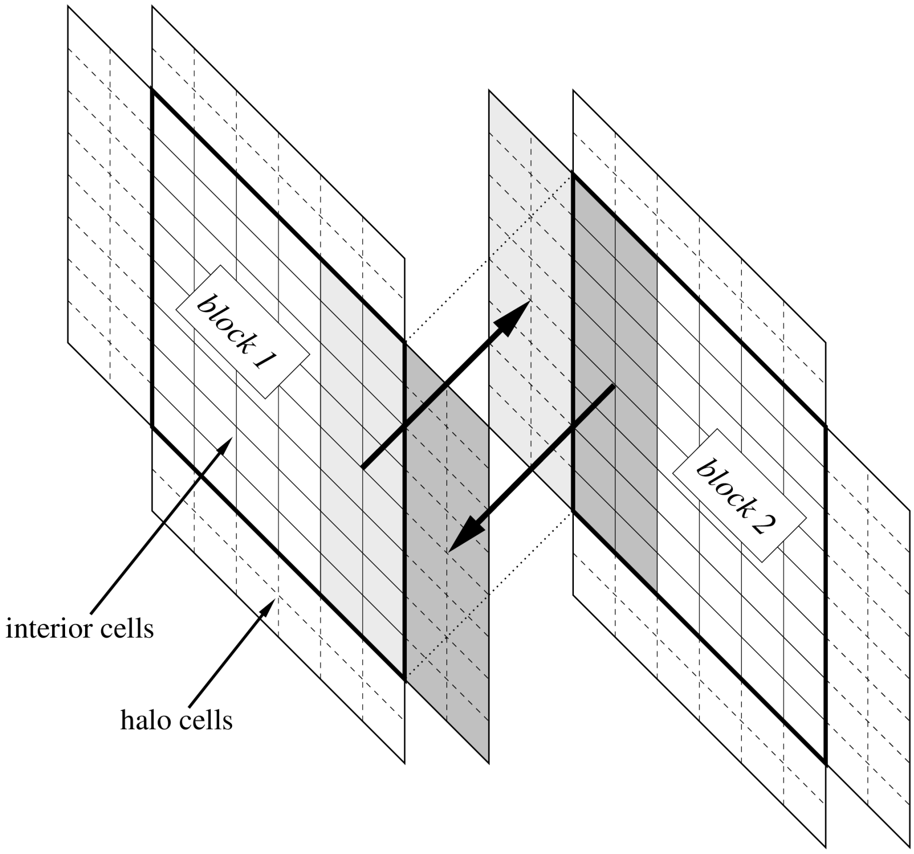


Figure 3.6: Sectional slice of j–k plane through the interior domain and halo cells of patch 1.

The treatment of the cut condition involves data exchange between two neighbouring blocks. The boundary between two blocks must be transparent to the flow, and all variables across the boundary should be continuous. The data exchange, which is done across the patches, is a two step operation, involving two neighbouring blocks (e.g. *block*1 and *block*2). As depicted in Fig. 3.6, halo cells of *block*1 are located in such way that they overlap the corresponding cells in the interior domain of *block*2.

Interested reader is referred to [66], where various types of computational grids and halo cells in CFD are described.

3.4 Steady and multi–frequency periodicity boundary condition

For some particular applications when the flowfield is periodic, it is sufficient to simulate the flow only within one repeating region of the whole computational domain [66]. The interaction of repeating region with the remaining physical domain is provided through the periodic boundary conditions. Herein we focus on rotational periodicity, where one periodic boundary is transformed into the other periodic boundary by the coordinate rotation. Figure 3.7 represents one repeating region or sector, which depicts two rotationally periodic boundaries, (*boundary 1* and *boundary 2*), where θ is the rotation angle between these two periodic boundaries, and $\underline{\Omega}$ is constant angular velocity given by Eqn. (2.54). For this work, we have assumed that the rotational

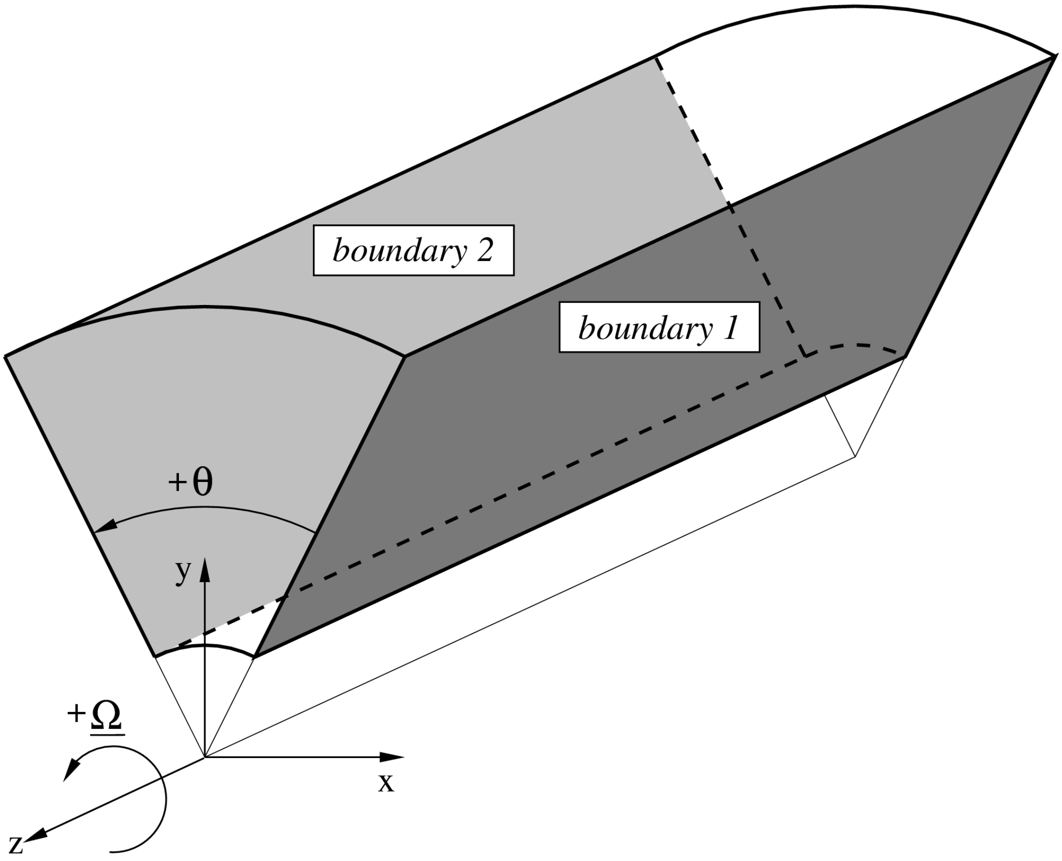


Figure 3.7: Sketch of a sector, with two periodic boundaries.

axis coincides with the z–axis, and all grid nodes are matching. The data exchange between two periodic blocks is done across the patches on the block surfaces, and it follows exactly the same principle that involves cut boundary condition.

Steady periodicity boundary condition depends only on the rotation of the coordinate system. This means that all scalar quantities (density, pressure, turbulent kinetic energy, and specific dissipation rate) are invariant with respect to the coordinate rotation. When such variables are copied from the interior domain of first block to halo cells of second block and vice versa, they remain unchanged. Moreover, all vector quantities (velocity or gradients of scalars) need to be transformed when data exchange between two periodic blocks takes place. The rotation matrix of transformation is the following:

$$\tilde{R} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.62}$$

The angle between the two periodic boundaries is positive in the counter–clockwise direction. The assumption is made based on the usage of right–handed coordinate system. The velocity vector transformation from periodic *boundary 1* ($\underline{\mathrm{u}}_{PB1}$) to periodic *boundary 2* ($\underline{\mathrm{u}}_{PB2}$) would therefore be the following:

$$\underline{\mathrm{u}}_{PB2} = \underline{\mathrm{u}}_{PB1} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.63}$$

The multi–frequency periodicity boundary condition allows to simulate single blade sector even for the unsteady flows in frequency–domain [107]. Hence, computational saving grows linearly with number of blades. This allows one to significantly reduce the size and cost of the computation. Periodic flows in turbomachinery applications satisfy a certain spatial periodicity in addition to temporal periodicity, meaning that the flow about one blade is the same as the flow about neighbouring blade with a time shift. Let us consider the rotor with 3 blades ($N_b = 3$) at time t and time $t + \Delta t$, depicted in Fig. 3.8, where:

$$\Delta t = \frac{T}{N_b}. \tag{3.64}$$

T represents total time required for one revolution. Rotor sees one revolution periodic perturbation. Blade 2 experiences same flow conditions of blade 3 every Δt time units, therefore, one can write the following condition:

$$\mathbf{u}(\theta + \theta_0, t) = \mathbf{u}(\theta, t + \Delta t), \tag{3.65}$$

where

$$\theta_0 = \frac{2\pi}{N_b}. \tag{3.66}$$

Assume that $\mathbf{u}(\theta, t)$ is expressed as a truncated Fourier series, same as Eqn. (2.64):

$$\mathbf{u}(\theta, t) = \hat{\mathbf{u}}_0(\theta) + \sum_{n=1}^{N_H} \left(\hat{\mathbf{u}}_{2n-1}(\theta) cos(n\omega t) + \hat{\mathbf{u}}_{2n}(\theta) sin(n\omega t)\right). \tag{3.67}$$

These boundary conditions must be applied in the frequency–domain. Once applied, HB solution Fourier coefficient variable $\hat{\mathbf{u}}$ is then transformed along the periodic boundaries, using Eqn. (2.74), to find the time–domain representation of HB Fourier coefficient $\tilde{u}$. The boundary condition that needs to be applied in frequency–domain is the following:

$$\begin{aligned} \hat{\mathbf{u}}_0(\theta + \theta_0) &= \hat{\mathbf{u}}_0(\theta), \\ \hat{\mathbf{u}}_{2n-1}(\theta + \theta_0) &= \hat{\mathbf{u}}_{2n-1}(\theta)\cos(n\theta_0) + \hat{\mathbf{u}}_{2n}(\theta)\sin(n\theta_0), \\ \hat{\mathbf{u}}_{2n}(\theta + \theta_0) &= -\hat{\mathbf{u}}_{2n-1}(\theta)\sin(n\theta_0) + \hat{\mathbf{u}}_{2n}(\theta)\cos(n\theta_0). \end{aligned} \tag{3.68}$$

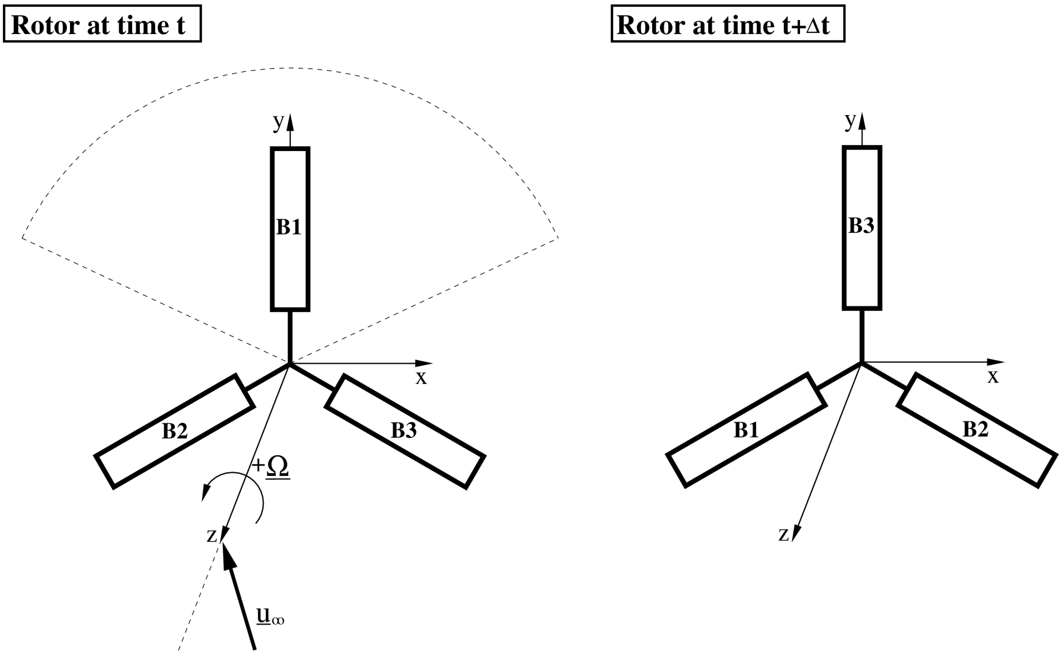


Figure 3.8: Rotor with 3 blades at time t and Δt.

3.5 Fully coupled integration of steady equations

After all spatial discretization has been done, a numerical solution algorithm converts the system of PDEs to a much larger coupled system of nonlinear algebraic equations. This system needs to be solved. For the steady problems it may be written in the form:

$$\mathbf{R_{\Phi}}(\mathbf{Q}) = 0, \tag{3.69}$$

where the array $\mathbf{R}_\Phi$ represents the residual, and the array $\mathbf{Q}$ unknown flow variables. The array $\mathbf{Q}$ is defined at the N_{cell} cell centers. It can be viewed as N_{cell} subarrays, where each subarray stores the N_{PDE} flow unknowns at a given cell center, resulting in its length $(N_{PDE} \times N_{cell})$. The array $\mathbf{R}_\Phi$ which stores the cell residuals, has the same structure as $\mathbf{Q}$. The N_{PDE} residuals for each cell are obtained by adding the convective fluxes $\mathbf{\Phi}^*_{c,f}$, the diffusive fluxes $\mathbf{\Phi}^*_{d,f}$ and the associated source terms.

The system of nonlinear algebraic equations above can be solved by time–stepping numerical method. In general, when solving the steady governing equations, the time–derivative terms are omitted, as represented by Eqn. (3.69). However, when solving the steady equations, the time–derivatives of the unknown flow variables may be re–introduced in order to use the time–stepping method as an effective iterative method.

One of the popular multistage time–stepping methods to solve RANS equations is an explicit Runge–Kutta method [108], where several stages are used to advance the solution and the residual is evaluated at intermediate stages. Runge–Kutta coefficients are used to weigh the residual at each stage. Such time–stepping scheme may be employed together with any spatial discretization scheme. Even though Runge–Kutta method is explicit, it features large stability bound at the expense of increased cost per time step. It may therefore be considered to have some characteristics of implicit schemes. Implicit character is further obtained by the convergence acceleration techniques such as local time stepping, implicit residual smoothing and multigrid, which are explained in Appendix B. Most commonly, this behaviour is associated with increased transfer of information across the numerical grid during a pseudo–time step, which is a characteristic of implicit methods.

For this work, the RANS and the turbulence closure equations are solved with a time–marching algorithm using the so–called strongly coupled approach [109, 110, 111], where the two sets of equations are time–marched simultaneously. The unknown flow vector $\mathbf{Q}$ is solved iteratively by computing Eqn. (3.69). Re–introduced time–derivative, which may be called a fictitious time–derivative, is added to the system of equations, given by Eqn. (3.69), and is premultiplied by the cell volumes:

$$V\frac{\partial \mathbf{Q}}{\partial \tau} + \mathbf{R}_{\mathbf{\Phi}}(\mathbf{Q}) = 0, \tag{3.70}$$

where τ denotes fictitious time and V are the grid cells volumes. V is the diagonal matrix, which may be viewed as a block–diagonal matrix of size $(N_{cell} \times N_{cell})$ with each block being the identity matrix of size $(N_{PDE} \times N_{PDE})$ multiplied by the volume of the cell the block refers to. Fictitious time–derivative is discretized with a four–stage Runge–Kutta scheme, where the numerical solution is marched in pseudo–time until the steady–state solution is achieved. The acceleration methods may greatly enhance the convergence rate and improve the stability of Runge–Kutta scheme. Therefore, local time–stepping, variable–coefficient central implicit residual smoothing and a full–approximation scheme multigrid algorithm were all used for this work.

Steady Runge–Kutta smoother reads:

$$\begin{array}{lcl} \mathbf{W}^0 & = & \mathbf{Q}_l \\ \mathbf{W}^m & = & \mathbf{W}^0 - \alpha_m \Delta\tau V^{-1} L_{IRS}[\mathbf{R}_{\Phi}(\mathbf{W}^{m-1}) + \mathbf{f}_{MG}] \\ \mathbf{Q}_{l+1} & = & \mathbf{W}^{NS}, \end{array} \tag{3.71}$$

where $\Delta\tau$ denotes the local pseudo–time–step, l is the Runge–Kutta cycle counter, and m the Runge–Kutta stage index. α_m represents the m^{th} Runge–Kutta coefficient. The symbol L_{IRS} denotes the implicit residual smoothing operator, and $\mathbf{f}_{MG}$ denotes the multigrid forcing function, which is non–zero when the smoother (3.71) is used on a coarse level after a restriction step.

When solving the RANS and $k-\omega$ SST equations with the smoother defined by Eqn. (3.71) the convergence rate of the solution process is very poor. This is due to the stiffness of the iterative operator caused by the large negative source terms of the turbulence model, particularly $-D_K$ and $-D_\omega$, given by Eqn. (2.51). To overcome this problem, a semi-implicit integration strategy may be used. This is done by evaluating the negative source terms of the turbulence equations at stage m rather than at stage $m-1$. This approach was adopted in [109] in order to develop an efficient strongly coupled MG iteration for the compressible RANS equations coupled with the standard $k-\omega$ model. Authors of [109] also treat implicitly the negative source term proportional to $-\nabla \cdot \underline{v}$ when the velocity divergence is positive. The turbulent multigrid integration implemented for this work, uses a similar approach but also presents important differences with the adopted approach of [109]. The differences are primarily due to the modelling differences of the standard $k-\omega$ and the $k-\omega$ SST models. The negative source terms of the k and ω equations which are treated implicitly in the Runge–Kutta smoother, lead to the following two iterations to update ρk and $\rho\omega$, respectively:

$$\begin{aligned}(1+\alpha_m\Delta\tau\Delta^+)(\rho k)^m = (\rho k)^0 + \alpha_m\Delta\tau\{\Delta^+(\rho k)^{m-1} - \beta^*[(\rho k\omega)^m - (\rho k\omega)^{m-1}] \\ - V^{-1}R_k(\mathbf{W}^{m-1})\},\end{aligned} \tag{3.72}$$

$$\begin{aligned}(\rho\omega)^m = (\rho\omega)^0 - \alpha_m\Delta\tau\{\frac{\gamma\Delta^+}{\nu_T}[(\rho k)^m - (\rho k)^{m-1}] + \beta[(\rho\omega^2)^m - (\rho\omega^2)^{m-1}] \\ + V^{-1}R_\omega(\mathbf{W}^{m-1})\}.\end{aligned} \tag{3.73}$$

Please note, the implicit residual smoothing operator and the multigrid forcing term have been temporarily omitted in these two equations for simplicity. The symbols R_k and R_ω denote the complete cell residual arrays of ρk and $\rho\omega$ respectively, and $\Delta^+ = \max(0, \frac{2}{3}\nabla \cdot \underline{v})$. Equations (3.72) and (3.73) form a system of two quadratic equations in the unknowns $(\rho k)^m$ and $(\rho\omega)^m$. Linearising the quadratic terms, a system of two linear equations in the unknowns $(\rho k)^m$ and $(\rho\omega)^m$ is obtained. The Runge–Kutta smoother for updating both the RANS and the $k - \omega SST$ variables can be written as:

$$\begin{array}{rcl} \mathbf{W}^0 & = & \mathbf{Q}_l \\ (I + \alpha_m \Delta\tau A)\mathbf{W}^m & = & \mathbf{W}^0 + \alpha_m \Delta\tau A \mathbf{W}^{m-1} \\ & & -\alpha_m \Delta\tau V^{-1} L_{IRS}[\mathbf{R}_\Phi(\mathbf{W}^{m-1}) + \mathbf{f}_{MG}] \\ \mathbf{Q}_{l+1} & = & \mathbf{W}^{NS}. \end{array} \tag{3.74}$$

The matrix A has size $(N_{cell} \times N_{cell})$ and is block–diagonal. The only non–zero entry of each $(N_{PDE} \times N_{PDE})$ block on the diagonal of A are those of the bottom right (2×2) partition. The expression of matrix A is as follows:

$$A(6:7, 6:7, i, j, k) = \begin{bmatrix} (\Delta^+ + \beta^* \omega) & \beta^* k \\ 0 & \gamma\Delta^+ + 2\beta\omega \end{bmatrix}. \tag{3.75}$$

Matrix A is upper triangular, and as a consequence $\rho\omega$ equation can avoid costly matrix inversions by updating $\rho\omega$ and ρk equations in succession. Further detail on this aspect is described in [26].

During the update process of Runge–Kutta smoother, ω is limited with a minimum threshold value. This value is based on the production term P_d of Eqn. (2.50), as in [109]. In this case unphysically low values of ω can be avoided. Furthermore, when using multigrid (section B.0.3), the residuals of the ω equation are limited before being restricted to a coarser grid as proposed in [109]. The turbulent eddy viscosity is computed on each grid level, whereas the production terms P_d and Δ^+ are computed only on the finest grid level and are restricted on the coarser levels by the restriction operator. In multigrid, flow solution is also updated by the prolongation operator, therefore the prolongations of k and ω corrections are limited as proposed in [111].

3.6 Fully coupled integration of time–domain equations

Typical technique to solve large coupled system of nonlinear algebraic equations for unsteady flows is called dual time–stepping, introduced in [112]. Using this method, the physical time–derivative term which appears in URANS equations, regardless the frame of reference (2.43) or (2.55), is discretized using a second order backward finite–difference. Moreover, at each new physical time–level $n + 1$, the whole set of nonlinear algebraic equations is solved using an explicit approach, with the same integration technique as used for steady problems, where a fictitious time–derivative has been introduced. The second order backward finite difference discretization of the physical time–derivative of the unknown flow state, and the introduction of the fictitious time derivative $(d\mathbf{Q}/d\tau)$, yield the following expression:

$$V\frac{d\mathbf{Q}^{n+1}}{d\tau} + V\frac{3\mathbf{Q}^{n+1} - 4\mathbf{Q}^n + \mathbf{Q}^{n-1}}{2\Delta t} + \mathbf{R}_\Phi(\mathbf{Q}^{n+1}) = 0. \tag{3.76}$$

The array $\mathbf{Q}$ which stores the unknown flow variables, can be viewed as made up of N_{cell} subarrays, each of which stores the N_{PDE} flow unknowns at each particular physical time. Its overall length is $(N_{PDE} \times N_{cell})$. The array of the cell residuals $\mathbf{R}_\Phi$ has the same structure as $\mathbf{Q}$. The diagonal matrix V, which stores the volumes of the grid cells, may be viewed as a block–diagonal matrix of size $(N_{cell} \times N_{cell})$, with each block being the identity matrix of size $(N_{PDE} \times N_{PDE})$ multiplied by the volume of the cell the block refers to. Please note that the matrix V is independent of the physical time–level denoted by the superscripts $^{n+1}$,^{n} and $^{n-1}$, as for this work only rigid–body grid motion has been considered. The symbol Δt denotes the user–given physical time–step. Equation (3.76) can therefore be viewed as a system of $(N_{PDE} \times N_{cell})$ ordinary differential equations in which the unknown is $\mathbf{Q}^{n+1}$, the flow state at time–level $n+1$. The calculation of $\mathbf{Q}^{n+1}$ is performed in the same manner as for the steady problems, iteratively by discretizing the fictitious time–derivative with a four–stage Runge–Kutta scheme, and marching the equations in pseudo–time until a steady state is achieved. Such steady–state is the flow solution for each particular physical time step. Also, for this case the convergence rate is greatly enhanced by using local time–stepping, variable–coefficient central implicit residual smoothing and a full–approximation scheme multigrid algorithm.

When the physical time–step Δt becomes significantly smaller than the pseudo time–step $\Delta\tau$, the solution procedure may become unstable. This was reported in [113], and thoroughly investigated in [114]. In [114] it was demonstrated that the instability is caused by the $\mathbf{Q}^{n+1}$ term. The study [114] elegantly solves the stability problem by treating the $\mathbf{Q}^{n+1}$ term of the physical time–derivative within the Runge–Kutta integration process implicitly. This strategy has also been adopted for this work. Multi–stage Runge–Kutta scheme, which discretizes the fictitious time–derivative of Eqn. (3.76) and considers the $\mathbf{Q}^{n+1}$ term at stage m rather than at stage $(m-1)$ yields the following unsteady Runge–Kutta smoother:

$$\begin{aligned} \mathbf{W}^0 &= \mathbf{Q}_l \\ [I + \alpha_m \Delta\tau(1.5/\Delta t I + A)]\,\mathbf{W}^m &= \mathbf{W}^0 - \alpha_m \Delta\tau V^{-1}[\mathbf{R}_\Phi(\mathbf{W}^{m-1}) \\ &\quad +V(-2\mathbf{Q}^n + 0.5\mathbf{Q}^{n-1})/\Delta t] \\ \mathbf{Q}_{l+1} &= \mathbf{W}^{NS}. \end{aligned} \tag{3.77}$$

As for steady Runge–Kutta smoother, m is the Runge–Kutta stage index, α_m is the m^{th} Runge–Kutta coefficient and l is the Runge–Kutta cycle counter. The symbol $\mathbf{Q}_l$ is the shorthand for $\mathbf{Q}_l^{n+1}$. As shown by [114], the stability of algorithm (3.77) is no longer dependent on the ratio $\Delta\tau/\Delta t$. However, the formulation above is still unsuitable when also implicit residual smoothing and multigrid acceleration techniques are employed. Both techniques would have to be applied to a residual term that vanishes at convergence, which is presently not the case. This can be solved by modifying the algorithm in the following way:

$$\begin{aligned} \mathbf{W}^0 &= \mathbf{Q}_l \\ [I + \alpha_m \Delta\tau(1.5/\Delta t I + A)]\,\mathbf{W}^m &= \mathbf{W}^0 + \alpha_m \Delta\tau(1.5/\Delta t I + A)\mathbf{W}^{m-1} \\ -\alpha_m \Delta\tau V^{-1} L_{IRS}[\mathbf{R}_\Phi(\mathbf{W}^{m-1}) &+ V(1.5\mathbf{W}^{m-1} - 2\mathbf{Q}^n + 0.5\mathbf{Q}^{n-1})/\Delta t + \mathbf{f}_{MG}] \\ \mathbf{Q}_{l+1} &= \mathbf{W}^{NS}. \end{aligned} \tag{3.78}$$

where L_{IRS} denotes the implicit residual smoothing operator, and $\mathbf{f}_{MG}$ is the multigrid forcing function. The algorithm given by Eqn. (3.78) is based on a point–implicit Runge–Kutta (PIRK) integration of the time–dependent NS equations. By setting $\Delta\tau/\Delta t = 0$, the standard fully explicit Runge–Kutta (FERK) integration method is retrieved. The turbulent PIRK integration significantly improves the stability of the fully coupled integration approach. Therefore, much larger CFL numbers may be selected than with the standard FERK integration. Consequently, this yields significant reductions of runtimes.

3.7 Fully coupled integration of harmonic balance equations

The major difference of HB RANS equations with respect to the TD RANS equations, is that the physical time–derivative of the TD equations is replaced by a volumometric source term, which is proportional to ω. Such set of nonlinear algebraic equations, resulting from the space–discretization of the system of equations 2.81, may be solved with the same technique as used for steady and time dependent problems.

The general pattern of series of computations may be written as:

```
do ih = 0, nharms
  do k = 1, ncellk
    do j = 1, ncellj
      do i = 1, ncelli
        Perform computations.
      end do
    end do
  end do
end do
```

`ncelli`, `ncellj` and `ncellk` respectively represent the number of cells in the i, j and k directions and `nharms` is the number of complex harmonics. When steady and TD solvers are used the `nharms` is zero, whereas, when HB solver is used, the `nharms` is larger than zero. Therefore, `ih` represents the flow field snapshot index which adds an extra loop layer to routines of the HB solver, increasing the number of dimensions of most arrays by one. For some arrays an extra dimension can be avoided in order to spare some memory. Four additional arrays need to be allocated to calculate and store matrix D defined by Eqn. (2.80), which is done only at the beginning of the calculation. Moreover, an array is also allocated to store the HB source term defined below. This source term represents the so–called HB overhead. The memory footprint of the HB solver is thus about $2N_H + 1$ times that of the steady solver. Each flow field snapshot uses about the same amount of memory as the steady solver.

Steady Runge–Kutta smoother, used for computing the sought HB flow solution $\mathbf{Q}_H$, solves the system of algebraic equations:

$$\mathbf{R}_{gH}(\mathbf{Q}_H) = \omega V_H D_H \mathbf{Q}_H + \mathbf{R}_{\Phi H}(\mathbf{Q}_H) = 0. \tag{3.79}$$

The array $\mathbf{Q}_H$, which stores the unknown flow variables, is constructed of N_{cell} sets of $(2N_H + 1)$ flow states. Each flow state refers to the physical times, defined as equally spaced points over one period, given by Eqn. (2.71). Therefore, the array $\mathbf{Q}_H$ can be written in the following manner:

$$\mathbf{Q}_H = [\mathbf{Q}'_1 \, \mathbf{Q}'_2 \ldots \mathbf{Q}'_{N_{cell}}]', \tag{3.80}$$

where the array $\mathbf{Q}_i$, with $i = 1, N_{cell}$, has the length $[N_{PDE} \times (2N_H + 1)]$. The elements of $\mathbf{Q}_i$ array contain the flow state at various positions of the equally spaced points over one period. Therefore, the first N_{PDE} corresponds to the beginning of period $t = t_0$, and the last N_{PDE} elements contain the flow state at $t = t_{2n_H+1}$, all the other N_{PDE} elements are in between these two points. The structure of the arrays $\mathbf{R}_{gH}$ and $\mathbf{R}_{\Phi H}$ is the same as that of $\mathbf{Q}_H$. A subarray $(\mathbf{R}_\Phi)_i$ is made up of a $(2N_H + 1)$ states of the cell residuals associated with the convective and diffusive fluxes, and the turbulent source terms, at the equally spaced physical times given by Eqn. (2.71). The residual subarray $(\mathbf{R}_g)_i$ includes also the source term $\omega V_i D\mathbf{Q}_i$, where V_i is the product of the volume of the i^{th} grid cell and $I_{N_{eqs}}$, the identity matrix of size $(N_{eqs} \times N_{eqs})$ with $N_{eqs} = [N_{PDE} \times (2N_H + 1)]$. The diagonal matrix V_H is a block–diagonal matrix with blocks given by the matrices V_i defined above, and the block–diagonal matrix D_H is defined as $D_H = I_{N_{cell}} \otimes D$.

The HB–counterpart of the turbulent steady smoother 3.74 is as follows:

$$\begin{aligned}
\mathbf{W}_H^0 &= (\mathbf{Q}_H)_l \\
[I + \alpha_m \Delta\tau_H A_H]\,\mathbf{W}_H^m &= \mathbf{W}_H^0 + \alpha_m \Delta\tau_H A_H \mathbf{W}_H^{m-1} - \\
&\quad \alpha_m \Delta\tau_H V_H^{-1} L_{IRS,H} [\mathbf{R}_{gH}(\mathbf{W}_H^{m-1}) + \mathbf{f}_{MG,H}]\,, \qquad m = 1, M \\
(\mathbf{Q}_H)_{l+1} &= \mathbf{W}_H^M .
\end{aligned} \tag{3.81}$$

There is one major difference with respect to steady smoother, the N_{cell} subarrays of pseudo time $\Delta\tau_H$ have now the length $(2N_H + 1)$ and they all contain the local time–steps for the $(2N_H + 1)$ flow states. The structure of the HB multigrid forcing term $\mathbf{f}_{MG,H}$ is the same as that of $\mathbf{Q}_H$. The matrix A_H may be viewed as a block–diagonal matrix of the dimensions $(N_{cell} \times N_{cell})$, where each block $A_{H,i}$ holds the same block–diagonal structure. The size of $A_{H,i}$ is $(N_{eqs} \times N_{eqs})$ where $N_{eqs} = [N_{PDE} \times (2N_H + 1)]$. $(2N_H + 1)$ non–zero $(N_{PDE} \times N_{PDE})$–entries of matrix A_H constructs the matrices $A_{H,i}$ for each particular flow state, referring to the specific time of the period, given by Eqn. (2.71). The structure of each block $A_{H,i}$ is the same as that of the matrix A in the turbulent steady smoother defined by Eqn. (3.74). Implicit residual smoothing HB operator $L_{IRS,H}$ features the same block structure as that of A_H.

When using the expression of Eqn. (3.75) for the update of ρk and $\rho\omega$ equations, the matrix premultiplying the term $\mathbf{W}_H^m$ at the second line of algorithm defined by Eqn. (3.81) has such structure, that for each grid cell, the update of the $[N_{PDE} \times (2N_H + 1)]$ unknowns does not require any matrix inversion. It is also noted that algorithm defined by Eqn. (3.81) uses a FERK treatment of the HB source term $\Omega V_H D_H \mathbf{Q}_H$. Based on the observations of the PIRK over the FERK integration for turbulent TD flow problems solved with the RANS and SST equations [26], it is anticipated that larger CFL numbers may be selected, when a point–implicit treatment of the HB source term [55] is used, thus further increasing the convergence rate of the HB equations. The HB PIRK integration, however, significantly increases the computational cost of each Runge–Kutta stage, because, for each grid cell, the update of the $[N_{PDE} \times (2N_H + 1)]$ unknowns requires the inversion of a matrix of size $[(2N_H + 1) \times (2N_H + 1)]$. The benefit of this approach depends on whether higher CFL numbers would mean faster convergence, and if this outweights the additional burden of the matrix inversion. It depends on case to case basis, whether the use of the HB FERK or PIRK integration is more beneficial. For all simulations reported in this thesis, the HB PIRK integration did not enable the use of CFL numbers higher than those used by the HB FERK approach of algorithm defined by Eqn. (3.81), therefore, only HB FERK approach has been used.

It should also be highlighted that the ratio of the computational cost of one HB FERK multigrid cycle and that of one steady multigrid cycle grows in a slightly superlinear fashion with N_H, due to construction of the HB source term $V_H D_H \mathbf{Q}_H$. This overhead, however, remains relatively small for low values of N_H, as highlighted in the numerical tests provided in subsection 6.2.5.

Chapter 4

Code parallelization

This chapter explains the details of the parallelism used by the COSA CFD code. The chapter begins with an outline of the general definition of High–Performance Computing and its role in CFD. The definition of parallelism, a brief description of different types of HPC machines, and parallel programming approaches are also provided. It continues with a general description of the MPI standard, and explains its usage in COSA. Furthermore, it also presents the general pattern of serial computations of COSA, and the most straightforward way to parallelize such pattern when using MPI. The assignment strategy of decomposed domain to MPI processes is explained, and the advantages and disadvantages of present approach are addressed. Point to point and collective communications are then described, as well as blocking or non–blocking communication protocols. It is precisely explained which MPI library routines are used for communications in COSA implementation. The chapter continues with the description of parallel MPI input/output capabilities and its role in COSA. Herein, different formats used for the input and output files of COSA are also described. The chapter is concluded with an outline of the two approaches for measuring parallel scalability (weak and strong scaling). Following the description of the strong parallel scalability approach, the results of such analysis, performed on the newly developed HB solver of COSA, are reported.

The parallel implementation of the 2D version of COSA has been primarily done at EPCC at the University of Edinburgh, therefore, the structure of the newly developed 3D solvers, by the author of this work, resembles the 2D version. All approaches that have been described in the following sections apply also for the 2D version of COSA. Parallel MPI input/output has also been restructured in this work, and is a shared contribution between EPCC and the author.

4.1 High Performance Computing

High–performance computing (HPC) generally refers to the development and use of supercomputers, parallel processing algorithms, and related software. It plays an important role in many aspects of modern society, and is without doubt key in enabling fast technological growth of already technologically advanced countries. HPC applications are widely used in numerous areas of academic, industrial and government sectors. In very simple terms, HPC is required to perform computations faster than using a single processor, by simultaneously solving a problem on multiple processors. It is driven by the need to obtain faster solutions, and therefore to tackle larger and more complex scientific and engineering problems, which can not be solved without parallelism. CFD represents one of the areas, where massive parallelism is a must in order to simulate and explore complex fluid dynamics problems. The rise in performance and capability of HPC machines in the past two decades has gone hand in hand with a rapid development of CFD technology. Figure 4.1 depicts exponential growth of supercomputing power of the most powerful commercially available computer system in the world (#1), the 500th most powerful commercially available computer system (#500) and the sum of supercomputing power for 500 most powerful commercially available computer systems [7]. The symbols represent the actual growth of the computing power, whereas the solid lines represent the future projections. The present most powerful HPC machines are delivering performance in the petaflop range, and according to the projections, the exascale range may be reached even before 2020, as shown in Fig. 4.1. The fast growth of computing power has enabled RANS CFD to be used as an accurate and fast high–fidelity design tool for various engineering applications, in order expensive experimental testing could be minimised. If the growth of computational power will continue with the same pace as in the past two decades, it will enable to deal with many complex fluid dynamics problems, which are currently thought to be impossible to solve with CFD on current HPC machines [115].

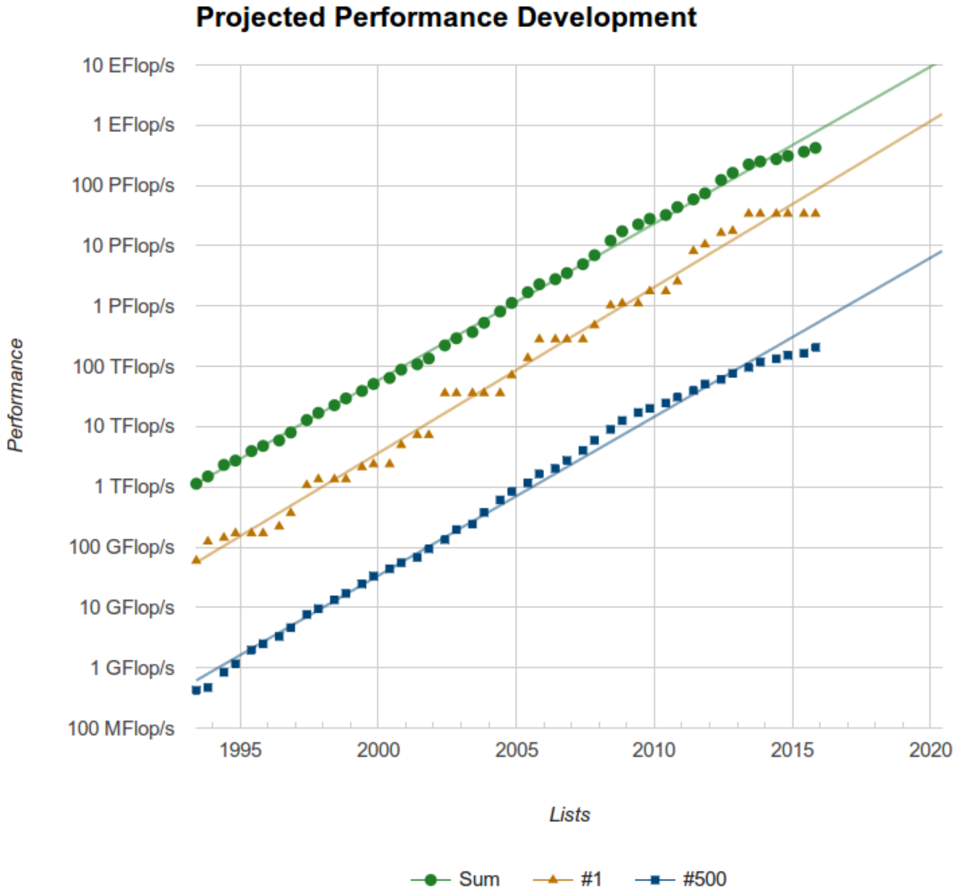


Figure 4.1: Growth of supercomputing power. Taken from [7].

Parallelism can be divided into two different types. The first is called instruction level parallelism, which is outside explicit user control. It is based on the compiler and the central processing unit (CPU) decisions, which instructions may be processed simultaneously. The second is the sort of parallelism, where multiple instructions are handled by multiple processors and usually explicitly scheduled by the user. In CFD we are particularly interested in the latter type of parallelism, the hardware that can be used with it and the programming instructions that enables it. There are two major reasons why to use parallelism. Using single core calculations for CFD applications is usually too slow to perform the required tasks in a reasonable amount of time, parallelism is used to obtain higher execution speeds, and thus reduce runtimes. Furthermore, usually when handling complex, computationally large problems, the available memory of a single system may be insufficient, therefore, parallelism enables access to more memory [116].

HPC machines, sometimes called supercomputers, can be referred to parallel computers, which are machines with more than one CPU that can be set to work on the same problem. Michael Flynn taxonomy [117] describes the amount of simultaneous control and data streams present in a parallel architecture. There are two dominating concepts in HPC today, the single instruction multiple data (SIMD) and multiple instruction multiple data (MIMD). In SIMD concept, multiple processing units work on the same instruction (same operation), but each operates on its own data item. Conceptually, the data to be operated on must be distributed over multiple processing units. All the processing units must be instructed to perform the same operations on their piece of the data. The performance improvements are obtained by operating single instruction on multiple pieces of data at once by multiple processing units. In SIMD concept, each processor is controlled by a single central control unit. Typical examples of SIMD concept are vector processors and graphics processing units (GPUs). In MIMD concept, multiple processors function asynchronously and independently. Different processors can simultaneously operate independent application programs. This enables one to have more freedom in the undertaken parallelisations, distributing the data as in the SIMD approach, or distributing separate work tasks, as required by the problem being tackled. Typical examples for MIMD paradigm are the shared memory and distributed memory parallel computers. Shared memory parallel computer consists of shared physical address space, which may be simultaneously accessed by all CPUs of that computer. There are two types of shared memory systems, uniform memory access (UMA) machines, where all physical memory locations are shared uniformly between all processors. Second type is called on cache–coherent nonuniform memory access (NUMA), where memory is physically distributed but logically shared. The memory access in this case is time dependent on the memory location relative to the processor. Its own local memory may be accessed much faster than non–local memory. Distributed memory parallel computers consist of a number of parallel computers, which are all interconnected with a network system. HPC machines often use an advanced network such as Gigabit Ethernet or InfiniBand. It is currently a common practice to build large distributed memory parallel computers, using small shared memory parallel computers. In this case, the individual nodes of the distributed memory system are featuring a hybrid parallel system, were each node is a shared memory computer, but the whole system is a distributed parallel machine [115, 118].

Parallel programming, with respect to serial programming, introduces additional sources of complexity. It is largely dependent on software tools and environments. Furthermore, additional problems such as communication, data partitioning and distribution, load–balancing, deadlocks, etc. are introduced. There are two main parallelisation approaches. The first approach is based on implicit parallelism, adopted by parallel languages (e.g. SISAL [119], PCN [120]) and parallelizing compilers. In this approach user does not specify compiler instructions and control the calculations and data distribution. Second approach is based on explicit parallelism, where the programmer is responsible for most of the parallelization such as task decomposition, mapping tasks to processors, and communication structure. Much better efficiency of parallel programming is obtained when using explicit parallelism, but it can be quite a challenging task. Fairly simple and commonly used approach of explicit parallelism is to write a set of compiler directives inside a sequential program, which indicate the places where the program can be parallelized, and how the data should be distributed. This is done in OpenMP [121], which is an extension to the programming languages C/C++ and Fortran. Its main objective is the parallel execution of loops. OpenMP works best with shared memory approach. Much more complex to program, but with the ability to create programs that will run efficient on both shared memory and distributed memory parallel computers, are the approaches used in distributed memory programming. One of the most important is Message Passing Interface (MPI) [122], which is a standardised and portable message–passing system for C/C++ and Fortran. As the vast majority of CFD programs require massive parallelism, MPI has been the primary method for parallelising CFD codes. As new HPC machine architectures also feature multi–cores and more levels of memory hierarchy, also hybrid parallelization approach which uses MPI and OpenMP is quite commonly used [116, 123, 124, 125].

4.2 Parallel computing with Message Passing Interface

MPI is a parallel programming paradigm, in which data is moved from the address space of one process to the address space of another process using communication among processes. MPI is not a programming language or an implementation, it is a specification for a library interface, for C/C++ and Fortran programming languages. All operations are expressed as functions, subroutines, or methods, according to the appropriate language bindings. The major goal of MPI is a degree of portability across a variety of HPC machines, and to achieve high performance on any of the platforms. One of the challenges of MPI was also to design an application programming interface (API), which allows an efficient communication without excessive memory to memory copying, and overlap of communication and computation. Furthermore, MPI offers reliable communication interface, where communication failures are resolved by the underlying communication subsystem. The semantics of the interface is language independent, offering convenient C/C++ and Fortran bindings for the interface. MPI applications can run on both distributed memory systems and the shared memory systems using exactly the same implementation [122, 126].

Any MPI program starts with the call to the MPI library routine `MPI_INIT`, which initialises the MPI execution environment, and completes with `MPI_FINALIZE`, which terminates MPI execution environment. All library routines from MPI must be executed between these two calls. Following the call to `MPI_INIT`, there is usually call to `MPI_COMM_SIZE`, the operation which queries the number of MPI processes currently running the program, and to `MPI_COMM_RANK`, which gives a unique identifier (or rank) of the particular MPI process. The processes executed in the MPI model have separate address spaces. Communication between two different processes works in such way that a portion of first process's address space is copied into second process's address space. More precisely, first process execute a send operation and the second process executes a receive operation. For this communication to be successful the minimum amount of information that needs to be specified by the sending process, is the data to be communicated and the destination (receiver rank) to send this data to. In very simple terms, the data can be described by a starting address and length in the memory of the sending process. The minimum required information on the receiver's side is the address, space in local memory where the received data is to be placed, and the identifier (rank) of the sender, so the receiving process can identify the process which sent the message. The minimum specified information might be adequate for some simple implementations, however, usually additional features also need to be specified. One important additional parameter is the tag (a number identifying the message) of the message, which allows a process to control which message it receives, thus, the receive operation specifying a tag will complete successfully only when a sent message with a matching tag arrives. Furthermore, the source or destination rank may also be specified on a receive operation as an additional screening parameter, and the actual length of the message received [127].

Let us examine the required information for send and receive operations more precisely. Firstly, as previously said, message buffers can be in very simple terms specified by the address and length, however, additional constraints must be taken into account. The message to be sent is usually not contiguous (stored linearly in memory), and the information content of a message needs to be independent of how these values are represented in various computers as collections of bits. MPI handles these constraints in such way, that the message buffer is defined by a triple information, address, count and datatype. Count describes number of elements of particular datatype (single precision real, double precision real, etc.) in buffer starting at particular address. This offers great flexibility in the values of datatype. Secondly, families of massages may be separated by the tag information for both send and receive operations. This gives the program the possibility to match messages in an ordered way. All the messages which do not have the desired tag are queued, until the program is ready to accept them. Finally, processes which belong to groups (called communicators in MPI) must be named. The processes within each group are identified by ranks. In an MPI implementation all processes belong to an initial group. If a group contains n processes, then the ranks are contiguous integers starting from 0 to $n-1$. The notions of context and group are combined in a communicator. Thus the source specified in a send or receive operation always refers to the rank of the process in the group identified with the given communicator. Let us now introduce basic send operation:

```
MPI_SEND(address, count, datatype, destination, tag, comm)
```

where the entries `(address, count, datatype)` describe `count` occurrences of items of the form `datatype` starting at the data `address`. `destination` denotes the rank of the process to send the data to, whereas `tag` is an integer used for message sorting, and `comm` identifies a group of processes and a communication context. The basic receive operation reads:

```
MPI_RECV(address, count, datatype, source, tag, comm, status)
```

where the additional argument `status` contains information about the actual message size, source and tag (filled in when the message is received by the MPI routine). `source` is the rank of the source of the message. Basic send and receive operations described above are most fundamental feature of MPI standard. However, the MPI standard introduces many additional advanced features. One of them is collective communications, which are performed by all the processes in a communicator. They are used either for the data movement operations, which rearrange the data among various processes, or for collective computation operations, such as maximum, minimum, sum, etc. Next, the MPI standard also features virtual topologies for convenience in programming, debugging and profiling mechanisms, multiple communication modes which include blocking and non blocking communications, support for parallel libraries which are completely independent of serial code and are inter–operable with other libraries. MPI can also run on heterogeneous networks, where the HPC machines may use different lengths and formats for various fundamental datatypes. Lastly, it is important to stress that not every aspect of a MPI program is specified by the MPI standard. There are many aspects which are left to the specific implementation, including process startup, which gives flexibility in how a MPI program is executed, and specifies the portability of the MPI program. Furthermore, the implementation may return many more number of error codes than specified by the MPI standard. Also, the amount of system buffering for messages and further issues, which may impact the performance, are implementation dependant [127].

4.3 Data decomposition

In the multi–block approach, the computational domain is decomposed into several smaller structured subdomains called blocks. Such grid topology represents a geometric domain decomposition and it enables a natural way for MPI parallelisation. The general pattern of serial computations of COSA within the code is the following:

```
do ib = 1, nblock
  do ih = 0, nharms
    do k = 1, ncellk
      do j = 1, ncellj
        do i = 1, ncelli
          Perform computations.
        end do
      end do
    end do
  end do
end do
```

where `nblock` represents the number of blocks of the computational domain. `ncelli`, `ncellj` and `ncellk` are respectively the number of cells in the i, j and k directions of the particular block. `nharms` is the number of complex harmonics of HB simulations defined in section 2.6. The parameter `nharms` is bigger than zero only when HB solver is used, and is zero in all other simulation types. The most straightforward way to parallelize the code with the structure represented above, is to assign each block (`ib`) to individual MPI process. Or in case where more blocks than available MPI processes exist, multiple blocks can be assigned to each MPI process. In order to map the decomposition of blocks, the algorithm to match the blocks to MPI processes for any number of blocks and MPI processes must be created. Such algorithm must also combine any global information that is required on the simulation state.

The parallelisation of the HB solver is essentially the same as that of the steady solver. There are two main differences with respect to the steady solver. The first difference is that `nharms` is bigger than zero when the HB solver is used, which means there is an extra dimension in nearly all subroutines. The second difference is that in the HB solver there is an additional HB source term, which does not exist in the steady solver. The HB source term adds two extra dimensions with respect to standard routine of the steady solver, because it consists of two nested loops of the flow field snapshots. As mentioned above, each block (`ib`) is assigned to one or multiple MPI processes, therefore, the structure of the parallel code with respect to the steady solver does not change. However, the memory of the HB solver is about $2N_H + 1$ times that of the steady solver, and about $2N_H + 1$ times the steady solver operations at one MPI process are performed.

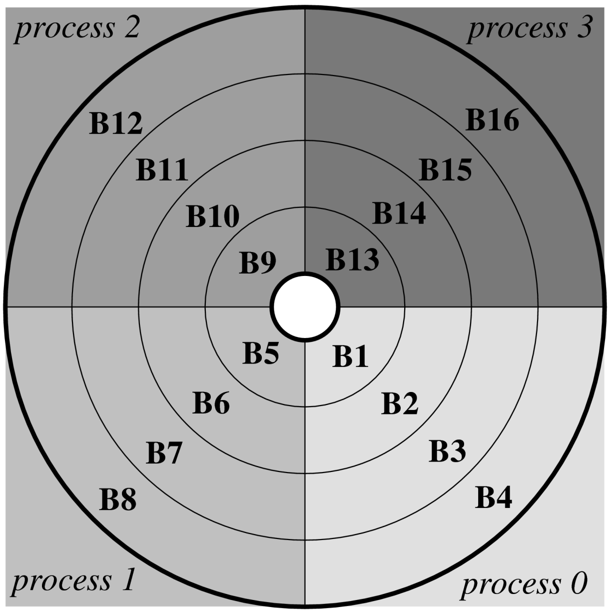


Figure 4.2: Simple representation of block assignment to MPI processes of a 2D computational grid around cylindrical body.

It should be pointed out that the parallelisation using MPI library is closely linked to the number of blocks in the computational domain, their size, and whether they are balanced. If using a small number of large–sized blocks, the parallelization cannot provide much benefit to computation. On the other hand, if using many small–sized blocks, the communication between blocks may become higher than computational load of each individual block. Therefore, the performance trade–off between block size and number of blocks must always be found. In reality, the situation in which the computational load of each individual block is lower than the communication between blocks is very unlikely to arise, as extremely large number of exceptionally small–sized blocks would be required. Presently, the assignment of blocks is done assuming individual blocks are of equal size. The existing algorithm assigns equal or as equal as possible number of blocks to each MPI process in a simple rank order, to ensure that all processes handle exactly the same or a similar number of blocks. Figure 4.2 depicts a simple 2D representation of block ordering of a grid with 16 blocks. The computational domain represents the flow past a cylindrical body, where all blocks feature equal number of cells in both directions. The innermost circle represents a body surface and the outermost circle depicts farfield boundary. If using 4 MPI processes, based on the simple rank ordering of blocks, blocks B1 to B4 are assigned to process 0, B5 to B8 to process 1, B9 to B12 to process 2, and finally B13 to B16 to process 3. If all blocks in the computational domain are of equal size, this strategy yields excellent load balance of the MPI simulation. The only drawback of this approach is that the grid generation around the complex geometries may be quite challenging, because the blocks must be balanced. For all calculations presented in this thesis, great effort has been made to perfectly balance all computational grids. If block balancing is not done, the scalability of the MPI code is restricted by the size of the biggest block of the grid. To simplify and accelerate the grid generation phase, the dynamic parallel load–balancing capability is now being developed in COSA, as a part of a different project. Load–balancing capability will assign the number of grid blocks to each MPI process according to the block size. This enables the MPI processes to use different numbers of blocks to ensure a comparable amount of work for each process.

4.4 Message Passing Interface communications

The MPI standard includes two types of communication, point to point and collective. Both types are used in the parallelisation of COSA. Regardless of its nature (point to point or collective), the communication can involve blocking or non–blocking protocols. Blocking protocol halts the program execution until the data exchange is completed, and thus the message buffer (slots in computer memory where the copy of the data is kept until it is received) is again safe for reuse. For sends, the data must be successfully sent or safely copied into the system buffer space. Whereas, for receives, the data must be safely stored in the receive buffer. In other words, the call to the send and receive functions does not return until the data transfer is finished. Non–blocking protocol, on the other hand, does not suspend the program execution, it rather initiates the send or receive operation but it does not wait for completion (e.g. message copying or the actual arrival of message). When using the non–blocking communication the programmer must ensure the buffer is free for reuse. The main purpose of the non–blocking protocols is to overlap calculation with communication in order to increase the performance of the application.

Various different message passing protocols of MPI exist, however, they are not defined by the MPI standard, instead they are defined by each individual programmer. The majority of MPI implementations use two different message protocols: a) the eager protocol, which is an asynchronous buffered approach allowing to accomplish send operation without confirmation from the receiving process. Usually it is used for small messages, as system buffer space must be preallocated. It has the advantage of synchronisation overhead being small; b) the rendezvous protocol, which is a synchronous, non–buffered approach, and requires a confirmation from the receiving process that the send operation has been successfully completed. It is usually used for large messages, as buffering the data does not make much sense due to the large amount of preallocated buffer space. In this case message envelope (message with any supplementary information, e.g. length, sender, tag, etc.) is immediately stored at the receiving process, but the actual message transfer waits until the receiving process buffer is available. Extra data copies to the message buffer are avoided, but the sender and receiver must synchronise for the data exchange to be completed [115, 118].

In point to point communication, the data is exchanged between two processes where one is sending and the other is receiving. The point to point communication operations involve a set of send and receive functions, that allow the communication of data with an associated tag. In COSA, we use point to point communications for the halo data exchange of the flow variables during each Runge–Kutta stage, and if using multigrid acceleration technique, also during each multigrid cycle. Furthermore, they are also used for the halo exchange of grid coordinates and volumes at the beginning of calculation or at the beginning of each physical time step when using dual time stepping approach. The halo data exchange of the flow variables during each Runge–Kutta stage and multigrid cycle may become computationally extremely expensive, as many messages must be sent and received. The overall computational cost of the communications can be reduced if the number of communications is reduced by increasing the size of each message. In COSA, the send and receive functionality is such that for each internal block boundary a single pair of arrays (one for each block of the pair) with all halo data is constructed. Once such pair is constructed, the non–blocking point to point communication occurs. Non–blocking communication is used to prevent the program from being slowed, as it allows to start sending and receiving several messages and to proceed with other operations. It also solves the problem of message buffering, as it waits that the receiving process provides a space for a message to be received into. For the non–blocking send operation `MPI_ISEND` routine is used, whereas for non–blocking receive operation `MPI_IRECV` is used. Let us now explain in more detail how the point to point non–blocking data exchange process in COSA routines works. Nonblocking send syntax is the following:

```
MPI_ISEND(address, count, datatype, destination, tag, comm, request).
```

The entries `(address, count, datatype)` describe `count` occurrences of items of the form `datatype` starting at the data `address`. The rank of the process is denoted by `destination` and `tag` is an integer used for message sorting. `comm` identifies a group of processes and a communication context. The argument `request` is the communication request, also known as a handle. The structure of `MPI_ISEND` is the same as for the blocking `MPI_SEND` operation, with the addition of a handle. The routine `MPI_ISEND` begins the non–blocking send operation, and it immediately returns. The handle determines whether the operation has been completed. The handle may be determined by the `MPI_WAIT` routine, which waits for a specified send to complete (identifying the specific operation to wait on using a provided handle). Once a send operation is complete, the request handle is set to `MPI_REQUEST_NULL`, which indicates a null handle. Another routine, `MPI_WAITANY`, can be used to wait on a set of handles. `MPI_WAITANY` will complete when a single communication operation associated with the group of handles has been passed. Also a `MPI_WAITALL` routine exist, which also waits on a set of pending communications using a set of handles, but this routine will only be completed, once all the operations associated with the group of handles passed to it have finished. The receiving process calls the `MPI_IRECV` routine to begin the non–blocking receive operation and, as with the send operation, it returns immediately. The syntax of non–blocking receive is the following:

```
MPI_IRECV(address, count, datatype, destination, tag, comm, request).
```

As one may notice `MPI_IRECV` routine has the same arguments as the routine `MPI_ISEND`.

In collective communication, the data is exchanged among all processes involved in a particular communicator. Collective operations may complete as soon as the call to the routine has finished, which is the case for blocking operation. Conversely a nonblocking call requires a separate completion call. Collective communication calls may use the same communicators as point to point communication. A message tag argument is not required here. In COSA, only blocking collective communication is used. It is required for calculation of distance from the closest wall for the $k-\omega$ SST turbulence closure, calculation of forces and residuals. Two MPI routines are used for all listed operations. First one is called `MPI_BCAST`, which broadcasts a message from the broadcasting process to all processes of the group. And the second required routine is called `MPI_ALLREDUCE`, which is in principle the same routine as `MPI_REDUCE`, except that the receiving data appears in the receive buffer of all the receiving processes. `MPI_REDUCE` combines the data passed to the function from all the MPI processes to one process. Additionally, also the routine `MPI_BARRIER` is used, which is one of the special collective operations. Its purpose is to ensure that every process has completed the computation. `MPI_BARRIER` routine is executed only at the end of calculation, just before MPI is finalised using `MPI_FINALIZE` routine.

4.5 Parallel input/output

The parallel input/output (I/O) capabilities of MPI represent an important aspect of the MPI standard. Historically, the I/O of almost every parallel application was using the following two approaches: a) multiple files multiple clients approach, where each process had to write into a separate file, or b) single file single client approach, where the data from all processes were gathered on a single master process, which then wrote the data into a single file. The reading of the files was done in the same manner. These two approaches were used as in most HPC machines the parallel I/O capabilities from multiple processes to a single file were not supported, or the I/O performance was poor. Furthermore, at the time when MPI standard was defined, the size of the CFD simulations was relatively small. The computational grids featured only up to hundreds of blocks, and simulations were typically run on tens of MPI processes. Therefore, if either of the two aforementioned approaches was used, this did not significantly affect the I/O performance. However, due to the sudden growth of CFD simulation size in the past two decades, the two approaches described above could significantly affect I/O performance. Multiple files multiple clients approach can cause issues when the simulation size is scaled up to a large number of MPI processes, as the data storage system may no longer be able to handle reading or writing efficiently. Moreover, when using single file single client approach, little benefit from the parallel file system is gained, as only a single process is doing all the I/O means. In the near–future, the size of the computational grids for 3D flow simulations may become extremely large, up to billions of grid cells and thousands of blocks. Therefore, it is crucially important to achieve high performance I/O. With the recent development of modern HPC systems, aforementioned limitations no longer exist. With properly configured I/O hardware and modern HPC file systems (e.g. Lustre, PVFS, GPFS), it is possible to achieve both high performance and having a single file which can be accessed by each MPI process directly (e.g. single file multiple clients or single file collective clients approaches). The I/O interface in MPI is designed to deliver high performance for various approaches of parallel I/O such as contiguous and noncontiguous accesses, collective I/O, etc. [128, 129].

All presently developed solvers of COSA read in simulation input file, a grid file and, when the simulation starts from the solution of a previous simulation, also a restart file. The solvers write a restart file and a set of solution files for flow visualisation and quantitative postprocessing. Input file is an ascii file and it contains all parameters which are required to perform the flow simulation. It also provides all information regarding computational grid, such as grid dimensions, boundary conditions and connectivity information. This file needs to be read by all MPI processes, and therefore does not use any special features of parallel I/O capabilities. Next, the grid, restart, and the solution file may be either in CGNS or plain data file binary format. The CFD General Notation System (CGNS) [130] provides a general, portable, and extensible standard for the storage and retrieval of CFD analysis data, therefore, it is an important part of every CFD computational code. However, as it was observed that pure MPI I/O parallelisation performs much better than parallel CGNS I/O for both reading and writing the files, we have decided to use pure MPI I/O parallelisation as a default of COSA, and to rather make conversion to the other file formats in a pre– or post–processing step. Due to this reason, a number of utility interface codes were developed during this project, to convert all mentioned binary files to CGNS standard and vice versa. An important utility interface code is the one which converts the CGNS file containing grid coordinates, inter–block grid connectivity and boundary definition of the grids generated by the commercial grid generators (e.g. NUMECA IGG, ANSYS ICEM, etc.), to a COSA format binary grid file and an ascii part of input file on grid information. Another important utility interface code converts COSA binary solution files to CGNS or TECPLOT (plt/szplt) format.

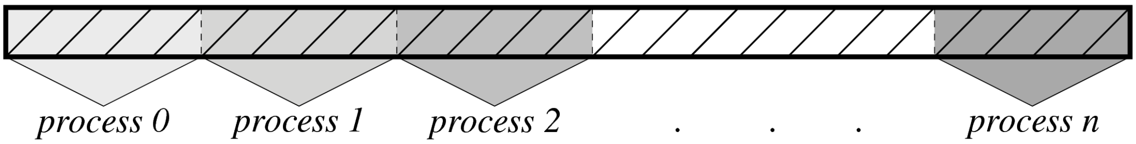


Figure 4.3: Sketch of a common data file, where each MPI process reads a chunk of the data.

All mentioned plain binary I/O data files contain data in contiguous format. Therefore, this structure is very appropriate for most straightforward MPI I/O parallelisation. The reading of the files is done with a standard MPI I/O functions, an open, a seek, a read, and a close. Simple example in Fig. 4.3 depicts n MPI processes, where each process needs to read $(1/n)$ part of the file. The collective MPI function for opening the files is called `MPI_FILE_OPEN`. After opening the file, each MPI process moves its local file pointer to the position where the data in the file needs to be read by that particular process. For this operation the function `MPI_FILE_SEEK` is used. In next operation each MPI process reads the data using the function `MPI_FILE_READ`, which reads the data from the position which was determined by the function `MPI_FILE_SEEK`. Finally, the file needs to be closed using the function `MPI_FILE_CLOSE`. The procedure for writing the files using MPI I/O is exactly the same as the one described above. The only difference is that instead of the function `MPI_FILE_READ` the function `MPI_FILE_WRITE` must be used. Using this function, each MPI process writes the data into the location, which was determined using the function `MPI_FILE_SEEK`.

The aforementioned MPI I/O capabilities were developed during this work. Previously, the MPI I/O was only used for writing files and was not optimal, as the individual data elements were written to the file one at a time. Writing of the data was optimised is such way that the data to be written were aggregated into arrays and then written all at once as explained above. The reading capabilities were previously using serial I/O, where each MPI process opened the file to be read, calculated the position of the file pointer and read the data. Previously, the grid file was in ascii format, this was replaced with a binary file. The size of a binary file is about three times smaller than the previously used ascii file, which is a great benefit in terms of saving disk space.

Table 4.1 shows the difference between the old and the optimised I/O capabilities. The analysis was performed on ARCHER HPC system [131], which is the latest UK national HPC service started in November 2013. The ARCHER hardware consists of the Cray XC30 MPP supercomputer, external login nodes, postprocessing nodes, and the associated filesystems. ARCHER phase 2 features 4920 compute nodes and each node has two 12–core Intel Ivy Bridge series processors giving a total number of 118, 080 processing cores. Each node has a total of 64 GB of memory, there are also fat memory nodes featuring 128 GB of memory. The system also features high–performance Lustre storage system. The analysis refers to a yawed wind HAWT test case solved with the HB solver using four complex harmonics $N_H = 4$. A 3.2 million cell–grid features 800 perfectly balanced blocks of the size 16 × 16 × 16 cells.

Table 4.1: Comparison of the old and optimised MPI I/O.

# cores	old I/O (s)	optimised I/O (s)	speed–up
20	85.2	29.7	2.9
50	65.4	25.4	2.6
100	58.6	23.5	2.5
200	57.0	24.9	2.3
400	64.4	25.4	2.5

First and second columns of Table 4.1, corresponding respectively to the old and optimised I/O, represent the time spent for reading and writing of all I/O files, without performing any computational work. The third column represents the speed–up defined as the ratio between the time taken to run a program with the old I/O and the time taken to run a program with the optimised I/O. From Table 4.1 it may be concluded that newly restructured I/O is performing more than 2.3–times better than the old I/O.

4.6 Parallel scalability

Scalability of a parallel algorithm is an important aspect of performance analyses. It is used to analyse how effectively parallel algorithms use an increased number of CPU cores. Generally, there are two types of scalability: so–called weak scalability and strong scalability [116]. The definition of weak scalability states that the problem size and number of cores grow in such a way that the amount of data per core remains constant. Linear scaling is achieved if the runtime remains constant while the workload is increased in proportion to the number of cores. The definition of strong scalability states that the problem size remains fixed and the number of cores are increased. The linear scaling is achieved if the speedup is equal to the number of cores used. Strong scalability assesses the efficiency of communications. Linear scaling with strong scalability at larger core counts is more difficult to achieve than linear scaling with weak scalability, since the number of communications in former analysis increases in proportion to the number of cores used. Speedup S_P is defined as the ratio between the time taken to run a program on a single core T_1 and the time taken on n number of cores T_n. Speedup may be written using the expression:

$$S_P = \frac{T_1}{T_n}. \tag{4.1}$$

In the ideal case the time $T_n = T_{ni}$, where T_{ni} represents the ideal time taken on n number of cores, and is given as:

$$T_{ni} = \frac{T_1}{n}. \tag{4.2}$$

We may also introduce the parallel efficiency E_P, which is the measure how far we are from the ideal speedup, hence:

$$E_P = \frac{S_P}{n}. \tag{4.3}$$

It must be pointed out that the definitions of Eqns. (4.1), (4.2) and (4.3) may cause problems in practice. One of the issues is that the size of the problem that is solved on a HPC machine may be quite large. Thus, it may therefore be too large to fit on a single core due to the memory constraints of a single node. Due to this reason, the definition of the T_1 must be slightly modified. Let us redefine T_1 to become the time taken to run a program on a minimum number of cores required, so the memory of the problem we are tying to solve can fit.

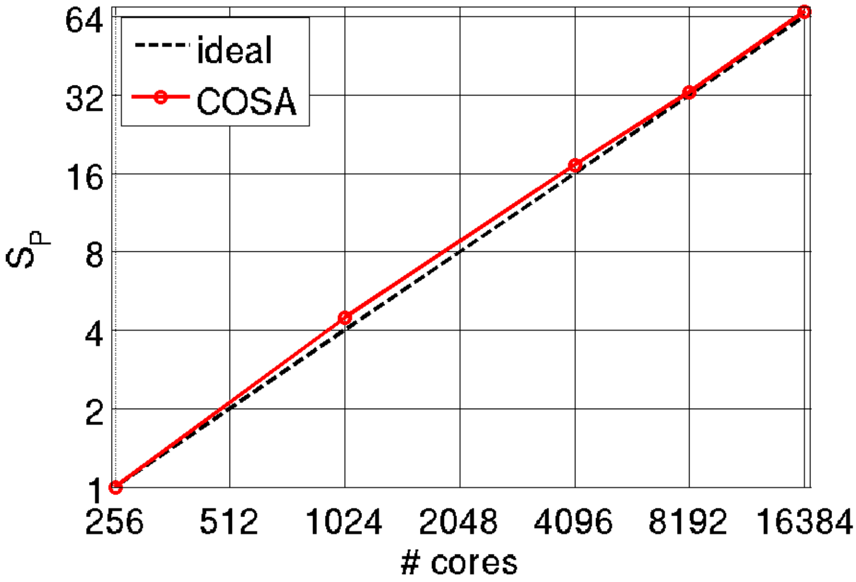


Figure 4.4: Parallel scalability analysis of algorithmic part of COSA HB MPI solver on ARCHER HPC system, using plunging wing test case.

Figure 4.4 depicts strong parallel scalability analysis of algorithmic part of COSA HB solver, performed on ARCHER HPC system [131]. The analysis refers to a plunging 3D wing test case solved with the HB solver using four complex harmonics $N_H = 4$. A 38 million cell–grid features $16,384$ perfectly balanced blocks of the size $16 \times 16 \times 16$ cells. For this test, we have turned off all I/O except for reading the input and mesh files. Figure 4.4 confirms that our newly developed HB solver achieves linear scaling at least up to $16,000$ cores. Further inspection of Fig. 4.4 reveals that in some parts of the curve COSA speedup is superlinear, which means that the speedup is larger than the number of cores used. In theory this occurrence is impossible, however, in practice superlinear speedup is not that uncommon. One possible reason for this occurrence is that when using lower number of cores, the program must access more memory of each node, and this could slightly slow down the program execution at lower core counts.

The strong parallel scalability test was also performed using a smaller HAWT test case. The analysis of algorithmic part of COSA HB solver was performed on ARCHER

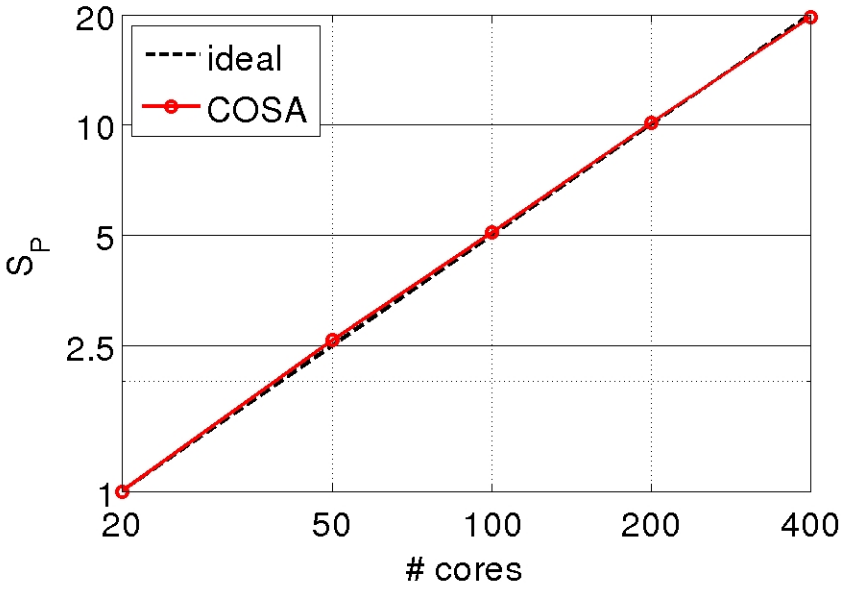


Figure 4.5: Parallel scalability analysis of algorithmic part of COSA HB MPI solver on ARCHER HPC system, using HAWT test case.

HPC system [131], and is depicted in Fig. 4.5. Please note that minimum required I/O capabilities were used for the scaling test. The test case is a HAWT in yawed flow, the one used to obtain data in Table 4.1. The HB solver and four complex harmonics $N_H = 4$ has been used. A 3.2 million cell–grid features 800 perfectly balanced blocks of the size $16 \times 16 \times 16$ cells. Figure 4.1 confirms the linear scaling of the HB solver up to 400 cores. Similar behaviour as in Fig. 4.4 has been observed also herein. The superlinear speedup at lower core counts is most likely associated with higher amount of memory used per node at lower core counts.

Chapter 5

Validation

This chapter presents numerical results used for the validation of newly developed 3D capabilities of COSA. At first a short summary of the previously published test cases used for the validation and verification of predicting capabilities of the 2D COSA solvers is described. Furthermore, the laminar delta wing, ONERA M6 wing, and S809 airfoil test cases are presented. These test cases served as a first step towards validation of the newly developed 3D capabilities. The laminar delta wing test case has been used to validate the calculation of the forces, moments and the solution accuracy of the newly developed laminar solver against the CFL3D CFD code. The ONERA M6 wing, a well establish reference case to validate external flow CFD methods, has been used to validate the accuracy of the $k-\omega$ SST turbulence model at transonic flow conditions. The computed solution has been compared against the experimental data and two well established CFD codes, CFL3D and NUMECA fine, using the SA and $k-\omega$ SST turbulence models, respectively. The S809 test case was selected as the S809 airfoil constructs the NREL Phase VI wind turbine, which has been used for the extensive validation of newly developed capabilities of COSA in this section. Using the S809 airfoil, the predictive capabilities of the $k-\omega$ SST turbulence model and its accuracy have both been tested. It has been assessed whether the $k-\omega$ SST model has the ability to predict the stall point and fully separated flows. The computed data have been compared with the experimental data and against the CFD code OVERTURNS, which uses the fully turbulent SA model, and the $\gamma - \overline{Re_{\theta t}} - SA$ transition model. The comparison with the transition model has been done in order to investigate whether the agreement between the CFD and the experimental data could be improved for the post–stall prediction.

The main validation consists of the H–Darrieus vertical–axis wind turbine, oscillating wing and several operating regimes of the upwind configuration of the NREL Phase VI test cases. The H–Darrieus vertical–axis wind turbine and oscillating wing test cases are both used to validate highly separated unsteady flows of moving airfoils, and are both compared against FLUENT. Additionally, the H–Darrieus vertical–axis wind turbine power curve has been compared with the experimental data. The upwind configuration cases of the NREL Phase VI wind turbine have been considered to extensively validate the newly developed prediction capabilities of the COSA CFD system for the analyses of HAWT flows. Several operating regimes in straight wind have been considered, as well as conditions in yawed wind flow. All calculations have been compared with the extensive experimental data set from the NREL Phase VI campaign, straight wind calculations have also been compared against commercial CFD code NUMECA.

5.1 Summary of previously published validation cases

The predicting capabilities of the steady 2D Euler solver and the second order accuracy of the convective flux discretization are described in [132]. A sequence of numerical solutions of an inviscid steady 2D internal flow problem computed with progressively more refined grids and the associated analytical solution, have been compared. The predicting capabilities of the 2D time–domain Euler solver and the second order accuracy of the time– and space–discretization have been demonstrated by considering a time–dependent problem resulting from the superposition of a uniform freestream and a steady vortex. The verification has been performed by comparing a sequence of numerical solutions of this problem computed with progressively more time– and space–refined grids and the associated analytical solution [22]. The assessment of the predicting capabilities and the second order accuracy of the 2D time–dependent laminar solver has been based on the analysis of the vortex shedding behind a circular cylinder and the use of Richardson's extrapolations. The viscous laminar predicting capabilities have also been verified by comparing a computed 2D laminar flat plate boundary layer and the analytical solution of Blasius [22]. The predicting capability of the code for 2D inviscid problems with moving grids is verified in [52], which compared the time–dependent pressure difference across a pitching flat plate in a uniform freestream to Theodorsen's analytical solution. In [26], the predicting capabilities of the turbulent solver have been assessed by considering two test cases. One was the flat plate turbulent boundary layer with freestream Mach number M_{fs} of 0.2 and Reynolds number Re of $6 \cdot 10^6$. The computed velocity profile normal to the flat plate is compared to Spalding's velocity profile, and the computed skin friction coefficient is compared to a typical analytical estimate. The other test case was the turbulent separated flow past the NACA4412 airfoil with $M_{fs} = 0.2$, Re based on the airfoil chord and the freestream velocity of $1.52 \cdot 10^6$, and angle of attack (AoA) of 13.87^o. The computed velocity profiles normal to the airfoil at several chordwise positions are compared with measured data. In both cases, an excellent agreement between computed and semi–empirical/measured data has been observed. The second order accuracy of the 2D TD solver for moving grid problems has been assessed by considering the inviscid time–dependent flow past a pitching NACA0012 airfoil [55].

For the verification of newly developed 3D capabilities, majority of these 2D validation and verification test cases have been repeated and compared with the 2D solutions. The 2D computational grids were extruded into the 3D grids, by lining up the same 2D grid along the airfoil span. The periodic boundary conditions were enforced in the spanwise direction, in order to simulate an infinite object.

5.2 Preliminary validation

5.2.1 Delta wing

The delta wing test case, is one of the test cases which are available on NASA CFL3D validation website [133]. The test case was created in order to demonstrate the CGNS capability implemented in CFL3D, which is a structured multiblock Navier–Stokes CFD code developed at NASA Langley Research Center for solving 2D and 3D fluid flow problems. This test case is considered here for validating the newly developed steady 3D laminar solver of COSA. The test case was selected, as it features laminar flow, therefore, while comparing forces, moments and flow solution, any uncertainty of the turbulence model can be removed. This is an excellent test case to verify whether the calculation of forces and moments is done properly, as it uses high angle of attack. The purpose of present test case is a) to validate the calculation of the forces and moments, b) to demonstrate solution accuracy of newly developed laminar solver.

The considered test case simulates a subsonic laminar flow past a 75° swept delta wing at 20.5° degrees angle of attack. The flow features free–stream Mach number of 0.3. The Reynolds number based on the root chord c, and the free–stream velocity is 0.5 million. The flow passes the leading edge then it rolls up and creates a vortex, which remains behind the wing for a long time. The reference area of the half wing is $0.13398c^2$. Complete description of the test case, input files and solution files are all provided on NASA CFL3D validation website [133]. The COSA simulation has been performed using the MG solver with 3 grid levels. CFL number was set to 4. Other numerical parameters were set as similar as possible to those of the CFL3D analysis.

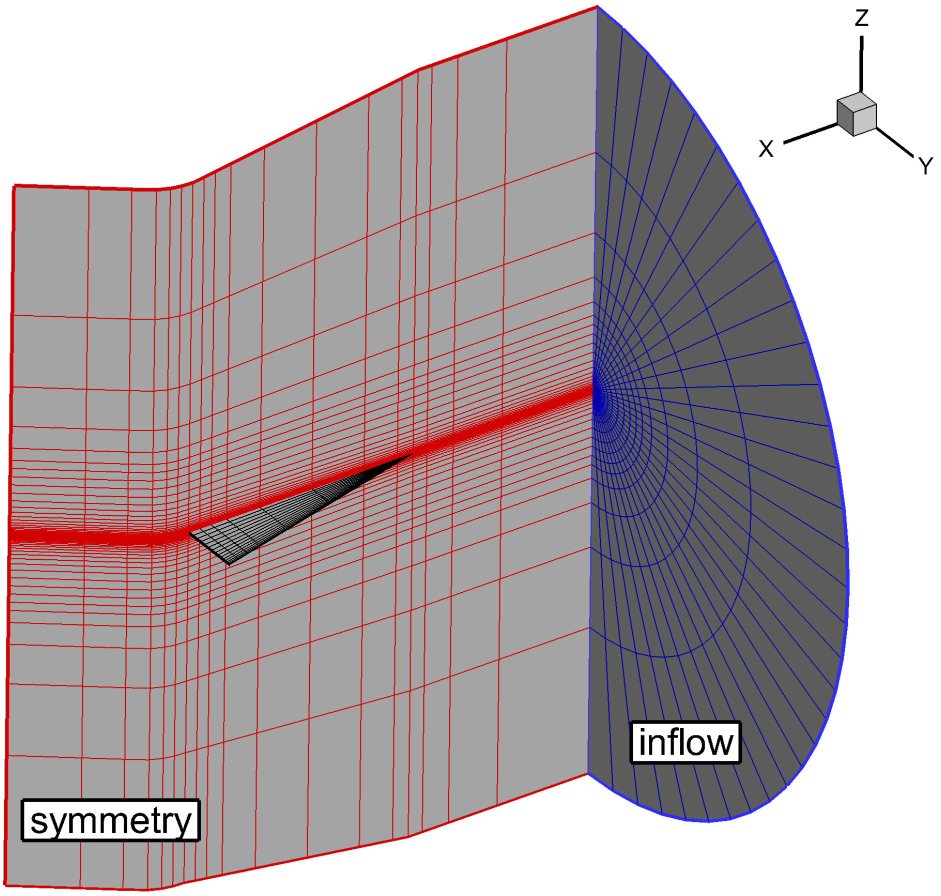


Figure 5.1: Representation of computational domain of the delta wing test case. Only every second line in all three directions is plotted.

Figure 5.1 depicts the computational grid, which was obtained from [133]. For visual clarity, only every second line of all three grid line sets is plotted. It is a single–block grid and has the size of $36 \times 64 \times 64$ cells. Only half of the problem is simulated, using symmetry boundary condition at midspan. The delta wing has zero thickness and features 16 cells in chord–wise direction and 32 cells in spanwise direction. The distance d_w of the first grid points off the wing surface from the surface itself is about $2 \cdot 10^{-4}c$.

Three force coefficients and a moment coefficient have been compared in the calculations performed by the steady laminar solver of CFL3D and COSA. Table 5.1 reports the lift C_l and drag C_d coefficients, as well as the force coefficient in y–direction C_y, and the moment coefficient in y–direction C_{my}. Table 5.1 shows an excellent agreement between COSA and CFL3D for all forces and moment coefficients.

Table 5.1: Comparison of forces and moment coefficients of CFL3D and COSA.

	C_l	C_d	C_y	C_{my}
CFL3D	0.7960	0.3066	0.0056	−0.2884
COSA	0.7956	0.3062	0.0057	−0.2876

Figure 5.2 shows the contour slices of the vorticity magnitude at four chordwise positions (30 %, 50 %, 70 % and 90 % of the root chord). The left semi–span represents the solution of CLF3D, whereas the right one represents the COSA solution. Inspection of the vorticity contours of the two different CFD codes reveals that for the positions 30 %, 50 % there are no visual differences between the CFL3D and COSA vorticity contours. For the positions 70 % and 90 %, the vortex computed with COSA appears to be slightly more dissipated than the one computed with CFL3D. Apart from that, the two codes show an excellent agreement.

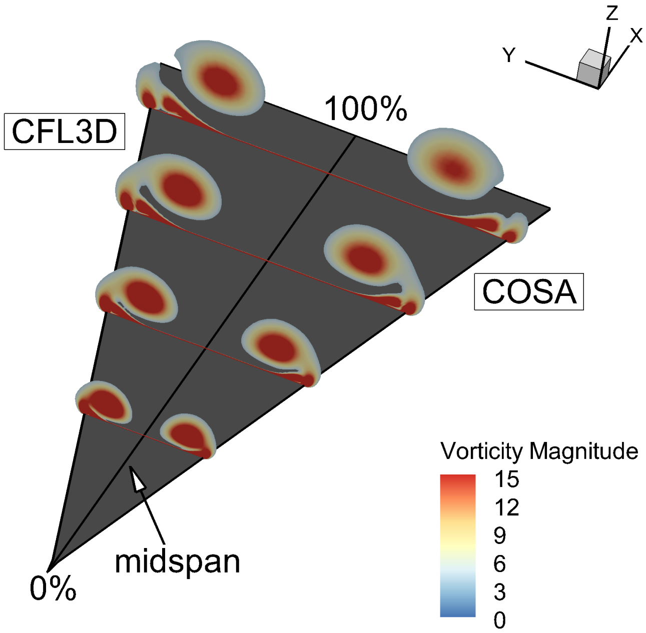


Figure 5.2: Slices of the vorticity magnitude at four chordwise positions, 30 %, 50 %, 70 % and 90 % of the root chord. The left semi–span represents the CFL3D solution, whereas the right semi–span represents the COSA solution.

The mean residual convergence histories of the two steady laminar analyses over the 600 MG cycles are reported in Fig. 5.3. One referring to the CFL3D computation and second to the COSA computation. The variable on the x–axis is the number of MG cycles, and the variable Δl_r on the y–axis is the logarithm in base 10 of the normalized RMS of all cell–residuals of the five RANS equations. For both simulations, each residual history curve is normalized by the RMS value at the first MG cycle. The convergence histories of both analyses are fairly close to each other, which confirms that the implementation of laminar solver has been done properly.

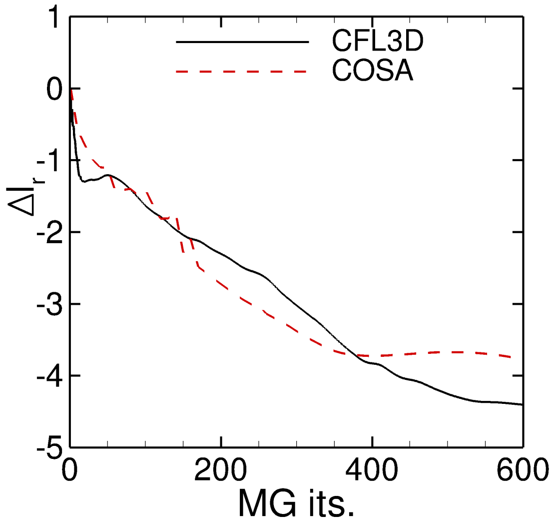


Figure 5.3: Residual convergence histories of the CFL3D and COSA calculations.

5.2.2 ONERA M6 wing

The ONERA M6 wing case has been established as a reference case to validate the external flow CFD methods. Some of the latest works using this test case for validation include [134, 135, 136]. The combination of simple geometry and complexities of transonic flow such as the local supersonic flow, shocks, and turbulent boundary layers separation, make the test case very attractive for CFD validation. The experiments have been carried out in a wind tunnel at various transonic Mach numbers and various angles of attack. The wind tunnel experiments are documented in [137]. Figure 5.4, obtained from [8] and available in [137], depicts the experimental setup of the ONERA M6 wing inside the wind tunnel. The CFD simulations at one flow field condition and the complete description of the test case is available at NPARC Alliance Validation Archive [8] and NASA CFL3D validation [133] websites. Herein the ONERA M6

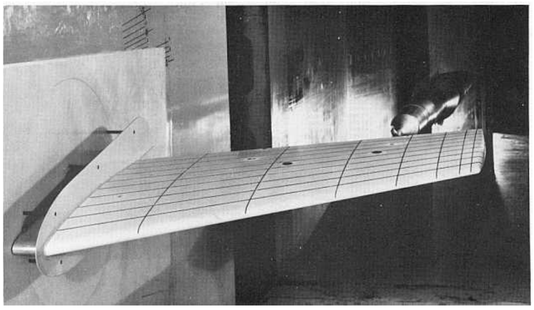
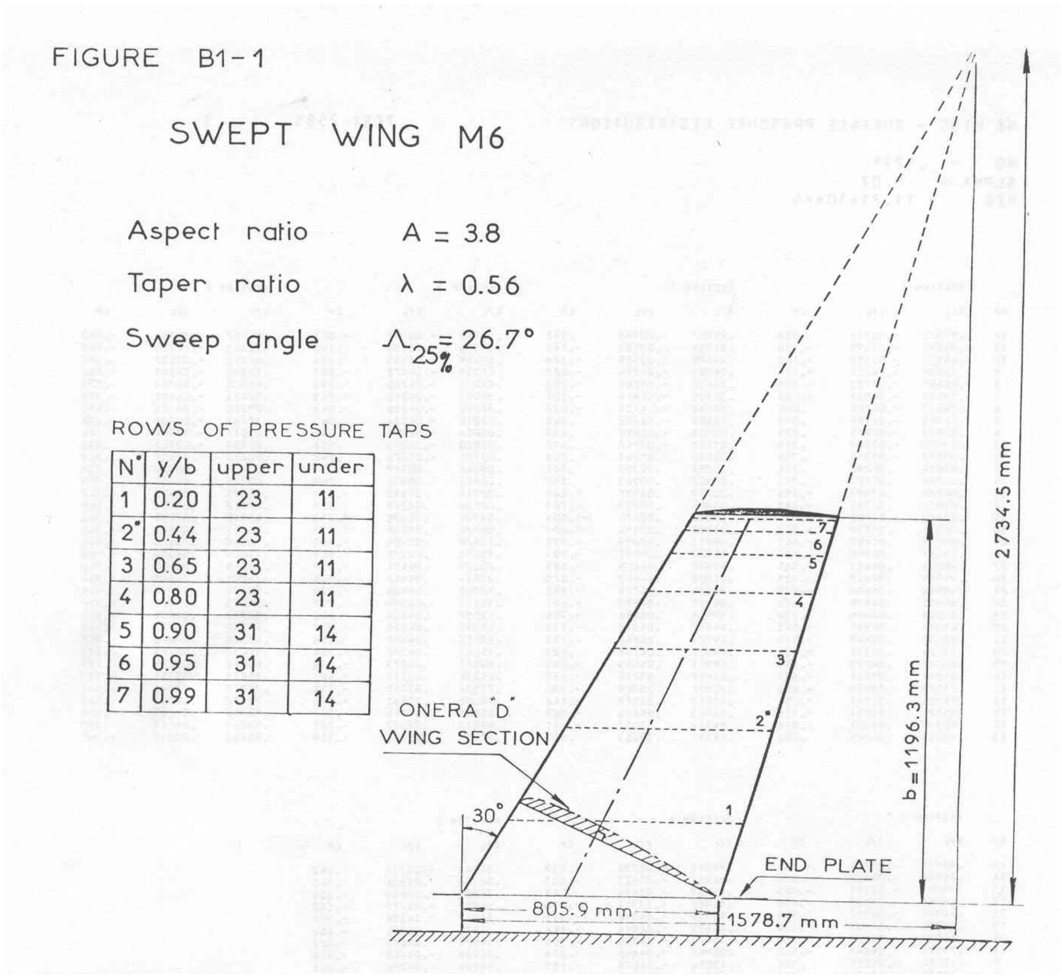


Figure 5.4: Experimental setup of the ONERA M6 wing. Taken from [8].



N°	y/b	upper	under
1	0.20	23	11
2*	0.44	23	11
3	0.65	23	11
4	0.80	23	11
5	0.90	31	14
6	0.95	31	14
7	0.99	31	14

Figure 5.5: Geometry of the ONERA M6 wing. Taken from [8].

wing case has been used to confirm the accuracy of the newly implemented 3D steady turbulent RANS solver, which uses the $k - \omega$ SST turbulence model.

The ONERA M6 wing is a swept, semi–span wing with no twist, which uses a symmetric ONERA D airfoil. The geometry of the ONERA M6 wing is depicted in Fig. 5.5, which is available in [137] and is obtained from [8]. Most important geometric properties are summarised in the Table 5.2.

The computational grid adopted for the flow simulations reported below is that available on the validation web site of the NASA CFD code CFL3D [133]. It is depicted in Fig. 5.6, and it consist of one zone wrapped as a C–grid about the wing leading

Table 5.2: Geometric properties of the ONERA M6 wing.

Semi–span (b)	1.1963 m
Mean aerodynamic chord (c)	0.64607 m
Leading–edge sweep	30.0°
Trailing–edge sweep	15.8°
Aspect ratio (AR)	3.8

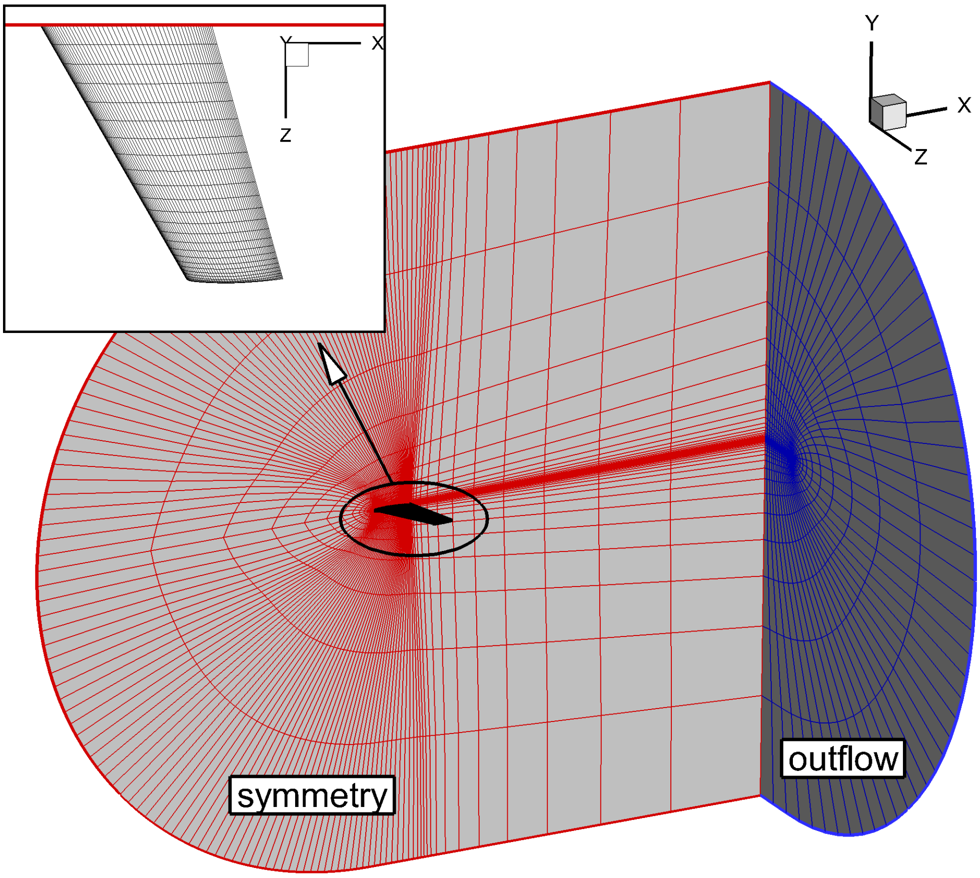


Figure 5.6: Representation of the computational domain and the wing surface grid of the ONERA M6 wing. Only every second line in all three directions is plotted.

edge. Only every second line of all three grid line sets is plotted, for visual clarity. Single–block grid features the dimensions of $289 \times 65 \times 49$ cells. Symmetry boundary condition at midspan is used, in order to simulate only semi–span wing. The semi–span wing features 256 cells in chord–wise direction and 48 cells in spanwise direction. The distance d_w of the first grid points off the wing surface from the surface itself is about $2.5 \cdot 10^{-6}c$ at the leading edge and about $5 \cdot 10^{-6}c$ at the trailing edge. The grid is nondimensionalized by the semi–span, therefore, the mean aerodynamic chord has the size $c = 0.54b$.

In this work, the turbulent flow field past the ONERA M6 wing corresponding to the condition of a transonic free stream of Mach 0.84 at the angle of attack of 3.06° is considered. The Reynolds number based on the mean aerodynamic chord length is 11.72 million. These operating conditions match one of the the wind tunnel tests reported in [8]. The COSA simulation has been performed using the MG solver with 2 grid levels and the CFL ramping, where the final CFL number has been set to 2.

Firstly, the values of the lift coefficient C_l, the drag coefficient C_d, the force coefficient in spanwise direction C_z, and the pitching moment coefficient C_{mz}, obtained with three different CFD simulations, are compared in Table 5.3. The results corresponding to CFL3D code are available on the NASA CFL3D validation website [133], and were produced using the SA turbulence model. While the COSA and NUMECA code both use the $k - \omega$ SST turbulence model. In order to remove any uncertainty on the flow solution between COSA and NUMECA, we have used in both codes as similar numerical parameters as possible. NUMECA is a commercial CFD and multiphysics

Table 5.3: Comparison of the forces and moment coefficients of CFL3D, COSA and NUMECA.

	C_l	C_d	C_z	C_{mz}
CFL3D (SA)	0.266	0.0172	0.014	0.1013
COSA (SST)	0.267	0.0154	−0.105	0.1020
NUMECA (SST)	0.265	0.0170	−0.105	0.1011

analysis and optimisation system. The calculations were run using FINE Turbo Flow Integrated Environment for rotating and non–rotating flow analysis in external and internal turbomachinery applications [138]. Overall, the comparison of the results between all three codes is in excellent agreement. One may only notice that the value of C_z significantly varies between CFL3D and the other two codes. This appears to be due to the fact that the CFL3D code is using the SA turbulence model, whereas the other two codes are both using the $k-\omega$ SST turbulence model. Furthermore, comparing the value of C_d of COSA and NUMECA, it is evident that there is some deviation of the C_d, even when using the same turbulence model. These differences probably appear due to the different settings of the $k-\omega$ SST model. It is difficult to match all turbulence model parameters, as all details of NUMECA implementation are not publically available.

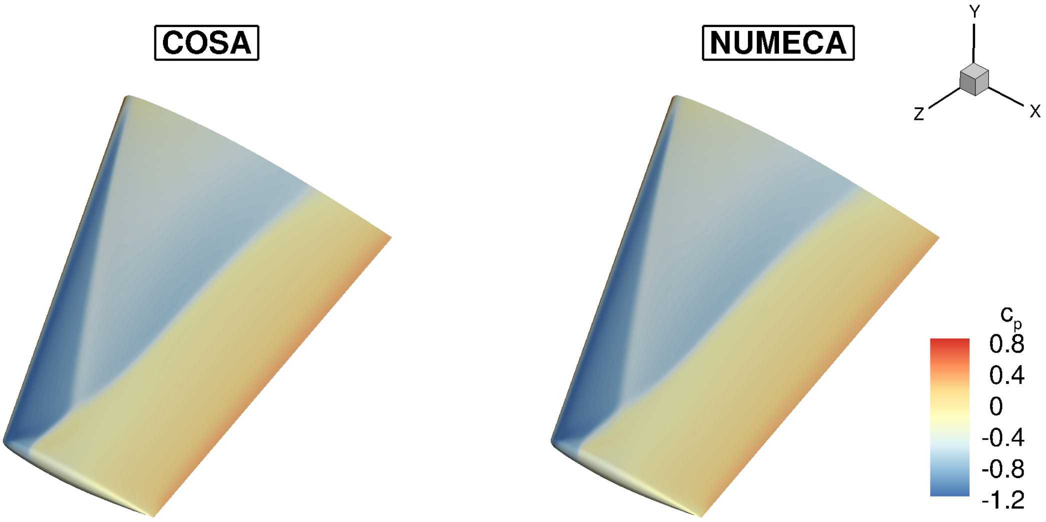


Figure 5.7: Contours of the pressure coefficient c_p along the wing span. Left plot: The COSA result. Right plot: The NUMECA result.

Figure 5.7 provides the contour plot of the pressure coefficient c_p over the ONERA M6 wing upper surface for the COSA and NUMECA solutions. The definition of c_p is:

$$c_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty u_\infty^2}, \tag{5.1}$$

where p and p_∞ denote local and freestream static pressure respectively. The blue colour on the upper surface of both wings corresponds to the low–pressure region associated with the formation of the two shock waves which coalesced into one. Visual comparison of the COSA (left) and NUMECA (right) solutions reveals there are no differences among the static pressure field between the two CFD codes.

Figure 5.8 represents the comparison of the wing static pressure coefficient for COSA, NUMECA and the experimental data. Seven spanwise positions are compared, 20 %, 44 %, 65 %, 80 %, 90 %, 95 % and 99 % of the semi–span. Inspection of the COSA

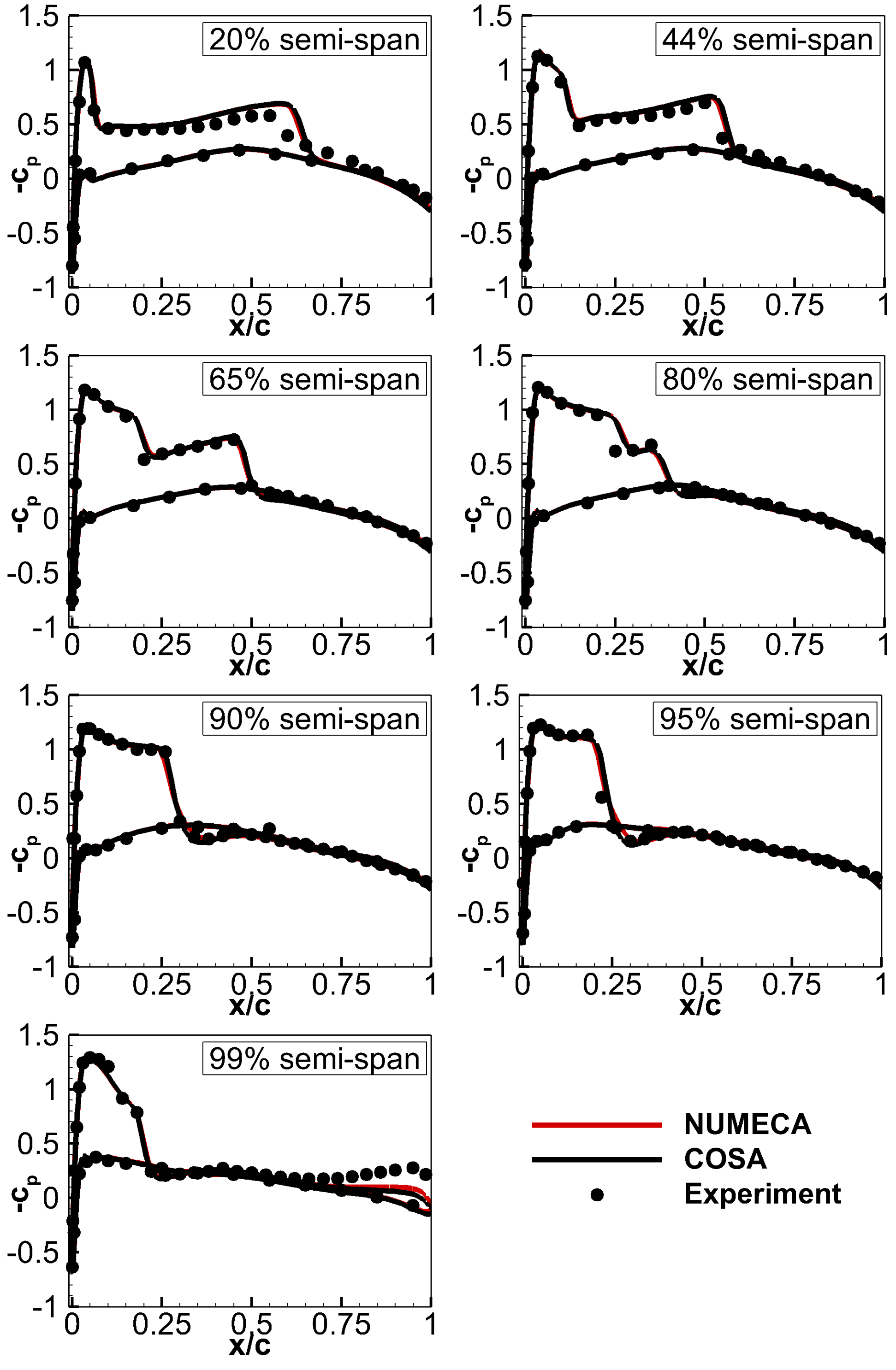


Figure 5.8: Comparison of the pressure coefficient c_p distributions along the chord at seven spanwise positions for two CFD results and experimental results.

and NUMECA results reveals that the two CFD results are in excellent agreement at all spanwise positions. In general, there is also a good agreement between the two CFD results and the experimental data. Inspecting the results more precisely, one may notice that at position 20 % of semi–span the shock computed with COSA and NUMECA is stronger than in the experiment. It is also evident, that at position 80% of semi–span, the two numerical solutions are unable to predict double shock formation. The discrepancy between numerical and experimental results could occur due to many reasons. One of the possible reasons is certainly the usage of turbulence modelling. Furthermore, the grid used in the computation is quite coarse, therefore, using finer grid may improve prediction of the shock formation at the positions 20 % and 80 % of the semi–span. Another reason for overpredicted shock computed by the two CFD codes at position 20% of semi–span, could also be the usage of the symmetry boundary condition at midspan. Based on Fig. 5.4, which depicts the experimental set–up of the ONERA M6 wing, the wing root was not mounted directly on the wind tunnel wall, but rather on a short root planform, with an endplate between the planform and the wing. Therefore, the symmetry condition can not exactly reproduce the flow physics of the semi–span wing mounted in the wind tunnel during the experiment.

5.2.3 S809 airfoil

The S809 airfoil is a 21 % thick, laminar–flow airfoil, specially designed for the HAWT applications. It is employed by the NREL Phase VI wind turbine [9]. The airfoil was deliberately designed to feature a short region of adverse pressure gradient along the upper surface or so called "transition–ramp", which enables smooth transition from laminar to turbulent flow [139]. When the airfoil was designed, the two main objectives were: a) to achieve a restrained, relatively low, maximum lift coefficient and b) to achieve low drag coefficients for the Reynolds number of 2.0×10^6, over a range of lift coefficients varying from 0.2 to 0.8. Herein, the S809 airfoil is used to explore, how well may the $k - \omega$ SST turbulence model predict the stall point and the separated flow. As pointed out in [69], the common problem of the two–equation turbulence models is their failure to correctly predict the onset and amount of separation in adverse pressure gradient flows.

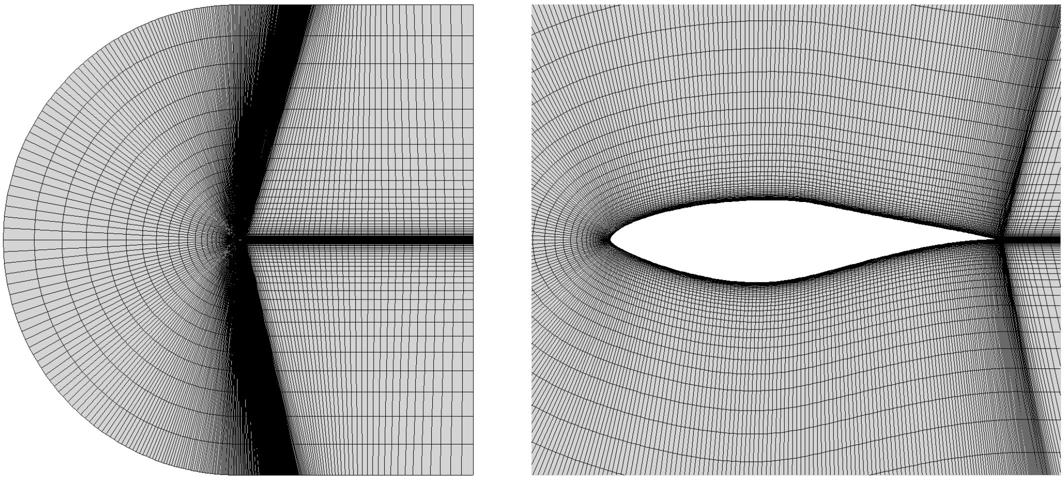


Figure 5.9: Grid for the S809 airfoil analysis. Left plot: complete domain. Right plot: near–airfoil area.

The 49, 152–cell C–grid adopted for the simulations of this section has 512 mesh intervals along the airfoil, 64 intervals in the grid cut, and 96 intervals in the normal–like direction. The farfield boundary is placed at about 50 chords from the airfoil, and the distance d_w of the first grid points off the airfoil surface from the surface itself is about $10^{-6}c$. Figure 5.9 provides the complete view of the adopted grid, and an enlarged view in the airfoil region. Simulations were run at the Reynolds number based on the airfoil chord and the freestream velocity 2×10^6, and the Mach number of 0.1, in order to avoid compressibility effects. The range of the AoAs has been simulated, varying from $-3°$ to $21°$. The CFL number of all simulations has been set to 4, the number of MG levels has been set to 3, and the CFL ramping has been used.

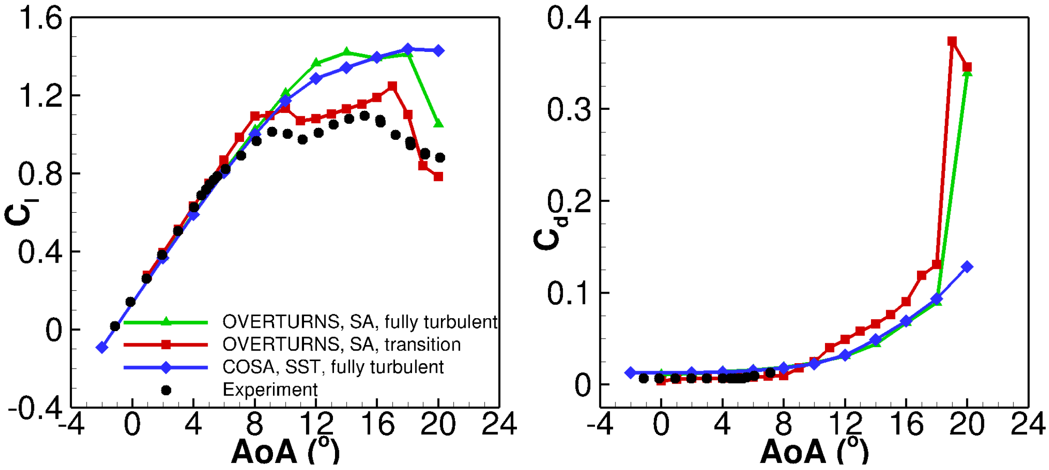


Figure 5.10: The lift and drag coefficients vs. angle of attack of the S809 airfoil.

Figure 5.10 represents the predicted lift and drag coefficients for various values of AoA. Plotted results are those obtained with COSA using the $k - \omega$ SST turbulence model and the experimental data reported in [139]. Additionally, the results obtained using the structured mesh solver OVERTURNS [140] were also added to the lift and the drag coefficient plots. These results were obtained using similar computational grid as the one used with COSA. The calculations performed with OVERTURNS, available in [141], were run both with the fully turbulent SA model, and the $\gamma - \overline{Re_{\theta t}} - SA$ transition model. Experimental data reveals that the airfoil starts stalling at about $AoA = 9°$. In the pre–stall region, the results of the fully turbulent calculations of COSA and OVERTURNS are fairly close to the experimental data for both the C_l and C_d. The transition model calculation of OVERTURNS slightly improves the prediction of both quantities in that region. However, close to the stalling point, the transition model seems to overpredict the C_l. In the post–stall region, the fully turbulent calculations of COSA and OVERTURNS both overpredict the C_l and delay the stalling point, regardless of the turbulence model used. Conversely, the transition calculation of OVERTURNS greatly improves the prediction of the C_l in that region. Furthermore, with the transition model the stalling point occurs at the AoA much closer to that in the experiment.

To further examine the agreement between COSA and the experimental data, the pressure coefficient profiles have been compared. The three subplots of Fig. 5.11 represent the c_p profiles of COSA and the experimental data, for the three different values of AoA. Overall, these plots highlight there is a very reasonable agreement between COSA and the experimental data in the pre–stall region of the airfoil. Examining each AoA more precisely, one may notice there is an excellent agreement for $AoA = 1°$, except

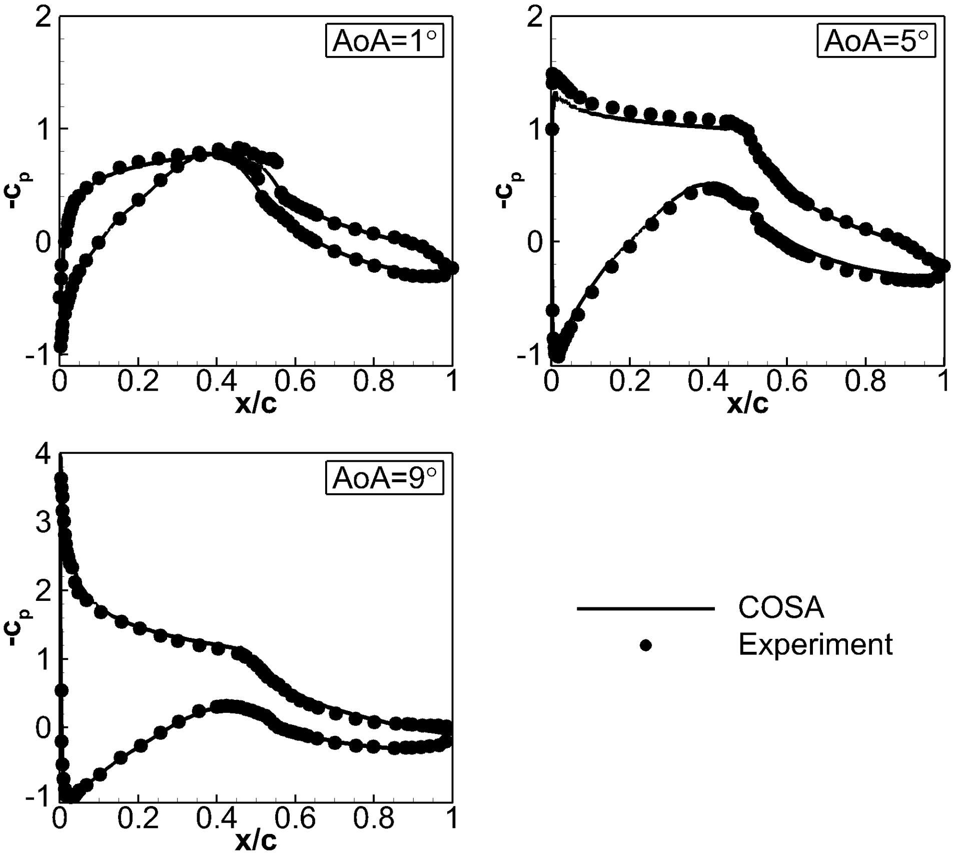


Figure 5.11: Pressure coefficient c_p of the S809 airfoil for the three different values of *AoA*.

in the small area of the upper surface around the position $x/c = 0.55$. At $AoA = 5°$, COSA slightly overpredicts the pressure coefficient in the transitional region, which is indicated with the prominent dip in pressure on the upper surface. Furthermore, at $AoA = 9°$, which corresponds to the experimental maximum lift coefficient, there is again an excellent agreement between COSA and the experimental data.

From the results presented above, it can be concluded that the fully turbulent calculations performed with either the $k-\omega$ SST or SA turbulence models, give very satisfactory results in the pre–stall region. This is first confirmed by the inspection of the lift and drag coefficients, which were compared with the experimental data. Furthermore, also the pressure coefficient profiles confirm there is a good agreement for all the examined *AoA*s, even the $AoA = 9°$, where a trailing–edge separation on the upper surface of the airfoil occurs. However, the prediction of the stall point is delayed in both COSA and OVERTURNS fully turbulent CFD calculations, and the post–stall characteristics are not in the best agreement with the experimental data. As shown in [141], it should be possible to improve predictions of the stall onset and post–stall characteristics when using transition models. However, as the transition models are beyond the scope of this thesis, all the calculations performed with COSA have been done using the fully turbulent $k-\omega$ SST turbulence model.

5.3 H–Darrieus vertical–axis wind turbine

A 2D model of an H–Darrieus wind turbine, sketched in Fig. 5.12, was considered herein. The rotor of the considered wind turbine has the radius R of 515 mm, and the blades are constructed using the NACA0021 airfoil with a chord of 85.8 mm. The blade is attached at 25 % chord from the airfoil leading edge. The operating condition considered for this study is characterised by the freestream velocity u_∞ of 9 m/s, and the rotational speed Ω of 550 RPM. The Reynolds number based on the airfoil chord is 1.7×10^5 and the Mach number associated with the circumferential speed of the rotor is 0.087. $TSR = \Omega R/u_\infty$ is the tip–speed ratio and for the considered operating conditions equals 3.3. This value corresponds to near maximum power operation, and unless otherwise stated, all results presented below refer to this value of TSR. This case study has been first reported and analyzed in studies [142] and [143], and later in several other studies, including [144].

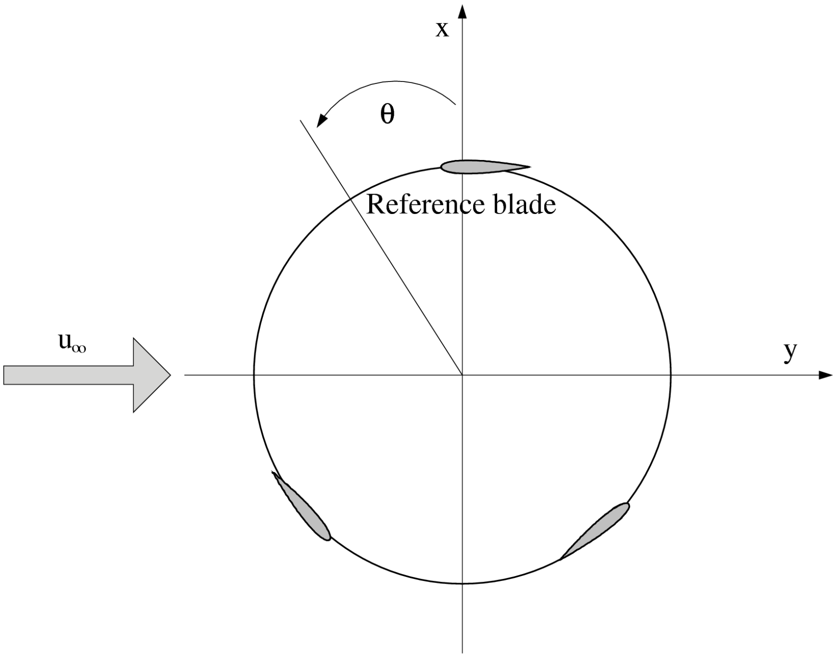


Figure 5.12: H–Darrieus rotor sketch.

Left and right subplots of Fig. 5.13 respectively report enlarged views of the grid around the rotor and the airfoil. The grid has 729,600 cells and features 448 cells around each airfoil. The distance d_w of the first grid points off the airfoil surface from the airfoil itself was $10^{-5}c$. The grid is made up of two subdomains: the circular region of radius $7R$ containing the three blades and consisting of 522,240 cells, and the annular region with inner radius of $7R$ and outer radius of $240R$ consisting of 207,360 cells. The identification of two distinct subdomains has been introduced to enable the CFD simulation of this rotor flow with the commercial ANSYS FLUENT CFD code. FLUENT uses a rotating and a stationary domain and requires a circular sliding interface. The FLUENT results presented below are obtained with the coupled pressure–based solver [145]. For COSA analyses the distinction between the two subdomains is irrelevant, since the entire grid moves with the rotor.

All COSA simulations have been performed using the MG solver with 3 grid levels, and CFL number has been set to 4.

Mesh refinement analysis, which has been performed by COSA, included the grid

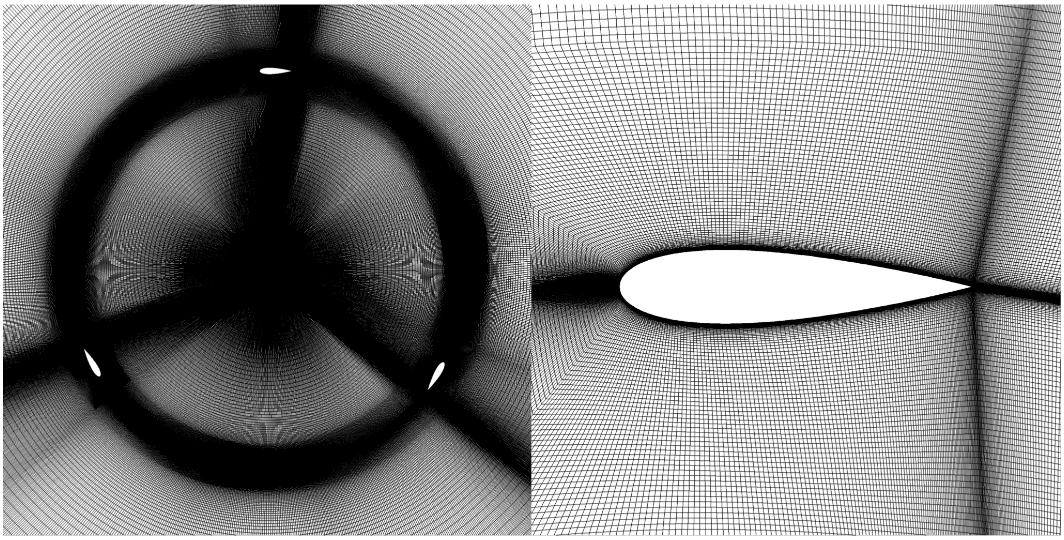


Figure 5.13: H–Darrieus rotor grid representation. Left: grid view in rotor region. Right: grid view in airfoil region.

under consideration and a finer one with twice as many grid lines in both directions. The analyses revealed that the present grid with 729, 600 cells gives a mesh–independent solution. In order to determine a time–step independent solution, four different TD simulations have been performed using a number of physical time–steps per period N_T of 1440, 720, 360, and 180. The study has revealed that 720 physical time–steps are required to obtain the independent solution of the time step.

The torque coefficient C_T per unit blade length has been used to monitor the convergence of the TD simulations to a periodic state, and is defined as:

$$C_T = \frac{T_D}{\frac{1}{2}\rho_\infty W_\infty^2 2R_D^2} \tag{5.2}$$

where T_D is the torque acting on the reference blade. The three TD simulations have been run until the maximum difference between C_T over the last two consecutive

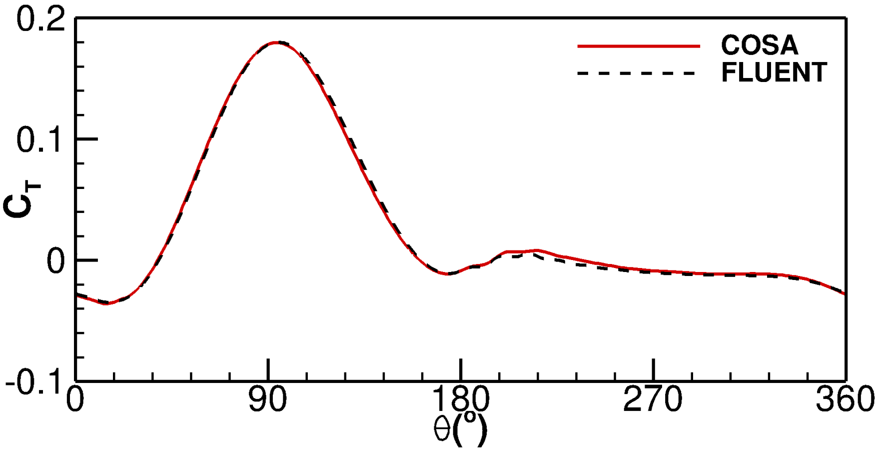


Figure 5.14: H–Darrieus rotor periodic profiles of torque coefficient of reference blade against azimuthal position θ computed with COSA and FLUENT simulations.

revolutions became about 0.1 % of their mean value over the latter period of the cycle pair.

Figure 5.14 reports the C_T profile comparison of the reference blade over one rotor revolution computed by the TD COSA and FLUENT analyses. The starting point of the reference blade of each revolution ($\theta = 0°$) is the position where the velocity of the reference blade and the absolute velocity of the wind are parallel and opposite. The C_T profile computed by the COSA refers to 720 intervals per period, and the one computed by the FLUENT to 900 intervals per period. An excellent agreement between the prediction of the two codes is observed. Some relatively small differences between the COSA and FLUENT predictions exist around the positions $\theta = 90^o$ and $\theta = 220^o$. The potential causes of these small differences could be due to the slight differences in the implementation of the turbulence model. The COSA and FLUENT solutions, however, are fairly close, which is also underlined by the fact that the mean torque predicted by the two codes differs by less than 0.15 %.

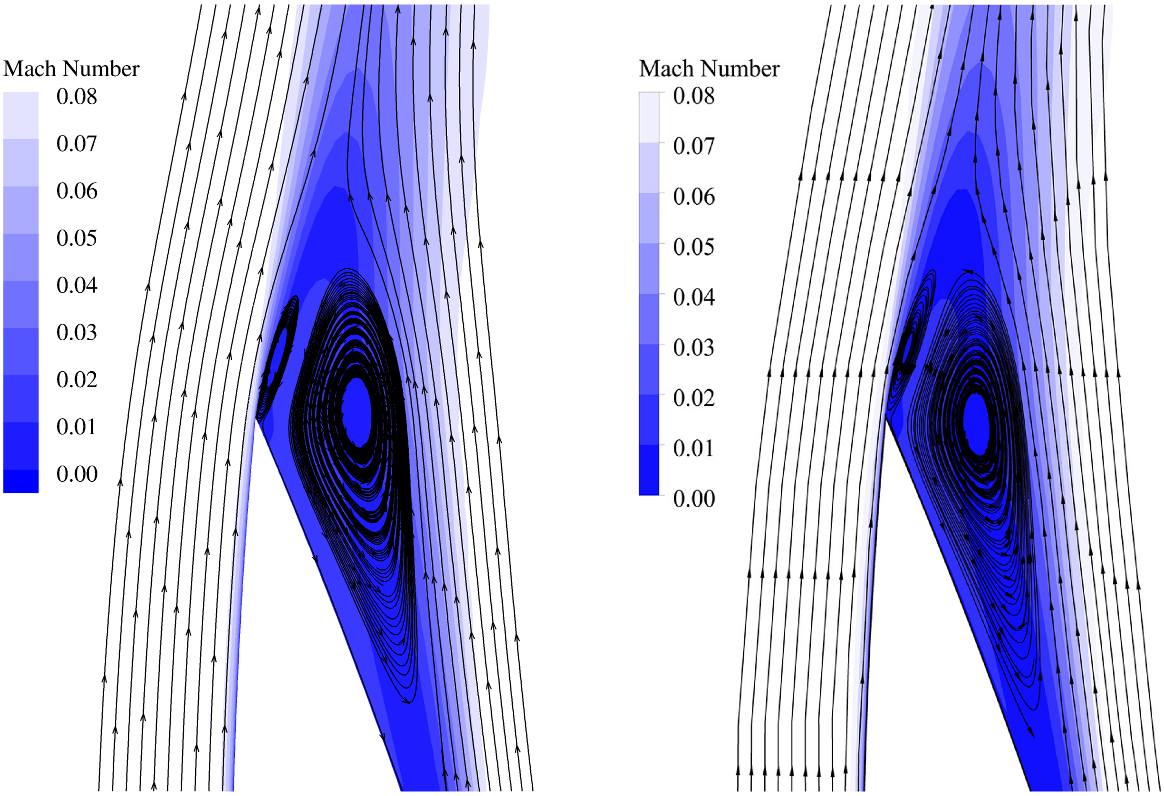


Figure 5.15: H–Darrieus rotor Mach contours and streamlines in the reference blade trailing edge region at azimuthal position $\theta = 99^o$ computed with COSA simulation (left) and FLUENT simulation (right).

Figure 5.15 depicts the comparison of the streamlines and Mach contours between the COSA simulation (left) and the FLUENT simulation (right), in the trailing edge region at azimuthal position $\theta = 99°$. One may notice that at this azimuthal position there is high level of stall associated with the highlighted flow separation. Despite this fact, an excellent agreement between the two CFD codes is observed.

Several additional TSRs have been computed with COSA simulations and compared with the experimental data in terms of power coefficient C_P, defined by the expression:

$$C_P = \frac{T \cdot \Omega}{\frac{1}{2}\rho A u_\infty^3}, \tag{5.3}$$

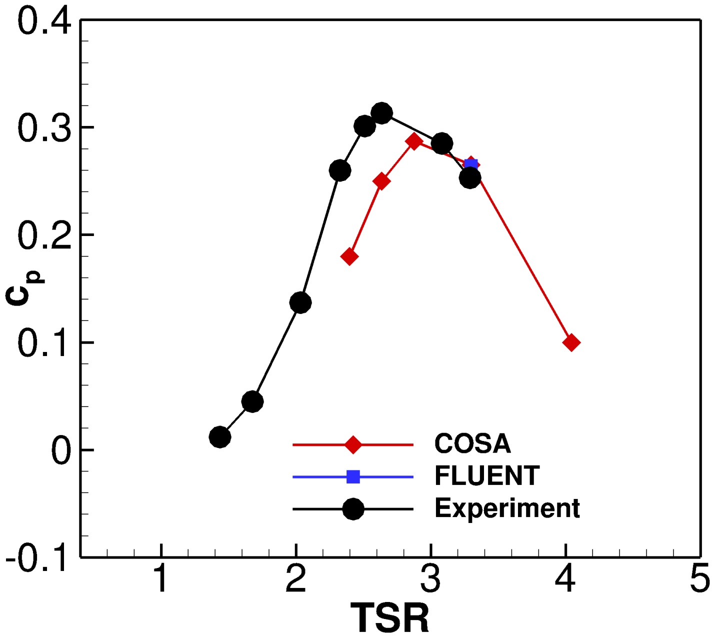


Figure 5.16: Comparison of the nondimensionalized power curves of the H–Darrieus rotor between COSA, FLUENT and experimental data.

where A is the frontal area of the rotor, given by the product of its diameter and height, and ρ is the standard density. The experimental data have been extracted from [143]. Please note, that in the experimental data, the correction due to wind tunnel blockage was not considered. Figure 5.16 shows the comparison of the rotor power curve predicted by the COSA simulations and experimental data. Operating point at $TSR = 3.3$ computed by the FLUENT simulation has also been added. The results indicate there is a reasonable agreement between the COSA simulations and experimental data. COSA is able to replicate the shape of the experimental power curve, however, the maximum C_P is predicted at slightly higher TSR as in the experiment. Furthermore, at most times, C_P of the experiment is slightly higher than in the CFD simulation. One of the reasons for these differences could probably be the wind tunnel blockage effects. Further analyses, which are not included in this work, also showed that the agreement between the power curve and COSA could be significantly improved by using low–speed preconditioner. Low–speed preconditioning is particularly important at the lower TSRs, where the level of dynamic stall is substantially higher than at higher TSRs. Nevertheless, the overall good agreement between the COSA solution and experimental data suggests the suitability of the COSA code for highly separated flows.

5.4 Oscillating wing

Oscillating wing device considered herein has been defined in section 6.1. Herein the operating condition characterised by a high efficiency of energy extraction in the turbulent flow regime is considered and compared to the FLUENT solution computed by the Kinsey and Dumas [23], for further verification of the COSA code.

Reynolds number based on the freestream velocity and the airfoil chord is $Re =$

0.5×10^6. The pitching center is at $x_p = 1/3$ of the chord from the LE, and the phase angle ϕ between heaving and pitching motions is 90.0^o. The heaving amplitude h_0 equals one chord and the pitching amplitude θ_0 is 75.0^o. This choice of parameters yields a value of the overall height h swept by the foil of 2.56 chords. The nondimensionalized frequency $f^* = fc/u_\infty$ for case A is 0.14, where f is the frequency in Hertz.

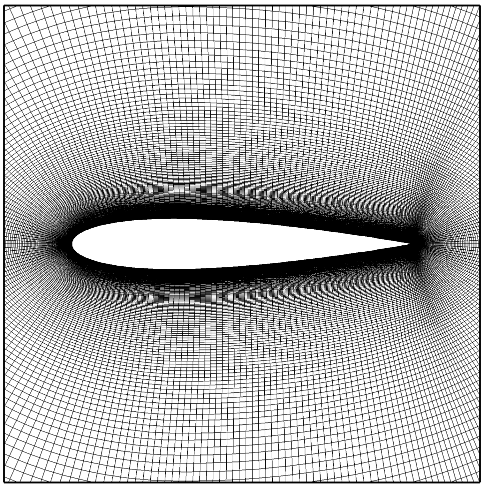


Figure 5.17: Grid for the NACA0015 airfoil.

The time–dependent 2D turbulent flow field past the oscillating wing was computed using structured multi–block non–deforming moving grid. The O–grid adopted for the simulations below is the one used in section 6.1.2 (coarse), and is depicted in Fig. 5.17. The grid features 256×256 cells with 256 intervals on the airfoil and 256 intervals in the normal–like direction. The farfield boundary in the airfoil plane was at about 50 chords from the foil, and the distance d_w of the first grid points off the foil surface from the foil itself was about $6 \times 10^{-6}c$. In all simulations the entire grid moved rigidly with the airfoil.

The required level of spatial and temporal refinement of the considered grid has been assessed in section 6.1.2, where the Reynolds number has been three times higher than the one considered for this simulation. It was concluded that 1024×1024 cells (fine) grid gives the grid–independent solution, and 512 intervals per period are sufficient to achieve a time–step independent solution. Relatively small differences between various space and time refinements were primarily due to small variations of the timing of the LEVS. Kinsey and Dumas [25] also performed mesh sensitivity analyses, however, they were only done using the SA turbulence model. They concluded that the medium grid with $42,200$ nodes gives the grid–independent solution, and 2000 intervals per period are required to achieve a time–step independent solution. Based on COSA grid independence analyses, it is very unlikely that the Kinsey and Dumas's medium grid would give the grid–independent solution also for the SST turbulence model. In order to make a fair comparison between COSA and FLUENT, 512 intervals per period were used in COSA and 2000 were used in FLUENT as these are sufficient to achieve a time–step independent solution in both cases. COSA also used similar spatial refinement than FLUENT.

At first, the periodic profiles of the overall power coefficient C_P, the heaving power coefficient C_{P_y} and the pitching power coefficient C_{P_θ} are reported respectively in the top, middle and bottom subplot of Fig. 5.18. The comparison is made between Kinsey

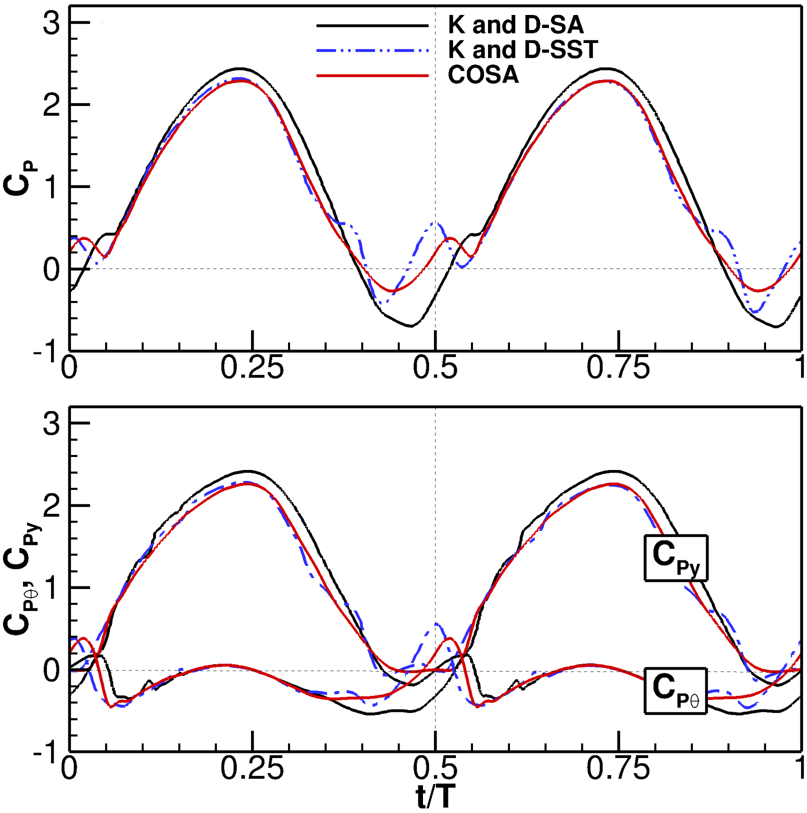


Figure 5.18: Comparison of the K and D–SA, K and D–SST and COSA solutions: overall power coefficient (top), heaving power coefficient (middle), and pitching power coefficient (bottom).

and Dumas's result using SA turbulence model (K and D–SA), Kinsey and Dumas's result using SST turbulence model (K and D–SST), and COSA solution, which uses SST turbulence model. Please note, all results of Kinsey and Dumas have been extracted and digitalized from [23]. The agreement of all power coefficient profiles between K and D–SST and COSA is fairly good. For all periodic profiles the differences between K and D–SST and K and D–SA are much higher than those observed between K and D–SST and COSA. Between the first 10 % and last 20 % of both semi–periods (region 1), the two C_P profiles computed by the SST are superimposed and are significantly lower than that of the SA solution. In the remainder of the cycle (region 2), there are some discrepancies between the two C_P profiles computed by the SST. However, the two periodic profiles are still relatively close, and are both significantly higher than that computed with SA solution. Looking at the C_{Py} plot, it is evident that the agreement between K and D–SST and COSA is fairly good during the whole oscillating cycle. Therefore, the component that is mostly responsible for the differences of the C_P profiles between the two SST solutions in region 2 is the $C_{P\theta}$. It is also observed that in region 1 all three $C_{P\theta}$ profiles are superimposed. The discrepancies in region 2 could be caused by various different reasons. As noted in grid refinement analyses in section 6.1.2, the sharp peak of C_{P_θ} is dependent on the level of spatial refinement, and these variations usually decrease as the grid is refined. Grid used by the COSA, and most likely, grid used by the FLUENT do not give the grid–independent solution. Different spatial refinement of the two grids, may therefore, be responsible for observed differences. Moreover, slightly different implementation of the COSA and FLUENT

$k-\omega$ SST turbulence model could also lead to small differences between the two solutions. Additionally, different setting of turbulent production limiters could also have an effect on the solution.

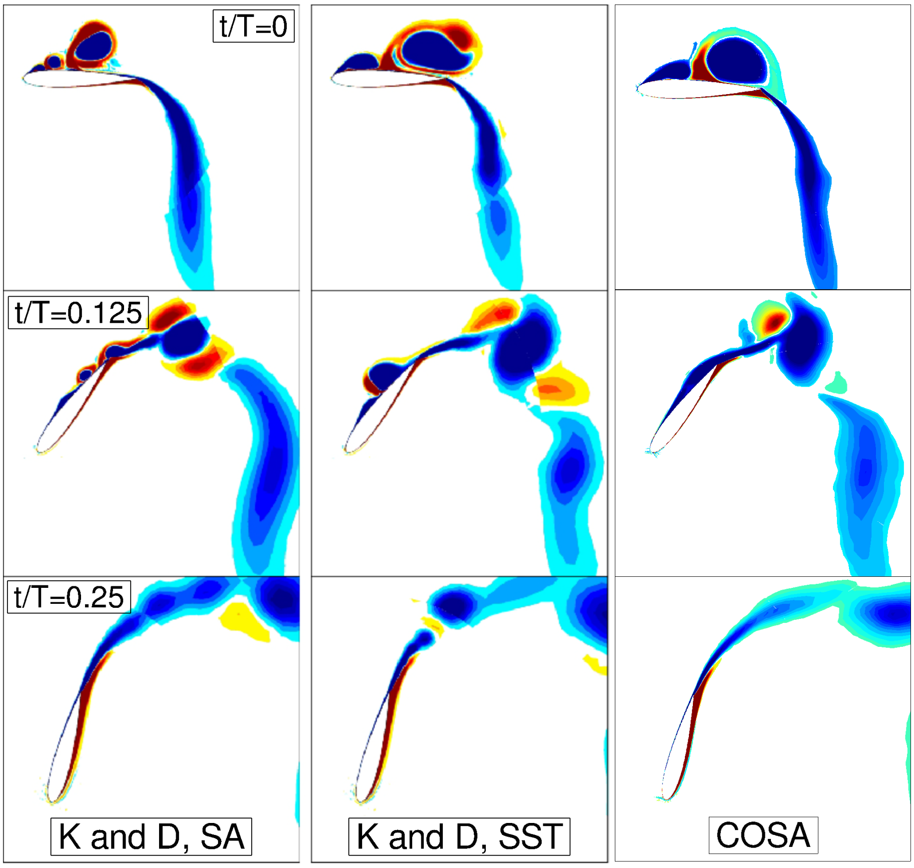


Figure 5.19: Comparison of vorticity time sequences at t/T=0, 0.125, 0.25, for K and D–SA, K and D–SST and COSA solutions (red: counter–clockwise vorticity, blue: clockwise vorticity).

Figure 5.19 represents the comparisons of K and D–SA, K and D–SST, and COSA vorticity field, when the airfoil is at positions 0 %, 12.5 % and 25 % of the oscillating cycle. Please note, all results of Kinsey and Dumas have been taken from [23]. At 0 % of the oscillating cycle the two SST solutions are in excellent agreement, the LEVS has travelled to the trailing edge in both cases. Only relatively small qualitative differences in the shape of the low–pressure bubble can be observed. When comparing the two SST solutions against the SA, it becomes apparent that the timing of the LEVS is completely different, LEVS in SA case has arrived only at mid–chord, whereas it is already at the trailing edge in the two SST cases. At 12.5 % and 25 % of the oscillating cycle, the three solutions are fairly similar, as also observed in Fig. 5.18.

Even though some discrepancies have been observed between the two SST solutions, the overall good agreement between the COSA and K and D–SST solution has been found. It has been observed that the discrepancies between the SA and SST turbulence models have been much higher, than those observed between the two SST models. This confirms the suitability of the COSA code for highly separated flows.

5.5 NREL Phase VI horizontal–axis wind turbine

Several upwind rotor configurations of the unsteady aerodynamics experiment of the NREL Phase VI wind turbine [9] have been considered herein for thorough validation of newly developed capabilities of the compressible NS solver COSA. The experiment has been conducted by the National Renewable Energy Laboratory (NREL) located at the National Wind Technology Center (NWTC) near Golden, Colorado, USA.

5.5.1 Physical and numerical set–up

The Phase VI rotor features two twisted and tapered blades, with the radius $R = 5.029\,m$. Their geometry is based on the S809 airfoil, and also features a non–linear twist distribution and a linear taper. More details about the blades can be found in [9]. The turbine is stall regulated and has the rated power of $19.8\,kW$. The cut–in wind velocity equals $v_\infty = 6\,m/s$, therefore, most of the experimental data exist for the velocities higher than the cut–in. Only the rotor is modelled in CFD calculations, both the tower and nacelle have been excluded.

Table 5.4: Operating conditions for the NREL Phase VI calculations.

$v_\infty\,(m/s)$	$\Omega\,(RPM)$	$\rho\,(kg/m^3)$	$\mu\,(kg/ms)$
7	71.9	1.246	1.769×10^{-5}
10	72.1	1.246	1.769×10^{-5}
13	72.1	1.227	1.781×10^{-5}
15	72.1	1.224	1.784×10^{-5}
20	72.0	1.221	1.786×10^{-5}
25	72.1	1.220	1.785×10^{-5}

Three different yaw angles have been considered: 0°, 10° and 30°. In all considered cases the rotor cone angle was set to 0°, and zero vertical shear has been assumed. For the yaw angle 0°, both blades experience the same inflow conditions regardless of the azimuthal position of the rotor. Therefore, only one blade can be modelled using steady periodicity boundary condition, thus, halving the computational cost. As the yaw 10° and 30° calculations have been performed with HB solver, only a single blade using multi–frequency periodicity boundary condition was modelled. As described in section 3.4, multi–frequency periodicity boundary condition allows to simulate a single blade for unsteady flows in frequency–domain. The computational cost with such boundary conditions is significantly reduced. The blade tip pitch angle was set to 3° towards feather. In this case the leading edge is pointing into the oncoming airflow. The rest of the operating conditions for all computed cases are summarised in Table 5.4. For the yaw angle 0°, all cases summarised in Table 5.4 have been computed, whereas for the yaw angles 10° and 30°, only operating condition at $7m/s$ has been considered. The Reynolds number based on the tip chord c, and the relative tip velocity varies between 0.96×10^6 and 1.10×10^6 for the lowest and the highest wind speed. The spanwise variation of the Reynolds number is not significant, as the length of the chord c is decreasing while the relative velocity v_r is increasing.

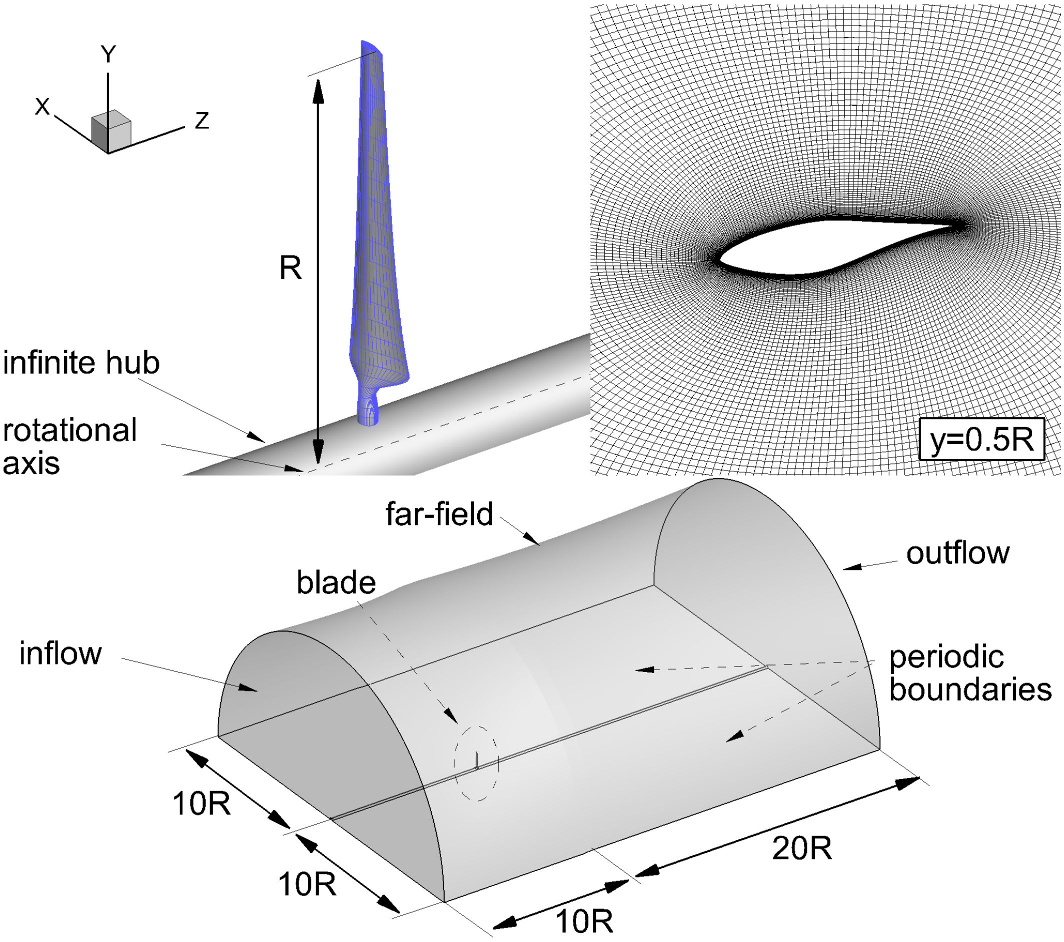


Figure 5.20: NREL Phase VI sector grid representation. Top Left: Blade view, only every fourth line in all directions is represented. Top right: near–airfoil area at 50 % of blade length. Bottom: complete domain.

Figure 5.20 depicts the three views of the sector grid used in the 0° yaw calculations. The 16–blocks grid has been generated using the NUMECA Autogrid5 grid generator and was split into 990 equal blocks of the size $32 \times 32 \times 16$ cells, resulting in the overall number of $16,220,160$ cells (medium). The grid topology around the airfoil consists of five blocks: the O–grid block, which is positioned around the airfoil and four H–grid blocks. First two H–grid blocks are positioned between the O–grid block and the inflow and outflow farfield boundaries. Whereas the other two are positioned between the O–grid block and the two periodic boundaries. The O–grid along the airfoil, depicted in the top right figure, consists of 256–cells on the airfoil and 96–cells in the normal–like direction. The distance d_w of the first grid points off the airfoil from the airfoil itself is about $1 \cdot 10^{-5}c$. This distance is constant along the whole blade span. The O–grid in the normal–like direction further expands for another 96–cells through the H–grid block, positioned between the O–grid block and the inflow. Similarly, the O–grid in the normal–like direction continues for 112–cells through the H–grid block, positioned between the O–grid block and the outflow. And lastly, the O–grid in the normal–like direction further continues for 48–cells, through the two H–grid blocks, positioned between the O–grid block and the two periodic boundaries. To ensure sufficient resolution in the tip region of the blade, the grid of the size 160×96–cells is placed at the tip. In spanwise direction, the grid has 128–cells on the blade surface and

96–cells from the tip to the lateral farfield. The grid on the blade surface was clustered towards the root and towards the tip. The maximum spacing in the spanwise direction on the blade surface was about $\Delta z = 1\cdot 10^{-1}c$. The minimum spacing achieved towards the tip was about $\Delta z = 1\cdot 10^{-5}c$, equal to the minimum distance from the airfoil surface in the plane of the airfoil itself. The grid in the lateral farfield was clustered towards the tip, with the minimum spacing of approximately $\Delta z = 1 \cdot 10^{-5}c$. The blade tip was modelled using sharp–tip topology. The grid coordinates are dimensional, and are expressed in m. The farfield boundaries have been positioned at $10R$ from the rotor centre at the inflow and in the lateral farfield boundary, and at $20R$ from the rotor centre at the outflow. Coarse grid has also been obtained from medium grid, by removing every second line in all three directions.

In all calculations characteristic boundary conditions are used for the inflow, outflow and farfield faces of the cylinder. A no–slip condition is applied on the blade in the rotating frame. In the 0° yaw case periodic boundary conditions are applied at the 180° cyclic boundaries, as depicted in the bottom plot of Fig. 5.20, and extrapolation boundary condition is applied at the infinite hub near the rotational axis. For the yaw angles 10° and 30°, the multi–frequency periodic boundary conditions are applied at the 180° cyclic boundaries. Instead of the infinite hub, a short hub considered as a zero–thickness cylindrical surface, has been created in the same fashion as the one depicted in the top left subplot of Fig. 6.33. The hub was surrounded by the computational grid from both sides, and inviscid wall boundary condition has been prescribed from both sides.

In all simulations, the minimum nondimensional wall distance y^+ was found to be smaller than 1 for the medium grid and about 1 for the coarse grid. To account for low–speed velocities of all considered cases, the low–speed preconditioning [92] has been used. All 0° yaw calculations have been performed using the steady $k - \omega$ SST solver of COSA, whereas the 10° and 30° yaw calculations have been performed using the HB $k - \omega$ SST solver of COSA. The CFL number of all cases has been set to 3, and the CFL ramping has been used. Due to numerical instabilities encountered with the MG solver, either with steady or HB solvers, all simulations were run using a single grid level and $30,000$ iterations were performed to compute the steady and HB solution. With this setup, the residuals of the NS and $k - \omega$ SST equations decreased by about 6 orders of magnitude, and all force and moment components converged within $25,000$ iterations.

5.5.2 Aerodynamic analyses of 0° yaw case

Table 5.5: Domain size study of the Phase VI wind turbine for the three different grids: reference, halved, doubled.

domain	inflow (R)	far–field (R)	outflow (R)	$\Delta F_z(\%)$	$\Delta M_z(\%)$
reference	10	10	20	–	–
halved	5	5	10	0.55	0.99
doubled	20	20	40	−0.08	−0.16

The sensitivity of the solution to the size of the computational domain has been examined by varying the distance of the farfield boundaries from the rotor centre. The medium grid described above has been used as a reference grid. Inflow, outflow and far–field faces of the cylinder have all been varied simultaneously. In the first case these dimensions were halved, and in the second case they were doubled. The near–field grid was held constant, so to increase or decrease the domain size, the clustering of points was varied only in the far–field region. Table 5.5 presents the domain dimensions in terms of rotor radius R and the % variation of the thrust (ΔF_z) and the torque (ΔM_z). The results reveal that the variation between the reference and halved domain was about 0.6 % for ΔF_z and nearly 1 % for ΔM_z. Whereas, it was less than 0.1 % for ΔF_z and less than 0.2 % for ΔM_z for the doubled domain. We may conclude, that the differences between the reference and doubled domain are very small. Whereas the differences between the reference and halved domain are non negligible. Therefore, the domain dimension of the reference grid is sufficient to minimise the effects of the imposed boundary conditions on the solution.

In all aerodynamic analyses for the yaw angle 0°, the experimental data for the integrated loads were averaged in such way that the azimuthal positions between 120° and 240° were excluded, as suggested in [35]. This was done in order to make a fair comparison between the experimental data and CFD calculations, thereby to remove any influence of the tower on the solution.

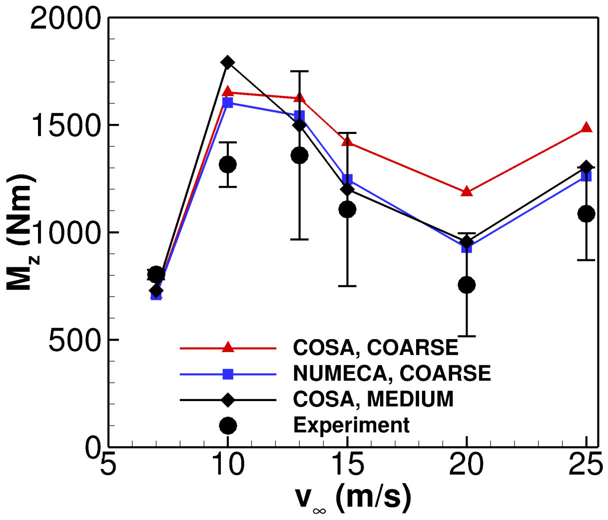


Figure 5.21: Power curve of the NREL Phase VI turbine. For the experimental data, sample minimum and maximum are also plotted.

Here, the aerodynamic torque has been assessed and compared with the experimental data and the commercial CFD code NUMECA FINE. Figure 5.21 depicts the two COSA solutions (medium and coarse), the experimental data which include error bars of the maxima and minima, and the NUMECA solution of the coarse grid. NUMECA FINE has also used the $k-\omega$ SST turbulence model and the low–speed preconditioning. The vast majority of numerical parameters between COSA and NUMECA have been matched, with the exception of numerical scheme. In COSA, the 2^{nd} order upwind scheme in conjunction with van Albada limiter has been used. In the attempt to perform the NUMECA calculations with the same scheme, the numerical instabilities were encountered, therefore, all the NUMECA calculations have been run using the central scheme. The CFL number has been set to 2, and all simulations were run using a single grid level, due to numerical instabilities when using the MG solver. With this set–up, the NUMECA calculations have performed $30,000$ iterations, in order to compute the steady solution. The residuals of the NS and $k-\omega$ SST equations decreased by about 5 orders of magnitude, and all force and moment components converged within $20,000$ iterations. As observed, the calculations performed with NUMECA converge slightly faster than those of COSA, eventhough the CFL number of the COSA simulations has been higher in all cases. This is probably due to the different numerical schemes used in both codes. It is generally known that in comparison to the upwind schemes, the central schemes are more robust, and thus, their convergence is faster. Looking

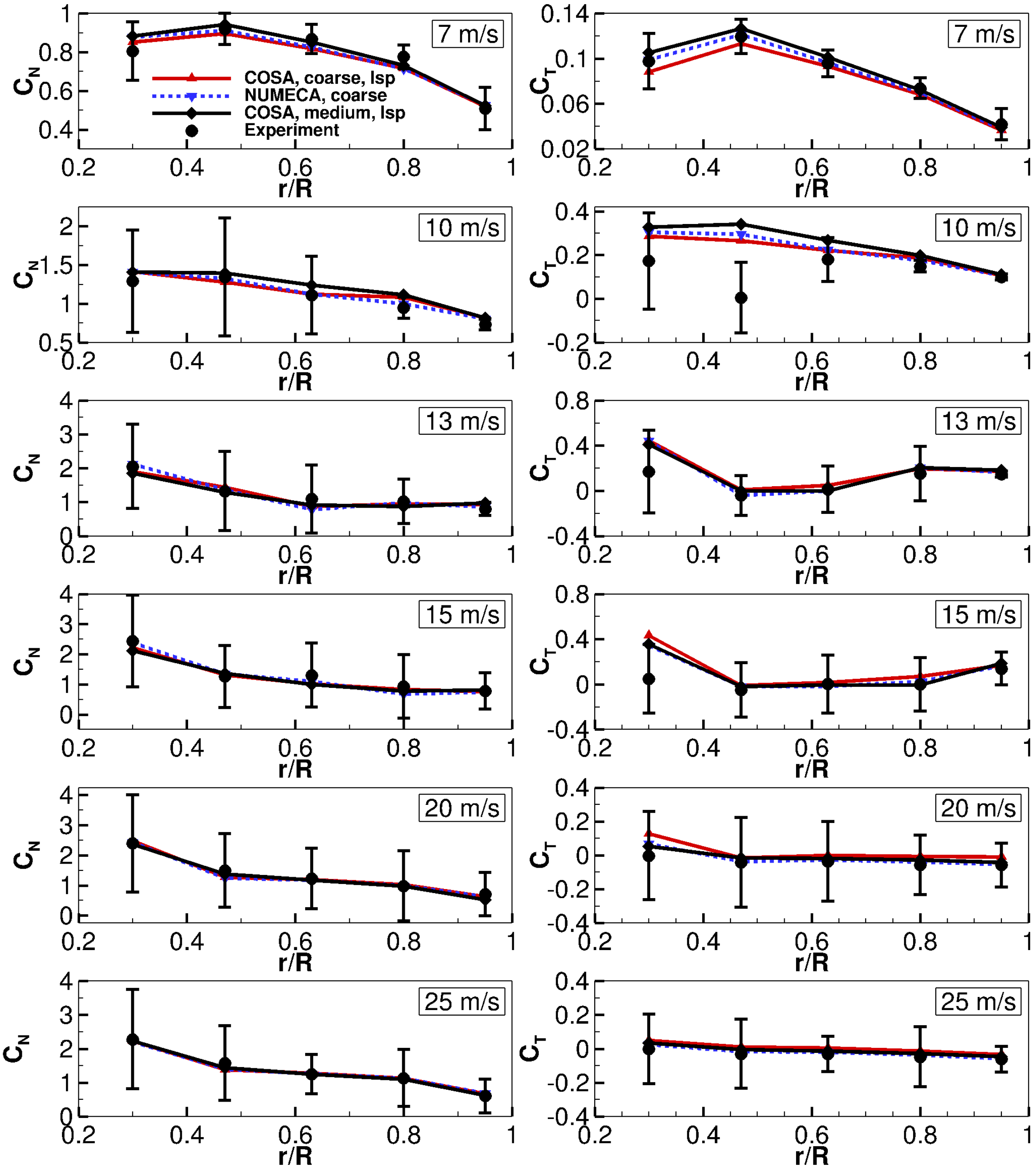


Figure 5.22: Spanwise distribution of the normal force coefficients (left) and the tangential force coefficients (right) for the six operating regimes of the NREL Phase VI blade: $7\,m/s$, $10\,m/s$, $13\,m/s$, $15\,m/s$, $20\,m/s$ and $25\,m/s$. For the experimental data, sample minimum and maximum are also plotted.

at Fig. 5.21, all three CFD simulations predict well the overall shape of the measured torque curve. For the lowest speed, the agreement between all thee numerical simulations and the experimental data is excellent. This is the condition, where the flow is completely attached. At 10 m/s, and 13 m/s, when stall appears, all three numerical simulations are still in good agreement, but they overpredict the aerodynamic torque, and thus the stall point. When the speed is greater than 15 m/s, the COSA solutions start to deviate. The COSA coarse grid solution predicts much higher torque than the COSA medium solution, this indicates that up to 13 m/s the COSA coarse grid is fairly grid–independent, but coarse grid–refinement becomes insufficient for velocities higher that 15 m/s. When looking at the NUMECA coarse grid solution for wind velocities higher than 15 m/s, it is also evident that the COSA and NUMECA solutions are no longer in good agreement. One reason could be the usage of different numerical schemes in the two codes. Furthermore, the low–speed preconditioners in the two codes are also different. COSA has used Merkle's type of low–speed preconditioner, whereas in NUMECA, Hakimi's preconditioner has been used. It is also important to mention that at higher wind speeds, the flow is heavily stalled, and therefore, the differences between COSA and NUMECA could also be due to the small differences in the implementation of the $k-\omega$ SST model. Some constants in NUMECA are also not publically available, and could therefore not be matched. One may notice that at higher wind speeds, the COSA medium grid is much closer to the experimental data than the COSA coarse grid. Even though the NUMECA coarse grid appears to be much closer to the experimental data than the COSA coarse grid, it is impossible to draw any conclusions, as the coarse grid–refinement appears to be very insufficient for higher wind speeds.

To provide the insight into the blade forces, the spanwise distribution of the normal (C_N) and tangential force (C_T) coefficients have been examined. The C_N and C_T were obtained integrating the pressure distributions around the airfoil, and represent the forces acting perpendicular and parallel to the airfoil chord, respectively. The integration procedure involved projection of the average pressure between two solution points onto the chord line, in order to obtain the C_N values, and projection of the average pressure onto an axis orthogonal to the chord to compute the C_T values. The definitions of C_N and C_T are provided respectively:

$$C_N = \sum_{i=1}^{n_s} \left(\frac{c_{p_i} + c_{p_{i+1}}}{2} \right) (x_{i+1} - x_i), \tag{5.4}$$

$$C_T = \sum_{i=1}^{n_s} \left(\frac{c_{p_i} + c_{p_{i+1}}}{2} \right) (y_{i+1} - y_i), \tag{5.5}$$

where the symbol n_s indicates the number of solution points along the airfoil. The first solution point was taken at the trailing edge ($x = 1c$), continuing in the direction over the upper surface of the blade, and then along the bottom surface, ending at the starting point at the trailing edge. The definition of c_p is provided below:

$$c_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty (u_\infty^2 + (r\Omega)^2)}, \tag{5.6}$$

where p and p_∞ denote the local and freestream static pressure respectively. Such definition of the pressure coefficient is based on the relative wind velocity and is a non–dimensional measure of the difference between the local and freestream pressure. At stagnation point c_p reaches its maximum value of 1. When plotting the pressure coefficient on the entire surface of the HAWT blade, it may be too complicated to use the above definition, therefore, in the denominator of Eqn. 5.6 the freestream wind velocity is usually used instead. In this case maximum value at the stagnation point is not known, and will vary along the span.

Using a similar integral procedure, one may determine also blade pitching moment coefficient (C_M). C_M represents the total moment about the 0.25 chord due to the C_N and C_T at each solution point with the vertical or horizontal distance from the pitch axis called the moment arm. The definition of the C_M is the following:

$$C_M = -\sum_{i=1}^{n_s} \left(\frac{c_{p_i} + c_{p_{i+1}}}{2} \right) \left[(x_{i+1} - x_i) \left(\frac{(x_{i+1} - x_i)}{2} + x_i - 0.25 \right) + (y_{i+1} - y_i) \left(\frac{(y_{i+1} - y_i)}{2} + y_i \right) \right] \tag{5.7}$$

In general, good agreement between all three simulations and the experimental data has been found for both the C_N and C_T, except for the C_T at 10 m/s. For 10 m/s, large deviations of the C_T have been observed at the inboard part of the blade. Some deviations of the C_T close to the root, have been also seen for the velocities 13 m/s and 15 m/s, but are still within the error bars of the experimental measurements. Looking at the behaviour of the C_T for the increasing wind speed, the data suggests that at 10 m/s, the numerical simulations fail to predict flow separation close to the root, and therefore overpredict C_T. For 13 m/s and 15 m/s, the C_T is still overpredicted close to the root, however, the data suggest that the numerical simulation is already able to predict some amount of stall, as the C_T is becoming closer to the experimental data. When the wind velocity further increases, the stalled area spreads, and the numerical simulations agree quite well with the experimental data.

Following figures represent the pressure coefficient c_p distributions of the three CFD simulations, alongside with the experimental data, at five spanwise positions, at $r/R =$ 0.3, 0.47, 0.63, 0.8 and 0.95. c_p is defined with Eqn. (5.6), and its definition is based on the relative wind velocity.

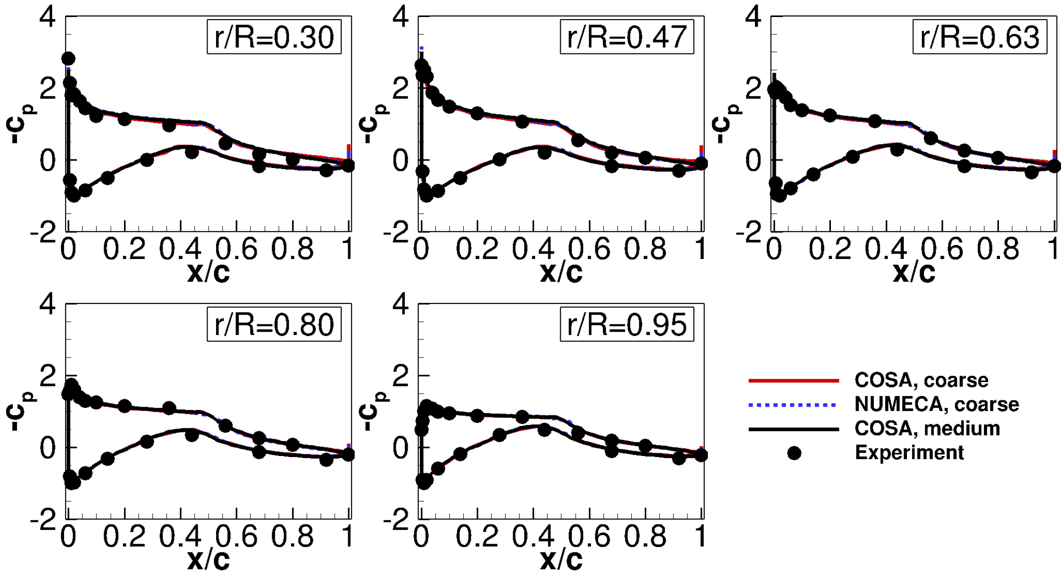


Figure 5.23: Pressure coefficient c_p distributions of the NREL Phase VI blade, for the 7 m/s case.

Figure 5.23 refers to the lowest considered wind speed, 7 m/s. Excellent agreement is found between all the three CFD simulations and the experimental data. No evident differences are observed between the three CFD solutions, indicating that the coarse grid is fairly grid–independent for this operating condition. Furthermore, the COSA and NUMECA agreement is excellent for this operating condition, despite of the different numerical schemes being used in the two codes.

Figure 5.24 refers to the operating condition corresponding to the wind speed 10 m/s. The COSA coarse, COSA medium and NUMECA coarse grid simulations are still in good agreement. However, as previously noted in Fig 5.22, the agreement between the experimental data and the three CFD simulations was poor in the inboard part of the blade. Looking at the c_p profiles in $r/R = 0.47$ subplot of Fig. 5.24, the experimental data clearly demonstrates there is a leading edge separation at position $r/R = 0.47$. The separation also affects the flowfield at position $r/R = 0.3$. It may also be noted that all CFD solutions fail to predict leading edge separation at position $r/R = 0.47$, but rather predict high suction peak.

Figure 5.25 refers to the operating condition corresponding to the wind speed 13 m/s. The experimental data suggests that the stalled area on the suction side of the blade is spreading, it now expands from the root of the blade until $r/R = 0.63$. Some differences within this area are observed between all the three CFD simulations. All three CFD solutions at positions $r/R = 0.8$ and 0.95 are still in good agreement with the experimental data. Although Fig. 5.21 shows that for higher wind speeds the torque computed with the NUMECA coarse grid simulation appears to be closer to the torque computed with COSA medium rather than COSA coarse, Fig. 5.25 indicates that this may be only a coincidence. Taking the area between the suction side and pressure side portion of c_p as a measure of the aerodynamic loading, one notice that the aerodynamic loading of the NUMECA coarse at the positions $r/R = 0.3$, 0.47 and 0.8 is bigger than for the COSA medium, whereas it is smaller for the other two positions.

Figure 5.26 refers to the operating condition corresponding to the wind speed 15 m/s. The experimental data suggests, that the stalled area on the suction side of the blade has now spread over most part of the blade. Only position at $r/R = 0.95$ is still unaffected. The CFD solutions predict stall well, except at position $r/R = 0.3$. It is also observed, that at position $r/R = 0.3$, all three simulations predict c_p differently. This suggests that for this operating condition, the flow conditions at this blade

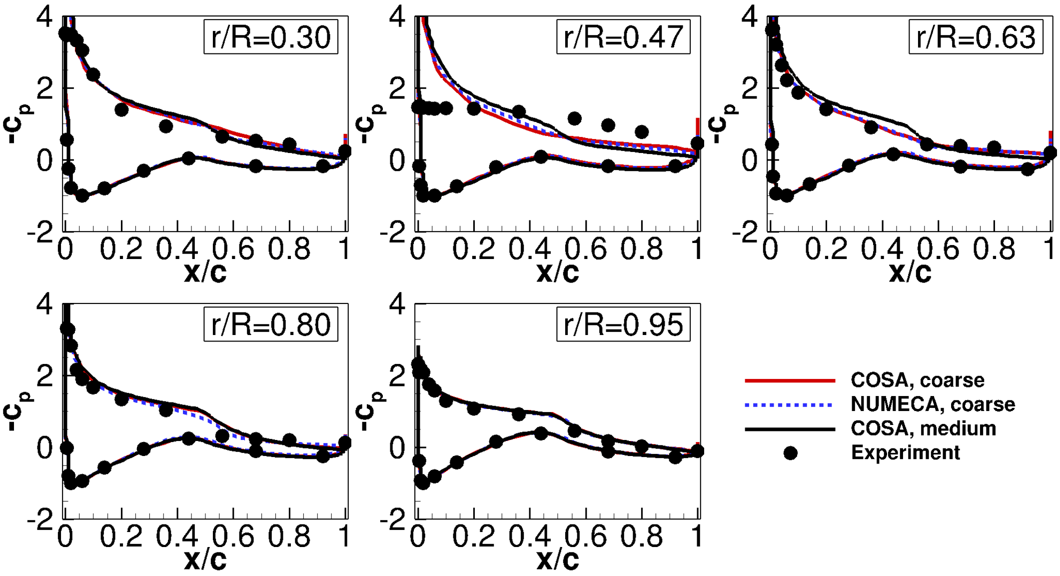


Figure 5.24: Pressure coefficient c_p distributions of the NREL Phase VI blade, for the 10 m/s case.

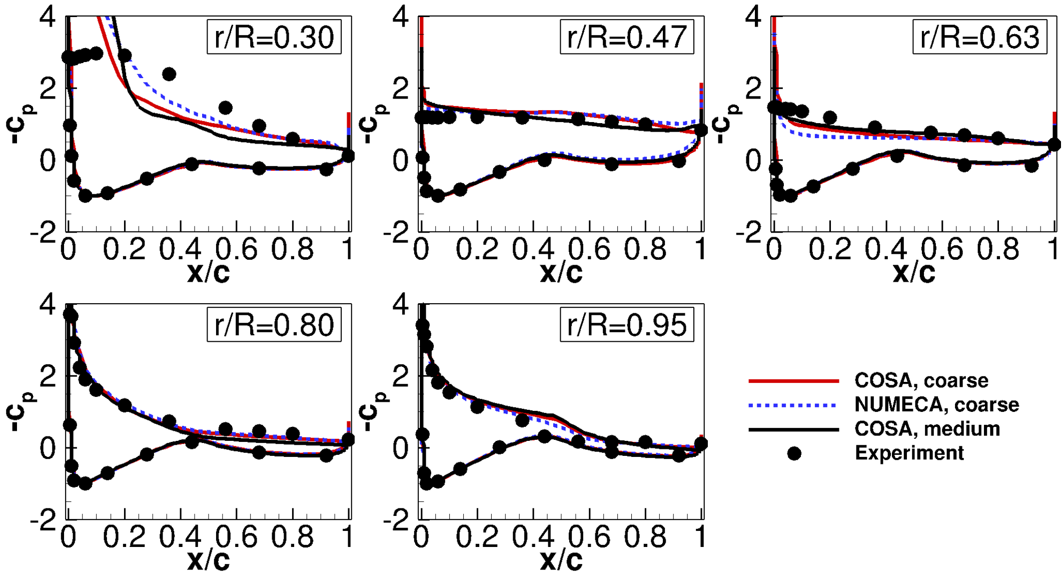


Figure 5.25: Pressure coefficient c_p distributions of the NREL Phase VI blade, for the $13\,m/s$ case.

length are very unstable.

Figure 5.27 refers to the operating condition corresponding to the wind speed $20\,m/s$. The data in all subplots suggest that the flow on the suction side of the blade, at this operating condition, is now fully stalled. Small differences can be observed between the three CFD simulations. However, the agreement between the experimental data and CFD simulations is now very good. For the positions $r/R = 0.3$ and 0.47, the COSA medium simulation is in better agreement with the experimental data than the other two CFD simulations.

Figure 5.28 refers to the operating condition corresponding to the wind speed

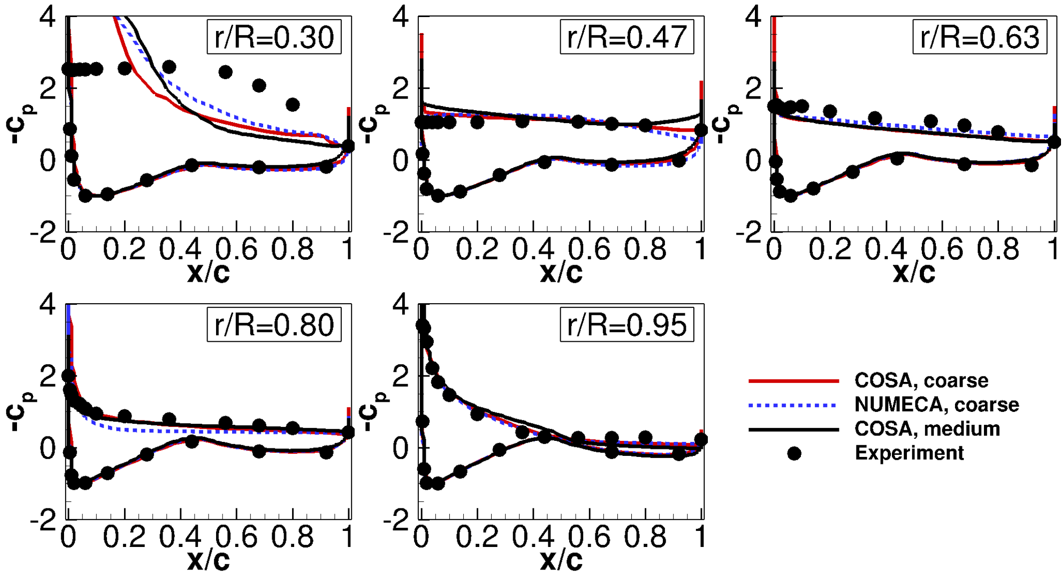


Figure 5.26: Pressure coefficient c_p distributions of the NREL Phase VI blade, for the $15\,m/s$ case.

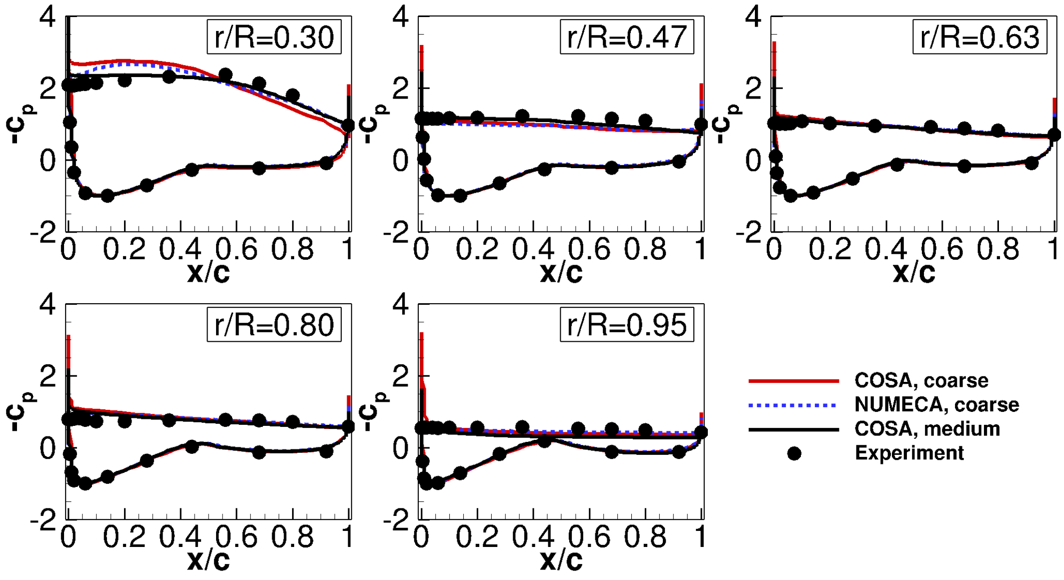


Figure 5.27: Pressure coefficient c_p distributions of the NREL Phase VI blade, for the $20\,m/s$ case.

$25\,m/s$. Also for this operating point, the flow on the suction side of the blade is under the fully stalled conditions. The three CFD simulations all show very good agreement with the experimental data. Very little visual differences are observed among the three CFD simulations, eventhough the torque prediction plot in Fig. 5.21 suggests otherwise.

Looking at Figs. 5.23–5.28, one may notice that at the tailing edge the discontinuities in pressure profiles may be observed. At 7 m/s and 10 m/s these discontinuities are still relatively small and may be observed only for the COSA and NUMECA coarse grid solutions. For the wind velocities 13 m/s–25 m/s, these discontinuities become much larger for both the COSA and NUMECA coarse grid solutions, and are now

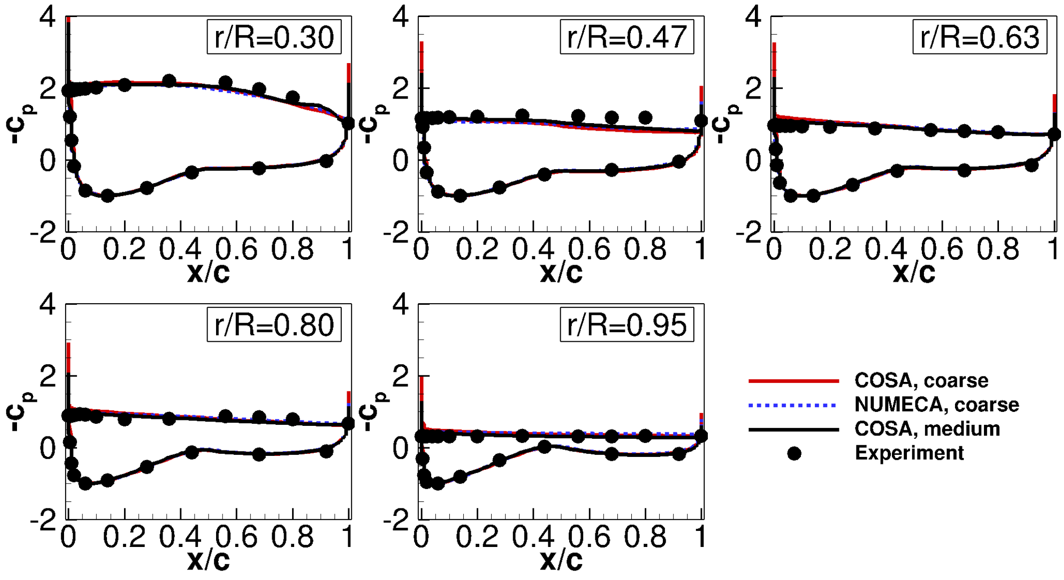


Figure 5.28: Pressure coefficient c_p distributions of the NREL Phase VI blade, for the $25\,m/s$ case.

visible also for the COSA medium grid solution. In all plots the biggest discontinuities are seen for the COSA coarse grid solution, following by the NUMECA coarse grid solution, and COSA medium grid solution. Cross comparison of the COSA coarse and NUMECA coarse grid solutions suggest, that either the wall boundary condition used in COSA causes higher discontinuities than the wall boundary condition used in NUMECA, or the numerical scheme of the NUMECA solver is less sensitive to these discontinuities. Furthermore, comparison of the COSA coarse and medium grid solution proposes the higher the grid refinement in rounded trailing edge area, the smaller the discontinuities.

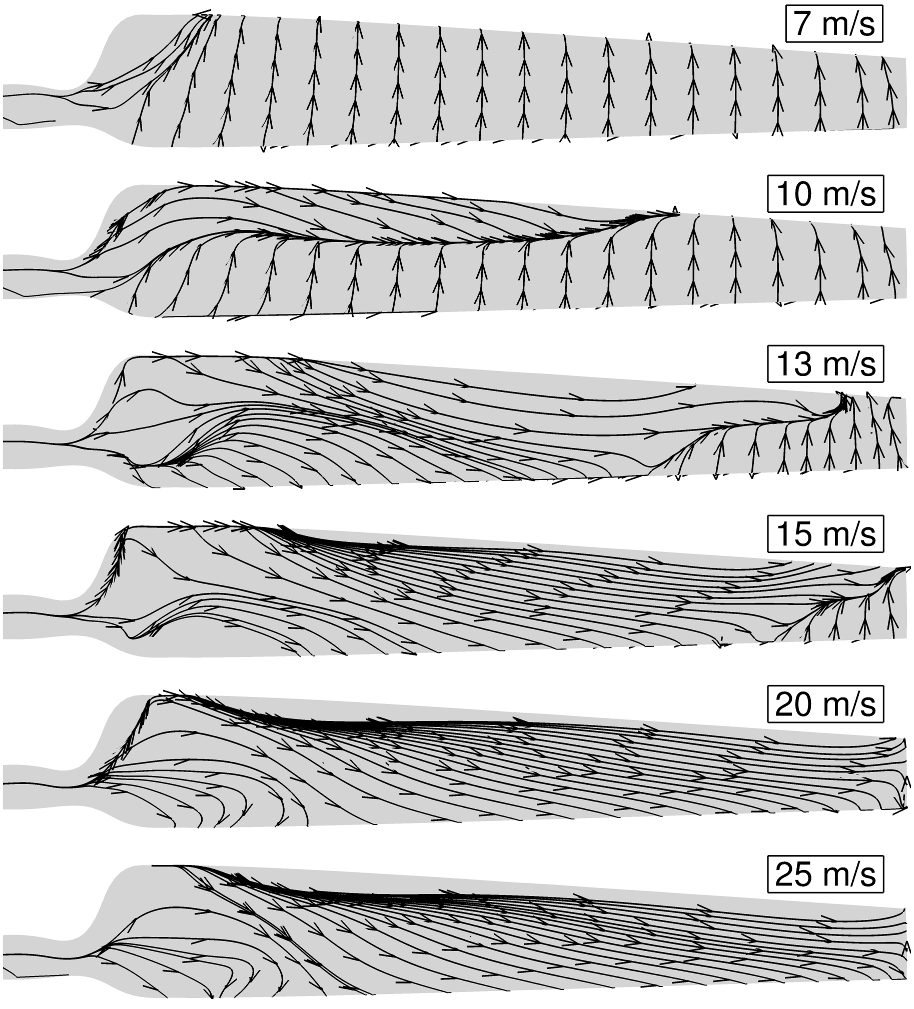


Figure 5.29: Skin friction lines on the suction side of the NREL Phase VI blade, computed with the COSA medium grid, for the six operating regimes: $7\,m/s$, $10\,m/s$, $13\,m/s$, $15\,m/s$, $20\,m/s$ and $25\,m/s$.

Figure 5.29 represents the comparison of the skin friction lines on the suction side of the NREL Phase VI blade for six operating conditions, all obtained with the COSA medium simulations. Looking at the $7\ m/s$ plot, the stream is well guided and kept aligned with the blade chord for the majority of the blade span, except for the blade sections close to the root and close to the tip. Close to the root, the skin friction lines turn towards the tip rather than being aligned with the chord, which indicates there is a small region of stalled flow conditions. Similarly, in the tip region, the skin friction lines turn towards the root, rather than being aligned with the blade chord.

This behaviour indicates the formation of the tip vortex, where the flow rolls over the tip from the pressure side to the suction side. The inclined skin friction lines are due to the flow acceleration, which is driven by the pressure difference between the suction and pressure sides of the blade. Passing from 7 m/s to 10 m/s, the small area of the stalled flow conditions near the root has now spread towards the tip, more than half blade is already affected. Furthermore, at 10 m/s, the radial pumping effect becomes apparent. At 13 m/s the region of stalled flow has nearly reached the tip region of the blade, whereas at 15 m/s the whole blade is affected. For 20 m/s and 25 m/s, the flowfield looks quite similar, the flow is separated everywhere. Skin friction lines close to the tip, as expected, are more inclined for 10 m/s, than in the 7 m/s case. This indicates that the tip vortex is becoming stronger. For higher velocities, the behaviour of the tip vortex seems to be heavily influenced by the stalled flow conditions. The comparison of the skin friction lines on the suction side of the NREL Phase VI blade between the three CFD codes, for all considered operating conditions, is provided in the appendix C.

In general, the obtained results in this subsection show good agreement between the numerical and experimental data, with the exception of the inboard part of the blade for the operating condition $10 m/s$. Some discrepancies were also found close to the root sections, for the operating conditions at $13 m/s$ and $15 m/s$. All these differences could be connected to very unstable flow conditions in this particular areas. Many reasons exist, why the flow behaviour here is not precisely captured with the CFD simulations. One of the possible reasons could be the lack of the laminar–to–turbulent transition. Furthermore, also the lack of accuracy in turbulence models for highly separated flows, which is a common problem for most linear eddy viscosity models, could be the reason for the observed differences.

5.5.3 Aerodynamic analyses of 10° and 30° yaw cases

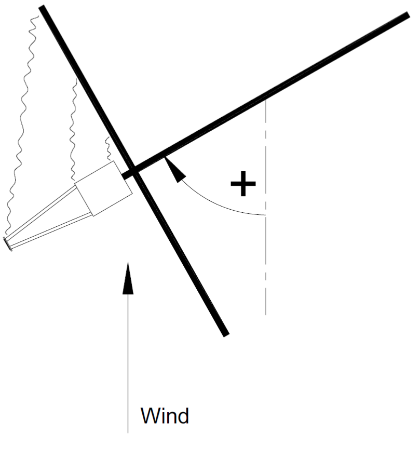


Figure 5.30: Top view of the boom and instrumentation enclosure wake interference. Taken from [9].

The periodic flow of the considered NREL Phase VI wind turbine has been computed herein, using the COSA HB solver, and validated against experimental data. The two yaw angles of $\delta = 10°$ and $\delta = 30°$ have been investigated, at the operating condition at 7 m/s, described in Table 5.4. The coarse grid described in subsection 5.5.1

has been used. All HB calculations have used the sector grid, and multi–frequency boundary conditions have been applied to the periodic boundaries.

In the yawed wind conditions, the experimental measurements were highly affected by the wake of the instrumentation boom and enclosures [9], as depicted in Fig. 5.30. The approximate area, which has been affected by the wake of the boom and instrumentation boxes or the tunnel flow with respect to the probe, is depicted in Fig. 5.31. Additionally, the experimental data have also been affected by the tower, and this region extends from about 120° and 240° azimuth for the 0° yaw case at wind velocity of 7 m/s [146]. In order to minimise uncertainties due to the affected measurements, the azimuthal positions at $\theta = 210°$, $\theta = 270°$ and $\theta = 330°$ have been considered. At $\theta = 210°$, some influence of the tower might still be visible, whereas the positions $\theta = 270°$ and $\theta = 330°$ are unaffected. Please note that all experimental data in this subsection were extracted and digitalized from [146], and are those conducted by the NREL in Phase VI wind turbine campaign [9].

At fist the COSA analyses for both yaw angles have been compared against the experimental data in terms of the normal (C_N) and tangential force (C_T) coefficients. The C_N and C_T were obtained by integration of the pressure distributions around the airfoil, using the Eqns. 5.4 and 5.5.

Figure 5.32 depicts spanwise distribution of the C_N and C_T for the yaw angle $\delta = 10°$. Only two HB calculations have been computed, as the solution between HB–1 and HB–2 does not change. HB simulations are denoted by the acronym HB, followed by the value of the retained number of harmonics $N_H = 1, 2$. To resolve the periodic flowfield for this operating condition, only one complex harmonic is required, as visible from all six subplots in Fig. 5.32. When comparing the HB–1 solution with the experimental data, an excellent agreement is found for both the C_N and C_T, for all three azimuthal positions. At all times the HB–1 data are within sample minimum and

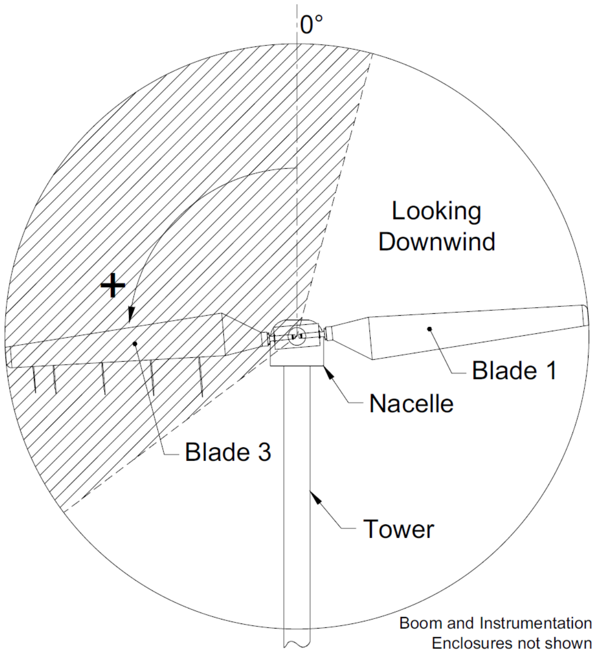


Figure 5.31: Front view of the Phase VI HAWT. Shaded area is significantly affected by the boom and instrumentation enclosure wake. Taken from [9].

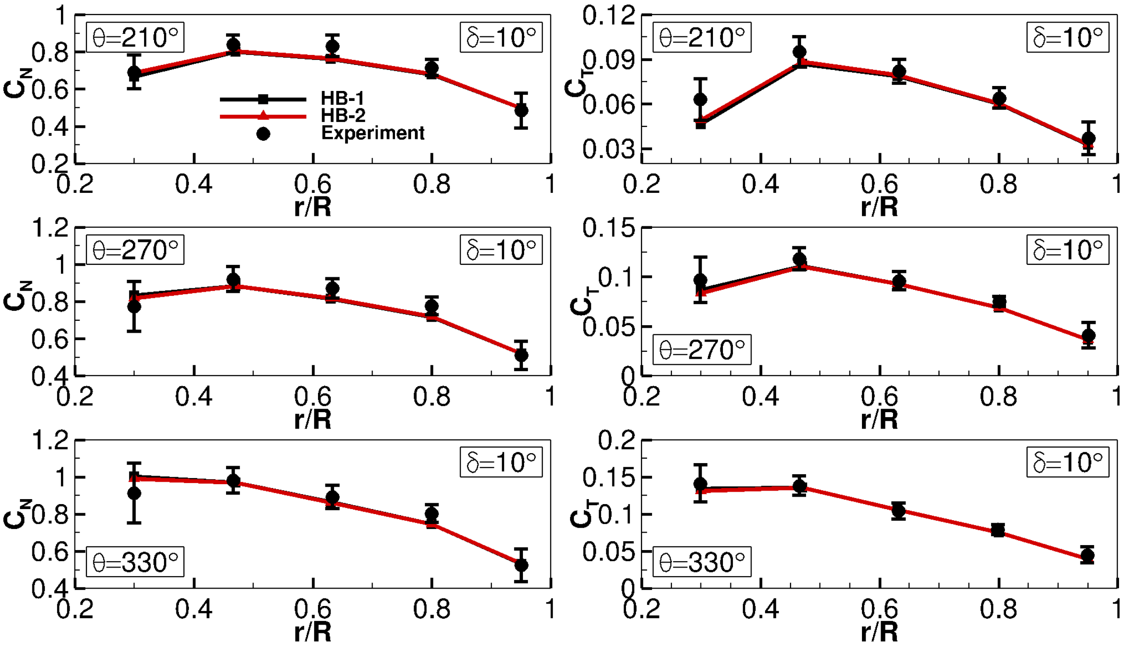


Figure 5.32: Spanwise distribution of the normal force coefficients (left) and the tangential force coefficients (right) at 7 m/s and the yaw angle $\delta = 10°$, for the three azimuthal positions of the NREL Phase VI blade: $\theta = 210°$, $\theta = 270°$, and $\theta = 330°$. For the experimental data, sample minimum and maximum are also plotted.

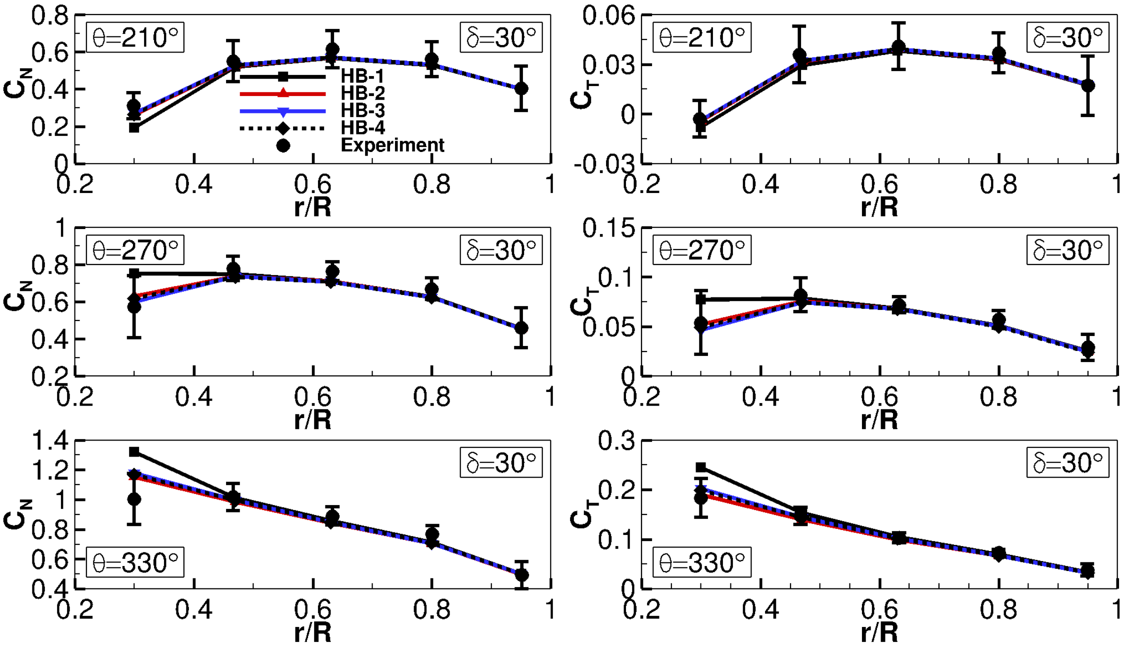


Figure 5.33: Spanwise distribution of the normal force coefficients (left) and the tangential force coefficients (right) at 7 m/s and the yaw angle $\delta = 30°$, for the three azimuthal positions of the NREL Phase VI blade: $\theta = 210°$, $\theta = 270°$, and $\theta = 330°$. For the experimental data, sample minimum and maximum are also plotted.

maximum region of the experimental data. The largest deviation of the C_N has been observed close to the root, at the azimuthal position $\theta = 330°$, whereas the largest deviation of the C_T has been observed close to the root, at the azimuthal position $\theta = 210°$.

Figure 5.33 depicts spanwise distribution of the C_N and C_T for the yaw angle $\delta = 30°$. Four HB calculations have been computed, in order to assess the minimum number of complex harmonics required. At this operating condition one complex harmonic is clearly insufficient, and to completely resolve the periodic flowfield for this operating condition, three complex harmonics are required. This is visible from the six subplots in Fig. 5.33. The HB–3 solution compares well with the experimental data. The comparison for the $\delta = 30°$ case is not worse than for the $\delta = 10°$ case. In some cases the data for the $\delta = 30°$ case seems to compare even better than for the $\delta = 10°$ case. The largest deviation of the C_N has been observed close to the root, at the azimuthal position $\theta = 330°$. For the C_T all spanwise positions agree well with the mean value of the experimental data.

Following figures represent the pressure coefficient c_p distributions of the HB–1 calculation, alongside with the experimental data, for the yaw angle of $\delta = 10°$, and azimuthal positions $\theta = 210°$, $\theta = 270°$ and $\theta = 330°$, at five spanwise positions, at $r/R = 0.3$, 0.47, 0.63, 0.8 and 0.95. c_p is defined with Eqn. (5.6), and its definition is based on the relative wind velocity.

Figure 5.34 refers to the c_p distributions of the yaw angle $\delta = 10°$, and the azimuthal position $\theta = 210°$. Excellent agreement is found between HB–1 simulations and the experimental data, for all five spanwise locations.

Figure 5.35 refers to the c_p distributions of the yaw angle $\delta = 10°$, and the azimuthal position $\theta = 270°$. The results for this azimuthal position are still in excellent agreement, especially for the spanwise locations $r/R = 0.47$-$r/R = 0.95$. c_p profile of the $r/R = 0.3$ seems to be slightly over predicted by the CFD simulation, however it still agrees well with the experimental data.

Figure 5.36 refers to the c_p distributions of the yaw angle $\delta = 10°$, and the azimuthal position $\theta = 330°$. Also for this azimuthal position, the agreement between HB–1 and

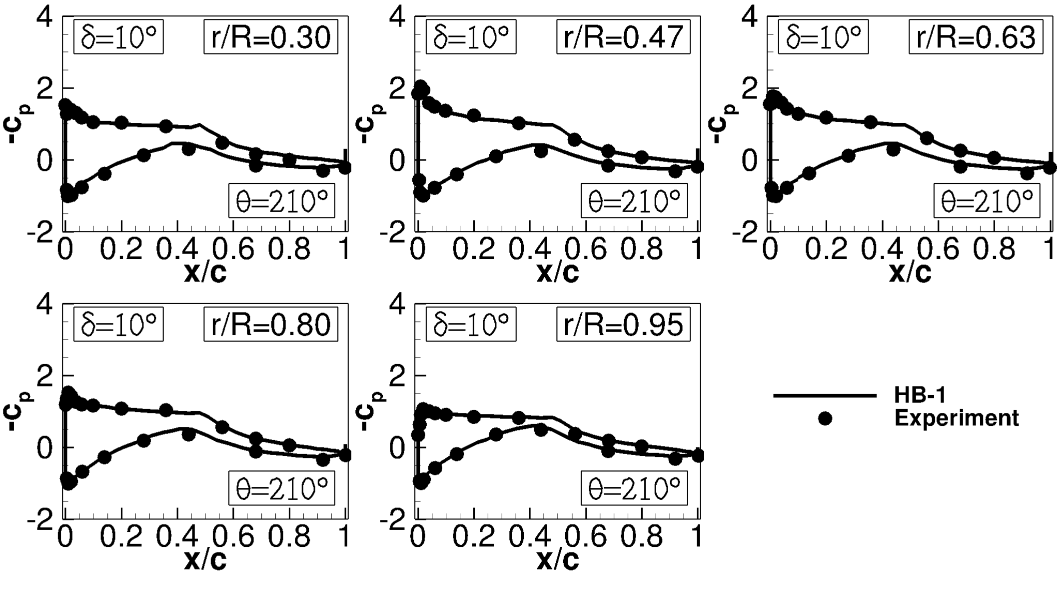


Figure 5.34: Pressure coefficient c_p distributions of the NREL Phase VI blade, for the 7 m/s case, the yaw angle $\delta = 10°$, and the azimuthal position $\theta = 210°$.

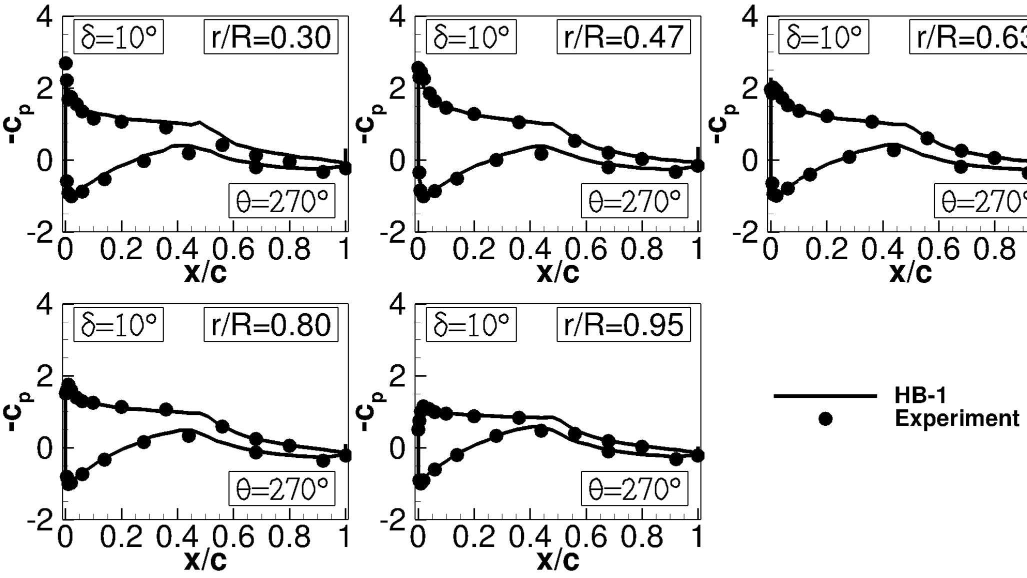


Figure 5.35: Pressure coefficient c_p distributions of the NREL Phase VI blade, for the $7\,m/s$ case, the yaw angle $\delta = 10°$, and the azimuthal position $\theta = 270°$.

the experimental data is excellent, for all five spanwise locations.

Following figures represent the pressure coefficient c_p distributions of the HB–3 calculation, alongside with the experimental data, for the yaw angle of $\delta = 30°$, and azimuthal positions $\theta = 210°$, $\theta = 270°$ and $\theta = 330°$, at five spanwise positions, at $r/R = 0.3$, 0.47, 0.63, 0.8 and 0.95. c_p is defined with Eqn. (5.6), and its definition is based on the relative wind velocity.

Figure 5.37 refers to the c_p distributions of the yaw angle $\delta = 30°$, and the azimuthal position $\theta = 210°$. For all five spanwise locations there is an excellent agreement between HB–3 simulation and the experimental data.

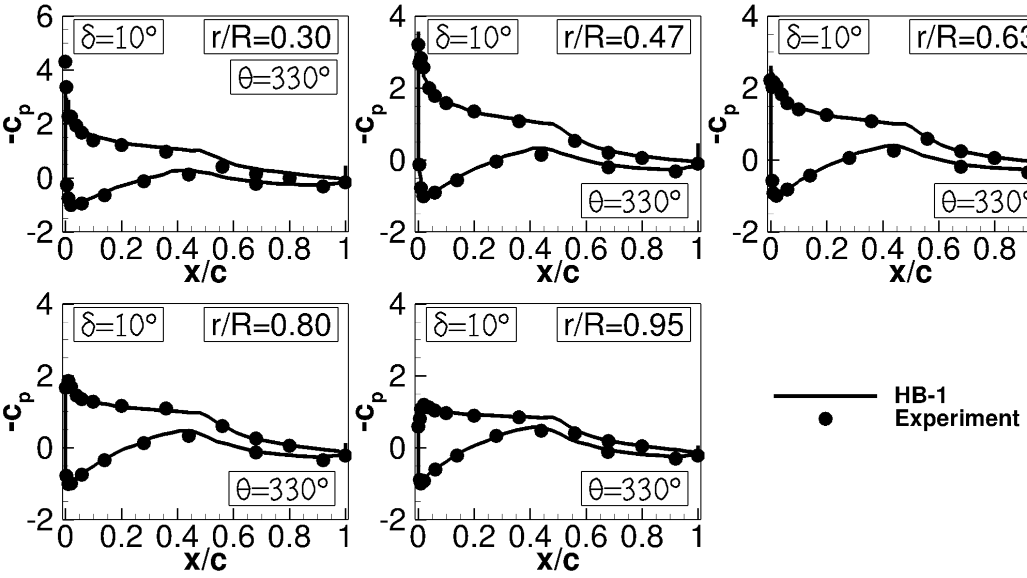


Figure 5.36: Pressure coefficient c_p distributions of the NREL Phase VI blade, for the $7\,m/s$ case, the yaw angle $\delta = 10°$, and the azimuthal position $\theta = 330°$.

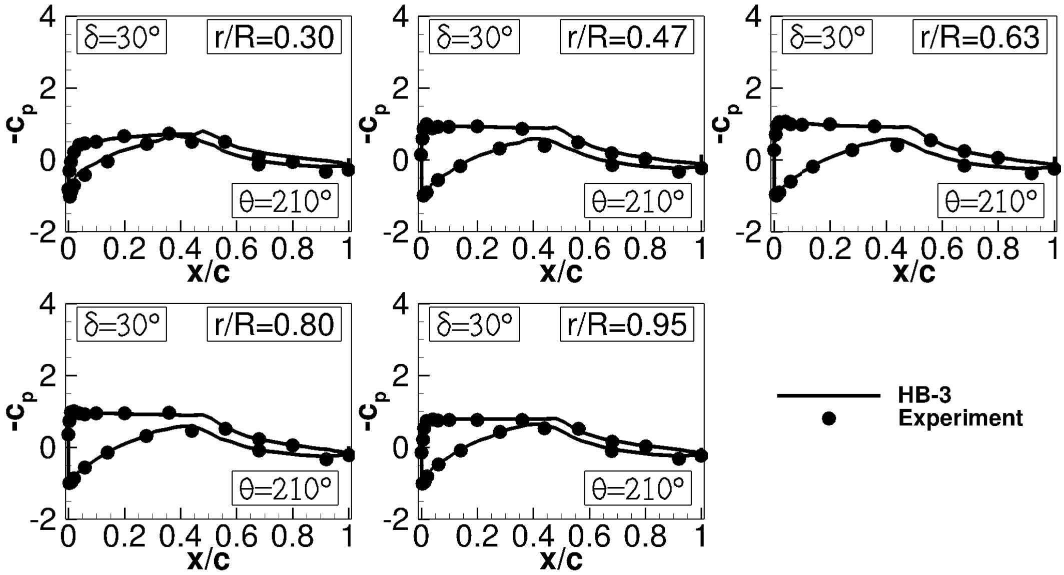


Figure 5.37: Pressure coefficient c_p distributions of the NREL Phase VI blade, for the $7\,m/s$ case, the yaw angle $\delta = 30°$, and the azimuthal position $\theta = 210°$.

Figure 5.38 refers to the c_p distributions of the yaw angle $\delta = 30°$, and the azimuthal position $\theta = 270°$. Other than slightly overpredicted c_p profile of the HB–3 simulation at the $r/R = 0.3$, the HB–3 results for this azimuthal position are still in excellent agreement with the experimental data, for the rest of the spanwise locations.

Figure 5.39 refers to the c_p distributions of the yaw angle $\delta = 30°$, and the azimuthal position $\theta = 330°$. Also for this azimuthal position, the agreement between the HB–3 solution and the experimental data is excellent, for all five spanwise locations. The only small discrepancy between the HB–3 and experimental data is observed at the

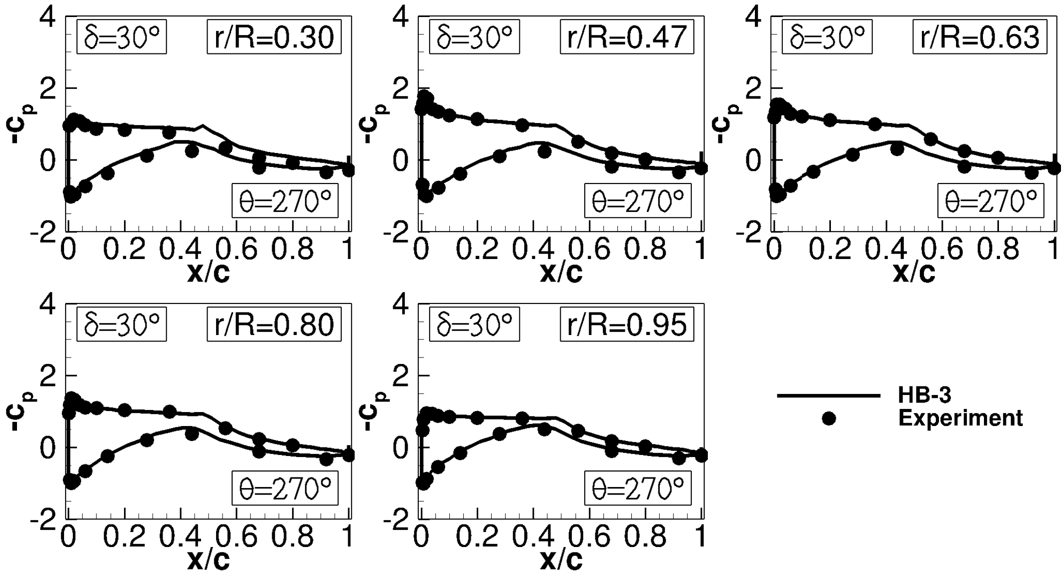


Figure 5.38: Pressure coefficient c_p distributions of the NREL Phase VI blade, for the $7\,m/s$ case, the yaw angle $\delta = 30°$, and the azimuthal position $\theta = 270°$.

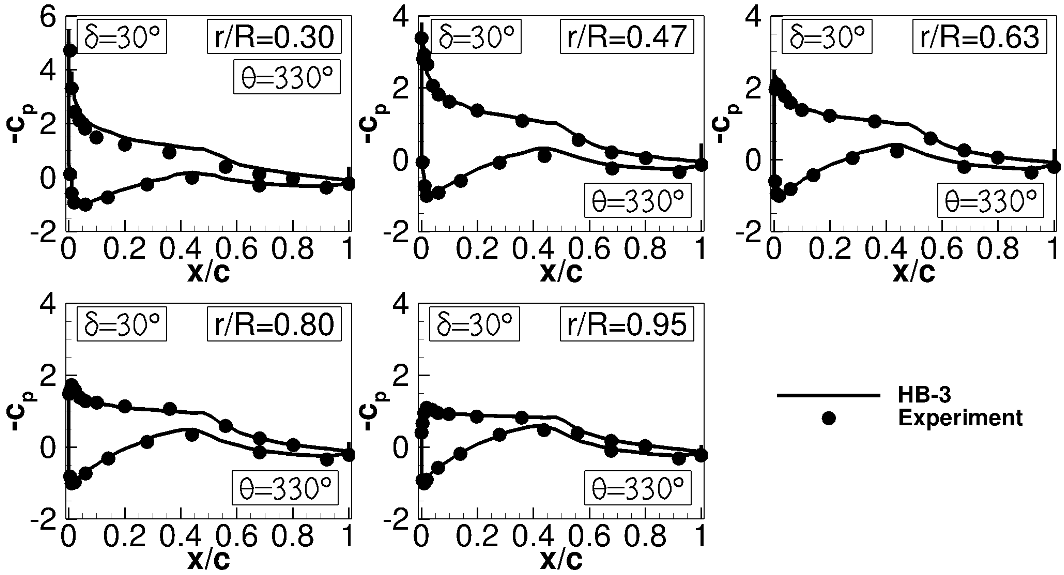


Figure 5.39: Pressure coefficient c_p distributions of the NREL Phase VI blade, for the $7\,m/s$ case, the yaw angle $\delta = 30°$, and the azimuthal position $\theta = 330°$.

suction side of the position $r/R = 0.3$. At this spanwise location the c_p is slightly overpredicted at the suction side of the airfoil.

The results obtained in this subsection show an excellent agreement between the numerical and experimental data for the operating condition in yawed wind flow at wind velocity $7\ m/s$. Two yaw angles of $\delta = 10°$ and $\delta = 30°$ at the azimuthal positions $\theta = 210°$, $\theta = 270°$ and $\theta = 330°$ have been analysed. These results have confirmed that the newly developed HB solver may capture well the unsteady HAWT flow effects, and it is well suited for yawed wind analyses.

Chapter 6

Results

This chapter focuses on numerical analyses of oscillating wing devices and horizontal–axis wind turbines. Both devices play a significant role in the development of efficient, clean, renewable energy generation systems. The chapter begins with the oscillating wing device basics and continues with the numerical set–up of oscillating wing device simulations. Two operating regimes are considered: one characterised by a high efficiency of energy extraction due to the occurrence of the leading edge vortex shedding (case A), and the other characterised by high efficiency of energy extraction in absence of the leading edge vortex shedding (case B). For both cases, 2D grid–independency study is first carried out, followed by a detailed numerical investigation into the impact of the flow three–dimensionality on the power generation efficiency. The numerical investigation is based on the comparative performance assessment of an infinite wing and two aspect ratio 10 wings, one featuring endplates, and the other featuring sharp tips. Under turbulent flow conditions the differences of unsteady hydrodynamic characteristics of the considered wing configurations are examined, discussed and quantified. The chapter continues with the definition of NREL 5–MW reference wind turbine, basic definitions of HAWTs and numerical set–up of the simulations of the 3D turbulent periodic flow past the rotating blade of NREL 5–MW reference wind turbine in yawed wind. This is a condition occurring when the freestream wind velocity is not orthogonal to the turbine rotor. The focus is first on a detailed aerodynamic discussion of this flow problem, followed by a detailed assessment of the actual benefits achievable by using the HB technology for the analysis of the wind turbine periodic aerodynamics. Time refinement analyses with the TD solver and spectral refinement analyses with the HB solver are performed to determine the speed–up of the HB simulation yielding a solution accuracy comparable to that of the fully resolved TD simulation.

6.1 Oscillating wing devices

6.1.1 Oscillating wing fundamentals

Motion description

An oscillating wing is defined as a foil experiencing simultaneous pitching $\theta(t)$ and heaving $h(t)$ motions. The following mathematical representation of the imposed motion is that adopted in [21]. Taking a pitching axis located on the chord line at position

x_p from the leading edge (LE), the foil motion is expressed as:

$$\theta(t) = \theta_0 \sin(\omega t) \rightarrow \Omega(t) = \theta_0 \omega \cos(\omega t) \tag{6.1}$$

$$h(t) = h_0 \sin(\omega t + \phi) \rightarrow v_y(t) = h_0 \omega \cos(\omega t + \phi) \tag{6.2}$$

where θ_0 and h_0 are respectively the pitching and heaving amplitudes, Ω is the pitching velocity, v_y is the heaving velocity, ω is the angular frequency and ϕ is the arbitrary phase angle between heaving and pitching. The freestream velocity is denoted by u_∞ and the angular frequency ω is linked to the vibration frequency f by the relationship $\omega = 2\pi f$. The prescribed oscillating motion is depicted in Fig. 6.1.

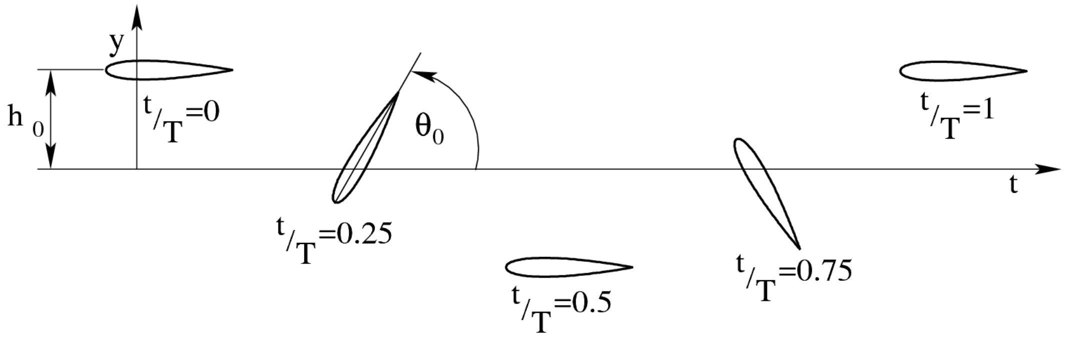


Figure 6.1: Prescribed motion of the oscillating wing for power generation.

Operating regimes

An oscillating symmetric foil can operate in two different regimes: propulsive or power–extracting mode. This distinction originates from the sign of the forces that the flow generates on the oscillating foil. Based on the imposed motion and the upstream flow conditions, the foil experiences an effective angle of attack (AoA) α and an effective velocity v_e given respectively by:

$$\alpha(t) = \arctan\left(-v_y(t)/u_\infty\right) - \theta(t) \tag{6.3}$$

$$v_e(t) = \sqrt{u_\infty^2 + v_y(t)^2} \tag{6.4}$$

The maximum values of α and v_e have a major impact on the amplitude of the peak forces in the cycle, and also on the occurrence of dynamic stall. The maximum effective AoA reached in the cycle is approximated by the modulus of its quarter–period value, that is $\alpha_{max} \approx |\alpha(T/4)|$. As explained in [21], the power–extracting regime (in a mean sense, over one cycle) occurs when $\alpha(T/4) < 0$, whereas the condition $\alpha(T/4) > 0$ corresponds to propulsive regime. The former and the latter conditions are represented respectively in the top and bottom sketch of Fig. 6.2, which provides a time–sequence viewed in a reference frame moving with the farfield flow at u_∞, so that the effective AoA $\alpha(t)$ is made visible from the apparent trajectory of the foil. The top sketch, also represents the resultant force $\underline{\mathrm{R}}$, which is constructed from typical lift and drag forces (right–hand side) and then decomposed into X and Y components (left–hand side). One sees that the vertical force component Y is in phase with the vertical velocity component v_y of the foil over the entire cycle. This implies that the wing extracts energy from the fluid as long as no energy transfer associated with the component X of the hydrodynamic force takes place. This is the case since the foil does not move horizontally. The hydrodynamic phenomena occurring during the wing oscillation are substantially more complex than the quasi–steady model discussed above. In some cases, for example, the efficiency of the energy extraction was shown to be heavily influenced by the occurrence of unsteady leading edge vortex shedding (LEVS) associated with dynamic stall and the phase between LEVS and foil kinematics.

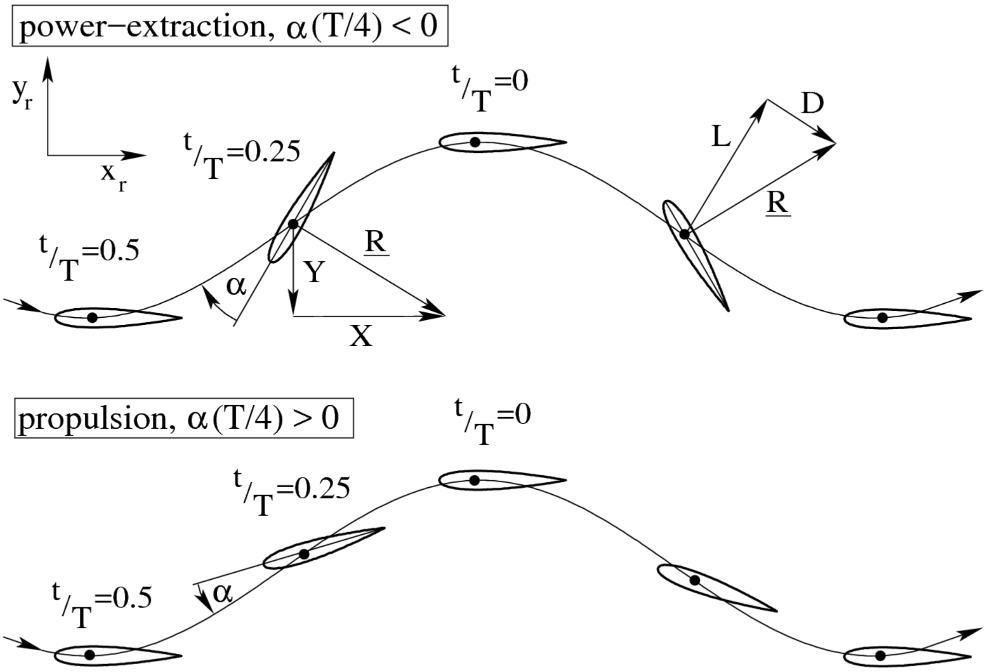


Figure 6.2: Foil motion in reference system moving with freestream velocity. Top sketch: power–extraction regime. Bottom sketch: propulsion regime.

Power and efficiency

The instantaneous power extracted from the flow is the sum of a heaving contribution $P_y(t) = Y(t)v_y(t)$ and a pitching contribution $P_\theta(t) = M(t)\Omega(t)$, where M denotes the hydrodynamic torque acting on the wing computed about the wing axis through the pitching center x_p. Denoting by c the foil chord, and z the coordinate along the wing span with origin at midspan, the power coefficient per unit wing length at position z is defined as:

$$C_{P_z}(t) \equiv P_z/(\frac{1}{2}\rho_\infty u_\infty^3 c) \tag{6.5}$$

where P_z is the sum of the instantaneous pitching and heaving power per unit wing length. The overall power coefficient, *i.e.* the nondimensional expression of the overall power extracted by the entire wing at time t is instead:

$$C_P(t) = \frac{1}{2l}\int_{-l}^{l} C_{P_z} dz \tag{6.6}$$

where l denotes the wing semispan. Introducing the time–dependent heaving force coefficient $C_Y(t)$ and pitching moment coefficient $C_M(t)$, defined respectively as $C_Y(t) = Y(t)/[\frac{1}{2}\rho_\infty u_\infty^2 c(2l)]$ and $C_M(t) = M(t)/[\frac{1}{2}\rho_\infty u_\infty^2 c^2(2l)]$, the nondimensional mean power produced over one cycle can be written as:

$$\overline{C}_P = \overline{C}_{P_y} + \overline{C}_{P_\theta} = \frac{1}{T}\int_0^T \left[C_Y(t)\frac{v_y(t)}{u_\infty} + C_M(t)\frac{\Omega(t)c}{u_\infty} \right] dt \tag{6.7}$$

In the analyses reported in section 6.1, use is also made of the heaving power coefficient per unit wing length $C_{P_{zy}}$ and the pitching power coefficient per unit wing length $C_{P_{z\theta}}$, obtained respectively by replacing P_z in Eqn. (6.5) with the instantaneous pitching and heaving power per unit wing length. Similarly, the time–dependent heaving power coefficient C_{P_y} and pitching power coefficient C_{P_θ} are obtained respectively by replacing C_{P_z} in Eqn. (6.6) with $C_{P_{zy}}$ and $C_{P_{z\theta}}$. For 2D problems, the expression of the time–dependent overall power coefficient C_P is provided by Eqn. (6.5), and the heaving and pitching power coefficients are computed using the same equation but considering separately the heaving and pitching power components.

The efficiency η of the power generation process is defined as the ratio of the extracted mean power $\overline{P} = \frac{1}{2}\overline{C}_P\rho_\infty u_\infty^3 c(2l)$ and the total available power $P_a = \frac{1}{2}\rho_\infty u_\infty^3 d(2l)$ in the oncoming flow passing through the swept area (the flow window):

$$\eta \equiv \frac{\overline{P}}{P_a} = \overline{C}_P\frac{c}{d} \tag{6.8}$$

where d is the overall vertical extent of the foil motion. This distance depends on the heaving and pitching motion parameters h_0, θ_0 and ϕ. The power extraction efficiency η defined by Eqn. (6.8) corresponds to the classical power coefficient obtained by means of Betz's analysis [147], which shows that the upper limit of η is $16/27 \times 100 \approx 59.3$ %. Therefore, Eqn. (6.8) provides the relationship between the mean power coefficient $\overline{C}_P$ defined by Eqn. (6.7) and Betz's theory power coefficient (η).

6.1.2 Physical and numerical set–up

The selected wing profile is the NACA0015 foil. The wing trajectory features a heaving and pitching motion component defined by Eqns. (6.1) and (6.2) respectively. Two operating conditions characterised by a high efficiency of energy extraction in the turbulent flow regime are considered. First high energy extraction efficiency operating condition (case A) is characterised by the occurrence of LEVS associated with the dynamic stall, described in [26]. Second high energy extraction efficiency operating condition (case B) does not experience the occurrence of LEVS and is described in [27].

Reynolds number based on the freestream velocity and the foil chord is $Re = 1.5 \times 10^6$, and is the same for both cases. The pitching center, which is at $x_p = 1/3$ of the chord from the LE, and the phase angle ϕ between heaving and pitching motions, which is 90.0^o, are also the same for both case A and case B. All other parameters are different for the two cases. The heaving amplitude h_0 of case A equals one chord and the pitching amplitude θ_0 is 76.3^o. This parameter choice for case A yields a value of the overall height h swept by the foil of 2.56 chords. The nondimensionalized frequency $f^* = fc/u_\infty$ for case A is 0.14, where f is the frequency in Hertz. The heaving amplitude h_0 of case B instead equals one and a half chord and the pitching amplitude θ_0 is 85^o. This choice of parameters for case B yields a value of the overall height h swept by the foil of 3.5 chords. The nondimensionalized frequency f^* for case B equals 0.16.

The time–dependent 3D turbulent flow fields past the oscillating wing were computed using structured multi–block non–deforming moving grids. In all simulations the entire grid moved rigidly with the wing. The 3D grid was obtained by extruding the 2D grid past the foil in the spanwise direction. The node coordinates of 2D and 3D grids were nondimensionalized by the foil chord.

The COSA code adopted for the analysis of this study is compressible, and therefore it requires prescribing the free stream Mach number on all far field boundaries. The free stream Mach number was set to 0.1, and this choice resulted in the maximum relative Mach number in the flow field never exceeding 0.3, the threshold above which compressibility effects may appear. Hence, the presented analyses do not include compressibility effects.

The required level of spatial refinement of the 3D grid in the foil plane was assessed by means of 2D simulations for case A and case B, using 256 time–intervals per oscillation cycle. More specifically, the periodic 2D flow field associated with the motion and flow parameters reported above was computed using four O–grids: one of dimension 256×256 (coarse) with 256 intervals on the foil and 256 intervals in the normal–like direction, and the other three of dimensions 512×512 (medium), 1024×1024 (fine), and 2048×2048 (extrafine). In all cases, the farfield boundary in the foil plane was at about 50 chords from the foil. On the extrafine grid level, the distance d_w of the

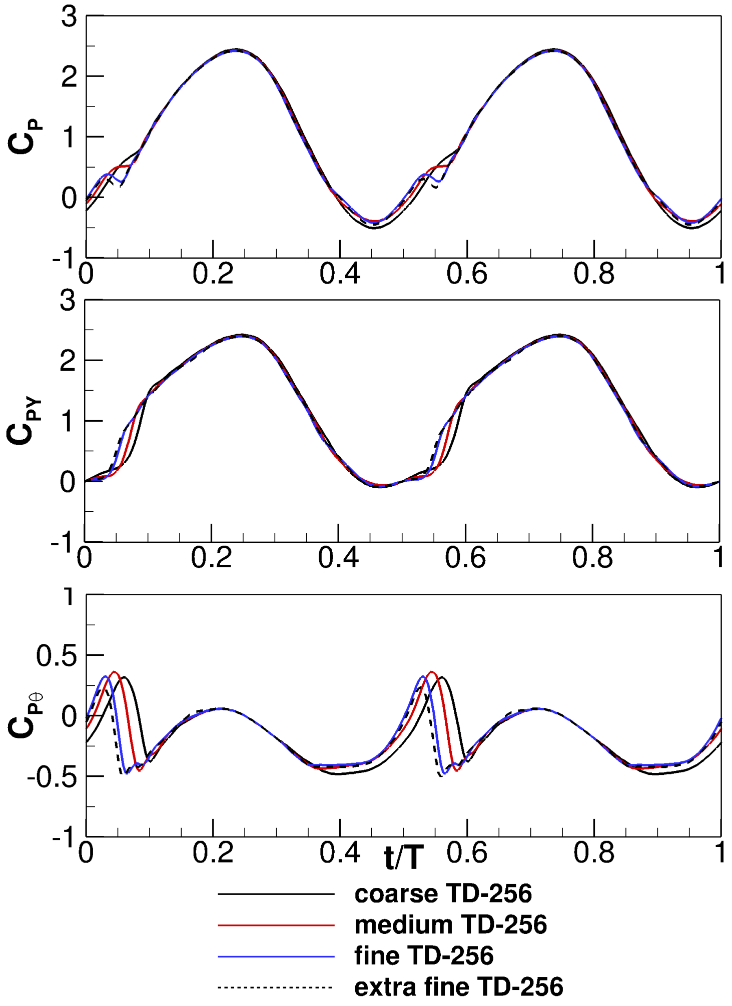


Figure 6.3: Mesh refinement analysis of case A: overall power coefficient (top), heaving power coefficient (middle), and pitching power coefficient (bottom) obtained using O–grid with coarse, medium, fine and extrafine refinement O–grids.

first grid points off the foil surface from the foil itself was about $8 \times 10^{-7}c$. The fine grid was obtained from the extrafine by removing every second line in both directions, and this approach was used recursively to also obtain the medium and coarse level grids. The periodicity error of the 2D simulations using these four grids and all other simulations of this study was assessed by monitoring the evolution of the heaving force coefficient C_Y. The simulations were run until the maximum difference between C_Y over the last two oscillation cycles became about 0.1 % of the maximum value of C_Y over the last cycle. It was chosen to monitor the periodicity error of C_Y because the vertical force component gives the highest contribution to the extracted power. The periodic profiles of the overall power coefficient C_P, the heaving power coefficient C_{P_y} and the pitching power coefficient C_{P_θ} resulting from the mesh refinement assessment for case A are reported respectively in the top, middle and bottom subplot of Fig. 6.3. Whereas the same plot for case B is reported by Fig. 6.4. The number 256 following the acronym TD indicates the number of time steps per period. It is noted that for case A some relatively small differences among the four C_P profiles exist in the first 10 % of the semi–period, and that such differences become progressively smaller as the grid refinement is increased. The same observation also holds for the four C_{P_y}

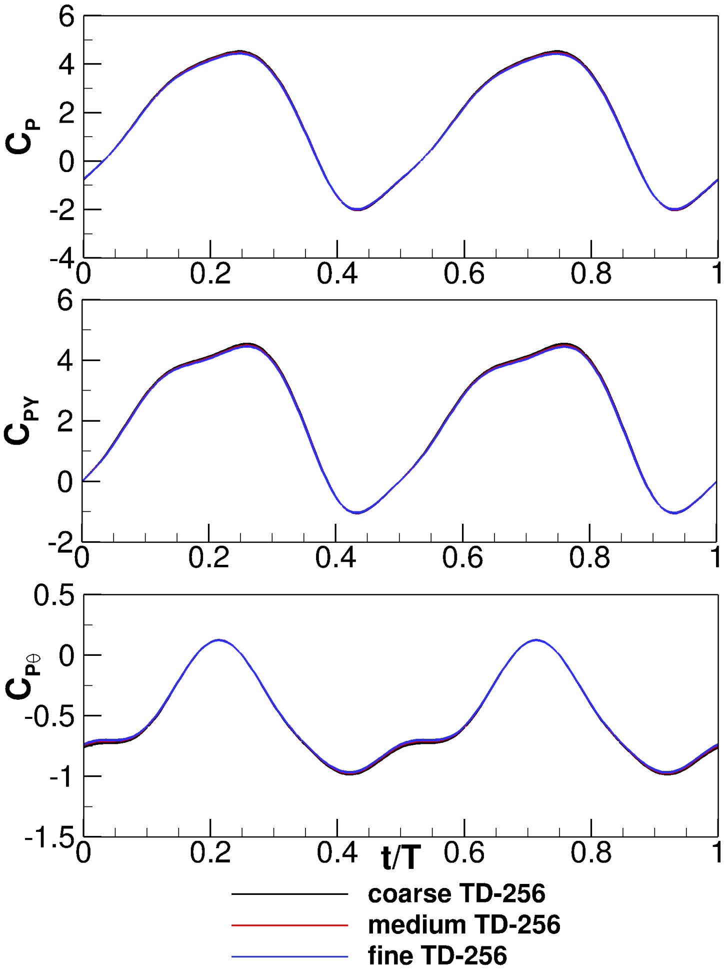


Figure 6.4: Mesh refinement analysis of case B: overall power coefficient (top), heaving power coefficient (middle), and pitching power coefficient (bottom) obtained using O–grid with coarse, medium and fine O–grids.

profiles. These discrepancies are caused by small variations of the phase of the LEVS associated with the considered regime [26] with respect to the foil oscillation. This occurrence is highlighted by the notable dependence of the position of the sharp peak of C_{P_θ} on the level of spatial refinement. As expected, these variations decrease as the grid is refined, and become very small when passing from the fine to the extrafine grid refinement, indicating that the 1024×1024 grid provides a fairly grid–independent solution. On the contrary, for case B no visual differences among coarse, medium and fine grid C_P profiles exist. Due to this reason, the extrafine grid computation was not performed. The same observation also holds for the three C_{P_y} and C_{P_θ} profiles.

Table 6.1: Mesh refinement analysis of case A and case B: mean overall, heaving and pitching power coefficients, and energy extraction efficiency η obtained using O–grid with coarse, medium, fine and extrafine refinement O–grids.

	case A				case B			
refinement	$\overline{C}_P$	$\overline{C}_{P_y}$	$\overline{C}_{P_\theta}$	$\eta(\%)$	$\overline{C}_P$	$\overline{C}_{P_y}$	$\overline{C}_{P_\theta}$	$\eta(\%)$
coarse	1.004	1.176	-0.172	39.19	1.577	2.100	-0.524	44.99
medium	1.015	1.176	-0.161	39.63	1.559	2.074	-0.516	44.48
fine	1.011	1.184	-0.173	39.45	1.546	2.058	-0.512	44.11
extra fine	0.998	1.188	-0.190	38.95	-	-	-	-

The mean values of the power coefficient profiles for case A and case B, depicted in Figs. 6.3 and 6.4 respectively, are reported in Table 6.1. The table also provides the efficiency η defined by Eqn. (6.8). One sees that for case A, the output featuring the highest sensitivity to the spatial refinement is the mean pitching power. However, due to significantly higher levels of heaving power, the variability of the overall mean power and the efficiency is significantly smaller. Some differences between the different spatial refinement also exist for case B. These differences are not visible in Fig. 6.4. Looking at the mean heaving and pitching power for case B, these relatively small differences are of the same order for both quantities. The differences are slightly higher between coarse and medium refinement, and they become smaller between medium and fine refinement. The same applies for the overall mean power and efficiency.

To assess the solution sensitivity to the level of temporal refinement the selected regime was simulated with the coarse–refinement grid using a number of time–intervals per period N_T of 128, 256, 512 and 1024. The periodic profiles of C_P, C_{P_y} and C_{P_θ} obtained with these four simulations are reported respectively in the top, middle and bottom subplot of Fig. 6.5 for case A, and in Fig. 6.6 for case B. For case A, similarly to the trend of spatial refinement, some relatively small differences among the four C_P profiles and among the four C_{P_y} profiles exist in the first 10 % of the semi–period. Such differences become progressively smaller as the time step decreases, and become practically negligible when passing from the TD–512 to TD–1024 analysis, indicating that 512 intervals per period are sufficient to achieve a solution independent of the time step. Similarly to what was highlighted in the assessment of the effect of the spatial resolution on the computed solution, the differences among the solutions obtained using 128, 256 and 512 steps per cycle are also due to small variations of the timing of the LEVS. These variations rapidly decrease as the grid is refined. Looking at Fig. 6.6,

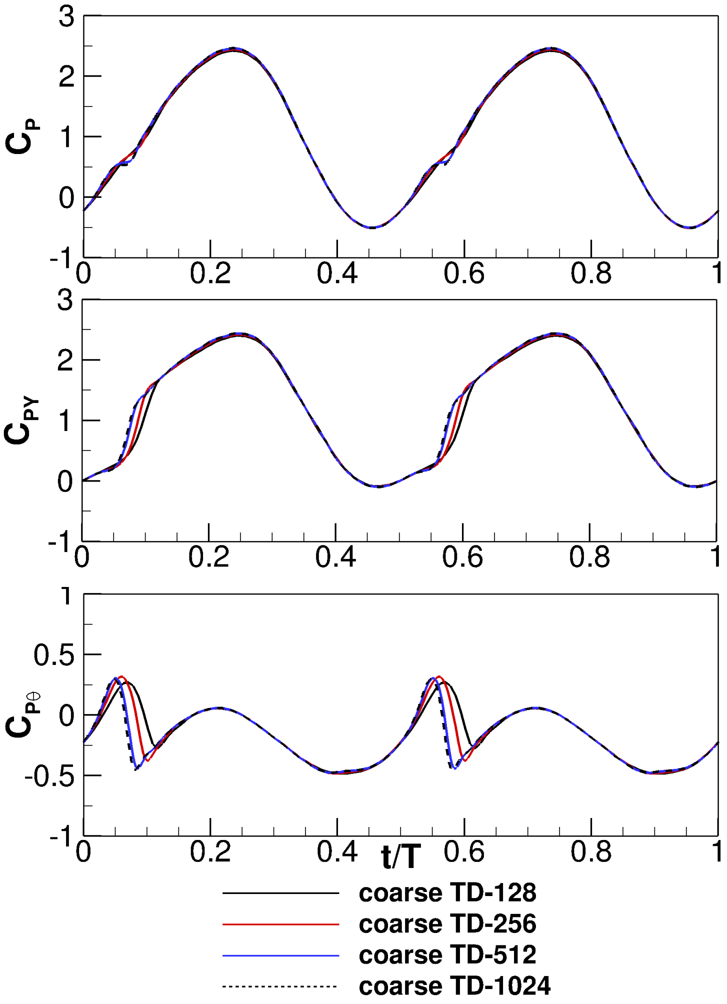


Figure 6.5: Time step refinement analysis of case A: overall power coefficient (top), heaving power coefficient (middle), and pitching power coefficient (bottom) obtained using 128, 256, 512 and 1024 steps per oscillation cycle.

no visual differences for different levels of temporal refinement for case B exist for all three periodic profiles, C_P, C_{P_y} and C_{P_θ}.

The mean values of the power coefficient profiles depicted in Figs. 6.5 and 6.6 are reported in Table 6.2, along with the efficiency η. Similarly as in the spatial refinement analyses, the output featuring the highest sensitivity to the refinement, for case A, is the mean pitching power. For case B, the differences, which are again not visible from the periodic profiles in Fig. 6.6, exist only between N_T of 128 and 256 and they become negligible when N_T is increased to 512.

The spatial and temporal mesh refinement analyses for case A highlight that the differences in case A, which are primarily due to small variations of the timing of the LEVS, are much bigger than in case B. The analyses suggest that a fully mesh–independent solution for case A is obtained using the 2D fine grid and 512 time intervals per cycle. Whereas for case B the 2D medium grid and 256 time intervals per cycle are sufficient to give a fully mesh–independent solution. However, to keep the computational cost of the 3D analyses within the size of the available resources, the 2D coarse grid was chosen as the foil planar mesh of the 3D grid, and 256 time intervals per cycle were used in the 3D simulations reported below. In the light of the findings

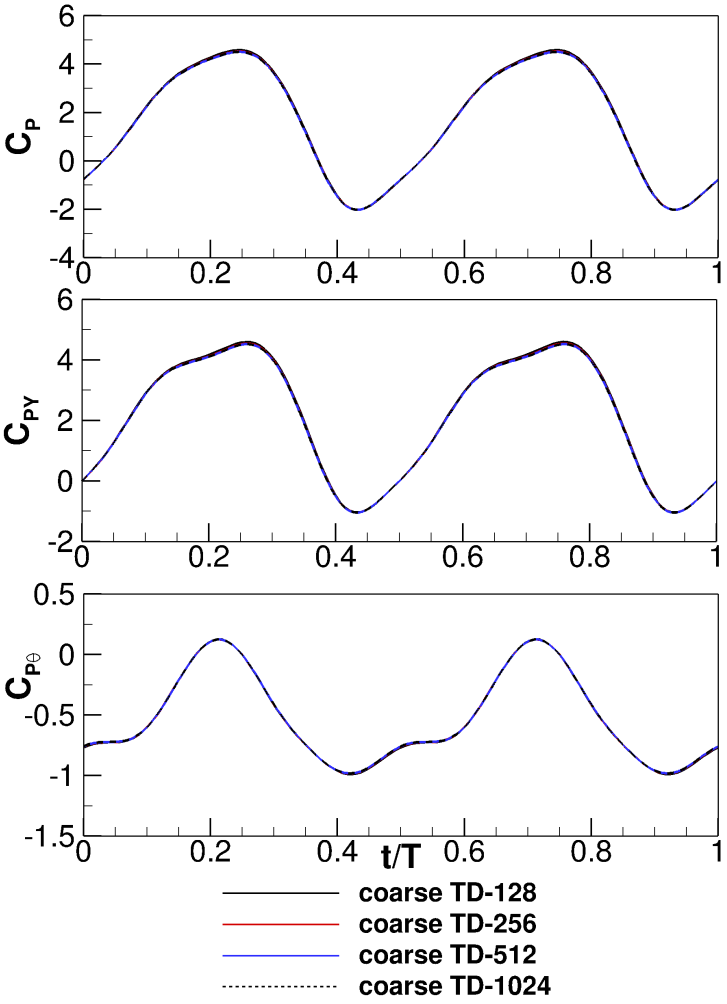


Figure 6.6: Time step refinement analysis of case B: overall power coefficient (top), heaving power coefficient (middle), and pitching power coefficient (bottom) obtained using 128, 256, 512 and 1024 steps per oscillation cycle.

Table 6.2: Time step refinement analysis of case A and case B: mean overall, heaving and pitching power coefficients, and energy extraction efficiency η obtained using 128, 256, 512 and 1024 steps per oscillation cycle.

	case A				case B			
N_T	$\overline{C}_P$	$\overline{C}_{P_y}$	$\overline{C}_{P_\theta}$	$\eta(\%)$	$\overline{C}_P$	$\overline{C}_{P_y}$	$\overline{C}_{P_\theta}$	$\eta(\%)$
128	0.984	1.149	-0.164	38.41	1.594	2.120	-0.527	45.49
256	1.004	1.176	-0.172	39.19	1.577	2.100	-0.524	44.99
512	1.015	1.203	-0.187	39.63	1.572	2.094	-0.523	44.85
1024	1.016	1.209	-0.191	39.66	1.573	2.095	-0.522	44.88

presented below, however, it is the author's view that the use of relatively coarse spatial and temporal grids for case A does not significantly affect the main conclusions of the investigations. For case B, the suggested spatial refinement is nearly grid–independent, and the temporal refinement is fully time–independent.

All 2D time–dependent simulations were performed using the MG solver with 3 grid levels. CFL ramping was used for all time steps, and the final CFL number reached equalled 4. 900 MG iterations were typically performed to compute the solution of each physical time. With this set–up, the residuals of the NS equations decreased by about 6 orders of magnitude at all physical times, and all force and moment components fully converged within 700 MG iterations.

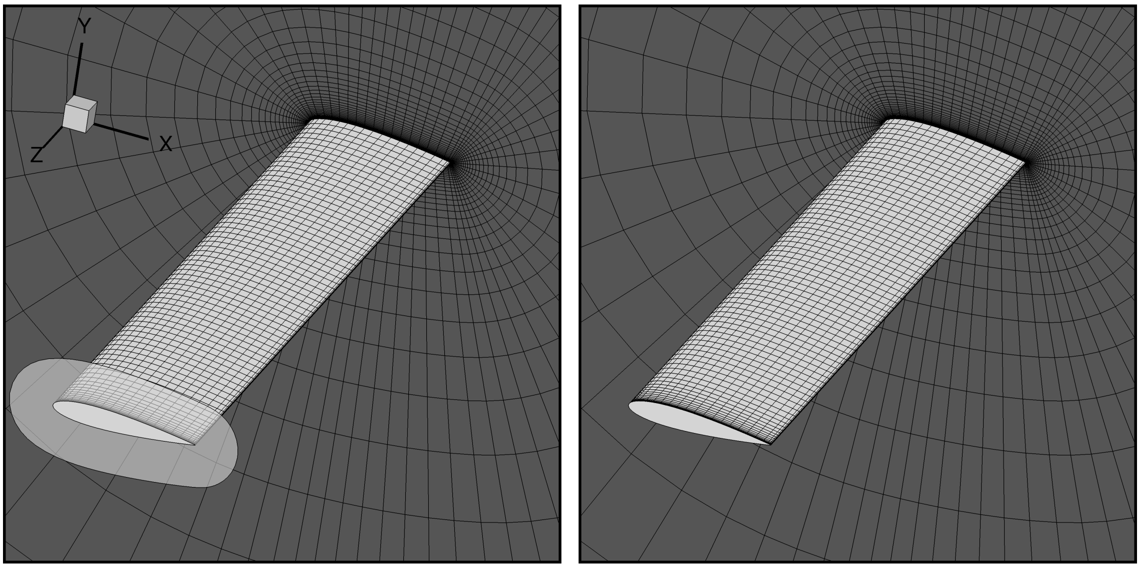


Figure 6.7: Surface mesh of wing and symmetry boundary (only every fourth grid line in all directions is reported). Top: wing with endplate. Bottom: wing with sharp tip.

The 3D simulations used a symmetry boundary condition at midspan to halve computational costs, and the 3D grid was built by stacking the 2D 256 × 256 O–grid in the spanwise direction from the midspan symmetry plane to the lateral farfield boundary, which was at 50 chords from the symmetry boundary. The AR of the wing was 10. Constant spanwise spacing $\Delta z = 0.02c$ was used from midspan to 90 % semispan, and from here the grid was clustered towards the tip achieving a minimum spacing $\Delta z = 6.4 \times 10^{-6}c$, equal to the minimum distance from the foil surface in the plane of the foil itself. The cell size increased again moving from the tip to the lateral farfield boundary. The grid featured 244 cells between the symmetry plane and the wing tip, and 144 cells between the wing tip and the lateral farfield boundary. The complete grid had $30,670,848$ cells.

Two wing tip topologies were considered, one with sharp tips, the other with endplates. The geometry of the endplate is depicted in Fig. 6.8. Careful grid design enabled the use of the same grid for both configurations, removing all uncertainty in the comparative analysis of these two configurations arising from using different grid topologies. A view of the surface mesh of the two 3D grids is provided in Fig. 6.7. In all simulations presented in this study, the minimum nondimensional wall distance y^+ was found to be smaller than one at all times of the periodic flow field.

The CFL number of all 3D simulations was set to 3. Due to numerical instabilities encountered with the MG solver, all simulations were run using a single grid level.

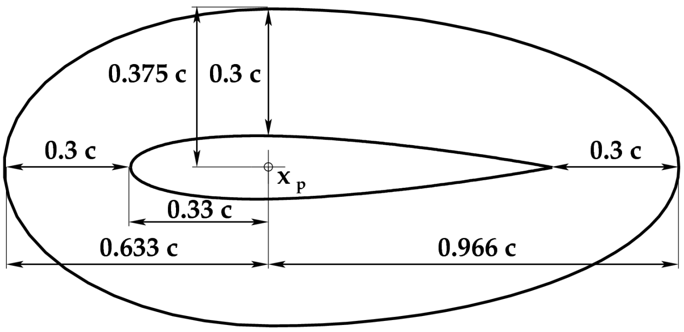


Figure 6.8: Endplate geometry.

The CFL ramping was used for all time steps, and $2,800$ iterations were performed to compute the solution of each physical time. With this set–up, the residuals of the NS equations decreased by about 4 orders of magnitude at all physical times, and all force and moment components fully converged within $2,500$ iterations. For both 2D and 3D analyses, the number of oscillation cycles typically required to achieve the 0.1 % periodicity error threshold on C_Y varied between four and ten, depending on the spatial and temporal refinement, and also on whether the simulation had been started from a freestream condition or from the solution of a simulation using the same grid but different temporal refinement.

As stated above, the number of MG iterations typically required in the 2D flow analyses for all force and moment components to converge was 700. The MG V–cycle consisted of 3 iterations on the fine grid, 3 iterations on the medium grid and 2 iterations on the coarse grid. Such number of MG iterations is comparable of about $2,400$ single–grid iterations. Given the fact that in the 3D calculation $2,500$ single–grid iterations were required for all force and moment components to fully converge, the runtime of the 3D calculations was not significantly penalised by not using MG in 3D analysis. It appears that MG method does not allow significant acceleration for high–Reynolds number flows. The difficulties encountered in solving such problems most likely occur due to the sophisticated 2–equation turbulence model employed for these calculations, and the very high aspect ratios of the computational grids required to resolve thin boundary layers at high Reynolds numbers. The stability and convergence rate of the explicit solver decreases as the grid aspect ratio increases. Such stability issues, when using sophisticated 2–equation turbulence models, could be limited by employing numerical stabilisation methods such as those reported in [148] and [149]. It was also observed that the convergence of the MG varies with the selected physical time step, the smaller the physical time step becomes, the faster the convergence.

6.1.3 Hydrodynamic analysis of case A

The infinite– and finite–span oscillating wing configurations analyzed herein share the same trajectory, which corresponds to that defined for case A in the previous subsection. The evolution of the main kinematic parameters of all wings over one oscillation cycle is depicted in Fig. 6.9. The plot shows the time–dependent values of the vertical position h of the wing, its angular position θ, the nondimensionalized heaving velocity v_y/u_∞, and the nondimensionalized pitching velocity Ω/Ω_{max}, with Ω_{max} being the maximum pitching velocity of the cycle. The figure also reports the effective AoA α computed with Eqn. (6.3). One notes that the maximum AoA is about 35^o. The four positions

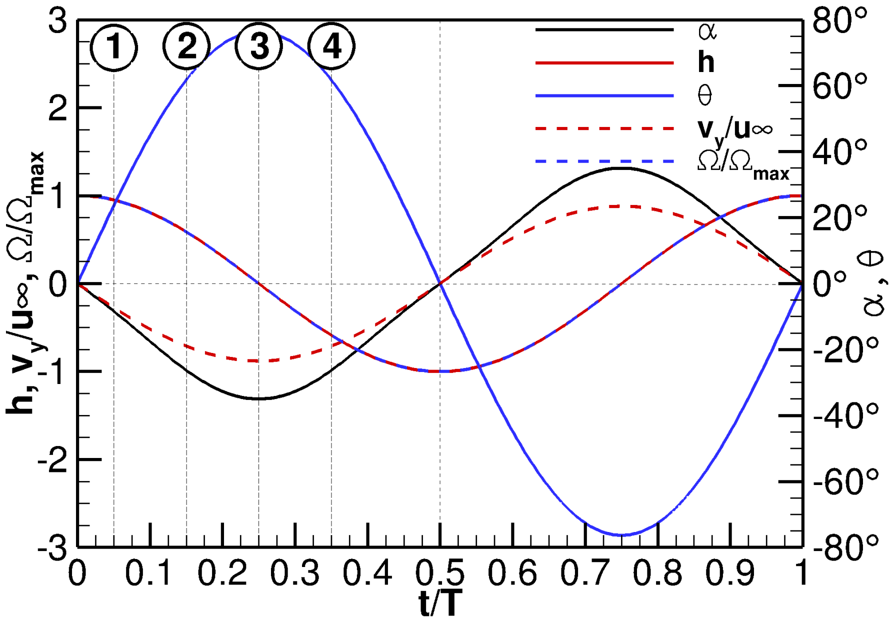


Figure 6.9: Kinematic parameters of the trajectory of the infinite– and finite–span wings of case A.

labelled 1 to 4 correspond to 5%,15%, 25% and 35% of the period respectively, and are those at which the flow field is examined in greater detail in the following analyses.

Table 6.3: Integral performance metrics of infinite wing and two AR 10 wings of case A. Columns 2 to 4: mean overall, heaving and pitching power coefficients; column 5: energy extraction efficiency η; columns 6 to 8: percentage variations of overall, heaving and pitching power coefficients of two AR 10 wings with respect to infinite wing values.

AR	$\overline{C}_P$	$\overline{C}_{P_y}$	$\overline{C}_{P_\theta}$	$\eta(\%)$	$\Delta\overline{C}_P(\%)$	$\Delta\overline{C}_{P_y}(\%)$	$\Delta\overline{C}_{P_\theta}(\%)$
∞	1.004	1.176	-0.172	39.19	–	–	–
10 EP	0.879	1.118	-0.239	34.32	-12.4	-4.9	-38.9
10 ST	0.835	1.074	-0.239	32.58	-16.8	-8.7	-38.9

The main integral performance metrics of the infinite wing, the AR 10 wing with endplates (EPs) and the AR 10 wing with sharp tips (STs) are reported and compared in Table 6.3. Columns 2 to 4 provide respectively the mean values of the overall power coefficient $\overline{C}_P$, the heaving power coefficient $\overline{C}_{P_y}$, and the pitching power coefficient $\overline{C}_{P_\theta}$; column 5 provides the overall efficiency η, whereas the percentage variations (Δs) of the three mean power coefficients of the AR 10 wings with respect to the reference values of the infinite wing are reported in columns 6 to 8. The infinite wing analysis is based on the 2D coarse grid TD–256 case A simulation, whereas the two AR 10 analyses are based on fully 3D TD–256 simulations using the $30,670,848$–cell grid described in subsection 6.1.2. One notes that $\overline{C}_P$ of the AR 10 wing with EPs is 12.4 % lower than that of the infinite wing, whereas $\overline{C}_P$ of the AR 10 wing with STs is nearly 17 % lower than that of the ideal infinite wing. The breakdown of the overall mean power into heaving and pitching power components for the three cases highlights that: *a*) the mean negative pitching power (a loss term) of both AR 10 wings increases by the same amount with respect to the ideal infinite wing case (about 39 %), *b*) the heaving power coefficient of both AR 10 wings decreases with respect to that of the infinite span wing: by about 4.9% for the wing with EPs, and by about 8.7% for the wing with STs. These observations highlight that 3D flow effects hit the overall energy extraction efficiency of this device in a complex manner, that appears not to depend only on the geometry of the wing tips.

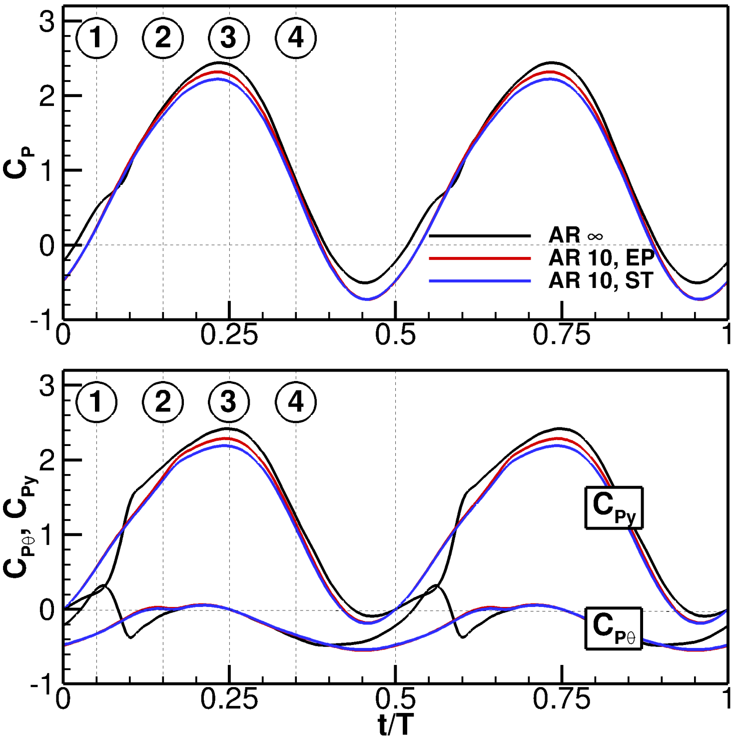


Figure 6.10: Overall power coefficient (top), heaving power coefficient (middle), and pitching power coefficient (bottom) of infinite wing and two AR 10 wings of case A.

The top subplot of Fig. 6.10 reports the C_P profiles of the three wings over one period, whereas the bottom subplot reports their C_{P_y} and C_{P_θ} profiles. In the first 10 % and last 15 % of both semi–periods (region 1), the AR 10 C_P profiles are superimposed and are significantly lower than that of the infinite wing. The C_P profiles of the AR 10 wings are lower than that of the infinite wing also in the remainder of the cycle (region 2), but the profile of the wing with EPs is higher than that of the wing with STs. The profiles of the pitching and heaving power coefficients in the bottom subplot of Fig. 6.10 show that in region 1 an increment of the heaving power of the two AR 10 wings with respect to the infinite wing is outweighed by a larger reduction of their pitching power. This explains why the profiles of the overall power coefficient of the finite span wings in region 1 are lower than that of the infinite wing. In region 2 the pitching power of all three wings is comparable, and the lower overall power of the two AR 10 is caused primarily by a reduction of the heaving power component, which is greater for the wing with STs.

To further investigate the dependence of the energy extraction efficiency of the finite span wing on the tip geometry highlighted by Table 6.3 and Fig. 6.10, the periodic profiles of the heaving power coefficient per unit wing length $C_{P_{zy}}$ and the pitching

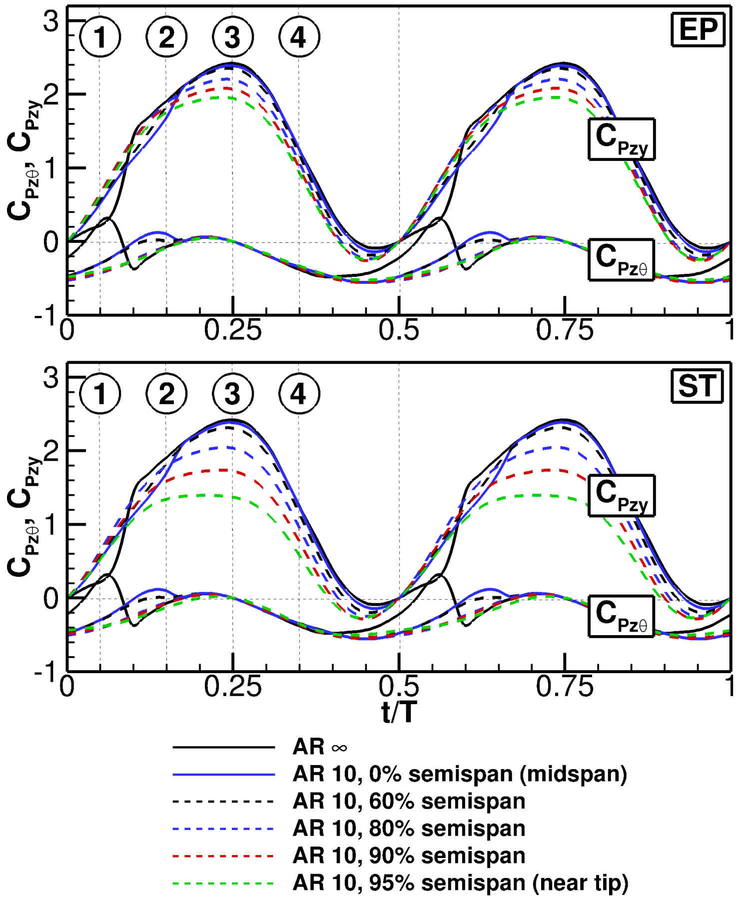
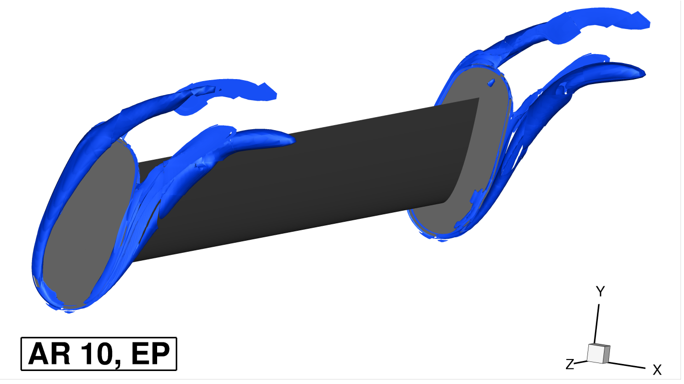


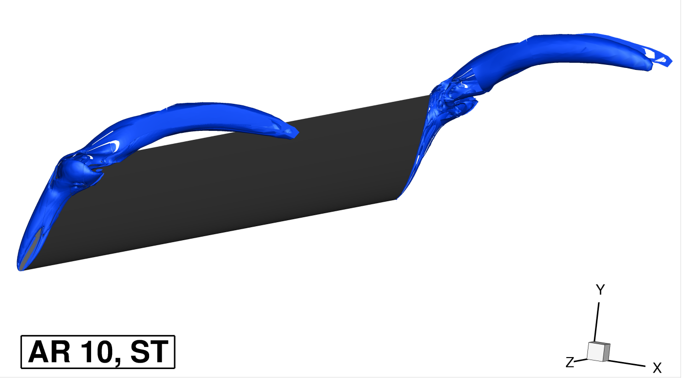
Figure 6.11: Overall power coefficient per unit wing length (top), heaving power coefficient per unit wing length (middle), and pitching power coefficient per unit wing length (bottom) of infinite wing and two AR 10 wings at five spanwise positions of case A.

power coefficient per unit length $C_{P_{z\theta}}$ of the AR 10 wings are cross compared at five spanwise positions in Fig. 6.11, which also reports the infinite wing profiles for reference. Between about 18 % and 50 % of both semi–periods, the heaving power coefficient of both finite wings from midspan to about 60 % semispan is only negligibly smaller than the corresponding infinite wing profile. It is observed, however, that throughout the period this power component decreases much more rapidly with respect to the ideal case from 60 % semispan to the tip region (95 % semispan) when the finite wing has STs. This performance difference is due mainly to the existence of stronger tip vortices featured by the finite wing without EPs. Consequently, the downwash lowering the effective AoA with a strength decreasing from tip to midspan is higher for the ST–wing. Note also that the largest differences between the heaving power of the finite span wings occur in the period range with maximum nominal AoA. The comparison of the $C_{P_{z\theta}}$ profiles of the three wings reported in Fig. 6.11 highlights several important phenomena. Firstly, the wing tip geometry does not appear to have a significant effect on the pitching power component of the two finite span wings, since the $C_{P_{z\theta}}$ profiles of the two AR 10 wings are extremely close at all reported spanwise positions. Secondly, although the pitching power profiles of both wings between about 18 % and 35 %

of both semi–periods coincide with the infinite wing profile at all reported spanwise positions, substantial qualitative and quantitative differences between the two finite wing and the infinite wing profiles exist over the remainder of the cycle at all spanwise locations. The infinite wing pitching power profile features a marked positive peak at about 6 % of both semi–periods, whereas at midspan of both AR 10 wings such peak has moved to about 14 % of the semi–period with greatly reduced strength. As one moves towards the wing tip, the peak disappears completely. As shown below, these important performance differences between the infinite and the finite wings are caused by a loss of favourable synchronisation between pitching motion and LEVS affecting the latter wings. It is also noted that the complete disappearance of the pitching power peak at the outboard sections of both finite wings in the first 20 % of the semi–periods occurs because LEVS rapidly decreases from about 60 % semispan to the wing tip.



a)



b)

Figure 6.12: Isosurface of vortex indicator $\lambda_2 = -0.1$ at 25 % of cycle of case A (position 3 in Fig. 6.9) for *a*) wing with endplates, and *b*) wing with sharp tips.

The vortex indicator λ_2 defined in [150] is used herein to visualise the flow patterns at the tips of the two finite wings. The isosurface $\lambda_2 = -0.1$ in the tip region of the wing with endplates and that in the tip region of the wing with sharp tips at 25 % of the oscillation cycle are shown in Fig. 6.12-*a* and Fig. 6.12-*b* respectively. The pattern of the λ_2 isosurface at the sharp tips, indicates that vorticity from the pressure side rolls down to the suction side to form a trailing vortex, which causes the downwash effect. The downwash leads to the aforementioned reduction of the effective AoA at the sections close to the tip, reducing C_{P_y}, as observed in the bottom plot of Fig. 6.11. The top plot of Fig. 6.12 shows that a tip vortex also exists for the wing with endplates. This vortex, however, originates at the edge of the endplate and is further away from the wing than the vortex of the wing with sharp tips, resulting in less pronounced downwash. Moreover, the vortex originating at the endplate is weaker than that originating at the sharp tip, because the driving pressure difference is smaller in the former case.

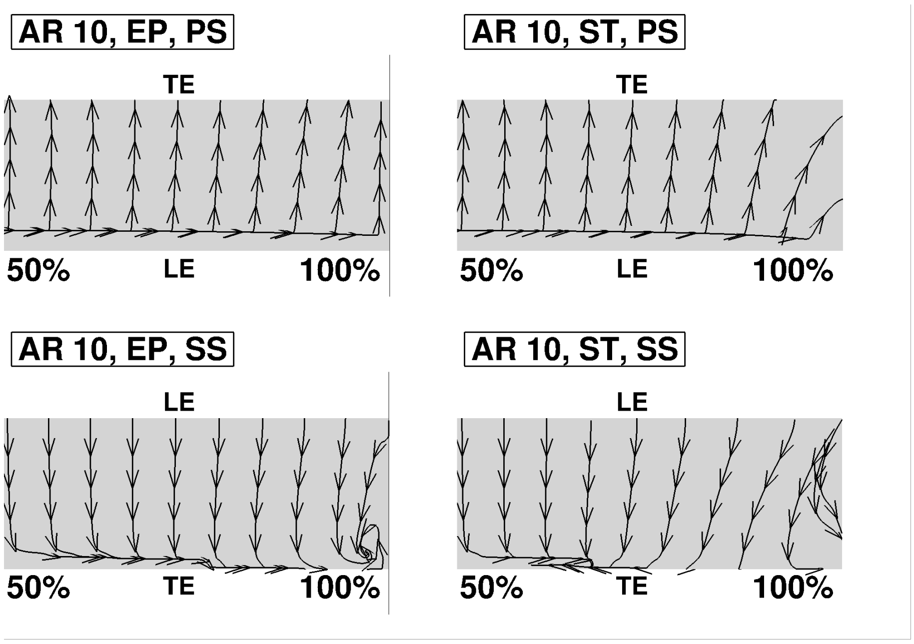


Figure 6.13: Skin friction lines on pressure side (PS) and suction side (SS) of wing with sharp tips and endplates of case A at 25% of cycle (position 3 in Fig. 6.9).

The comparison of the skin friction lines of the two finite span wings at 25 % of the vertical stroke from 50 % semispan to tip is reported in Fig. 6.13. The skin friction lines on the pressure side (PS) and suction side (SS) of the wing with EPs are depicted in the top left and bottom left subplots respectively, whereas those of the wing featuring STs are reported in the right subplots. One sees that the use of endplates results in the stream being much better guided and kept aligned with the wing chord. This is highlighted by the fact that the skin friction lines on the outboard portion of the wing have a more rectilinear path when using endplates. In the same region, conversely, the skin friction lines on the pressure side of the wing with STs deviate towards the tip rather than progressing towards the trailing edge (TE). This pattern denotes the local flow motion towards the tip where a strong tip vortex is formed. The comparison of the PS flow patterns (top subplots) of the two wings also shows that the stagnation line at the corner between the LE and the wing tip is closer to the LE in the case of the ST–wing. This is due to stronger downwash hitting the ST–wing, which reduces the lift force and thus the amount of work the near–tip sections can extract from the fluid stream. The bottom left subplot also shows a small 3D recirculation region on the wing SS at the corner between the wing endplate and the wing trailing edge. This denotes the existence of a corner stall region similar to that encountered in shrouded turbomachinery rotors and stators [151].

Contour slices of the z component of the flow vorticity at thirteen spanwise positions of the EP–enhanced wing, the ST–wing and the infinite wing at 5 % of the oscillation cycle (position 1) are reported respectively in the left, middle and right images of Fig. 6.14. Inspection of the vorticity contours of the two AR 10 wings reveals that the only significant difference between these two configurations is the presence of the

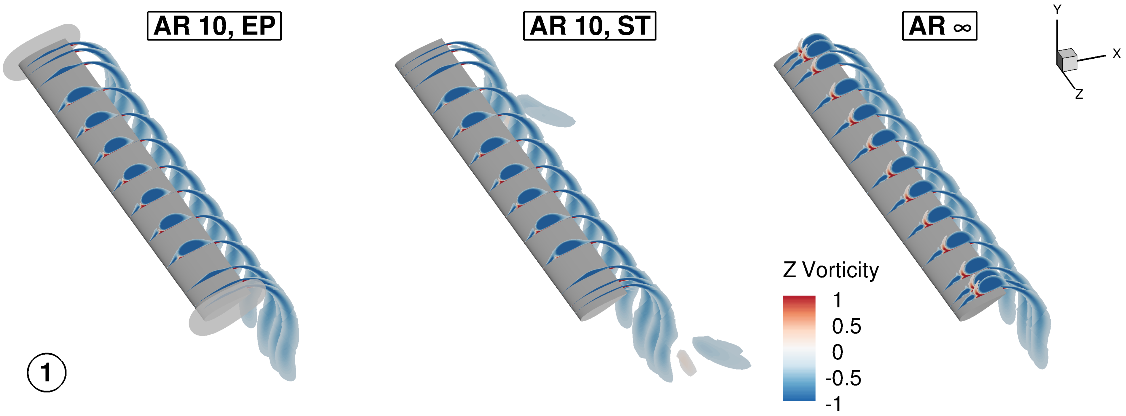


Figure 6.14: Contours of z component of flow vorticity along wing span of case A at 5 % of cycle (position 1 in Fig. 6.9). Left: wing with EPs; middle: wing with STs; right: infinite wing.

footprint of the tip vortex behind the trailing edge of the tip section of the wing with sharp tips. Moreover, for both wings the vortex associated with LEVS (blue vorticity region on the wing PS) is absent in the near–tip region, indicating a strong loss of coherent vortical structure due to finite wing effects. The comparison of the vorticity contours of the two AR 10 wings and the infinite wing highlights that the vortex associated with finite wings LEVS lags behind that of the infinite wing. This phase difference has an important effect on the variations of the generated power of the finite wings relative to the infinite wing, due to different static pressure fields on the wing surface.

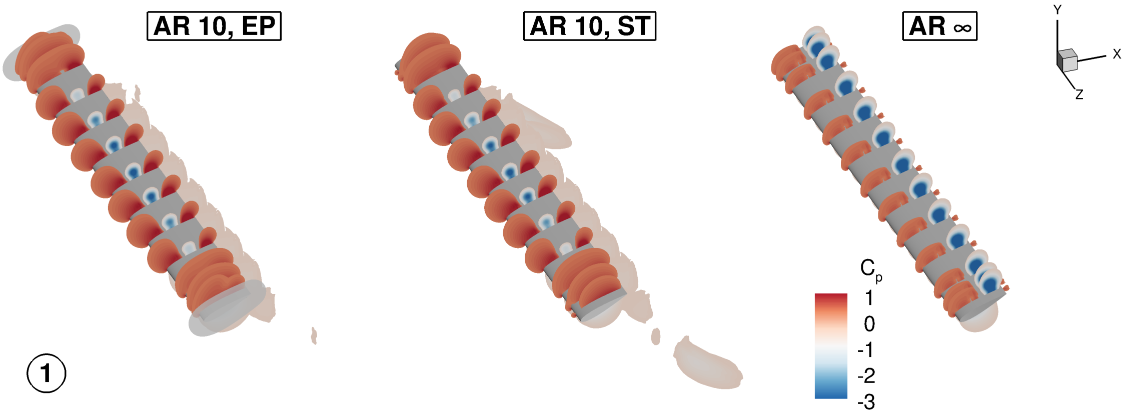


Figure 6.15: Contours of pressure coefficient along wing span of case A at 5 % of cycle (position 1 in Fig. 6.9). Left: wing with EPs; middle: wing with STs; right: infinite wing.

Figure 6.15 provides the contour slices of the pressure coefficient c_p, at the same spanwise positions of the three wings and the same point of the cycle used in Fig. 6.14. c_p is defined with Eqn. (5.6), and its definition is based on the relative wind velocity.

The blue colour in all three images corresponds to the low–pressure region associated with the passage of the high–kinetic energy vortex. It is observed that the low–pressure region on the top side of the infinite wing is wider than in the other two configurations, and, more importantly, it is farther away from the pitching center (i.e. closer to the TE). This results in a larger upward force acting on the rear of the infinite wing, and, in turn, in a higher positive (counter–clockwise) pitching moment. At this time of the period, the angular velocity of all wings is also positive and close to its maximum, as visible in Fig. 6.9. As a result, the pitching power of the infinite wing is higher than that of the two finite span wings in their midspan region. This is the reason why Fig. 6.11 shows that the peak of the pitching power of the infinite wing in the first 20 % of the semi–periods is significantly higher than that of the midspan sections of the *AR* 10 wings in the same region of the cycle. On the other hand, the wider extent of the low–pressure area on the top side of the infinite wing, due to a stronger vortex intensity, results in a reduction of the downward heaving force and, in turn, a reduction of the heaving power at this point of the cycle. For the opposite reason, the *AR* 10 wings have higher heaving power in this region of the cycle. As observed before, however, such higher heaving power is outweighed by the loss of pitching power. In the light of these phenomena, it can be concluded that the loss of favourable synchronisation between LEVS and pitching motion with respect to the infinite wing case results in a significant total power loss that is largely independent of the wing tip geometry.

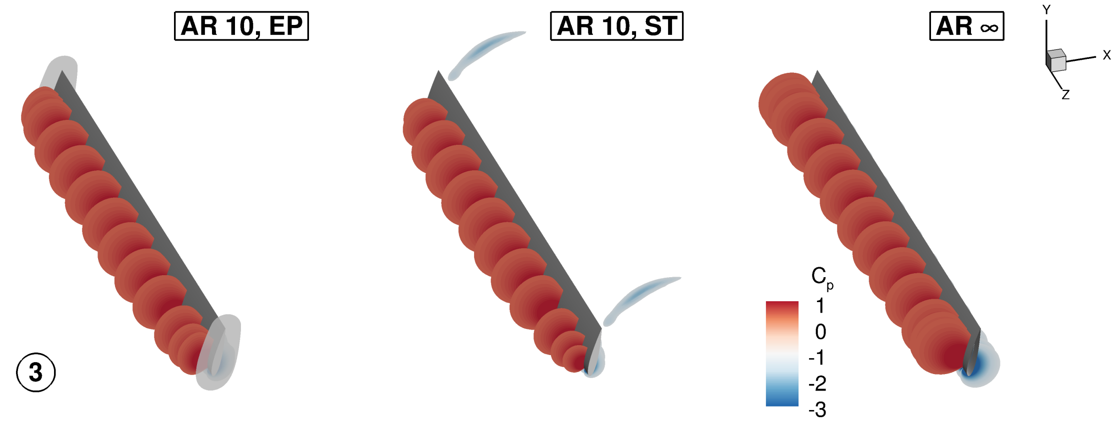


Figure 6.16: Contours of pressure coefficient along wing span of case A at 25 % of cycle (position 3 in Fig. 6.9). Left: wing with EPs; middle: wing with STs; right: infinite wing.

Contour slices of c_p at thirteen spanwise positions of the three wings at 25 % of the oscillation cycle (position 3) are reported respectively in the left, middle and right images of Fig. 6.16. No differences among the static pressure field of the three wings are observed from midspan to more than 60 % of the semispan. In the tip region of both *AR* 10 wings, however, the static pressure on the top side is lower than for the infinite wing. This is due to the tip load reductions associated with the formation of the tip vortex, resulting in a smaller downward force and heaving power. As expected, this loss depends on the tip geometry, as highlighted by the fact that the pressure acting on the top side of the EP–enhanced wing in the tip region is higher than that of the ST–wing.

The effects of the flow mechanisms discussed above are examined in a more quantitative fashion in Figures 6.18 and 6.17. The former provides the foil static pressure coefficient at midspan for the three wings at the positions labelled 1 to 4 in Fig. 6.9;

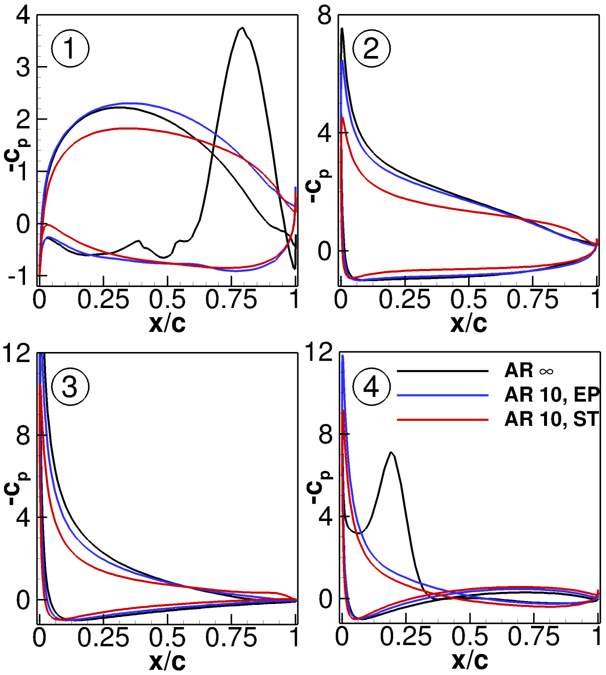


Figure 6.17: Pressure coefficient c_p of infinite wing, and at 95 % semispan section of AR 10 wings of case A at positions labelled 1 to 4 in Fig. 6.9.

the latter has the same structure but refers to the wing section at 95 % semispan. Inspection of the results of Fig. 6.18 confirms that neither qualitative nor quantitative differences exist between the flow pattern of the two AR 10 wings at midspan,

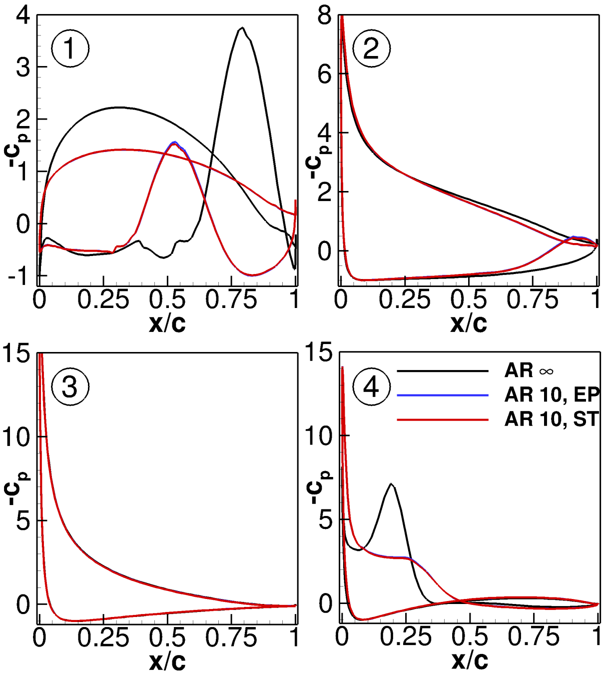


Figure 6.18: Pressure coefficient c_p of infinite wing, and at midspan of AR 10 wings of case A at positions labelled 1 to 4 in Fig. 6.9.

indicating that for this value of AR the performance of the midspan region is fairly independent of the wing tip geometry. At position 3, where the effects of LEVS are absent, the flow of the two finite wings is virtually two–dimensional, as indicated by the fact that all three profiles are superimposed. At positions 1 and 4, conversely, the infinite and finite wing c_p profiles differ substantially due to different patterns of the LEVS. This, as discussed, results in lower overall power generation of the AR 10 wings. At position 2 the relatively small differences between the infinite and the finite wings are due to effects of the delayed vortex being still perceived by the finite wings. The four subplots of Fig. 6.17 highlight the significant effect of tip design on the hydrodynamic performance of the oscillating wing. It is observed that the loading of the near–tip section, here taken as the area between the SS and PS of the wing, is higher when using EPs, due to the lower downwash caused by a weaker tip vortex pattern. The performance difference associated with the use of either tip geometry is particularly strong between positions 1 and 3, which define the interval in which the effective AoA ramps up towards its maximum.

6.1.4 Hydrodynamic analysis of case B

The trajectory of the infinite– and finite–span oscillating wing set–ups analyzed for case B, is defined in subsection 6.1.2 (case B). All configurations analyzed herein share exactly the same trajectory. Figure 6.19 represents the evolution of the main kinematic parameters of all wings over one oscillation cycle for case B. The time–dependent values of the vertical position h of the wing, its angular position θ, the nondimensionalized heaving velocity v_y/u_∞, and the nondimensionalized pitching velocity Ω/Ω_{max}, with Ω_{max} being the maximum pitching velocity of the cycle are all depicted herein. The effective AoA α computed with Eqn. (6.3) is also reported, and its maxima is about 29^o. Likewise in case A, the four positions labelled 1 to 4 correspond to 5%,15%, 25% and 35% of the period respectively, and are those at which the flow field is examined in

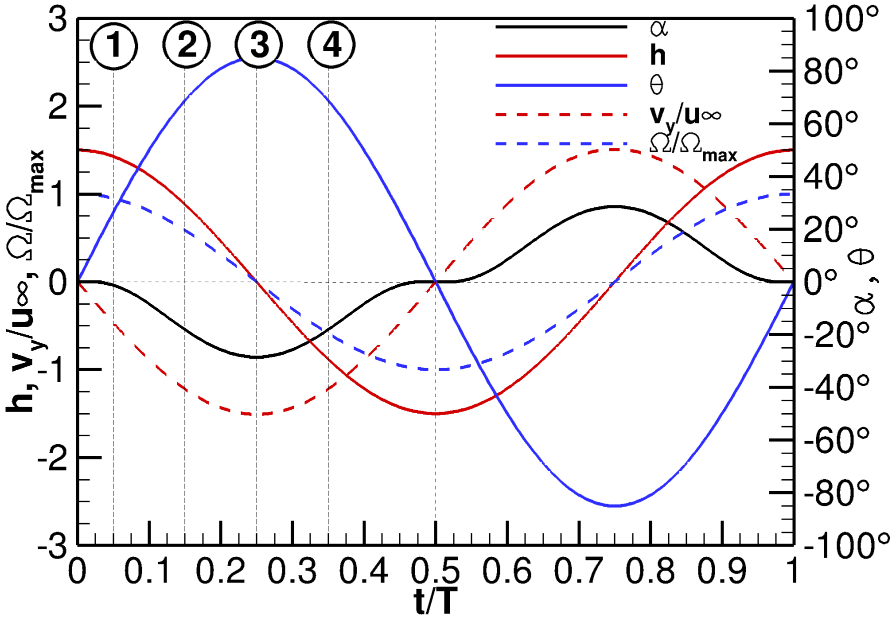


Figure 6.19: Kinematic parameters of the trajectory of the infinite– and finite–span wings of case B.

greater detail in the analyses below. The same positions as in case A were selected in order to precisely examine the differences between the two regimes. Main findings and conclusions are stated in the subsection 6.1.5.

Table 6.4: Integral performance metrics of infinite wing and two AR 10 wings of case B. Columns 2 to 4: mean overall, heaving and pitching power coefficients; column 5: energy extraction efficiency η; columns 6 to 8: percentage variations of overall, heaving and pitching power coefficients of two AR 10 wings with respect to infinite wing values.

AR	$\overline{C}_P$	$\overline{C}_{P_y}$	$\overline{C}_{P_\theta}$	$\eta(\%)$	$\Delta\overline{C}_P(\%)$	$\Delta\overline{C}_{P_y}(\%)$	$\Delta\overline{C}_{P_\theta}(\%)$
∞	1.577	2.100	-0.524	44.99	–	–	–
10 EP	1.335	1.848	-0.513	38.08	-15.3	-12.0	+2.1
10 ST	1.260	1.769	-0.510	35.95	-20.1	-15.8	+2.7

Table 6.4 reports and compares the main integral performance metrics of the infinite wing, the AR 10 wing with EPs and the AR 10 wing with STs. Columns 2 to 4 provide respectively the mean values of the overall power coefficient $\overline{C}_P$, the heaving power coefficient $\overline{C}_{P_y}$, and the pitching power coefficient $\overline{C}_{P_\theta}$. Column 5 provides the overall efficiency η, whereas the percentage variations (Δs) of the three mean power coefficients of the AR 10 wings with respect to the reference values of the infinite wing are reported in columns 6 to 8. The infinite wing analysis refers to the 2D coarse grid TD–256, whereas the two AR 10 analyses are referring to fully 3D TD–256 simulations

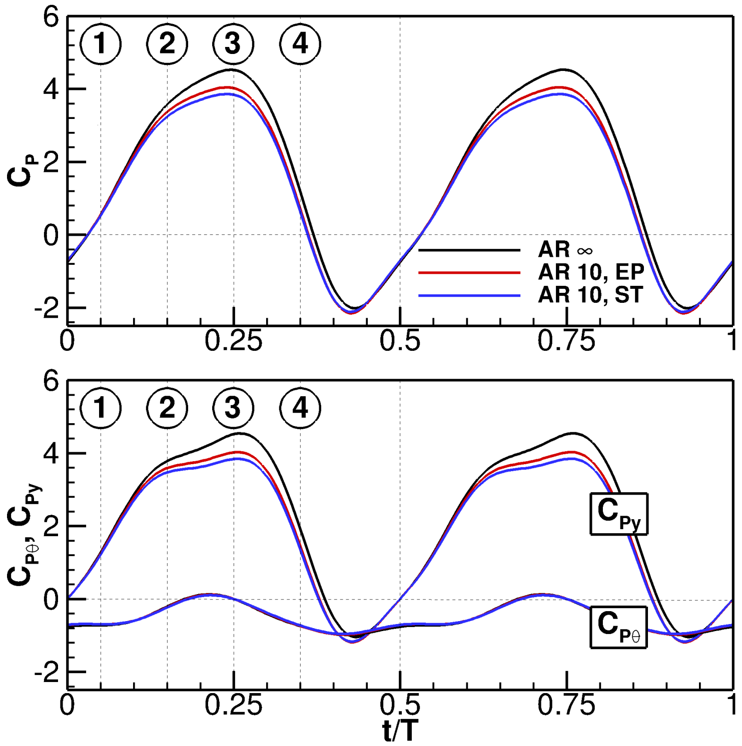


Figure 6.20: Overall power coefficient (top), heaving power coefficient (middle), and pitching power coefficient (bottom) of infinite wing and two AR 10 wings of case B.

using the 30, 670, 848–cell grid, described in subsection 6.1.2. The $\overline{C}_P$ of the AR 10 wing with EPs is 15.3 % lower than that of the infinite wing, whereas $\overline{C}_P$ of the AR 10 wing with STs is 21.1 % lower than that of the ideal infinite wing. The breakdown of the overall mean power into heaving and pitching power components for the three cases highlights that: *a*) the mean negative pitching power (a loss term) of both AR 10 wings slightly decreases by the comparable amount with respect to the ideal infinite wing case: 2.1 % for the wing with EPs and 2.7 % for the wing with STs, *b*) the heaving power coefficient of both AR 10 wings decreases with respect to that of the infinite span wing: by 12 % for the wing with EPs, and by about 16% for the wing with STs. These observations highlight that 3D flow effects that hit the overall energy extraction efficiency of this device, are mostly influenced by the heaving power and depend mostly on the geometry of the wing tips.

The C_P profiles of the three wings over one period, are depicted in the top subplot of Fig. 6.20, whereas the bottom subplot depicts their C_{P_y} and C_{P_θ} profiles. In the first 10 % and last 5 % of both semi–periods, both AR 10 and infinite wing C_P profiles are superimposed (region 1). In the remainder of the cycle (region 2), both AR 10 C_P profiles are significantly lower than that of the infinite wing. However, in the first half

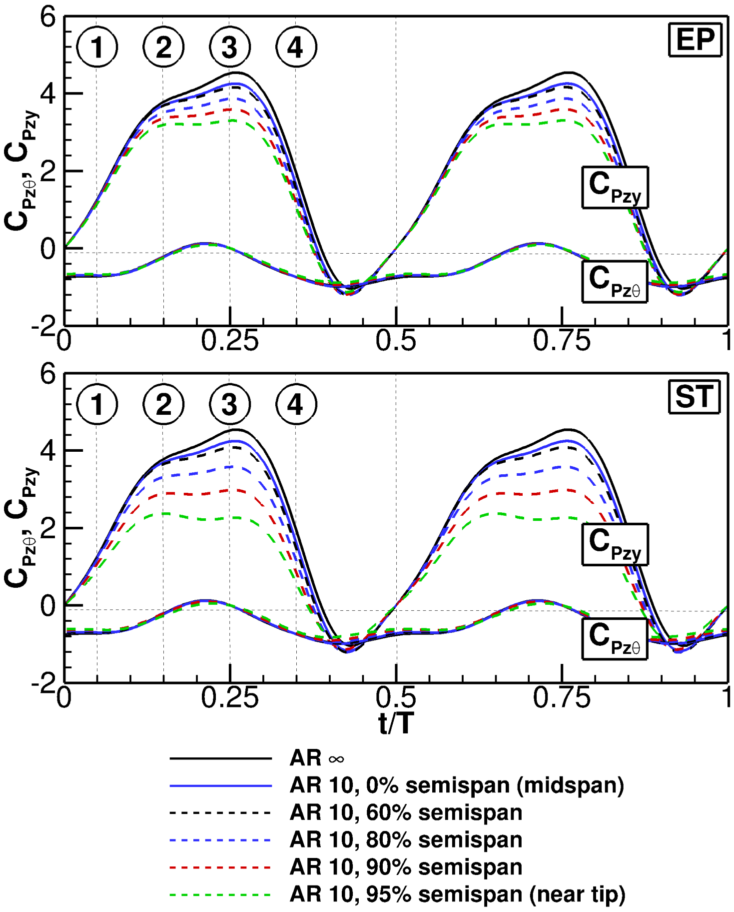
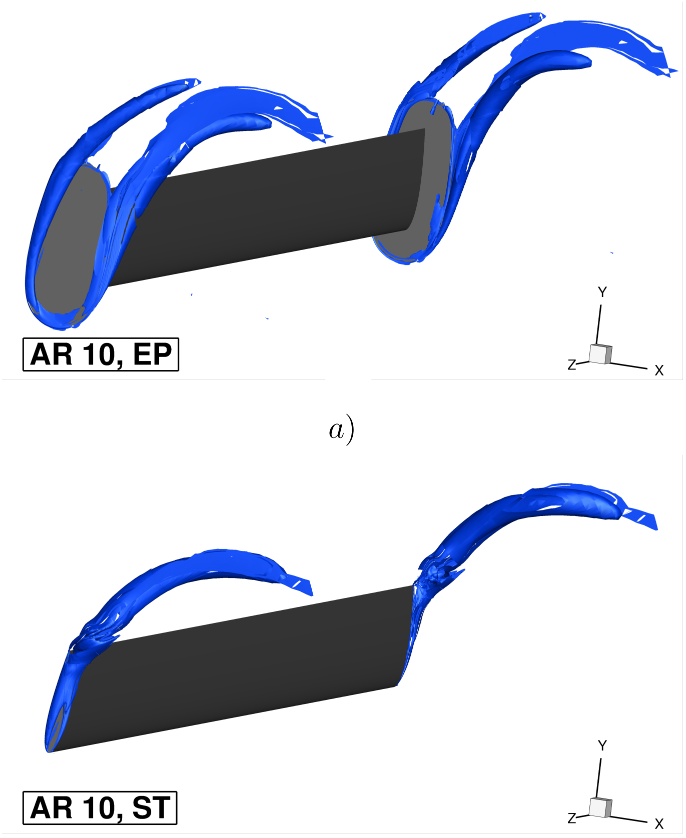


Figure 6.21: Overall power coefficient per unit wing length (top), heaving power coefficient per unit wing length (middle), and pitching power coefficient per unit wing length (bottom) of infinite wing and two AR 10 wings at five spanwise positions of case B.

of region 2 the profile of the wing with EPs is higher than that of the wing with STs. The profiles of the pitching and heaving power coefficients, depicted in the bottom subplot of Fig. 6.20, show that in region 1 there are no losses associated with the heaving power and the pitching power of the two AR 10 wings with respect to infinite wing. In region 2 the pitching power of all three wings is also comparable, and the lower overall power of the two AR 10 is caused primarily by a reduction of the heaving power component, which is greater for the wing with STs in the first half of region 2.

The dependence of the energy extraction efficiency of the finite span wing on the tip geometry for the AR 10 wings, highlighted by Table 6.4 and Fig. 6.20, has been further examined by the cross comparison of the periodic profiles of the heaving power coefficient per unit wing length $C_{P_{zy}}$ and the pitching power coefficient per unit wing length $C_{P_{z\theta}}$, at five spanwise positions. This is depicted in Fig. 6.21, where the infinite wing profiles are also added for reference. The periodic profile of the heaving power coefficient of both finite wings from midspan to about 60 % semispan is about the same than the corresponding infinite wing profile in region 1 and is slightly smaller than the infinite wing profile in region 2. Comparing the top and bottom subplot, it can be concluded that throughout the period this power component decreases much more rapidly with respect to the ideal case from 60 % semispan to the tip region (95 % semispan) in the STs case. Similarly as in case A, the difference between EPs and STs case occurs due to the existence of powerfull tip vortices in the STs case. Consequently, the ST–wing experiences more pronounced downwash, which is responsible for lowering the effective AoA with a strength decreasing from tip to midspan. Also in



$a)$

$b)$

Figure 6.22: Isosurface of vortex indicator $\lambda_2 = -0.1$ at 25 % of cycle of case B (position 3 in Fig. 6.19) for $a)$ wing with endplates, and $b)$ wing with sharp tips.

case B, the largest differences between the heaving power of the finite span wings occur in the period range with maximum nominal AoA and the highest heaving velocity. Furthermore, as depicted in Fig. 6.21, the wing end geometry does not appear to have a significant effect on the pitching power component of the two finite span wings, as the $C_{P_{z\theta}}$ profiles of the two AR 10 wings are extremely close at all reported spanwise positions.

Figures 6.22–*a* and 6.22–*b* respectively depict the isosurface of the vortex indicator $\lambda_2 = -0.1$ in the tip region of the wing with endplates and the wing with the sharp tips at 25 % of the oscillation cycle. In the bottom plot of Fig. 6.22 the λ_2 isosurface at the sharp tips visualises strong trailing vortex, which is formed when the vorticity from the pressure side rolls down to the suction side. This strong vortex causes the pronounced downwash effect, leading to the reduction of the effective AoA at the sections close to the tip, reducing C_{P_y}, as observed in the bottom plot of Fig. 6.21. The top plot of Fig. 6.22 highlights that a tip vortex is also present for the wing with endplates but is much weaker than in the STs case, as the driving pressure difference is much smaller here. This vortex originates at the edge of the endplate, further away from the wing than the vortex of the wing with sharp tips, resulting in less pronounced downwash.

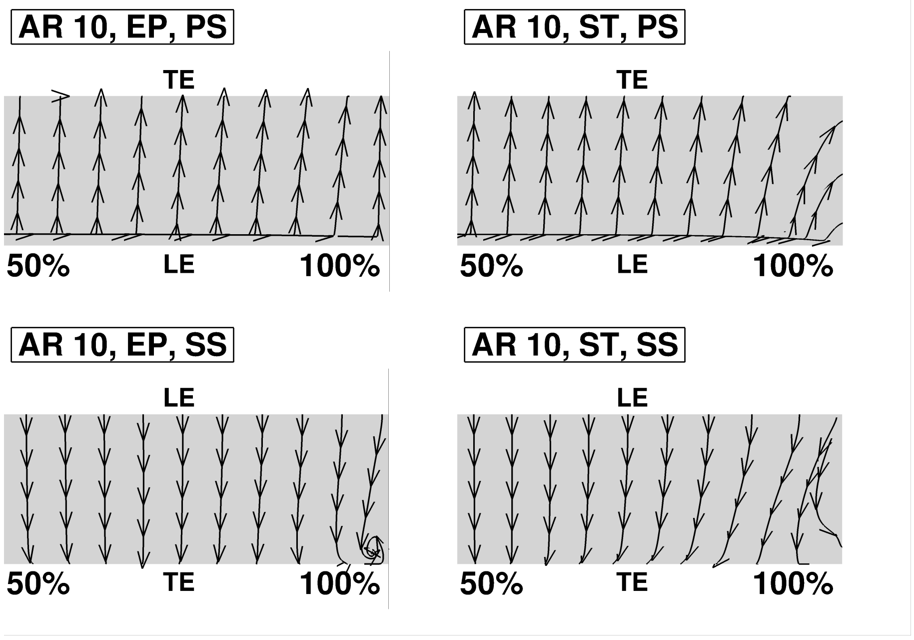


Figure 6.23: Skin friction lines on pressure side (PS) and suction side (SS) of wing with sharp tips and endplates of case B at 25% of cycle (position 3 in Fig. 6.19).

Figure 6.23 represents the comparison of the skin friction lines of the two finite span wings at 25 % of the vertical stroke (position 3) from 50 % semispan to tip. The skin friction lines on the pressure side (PS) and suction side (SS) of the wing with EPs are depicted in the top left and bottom left subplots respectively. Whereas those of the wing featuring STs are reported in the subplots on the right hand side. For the wing with EPs, the stream is much better guided and kept aligned with the wing chord. This is visible on the outboard portion of the wing, where the skin friction lines feature a more straight path when using endplates. Conversely, in the same region in the STs case, the skin friction lines on the pressure side of the wing turn towards the tip rather than being aligned with the chord. This behaviour indicates the formation of a strong tip vortex. Further confirmation of the strong tip vortex occurrence in STs case is also evident in the cross–comparison of the PS flow patterns (top subplots) of the two wings, which reveals that the stagnation line deviation towards the LE is much stronger in the case of the ST–wing. This is due to stronger downwash hitting the ST–wing, which reduces the lift force and thus the amount of work the near–tip sections can extract from the fluid stream. The bottom left subplot shows a small 3D recirculation region on the wing SS at the corner between the wing endplate and trailing edge. This denotes the existence of a corner stall region similar to that encountered in shrouded turbomachinery rotors and stators [151].

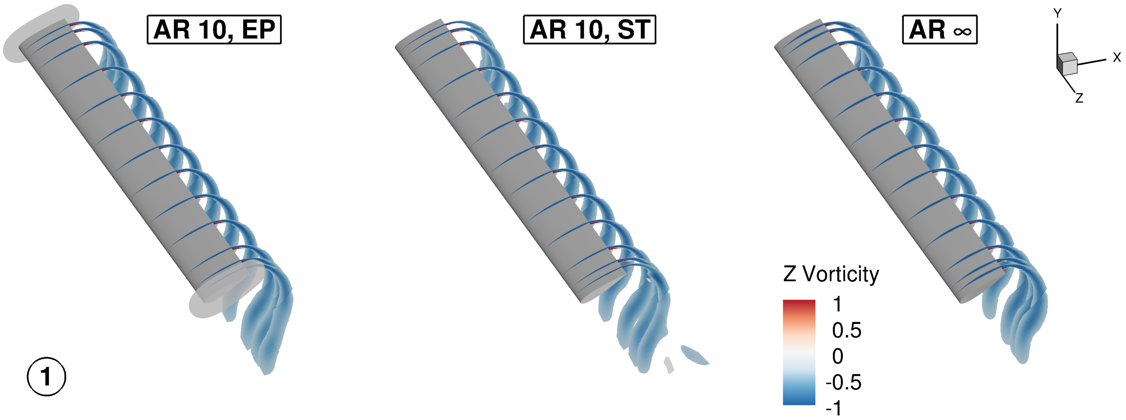


Figure 6.24: Contours of z component of flow vorticity along wing span of case B at 5 % of cycle (position 1 in Fig. 6.19). Left: wing with EPs; middle: wing with STs; right: infinite wing.

The three subplots of Fig. 6.24 report the z component contour slices of the flow vorticity at 5 % of the oscillation cycle (position 1), at thirteen spanwise positions for the three wing configurations. The left, middle and right images correspond respectively to the EP–enhanced wing, the ST–wing and the infinite wing. Cross–comparison of the vorticity contours of the two AR 10 and infinite wings reveals that no quantitative differences among the three wings exist. The only small differences between the two finite wings and infinite wing configurations are in the the near–tip region, where the boundary layer in the two finite wing cases is thinner than for the infinite wing case.

Figures 6.25 and 6.26 depict the pressure coefficient c_p contour slices, at the same thirteen spanwise positions of the three wings used in Fig. 6.24. c_p is defined with Eqn. (5.6), and its definition is based on the relative wind velocity. Figure 6.25 corresponds to 5 % of the oscillation cycle (position 1), whereas Fig. 6.26 represents the position at 25 % of the oscillation cycle (position 3). It is observed that for both positions, no differences exist among the static pressure field of the three wings from midspan to more than 60 % of the semispan. Furthermore, for both positions, the static pressure in the tip region, on the top side, of both finite wings is lower than for the infinite wing. As expected, this loss is dependent on the wing end geometry, as highlighted by the fact that the pressure acting on the top side of the EP–enhanced wing in the tip region is higher than that of the ST–wing. For position at 5 % of the oscillation cycle (position 1), the loss is relatively small for the EP–enhanced wing, and is slightly higher for the ST–wing. Relatively small losses for this position are due to the fact that both the effective AoA α and heaving velocity are close to its minimum.

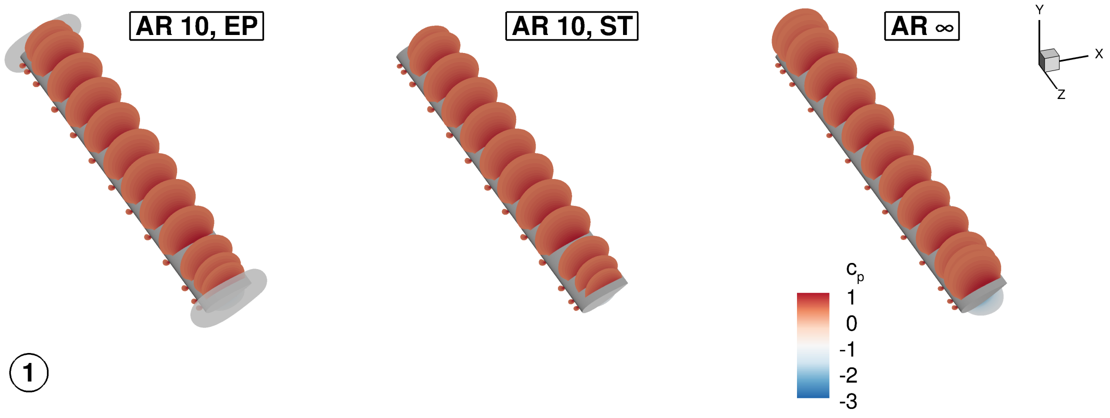


Figure 6.25: Contours of pressure coefficient along wing span of case B at 5 % of cycle (position 1 in Fig. 6.19). Left: wing with EPs; middle: wing with STs; right: infinite wing.

On the contrary, for position at 25 % of the oscillation cycle (position 3), both the EP–enhanced and ST–wing experience much bigger losses. Here, the differences in load reduction between EP–enhanced and ST–wing are clearly visible. As previously explained, these differences arise due to the occurrence of the strong tip vortex in the STs case, which is much weaker in the EPs case. The strong tip vortex in the STs case results in much smaller downward force and heaving power compared to the EPs case, leading to much bigger load reductions than in the EPs case.

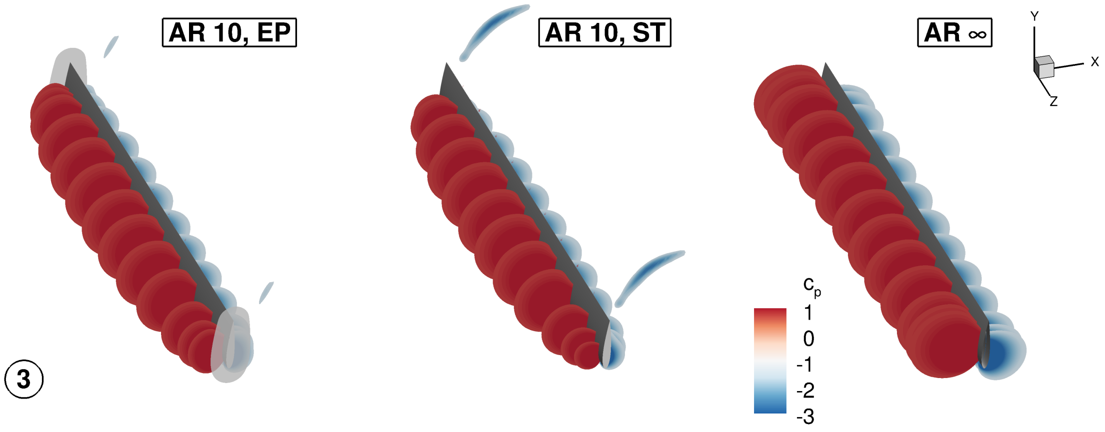


Figure 6.26: Contours of pressure coefficient along wing span of case B at 25 % of cycle (position 3 in Fig. 6.19). Left: wing with EPs; middle: wing with STs; right: infinite wing.

The differences in the flow behaviour between the three wing configurations discussed above, are examined in a more quantitative fashion in Figures 6.27 and 6.28. Both figures examine the foil static pressure coefficient for the three wings at the positions labelled 1 to 4 in Fig. 6.19. The former figure refers to the wing section at midspan, whereas the latter refers to the wing section at 95 % semispan. The results of Fig. 6.27 confirm that neither qualitative nor quantitative differences exist in the

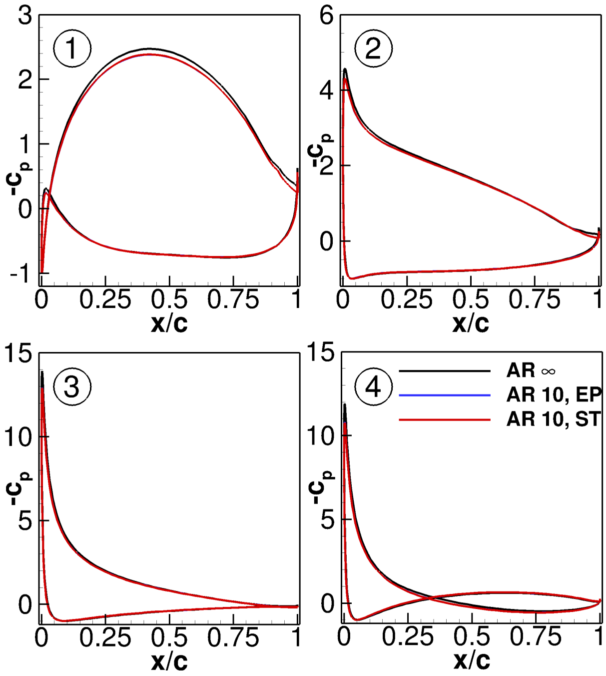


Figure 6.27: Pressure coefficient c_p of infinite wing, and at midspan of AR 10 wings of case B at positions labelled 1 to 4 in Fig. 6.19.

flowfield between the two AR 10 wings at midspan, meaning that the performance of the midspan region is fairly independent of the wing tip geometry. At position 25 % of the oscillation cycle (position 3), all three profiles are superimposed, which indicates

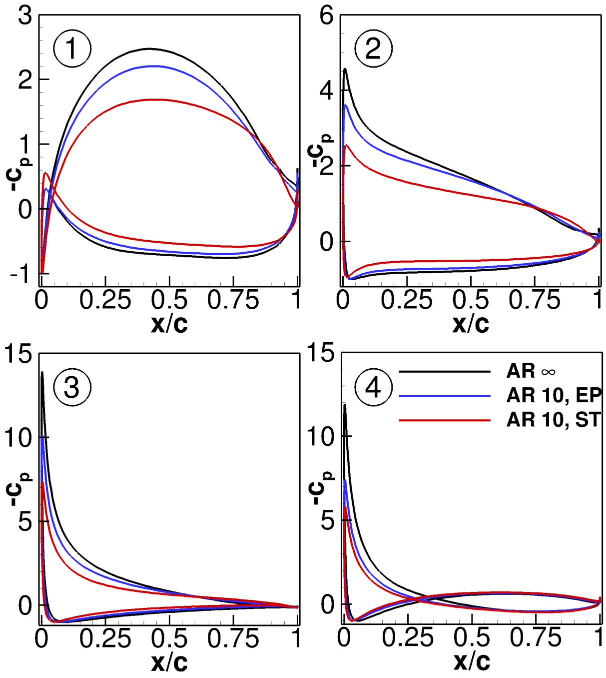


Figure 6.28: Pressure coefficient c_p of infinite wing, and at 95 % semispan section of AR 10 wings of case B at positions labelled 1 to 4 in Fig. 6.19.

that the flow of the two finite wings becomes two–dimensional. At positions 1, 2 and 4, the infinite and finite wing c_p profiles slightly differ, however, due to very small differences it is safe to say that the flow is fairly two–dimensional also for these three positions. Conversely, the four subplots of Fig. 6.28 highlight the significant effect of tip design on the hydrodynamic performance of the oscillating wing. The loading of the near–tip section, here taken as the area between the SS and PS of the wing, is much bigger when using EPs for all four positions. This is due to the lower downwash associated by a weaker tip vortex pattern. The performance difference between the two wing end geometries is particularly strong between positions 1 and 3, in the interval where the effective AoA ramps up towards its maximum.

6.1.5 Discussion

A detailed numerical investigation into the impact of flow three–dimensionality on the power generation efficiency of realistic oscillating wing configurations for renewable energy production has been presented for two different high energy extraction operating regimes. First operating regime (case A) was characterised by the occurrence of LEVS associated with the dynamic stall, whereas the second operating condition (case B) does not yield LEVS. The study for the two different cases was based on the comparative performance assessment of an infinite wing and two aspect ratio 10 wings, one featuring endplates, the other featuring sharp tips. The study has been performed using TD turbulent flow fields past the oscillating wing. HB method has not been used, as for the aerodynamic problems characterised by the occurrence of LEVS associated with the high levels of dynamic stall, such as high energy extraction operating regime A, the HB method does not seem to be gaining significant speedup with respect to the TD method [56]. In order to properly capture the physics of such highly non–linear problem, the number of complex harmonics becomes extremely large. Large number of complex harmonics in such highly non–linear problem, causes the numerical instabilities of the HB solver, and therefore, spoils its convergence. Poorly converged solution then prevents HB analysis to achieve a solution accuracy comparable to that of the TD solution. Unfortunately the HB method has its limitations, and it has been proved less suitable for the cases with higher amount of dynamic stall [56]. For case B, the HB method would most likely perform well, as this operating regime does not feature LEVS associated with the dynamic stall.

For case A, the mean overall power coefficient of the AR 10 wing with endplates and sharp tips are found to decrease respectively by about 12 % and 17 % with respect to that of the infinite wing. Whereas for case B, the drop of mean overall power coefficient of the AR 10 wing with endplates and sharp tips was about 15 % and 20 % with respect to that of the infinite wing. The finite wing losses in case A are caused both by the reduction of the effective AoA at the near–tip sections induced by the downwash associated with the tip vortices, and the loss of optimal synchronisation of LEVS and pitching motion of the wing. The latter phenomenon results in a lower efficiency of the finite span wing due to a significant loss of pitching power with respect to the ideal infinite wing. In case B the finite wing losses depend only on the reduction of the effective AoA at the near–tip sections caused by the downwash. Even though case B is not affected with the LEVS, and its losses purely depend on the strength of the tip vortex, reduction of the overall mean power coefficient of both endplate–enhanced wing and ST–wing with respect to the infinite wing are both slightly bigger than for case A. On the other hand, the efficiency of case A wing with endplates and sharp tips is respectively 34.4 % and 32.6 %, whereas in case B they are 38.08 % and 35.95 %, respectively. In the light of the efficiency, case B is still more efficient, eventhough the finite wing losses are slightly bigger. Higher losses for case B are associated with much higher peak heaving velocity as in case A. Peak heaving velocity for case B is nearly twice as that for case A. Furthermore, based on the analyses of case A, where the power reduction is also associated with the loss of favourable synchronisation between pitching motion and LEVS when considering finite wing effects, it appears advisable to design these devices avoiding regimes characterised by 2D LEVS, as to minimise losses due to finite wing effects. Alternatively one would have to perform the design optimisation of the wing kinematic parameters making use of costly 3D flow simulations, since the results of 2D optimisation appear to be unsuitable to yielding optimal efficiency of the 3D oscillating wing.

The loss due to tip vortex–induced downwash depends, for both cases, on the wing tip geometry, and is smaller for the wing with endplates. The pitching power loss, which is present only in case A, however, does not depend on the wing tip geometry, and hits in a qualitatively and quantitatively similar fashion both AR 10 wings.

The 12.4 % reduction of the overall mean power coefficient of the case A AR 10 endplate–enhanced wing with respect to the infinite wing (Tab. 6.3) is comparable to the 11 % reduction of the same wing reported by Kinsey and Dumas [25]. It should be noted, however, that significant differences between the analyses yielding the two efficiency loss estimates exist. The present simulation used a 30.7 million structured multi–block grid, the SST turbulence model, a Reynolds number of 1.5 million and a pitching amplitude of 76.3^o; the analysis of [25] used a 3.4 million unstructured grid, the Spalart–Allmaras turbulence model, a Reynolds number of 0.5 million and a pitching amplitude of 75.0^o. The closeness of the two results makes one wonder if the outcome of the comparative analysis of this paper would vary fairly little for Reynolds number between 0.5 and 1.5 million. Answering this question with confidence is presently difficult due to the lack of the analysis of the AR 10 wing with sharp tips in the study of [25] and also the aforementioned differences of the computational approach. It is also noted that the article [25] provides the comparative analysis of the finite wings with sharp tips and endplates for AR 5 and 7. Although the overall loss levels for these configurations are higher than for AR 10 (as expected), the qualitative differences of flow patterns of the two wing types for given aspect ratio appear to be similar to those observed in the present study. This may point to independence of the qualitative aspects of the present analysis on the Reynolds number for $0.5 \times 10^6 < Re < 1.5 \times 10^6$.

Arguably, the use of relatively coarse spatial and temporal discretizations of case A (coarse base 2D grid and 256 time–intervals per period) made herein might have introduced some uncertainty in the quantitative estimates of the case A results discussed in section 6.1.3. More precise quantification of the variations of the power coefficients with the wing AR and tip design may require larger grids and computational resources. Nevertheless, the author's view is that a higher resolution is unlikely to alter significantly the general findings of the analyses above, particularly the key ones on the loss of optimal synchronisation between LEVS and wing motion, and the consequent efficiency loss of both finite–span wings with respect to the infinite wing. Considering the case of the pitching power profile, this is because its variation with the spatial and temporal refinement in the 2D analysis (Figures 6.3 and 6.5) consists mainly of relatively small shifts of its peak values due to small variations of the phase between LEVS and foil motion, and does not correspond to any significant variation of LEVS vortical structures (these analyses have not been included for brevity). Conversely, the pitching power profile of the 3D wings on one hand and the infinite–wing on the other have substantially different patterns. (Figures 6.10 and 6.11). Such pattern alterations are due to variations of the phase between LEVS and wing motion which are much larger than those due to varying spatial and temporal refinement. Moreover, the efficiency loss of the two finite–span wings appears to be fairly independent of the tip geometry (column 8 of Table 6.3), which makes one assume that a higher refinement of the tip region flows is unlikely to change the important physical finding on the loss of optimal synchronisation between LEVS and wing motion due to finite–wing effects. Such a higher refinement may instead be advisable for further verification of the quantitative dependence of the efficiency loss on the tip geometry for case A.

6.2 Horizontal–axis wind turbines

All reported analyses in this section refer to the yawed flow past the blades of the NREL offshore 5–MW baseline rotor and are based on 3D NS CFD. NREL 5–MW baseline horizontal–axis wind turbine is a virtual model of the conventional three–bladed upwind variable–speed variable blade–pitch–to–feather–controlled turbine, developed at National Renewable Energy Laboratory (NREL) [39]. The model was created using best available and most representative data of various published documents of wind turbine manufacturers, and publicly available data from previously developed conceptual models in the WindPACT, RECOFF, and DOWEC projects. The model of NREL 5–MW baseline wind turbine has been widely used as a reference by many research teams worldwide. Its purpose is to standardise the detailed specifications of a large representative utility–scale onshore and offshore multimegawatt turbines, and to quantify the benefits of such wind energy technologies. The NREL 5–MW baseline wind turbine rotor features three twisted blades, with the radius $R = 63\,m$. The blades are constructed using two types of airfoils, the inner part consists of DU airfoils and the outer part uses the NACA64 family of airfoils. The actual rotor design is based on the Energy Research Centre of Netherlands 6 MW offshore turbine design project DOWEC [152, 153, 154]. The cut–in, rated and cut–out wind velocities of the NREL 5–MW baseline turbine are respectively $3\,m/s$, $11.4\,m/s$ and $25\,m/s$. The cut–in and rated rotor speeds equal 6.9 RMP and 12.1 RPM, respectively.

6.2.1 Aerodynamics of horizontal–axis wind turbines

Aerodynamic loads

Figure 6.29 depicts the typical HAWT's airfoil twisted by the local twist angle γ and the blade sectional forces. The force in the x–direction per unit blade length F_{x_y} is the tangential force component that is used for the calculation of useful torque. The force in the z–direction per unit blade length F_{z_y} is the axial force component that results in rotor thrust. The overall tangential and axial forces on the blade are defined respectively:

$$F_x = \int_0^R F_{x_y} dy, \tag{6.9}$$

$$F_z = \int_0^R F_{z_y} dy. \tag{6.10}$$

The exerted torque per unit blade length on the blade section M_{z_y} is computed as:

$$M_{z_y} = -\oint_S (p\underline{\mathrm{n}}dS) \times \underline{\mathrm{r}}_a + \oint_S \left(\underline{\underline{\tau}} \cdot \underline{\mathrm{n}}dS\right) \times \underline{\mathrm{r}}_a, \tag{6.11}$$

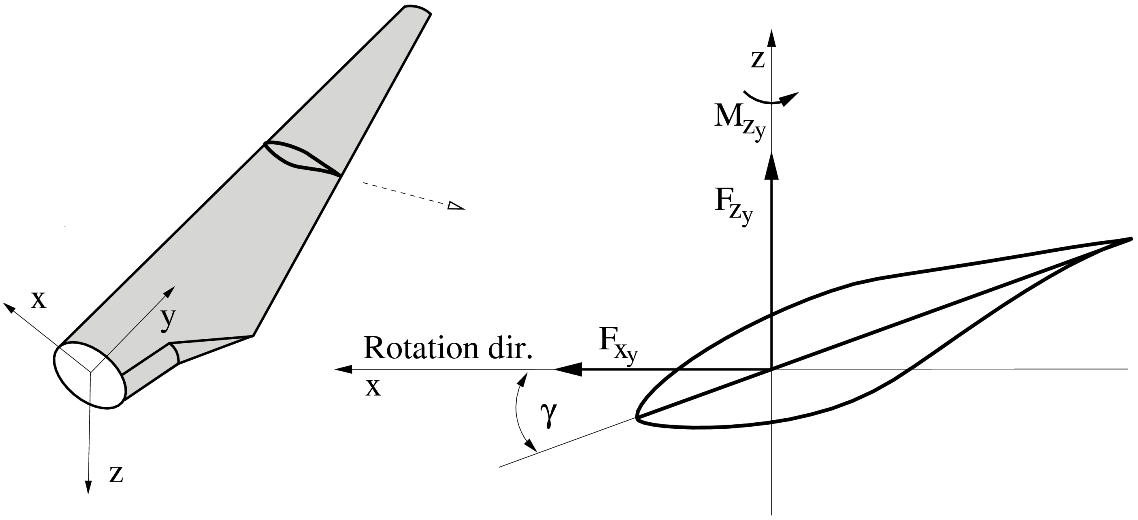


Figure 6.29: Representation of computed HAWT's airfoil forces.

where $\underline{r}_a$ represents the position vector, which originates from the origin of the coordinate system and is defined to the point where the force is applied, and dS is an infinitesimal segment of the curve defining the airfoil. The overall torque on the blade is computed as follows:

$$M_z = \int_0^R M_{z_y} dy. \tag{6.12}$$

The overall aerodynamic forces and moments acting on a HAWT blade are depicted in Fig. 6.30. Overall tangential, radial and axial forces are denoted by F_x, F_y and F_z, respectively. Overall aerodynamic moments are as follows, the moment around x–axis M_x represents out–of–plane bending moment, the moment around y–axis M_y is the torsional moment, and the moment around z–axis M_z, which results in exerted torque, represents in–plane bending moment.

In the analyses below, the moment M_x is applied in the point (0,10,0) which corresponds to the blade length of the first aerodynamic profile, whereas the moments M_y and M_z are applied in the origin of the coordinate system (0,0,0).

In the sections below, use is also made of the 2D airfoil forces and torque. F_{x_y} is the tangential force component per unit blade length that is used for the calculation of useful 2D torque. F_{z_y} is the axial force component per unit blade length that results in 2D thrust. M_{z_y} denotes the exerted torque per unit blade length on the blade section.

Tip–speed ratio

The tip–speed ratio (TSR) is an important rotor design parameter, and is defined as the ratio between the linear velocity at the tip of the blade, and the freestream wind velocity:

$$TSR = \frac{\Omega \cdot R}{u_\infty}, \tag{6.13}$$

where Ω represents the angular velocity of a rotor and R is the rotor radius.

The turbine rotor design typically starts with the TSR corresponding to the maximum power coefficient [28], however, this TSR is not the optimal for the HAWT operation. The optimal TSR depends on many factors, such as the number of blades, torque, mechanical stress, aerodynamics and noise. Turbines with lower number of

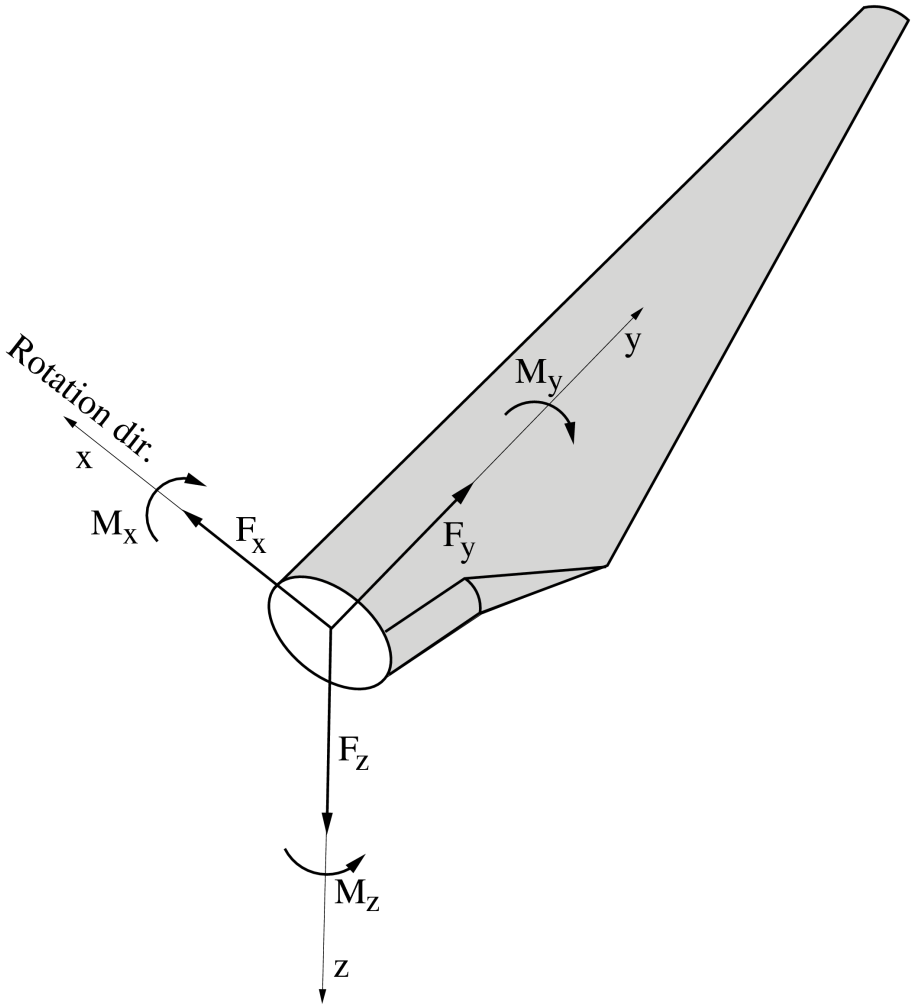


Figure 6.30: Representation of computed HAWT's forces and moments.

blades feature higher optimal TSR. Furthermore, when using higher TSR, the desired power is generated using lower torque, leading to a reduced weight of the rotor and drive train. Moreover, increasing TSR decreases rotor solidity, meaning that less material is required for the production of rotor blades. On the other hand, higher TSR produces greater aerodynamic noise emission, which can be a significantly disturbing factor in urban environment. Higher TSR also causes an increase in the centrifugal and aerodynamic forces, which may lead to difficulties in maintaining the structural integrity and preventing blade failure [155]. Nowadays, common design TSR for two–bladed rotor is typically in between 9 and 10, whereas for the three–bladed rotor is usually from about 6 to 8 [156].

Power and efficiency

The blade power is defined as the product of the rotor angular speed and the exerted torque on the blade:

$$P_n = \Omega \cdot M_z. \tag{6.14}$$

The rotor power is the sum of the power of all blades:

$$P = \sum_{n=1}^{n_{blades}} P_n, \tag{6.15}$$

where the symbol n_{blades} denotes total number of blades.

The efficiency η of the HAWT rotor is defined as the ratio of the extracted power P and the total available power $P_a = \frac{1}{2}\rho_\infty u_\infty^3 A_{rot}$ in the oncoming flow passing through the rotor swept area A_{rot}:

$$\eta \equiv \frac{\overline{P}}{P_a}. \tag{6.16}$$

The power extraction efficiency η defined by Eqn. (6.16) corresponds to the classical power coefficient obtained by means of Betz's analysis [147], which shows that the upper limit of aerodynamic efficiency is 16/27 (about 0.59).

The state–of–the–art commercial HAWTs have already reached very high values of aerodynamic efficiency. The performance coefficient in 80s was about 0.44, whereas it already reached the value of about 0.50 by the mid 2000s. This value is already quite close to the theoretical Betz's limit. It is now crucially important to replace the reliance on the low–fidelity optimisation tools with the high–fidelity numerical tools. Low–fidelity optimisation tools fail to predict flow unsteadiness, which is one of the crucial factors when designing the HAWT rotors. The high–fidelity numerical tools will enable new era of the aerodynamic design of novel HAWT rotors, and will improve aerodynamic efficiency of the existing rotor models, which are currently being manufactured for the large–scale production.

6.2.2 Horizontal–axis wind turbines in yawed wind condition

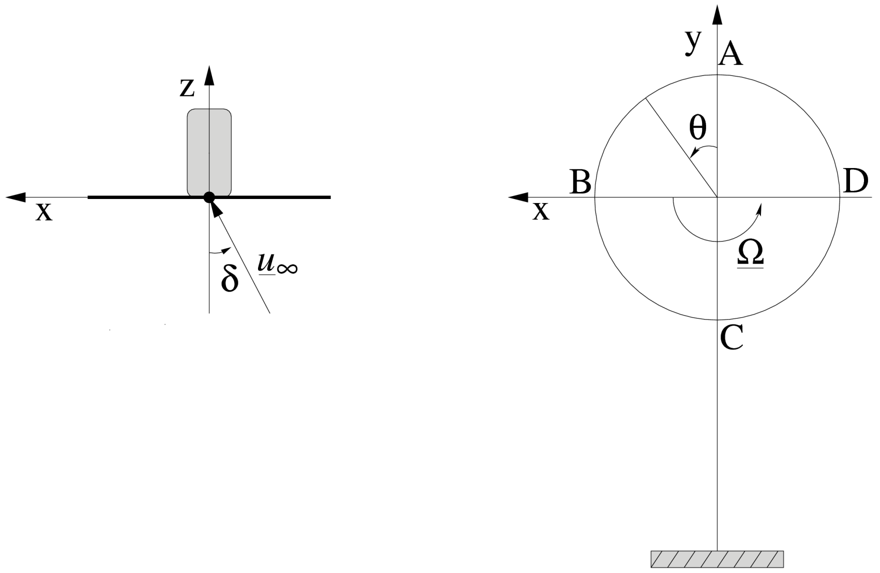


Figure 6.31: Schematic views of the HAWT in yawed wind. Left plot: top view; Right plot: front view.

The periodic flow condition in yawed wind affected by the airfoils of the NREL 5–MW blade depends on the following parameters: the freestream wind speed u_∞, turbine rotational speed Ω, yaw angle δ, which is the angle between u_∞ and the normal to the rotor plane, chord c of the airfoil and its distance r from the rotational axis. Figure 6.31 depicts the top and front schematic views of a typical HAWT rotor in yawed wind, and the aforementioned parameters. The azimuthal position of a blade is described by the angle $\theta = \omega t$. θ is considered to be zero when the blade is vertical and descending (position A).

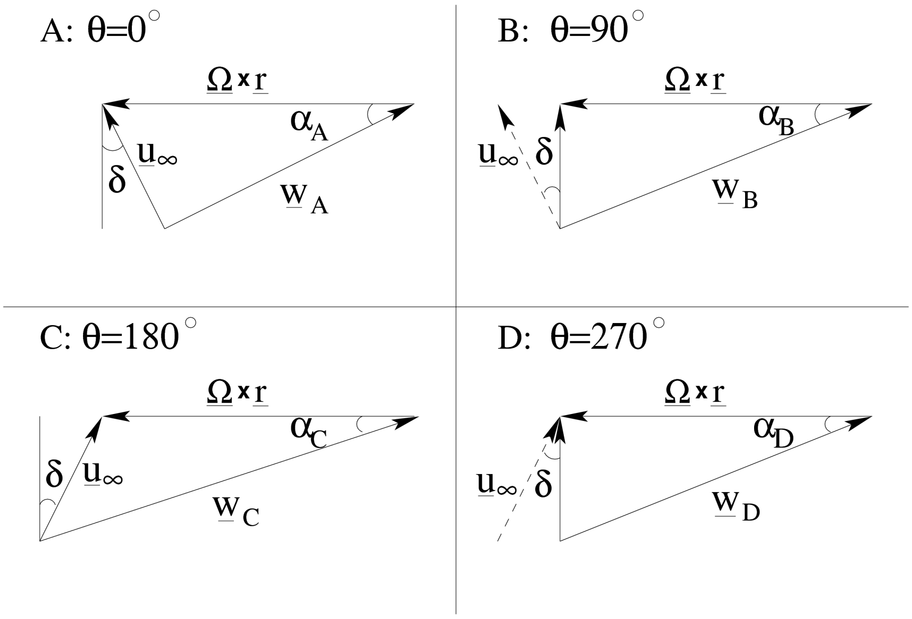


Figure 6.32: Velocity triangles of the HAWT's blade airfoil section for positions A to D, labelled in Fig. 6.31.

Figure 6.32 depicts the velocity triangles for blade airfoil sections at arbitrary distance r from the rotational axis, for the four labelled positions (A to D) in the front view plot of Fig. 6.31. Positions A, B, C, D, correspond to azimuthal angles $\theta = 0°$, $\theta = 90°$, $\theta = 180°$ and $\theta = 270°$ respectively. The modulus of the axial velocity component, $|\underline{u}_\infty \cos(\delta)|$, and that of the entrainment velocity, $|\underline{\Omega} \times \underline{r}|$, remain constant during the revolution, therefore, they are the same in all four triangles. The symbols $\underline{w}_i$ and α_i, where $i = A, B, C, D$, denote respectively the relative wind velocity and the AoA, defined as the angle between the relative wind and the chord line. The AoA can be used to control the aerodynamic force of the blade. Both parameters vary with the azimuthal position θ. The velocity triangles are all contained in the plane tangent to the cylinder of distance r centered on the rotational axis. The radial velocity component (*i.e.* along the blade axis) is, therefore, neglected. However, the magnitude of the radial component varies with azimuthal position and is such that no component is discarded when the blade is at the two vertical positions (A and C), and the entire radial component $\underline{u}_\infty \sin(\delta)$ is neglected when the blade is at the two horizontal positions (B and D). When the blade is at positions A and C, the entire vector $\underline{u}_\infty$ is contained in the tangent plane, whereas, when the blade is at the positions B and D, the radial component of $\underline{u}_\infty$ is orthogonal to the tangent plane. In this 2D kinematic environment, the axial and circumferential components of the farfield wind velocity perceived by a blade airfoil section can be written respectively:

$$w_z = u_\infty \cos(\delta) , \tag{6.17}$$

$$w_\theta = \Omega R - u_\infty \sin(\delta) \cos(\Omega t). \tag{6.18}$$

w_z and w_θ are the components of a time–dependent velocity vector $\underline{w}$. The angle of the relative wind ϕ, formed by $\underline{w}$ and the rotor plane may be written:

$$\phi = \arctan(w_z / w_\theta). \tag{6.19}$$

HAWT's blades are twisted by the local twist angle γ, therefore, the airfoil AoA α can be calculated by subtracting the twist angle from the angle of relative wind:

$$\alpha = \phi - \gamma. \tag{6.20}$$

The positions B and D are those, at which the flow field is examined in greater detail in the following analyses.

6.2.3 Physical and numerical set–up

The selected operating condition of the upwind rotor for all CFD simulations below was that corresponding to the rated operating conditions of the NREL 5–MW baseline wind turbine. The considered rotor speed Ω is $12.1\,RPM$, the freestream wind velocity v_∞ equals $11.4\,m/s$, and the TSR equals 7.0. Only the rotor is modelled, while the tower and nacelle have both been excluded. In all considered cases, the rotor cone angle was set to 0° and zero vertical shear has been assumed. The blade tip pitch angle was set to 0° towards feather. The Reynolds number based on the standard density of $1.225 kg/m^3$, the tip chord, and the relative tip velocity is 6.5×10^6. The spanwise variation of the Reynolds number is not negligible, as it varies between about 6×10^6 and 12×10^6. Two different cases are considered herein, zero–yaw and yawed wind case. Zero–yaw case was computed using the steady solver. Due to the fact that all three blades in zero–yaw case experience the same inflow conditions regardless of the azimuthal position of the rotor, only one blade was modelled using steady periodicity boundary condition. In yawed wind case, the considered yaw angle δ is 20°. Yawed wind simulations were performed using the TD and HB solvers. In time–dependent calculations the whole rotor had to be modelled, whereas in HB calculations only single blade was modelled using multi–frequency periodicity boundary condition. As explained in section 3.4 multi–frequency periodicity boundary condition allows to simulate a single blade for unsteady flows in frequency–domain, which significantly reduces computational size and cost of the simulation.

The computational grid used in the analyses below, depicted in Fig. 6.33, was generated in ICEM CFD and consists of 5270 equal blocks per one sector of the size $16 \times 16 \times 16$ cells, resulting in the overall number of $21,585,920$ cells (medium). The grid topology consists of the O–grid, positioned around the airfoil and four H–grids positioned between the O–grid and the inflow and outflow farfield boundaries and between the O–grid and the two periodic boundaries. The O–grid along the airfoil, depicted in the top right subplot of Fig. 6.33, consists of 256–cells on the airfoil and 112–cells in the normal–like direction. The distance d_w of the first grid points off each particular airfoil from the airfoil itself is about $2 \cdot 10^{-6} c$ and is constant along the whole blade span. The O-grid in the normal–like direction further expands for another 64–cells through the H–grid block, positioned between the O–grid block and the inflow, whereas, the H–grid positioned between the O–grid block and the outflow, expands for another 192–cells. Moreover, the O–grid in the normal–like direction further expands for 48–cells, through the two H–grid blocks, positioned between the O–grid block and the two periodic boundaries. The grid of the size 128×128–cells is placed at the tip to ensure the sufficient resolution in the tip region of the blade. In spanwise direction, the grid features 128–cells on the blade surface and 128–cells from the tip to the lateral farfield. The grid on the blade surface in the spanwise direction was clustered towards the root and towards the tip, reaching the maximum spacing of about $\Delta z = 2 \cdot 10^{-1} c$. The minimum spacing achieved towards the tip was about $\Delta z = 2 \cdot 10^{-6} c$, equal to

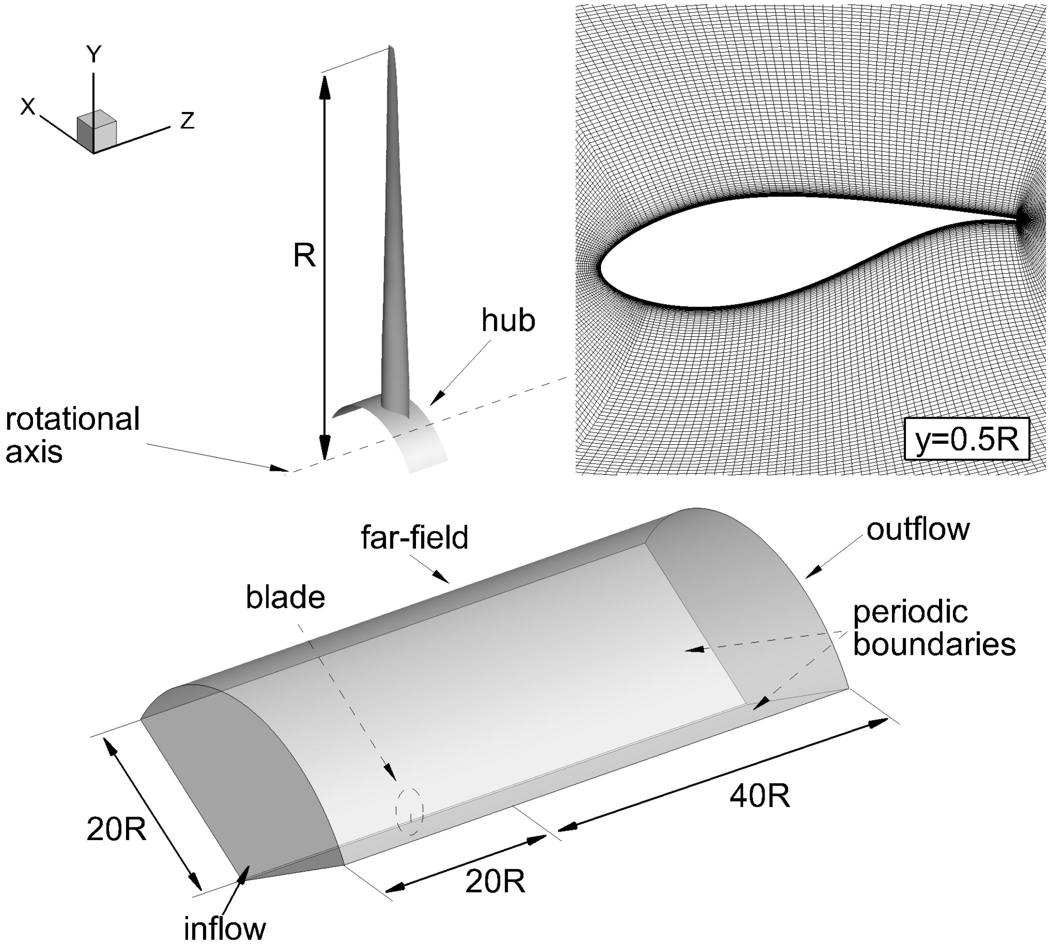


Figure 6.33: NREL 5–MW baseline wind turbine sector grid representation. Top Left: Blade view. Top right: near–airfoil area at 50% of blade length. Bottom: complete domain.

the minimum distance from the airfoil surface in the plane of the airfoil itself. The grid in the lateral farfield was clustered towards the tip, with minimum spacing of approximately $\Delta z = 2 \cdot 10^{-6} c$. The blade tip was modelled using sharp–tip topology. The grid coordinates are dimensional, and are expressed in m. The farfield boundaries have been positioned at $20R$ from the rotor centre at the inflow and in the lateral farfield boundary, and at $40R$ from the rotor centre at the outflow, as suggested in [42], where extensive grid independence study for the NREL 5–MW baseline wind turbine has been performed using compressible structured solver OVERFLOW2. By removing every second line of the medium grid in all three directions, the coarse grid has also been obtained.

For steady and HB calculations, periodic boundary conditions are applied at the 120° cyclic boundaries, as represented in the bottom plot in Fig. 6.33. For time–dependent calculations, the sector grid was rotated and copied for 120° and 240°, so the whole rotor grid has been obtained. In the farfield, the characteristic boundary conditions are used for the inflow, outflow and farfield faces of the cylinder. A no–slip condition was applied on the blade in the rotating frame, whereas, inviscid wall condition was applied at the hub near the rotational axis. The hub was considered as a zero–thickness cylindrical surface, as depicted in top left subplot of Fig. 6.33, and was surrounded by the computational grid from both sides.

The minimum nondimensional wall distance y^+ was kept smaller than one for the medium grid and about one for the coarse grid. All calculations have been performed using the $k-\omega$ SST turbulence model. The low–speed preconditioning was not required for considered operating condition. The CFL number has been set to 3, and CFL ramping has been used. All simulations have been performed using a single grid level, as numerical instabilities were encountered in the attempt of using the MG solver.

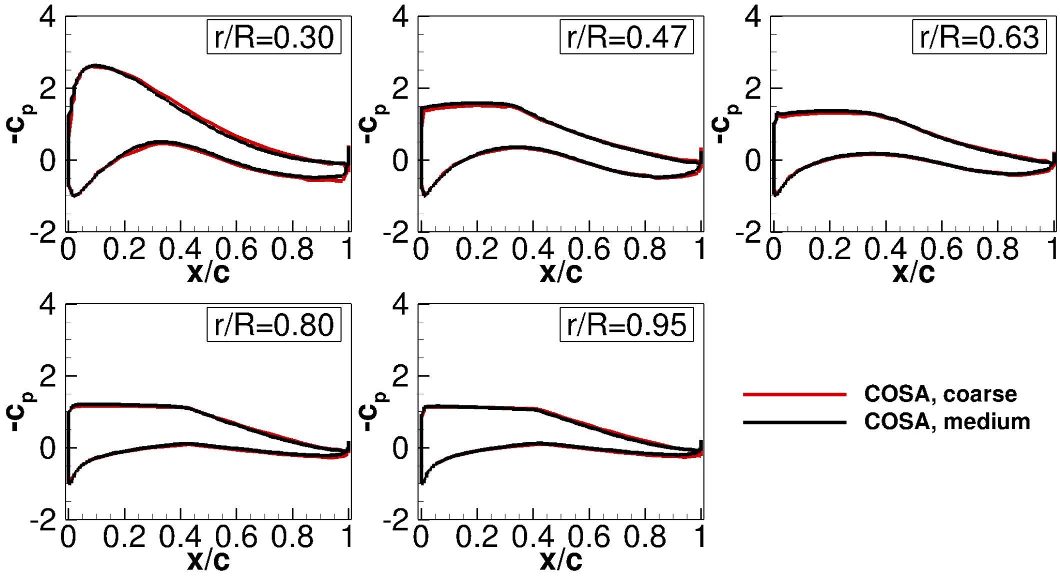


Figure 6.34: Pressure coefficient c_p distributions of the 5–MW baseline wind turbine blade for the steady case.

Mesh refinement analyses with the medium and coarse grid have been carried out, using the steady solver of COSA. The thrust (F_z) and torque (M_z) for single blade have been compared among the medium and coarse grid–refinement, and are reported in Table 6.5, along with the computed power of all three blades P. Table 6.5 shows reasonable agreement of medium and coarse grid–refinement. The output featuring the highest sensitivity is the thrust F_z, which is overpredicted by about 6 % with the coarse grid–refinement. On the other hand, the M_z is only approximately 2 % bigger when using the coarse grid. The pressure coefficient c_p distributions have also been compared, as the pressure force component has the biggest contribution to the overall forces, and are depicted in Fig. 6.34. c_p is defined with Eqn. (5.6), and its definition is based on the relative wind velocity. Looking at Fig. 6.34, one may notice there are slight

Table 6.5: NREL 5–MW baseline turbine: Comparison of thrust (F_z) and torque (M_z) for the single blade and sum of the three blades power (P) for COSA medium and coarse refinement.

	$F_z\ (kN)$	$M_z\ (kNm)$	$P\ (MW)$
medium	215.6	1215.3	4.62
coarse	229.9	1241.4	4.72

differences between the medium and coarse grid–refinement for the position at 30 % of the blade length. However, for all other positions the two solutions are fairly close. The analyses suggest that the solution computed with the coarse grid–refinement presents a fairly independent spatial grid–refinement. Based on the analyses above, only slight differences were observed when comparing the solution computed by the coarse and medium grid–refinements. The coarse grid was selected for the following analyses.

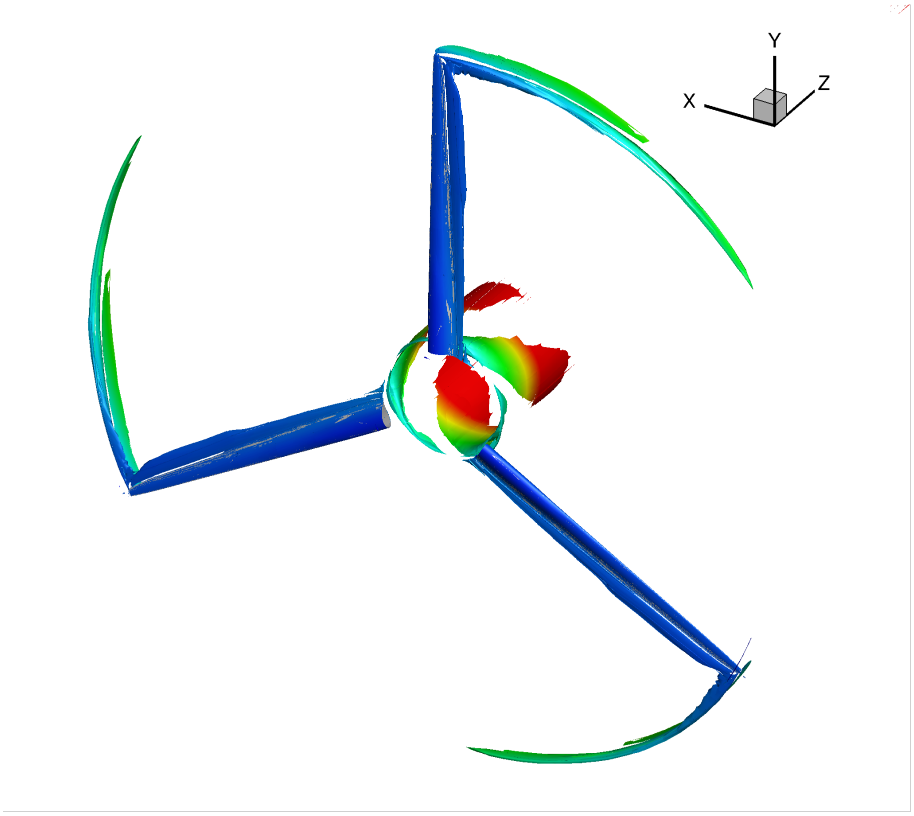


Figure 6.35: Isosurface of the vortex indicator λ_2 for the coarse grid of the steady 5–MW baseline wind turbine rotor simulation coloured with contours of distance normal to the rotor plane.

Figure 6.35 depicts the vortex indicator λ_2 defined in [150]. The isosurface $\lambda_2 = -4 \times 10^{-5}$ clearly shows the tip and root vortices. Due to the relatively coarse grid in the wake region, the expansion of the blade wake, associated with the extraction of the kinetic energy of the wind, is not well resolved for such grid refinement.

The coarse refinement simulation has also been performed with the NUMECA FINE CFD code. The comparison of the computed thrust and torque for the single blade between the COSA and NUMECA coarse grid–refinement is reported in Table 6.6, along with the computed power of all three blades P. The thrust F_z computed with NUMECA is approximately 2 % lower than the thrust F_z computed with COSA, while the torque M_z is about 1 % lower when computed with COSA compared to NUMECA. This results in slightly bigger power output of the NUMECA simulation. The comparison of the c_p distributions is also depicted in Fig. 6.36. The c_p profiles of COSA and NUMECA are in excellent agreement for all five considered spanwise positions.

Table 6.6: NREL 5–MW baseline turbine: Comparison of thrust (F_z) and torque (M_z) for the single blade and sum of the three blades power (P) for COSA and NUMECA coarse refinement.

	$F_z\,(kN)$	$M_z\,(kNm)$	$P\,(MW)$
COSA, coarse	229.9	1241.4	4.72
NUMECA, coarse	226.0	1253.3	4.76

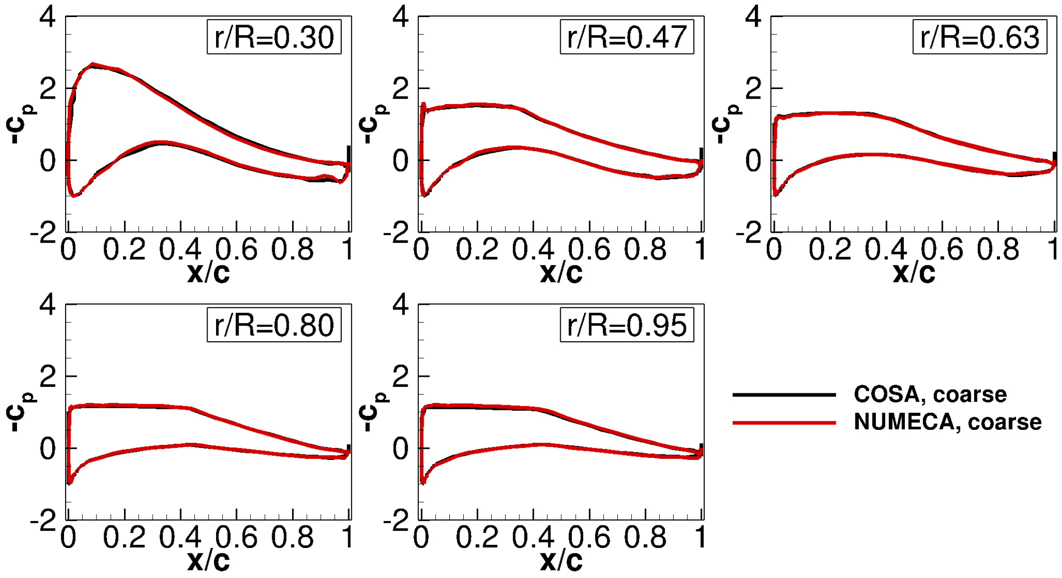


Figure 6.36: Pressure coefficient c_p distributions of the 5–MW baseline wind turbine blade for the steady case, performed with COSA and NUMECA.

6.2.4 Aerodynamic analyses

The coarse grid described in subsection 6.2.3 is used to determine the periodic flow of the considered NREL 5–MW baseline wind turbine. The HB calculations were performed using the sector grid and multi–frequency periodic boundary conditions, whereas the TD calculations simulated the whole rotor. The reference blade is referred as the one positioned at the beginning of the revolution ($\theta = 0°$).

The minimal time–resolution of the COSA TD analysis, required to obtain a solution independent physical time–step, was assessed by means of the selected yawed wind case, described in subsection 6.2.3. Three different simulations have been performed, using a number of time–intervals per period N_T of 180, 360 and 720. The aerodynamic loads of F_x, F_z, M_x, M_y and M_z were used to monitor the convergence of the TD simulations to a periodic state. All three TD simulations have been run until the maximum difference between F_x, F_z, M_x, M_y and M_z differences over two consecutive revolutions became less than 0.1% of their maxima over the latter revolution of the cycle pair. The number of revolutions typically required to achieve the 0.1 % periodicity error threshold on all monitored variables varied between four and twelve revolutions, depending on whether the simulation had been started from a freestream condition or from the solution of a simulation using the same grid but different temporal refinement. The

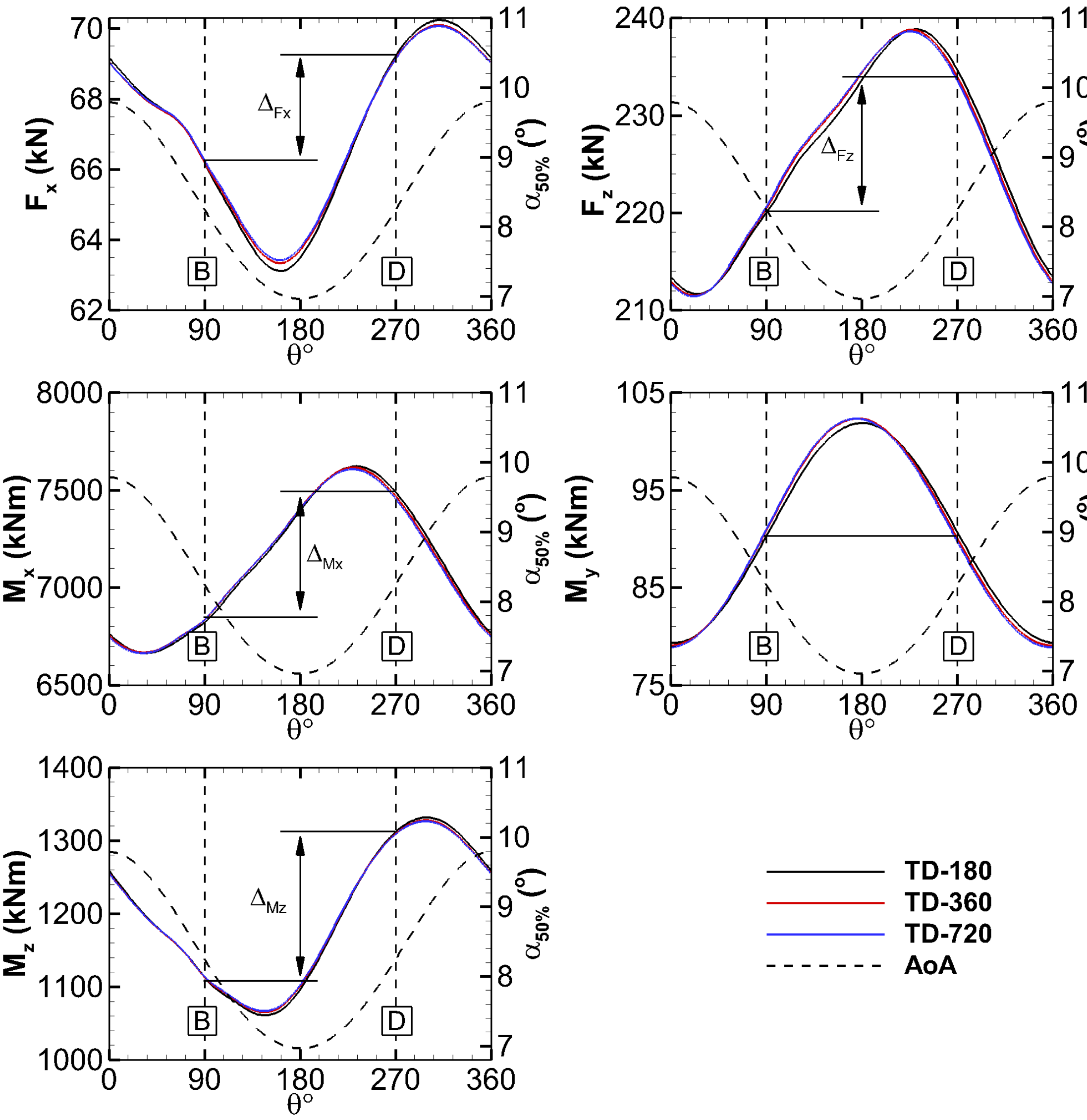


Figure 6.37: Time step refinement analysis of the 5–MW baseline wind turbine in yawed wind, obtained using 180, 360 and 720 steps per oscillation cycle. Top left: tangential force (F_x); top right: axial force (F_z); middle left: out–of–plane bending moment (M_x); middle right: torsional moment (M_y); bottom left: in–plane bending moment (M_z).

top left, top right, middle left, middle right and bottom left subplots in Fig. 6.37 report respectively the periodic profiles of the tangential force component F_x, the axial force component F_z, the out–of–plane bending moment M_x, the torsional moment M_y, and the in–plane bending moment M_z for the reference blade, obtained with these three simulations. These periodic profiles of the reference blade over one rotor revolution computed by the three TD analyses are plotted against the azimuthal position θ of the same blade. Some differences for all F_x, F_z, M_x, M_y and M_z periodic profiles among N_T 180 and N_T 360 exist. Such differences become practically negligible when passing from N_T 360 to N_T 720, indicating that 360 intervals per period are sufficient to achieve a temporally independent solution. The TD simulation using N_T 360 is therefore taken as the reference result of the TD simulation.

Five subplots in Fig. 6.37 also report the estimate of the time–dependent AoA α, defined by Eqn. (6.20), at 50 % of the blade length. It can be noted that α reaches its maxima at the beginning of the period ($\theta = 0°$), when the reference blade is at the vertical position, and the blade velocity and the yawed wind velocity component have opposite directions. Moreover, the α reaches its minima at the period midpoint ($\theta = 180°$), where the blade is at the vertical position, and the blade velocity and the yawed wind velocity components have the same direction. The inspection of the flow solution shows that the dynamic stall is not present anywhere on the blade, but nevertheless all force cycles except M_y are significantly hysteretic, as depicted on the five subplots of Fig. 6.37. To emphasise the significant hysteresis curves variations in the four subplots, the two labelled positions B and D are considered. These two positions denote the $\theta = 90°$ and $\theta = 270°$, respectively, and feature the same AoA α. The symbols Δ_{Fx}, Δ_{Fz}, Δ_{Mx} and Δ_{Mz} denote respectively the F_x, F_z, M_x and M_z difference between positions B and D. Such differences occur due to the yaw misalignment to the wind flow. Yawed flow produces a velocity component parallel to the plane of rotation $\underline{\mathrm{u}}_{\infty} sin(\delta)$, which induces an inflow gradient across the rotor blade, which strength depends on the azimuthal position of the blade. As shown later on, the inflow gradient is particularly strong when the blade is at position B, because $\underline{\mathrm{u}}_{\infty} sin(\delta)$ points in the same direction as radial velocity component, which is induced due to rotation and always acts radially outward. Furthermore, $\underline{\mathrm{u}}_{\infty} sin(\delta)$ and radial velocity component due to rotation have the opposite directions when the blade is at position D, meaning that the inflow gradient is weaker than in position B, and this results in higher AoA. Moreover, higher loading at position D in four subplots also indicates that the 2D relative velocity component at position D is more aligned with the airfoil plane than at position B.

To assess the minimum number of required harmonics in order to resolve the TD problem with the HB solver, and attaining a time–resolution comparable to that of the TD–360 calculation, four HB simulations have been performed. These simulations are denoted by the acronym HB, followed by the value of the retained number of harmonics $N_H = 1, 2, 3, 4$. The hysteretic cycles of the F_x, F_z, M_x, M_y and M_z computed by the four HB analyses and TD–360 are plotted against the estimated time–dependent AoA α at 50 % of the blade length in the five subplots in Fig. 6.38. Inspection of the five subplots in Fig. 6.38 reveals that three complex harmonics are sufficient to achieve the same time–resolution of forces and moments as the TD–360 simulation. This is highlighted by the fact that the TD–360 and the HB–3 solutions are superimposed. The noticeable size of the hysteresis loops in Fig. 6.38 also highlights that the yawed wind condition is responsible for the relatively high level of nonlinearity in the periodic flow field, and therefore, requires the use of nonlinear CFD. The use of the linear CFD is likely to yield insufficiently accurate estimates of the time–dependent loads required for reliable fatigue and aeroelastic analysis and design of HAWT blades. The F_x and F_z loops highlight a periodic variation of about ±5% and ±6%, respectively, whereas the variation of the M_x, M_y and M_z with respect to its mean value is about ±7%, ±13% and ±11%, respectively. In view of structural mechanics, these numbers point to significant variations of the blade loads caused by the yawed wind regime. Cyclically increasing and decreasing aerodynamic loads on the blade contributes considerably to material fatigue, particularly close to the root, where the combined effect of the three most significant components of the stress (normal stress due to M_x, normal stress due to M_z, and shear stress due to torsion M_y) is the highest. The AoA α achieves its maxima when the blade is vertical and the blade velocity and the yawed wind velocity components have the opposite direction, whereas it takes its minima when the blade is vertical and the blade velocity and the yawed wind velocity components have the

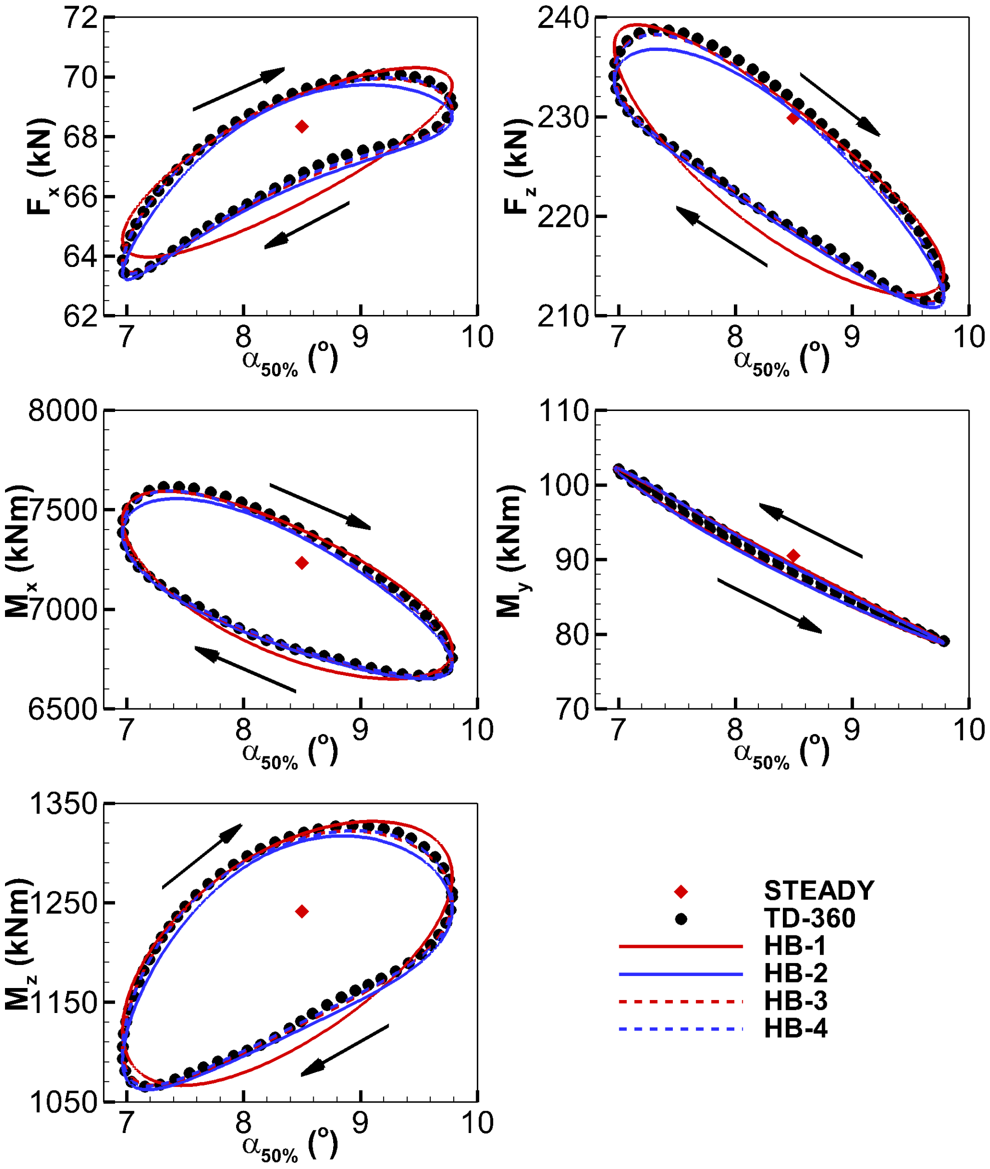


Figure 6.38: Hysteretic loops of yawed wind periodic flow and steady point of zero yaw condition of F_x, F_z, M_x, M_y and M_z, computed with steady simulation, four HB simulations and TD–360 simulation, for the 5–MW baseline wind turbine blade analyses. Top left: tangential force (F_x); top right: axial force (F_z); middle left: out–of–plane bending moment (M_x); middle right: torsional moment (M_y); bottom left: in–plane bending moment (M_z).

same direction. Therefore, Fig. 6.38 highlights that the maximum of F_x and M_z occurs when the blade moves against the direction of the yawed wind component, whereas the minimum occurs when the blade moves in the direction of the yawed wind component. On the contrary, the F_z, M_x and M_y components experience their maximum, when the blade moves in the direction of the yawed wind component, whereas the minimum occurs when the blade moves against the yawed wind component. Five subplots in Fig. 6.38 also depict the steady values of the F_x, F_z M_x, M_y and M_z, computed by the zero–yaw simulation. For all components except for M_y it is evident there is a reduction of their mean values when the turbine experiences the yawed wind condition. The mean value of the periodic flow for M_z, which results in blade torque, decreases by about 3.6% with respect to the zero–yaw simulation.

To further investigate the variation of the hysteresis loops, and to assess the spanwise dependence of the blade loads, four spanwise positions have been examined and compared with the overall values per unit blade length. Four blade lengths, one close to the root section at $r/R = 0.3$, one at the midspan section at $r/R = 0.5$, one at the $r/R = 0.7$ section and one close to the blade tip at position $r/R = 0.95$, were selected

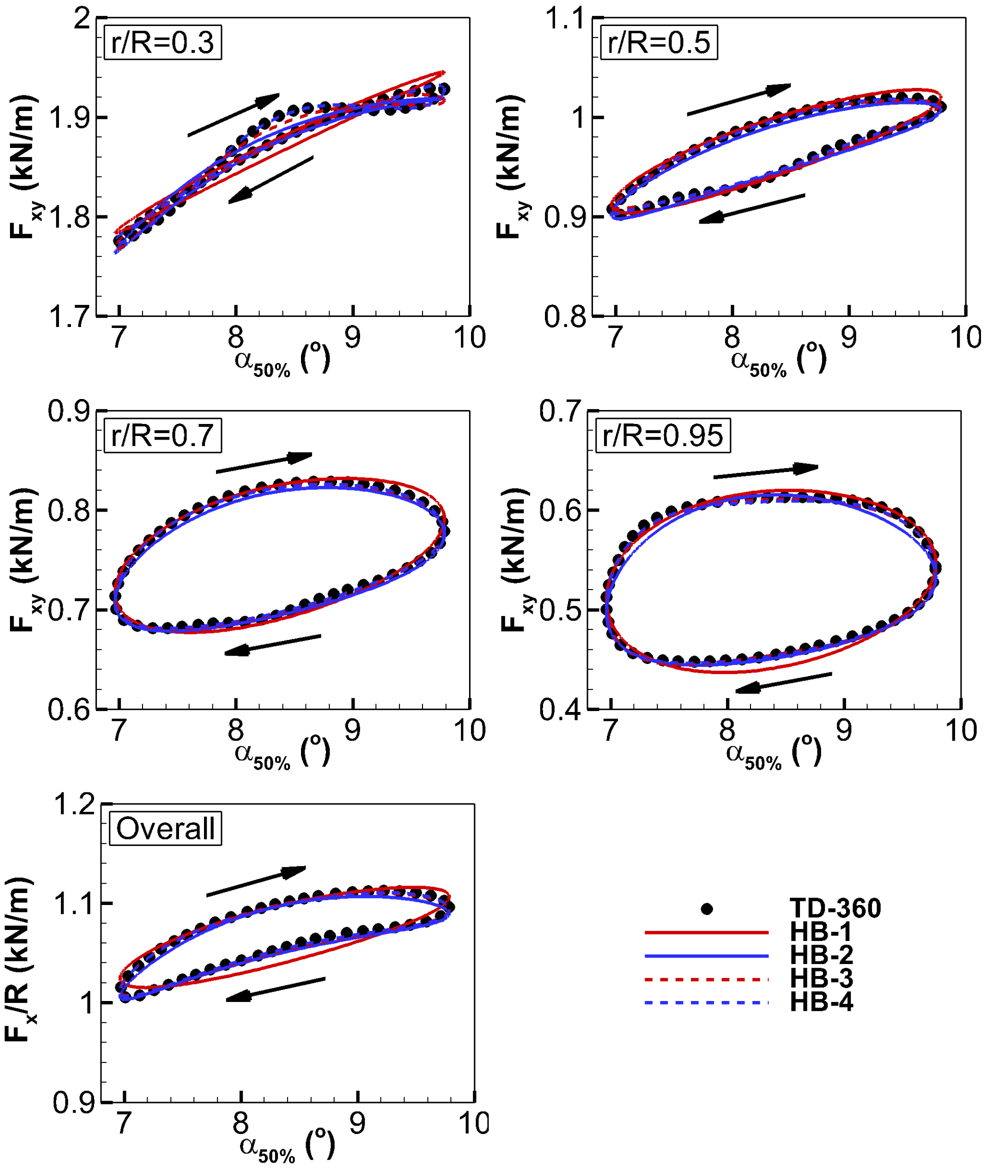


Figure 6.39: Hysteretic loops of the force in x–direction per unit blade length for yawed wind periodic flow at the four spanwise positions and the overall value, computed with four HB simulations and TD–360 simulation, for the 5–MW baseline wind turbine blade analyses. Top left: blade section at $r/R = 0.3$; top right: blade section at $r/R = 0.5$; middle left: blade section at $r/R = 0.7$; middle right: blade section at $r/R = 0.95$; bottom left: overall force per unit blade length.

for the analyses.

Figure 6.39 depicts the tangential force per unit blade length (F_{x_y}) for the four spanwise positions and the overall tangential force per unit blade length on the blade (F_x/R). The hysteresis loops highlight the variation of the F_{x_y} component with respect to its mean value of about ±4% for the blade section at $r/R = 0.3$, ±6% for $r/R = 0.5$ ±10% for $r/R = 0.7$ and ±15% for $r/R = 0.95$. Comparing the F_{x_y} with F_x/R, one can notice that the mean value of F_{x_y} at $r/R = 0.3$ is almost two–fold larger than F_x/R. While the mean values of F_{x_y} at $r/R = 0.5$, $r/R = 0.7$ and $r/R = 0.95$ are

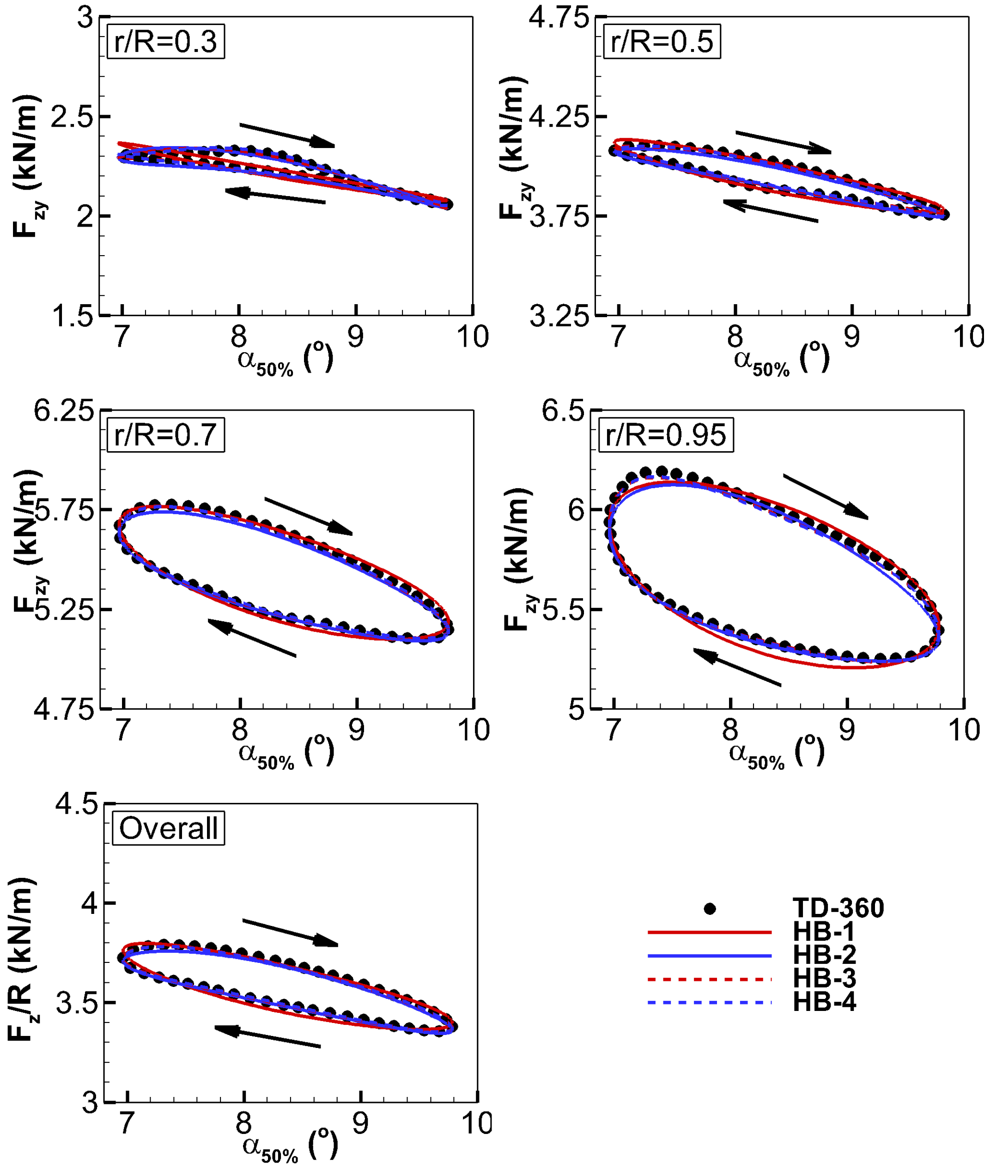


Figure 6.40: Hysteretic loops of the force in z–direction per unit blade length for yawed wind periodic flow at the four spanwise positions and the overall value, computed with four HB simulations and TD–360 simulation, for the 5–MW baseline wind turbine blade analyses. Top left: blade section at $r/R = 0.3$; top right: blade section at $r/R = 0.5$; middle left: blade section at $r/R = 0.7$; middle right: blade section at $r/R = 0.95$; bottom left: overall force per unit blade length.

lower by 10 %, 30 % and more than 50 %, respectively, than F_x/R. One notes that the smallest variations of the hysteresis curves occur in the inboard part of the blade, and they progressively become larger when moving towards the tip.

Figure 6.40 depicts the axial airfoil force per unit blade length (F_{z_y}) for the four spanwise positions and the overall axial force per unit blade length on the blade (F_z/R). The variation of the F_{z_y} component with respect to its mean value is about ±6% for the blade section at $r/R = 0.3$, ±5% for $r/R = 0.5$ ±6% for $r/R = 0.7$ and ±8% for $r/R = 0.95$. The mean value of F_{z_y} at $r/R = 0.3$ is nearly 40 % lower as of F_z/R.

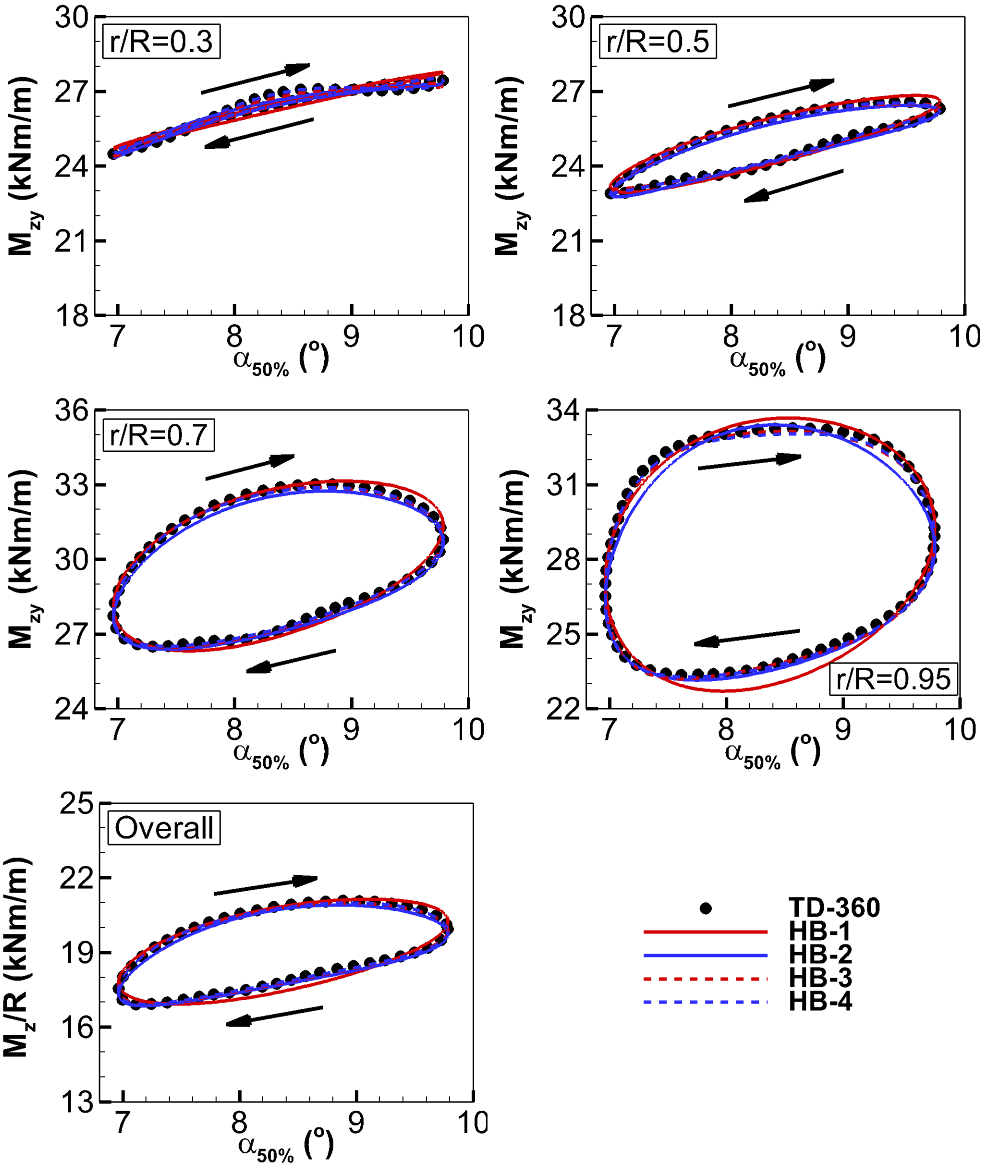


Figure 6.41: Hysteretic loops of the torque per unit blade length for yawed wind periodic flow at the three spanwise positions and the overall value, computed with four HB simulations and TD–360 simulation, for the 5–MW baseline wind turbine blade analyses. Top left: blade section at $r/R = 0.3$; top right: blade section at $r/R = 0.5$; middle left: blade section at $r/R = 0.7$; middle right: blade section at $r/R = 0.95$; bottom right: overall force per unit blade length.

Whereas the mean values of F_{z_y} at $r/R = 0.5$, $r/R = 0.7$ and at $r/R = 0.95$ are about 10 %, 50 % and nearly 60 % higher, respectively, than F_z/R. Similarly as for the F_{x_y} component, the smallest variations of the hysteresis curves occur in the inboard part of the blade, and they become progressively larger in the outboard part of the blade.

Figure 6.41 depicts the torque per unit blade length (M_{z_y}) for the three spanwise positions and the overall torque per unit blade length on the blade (M_z/R). The variation of the M_{z_y} with respect to its mean value is about ±6% for the blade section at $r/R = 0.3$, about ±7.5% for $r/R = 0.5$, about ±11% for $r/R = 0.7$, and about ±17% for $r/R = 0.95$. The mean value of M_{z_y} is in comparison with M_z/R about 38 % lower at $r/R = 0.3$, about 31 % higher at $r/R = 0.5$, more than 56 % higher at $r/R = 0.7$, and nearly 50 % higher at $r/R = 0.95$. This indicates that the section at $r/R = 0.7$ contributes the most to the power output, whereas the section at $r/R = 0.3$ contributes the least. The contribution of the section at $r/R = 0.95$ to the power production is smaller than at $r/R = 0.7$ due to various reasons. One of the reasons is that the airfoil chord, which is decreasing from midspan towards the tip, decreases

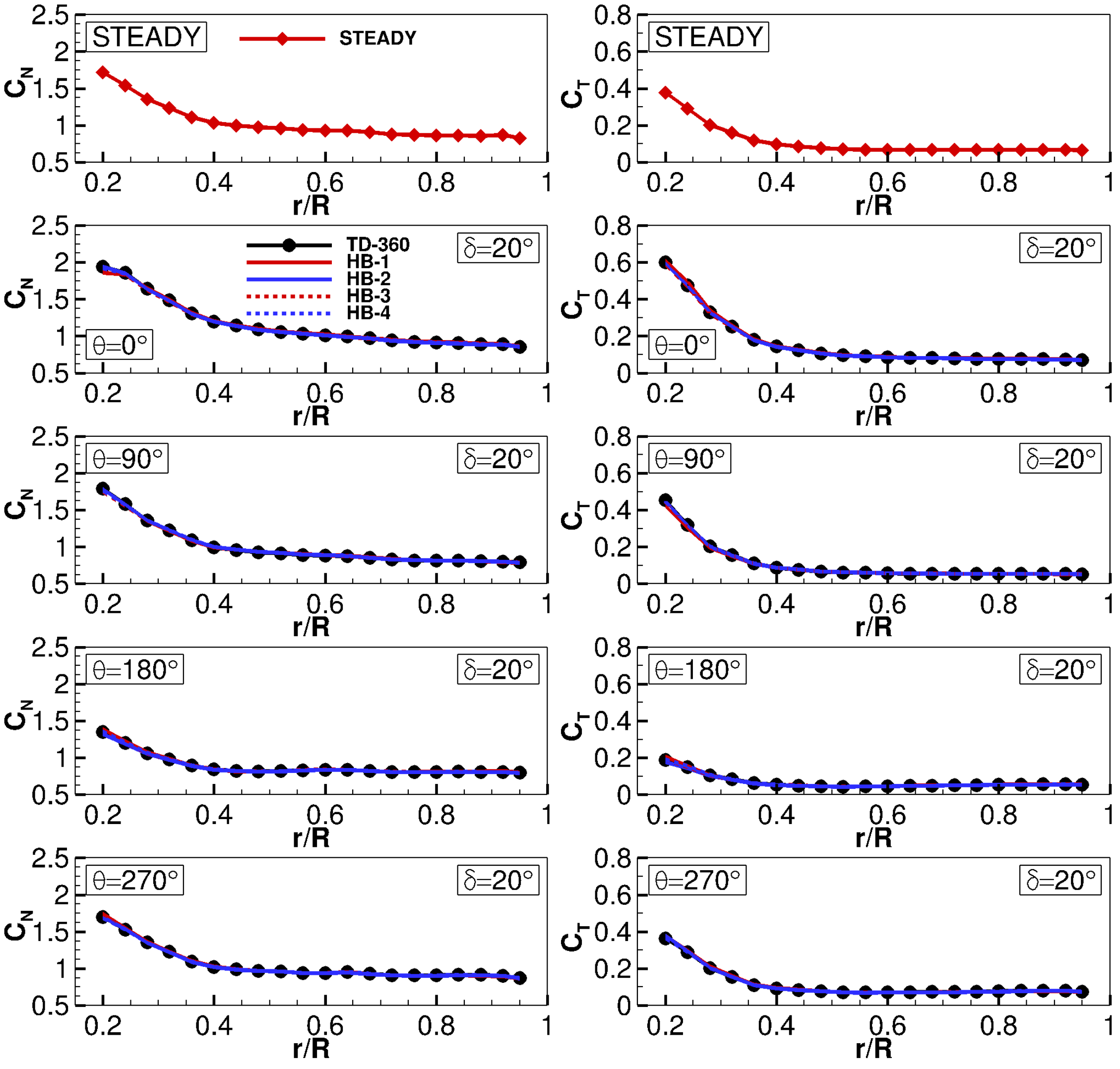


Figure 6.42: Spanwise distribution of the normal force coefficients (left) and the tangential force coefficients (right) at 11.4 m/s and the yaw angle $\delta = 20°$, for the three azimuthal positions of the NREL–5MW blade: $\theta = 0°$, $\theta = 90°$, $\theta = 180°$ and $\theta = 270°$.

more rapidly near the tip. At $r/R = 0.95$ the airfoil chord is nearly 60 % smaller than that at $r/R = 0.7$. Moreover, in the tip region the wing is strongly affected by the strong tip vortex, which causes the downwash effect. The downwash leads to the reduction of the AoA at the airfoil sections close to the tip, reducing the M_{z_y}.

Figure 6.42 depicts spanwise distribution of the normal (C_N) and tangential force (C_T) coefficients. The C_N and C_T were obtained by the integration of the pressure distributions around the airfoil, using the Eqns. 5.4 and 5.5. The top subplots of Fig. 6.42 refer to steady simulation in straight wind, bottom eight plots refer to yawed flow operating condition for the four azimuthal positions at $\theta = 0°$ (position A), $\theta = 90°$ (position B), $\theta = 180°$ (position C) and $\theta = 270°$ (position D), denoted in Fig. 6.37, and are computed with four HB calculations and the TD–360 calculation. From bottom eight subplots in Fig. 6.42 it may be observed that only one complex harmonic is required to resolve the radial distribution of the C_N and C_T at four azimuthal positions. This result suggests that the plot of radial distribution of the C_N and C_T is less sensitive

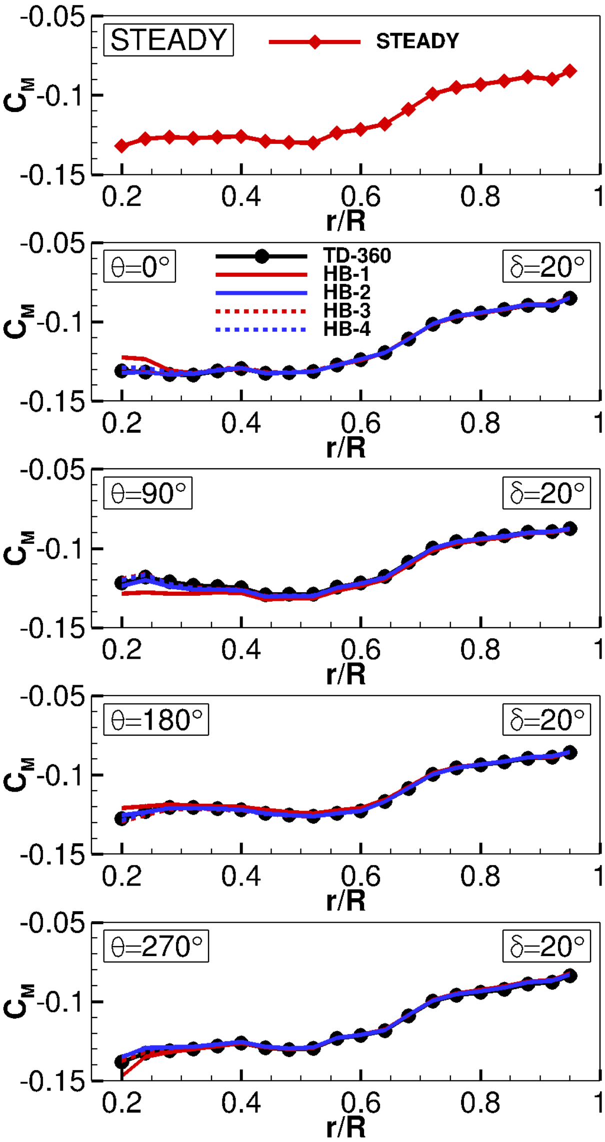


Figure 6.43: Spanwise distribution of the blade pitching moment coefficient C_M at 11.4 m/s and the yaw angle $\delta = 20°$, for the four azimuthal positions of the NREL–5MW blade: $\theta = 0°$, $\theta = 90°$, $\theta = 180°$, and $\theta = 270°$.

than hysteresis loop plots at four spanwise positions above, where at least three complex harmonics were required to achieve the same time–resolution of forces and moments as in the TD–360 simulation. From Fig. 6.42 it may also be observed that the highest variation of the unsteady C_N and C_T w.r.t. to the steady ones is in the inboard part of the blade.

Figure 6.43 depicts spanwise distribution of the blade pitching moment coefficient (C_M). The C_M was obtained by the integration of the pressure distributions around the airfoil, using the Eqn. (5.7). The top subplot of Fig. 6.43 refers to steady simulation in straight wind, bottom four plots refer to yawed flow operating condition for the positions A, B, C and D, denoted in Fig. 6.37, and are computed with four HB calculations and the TD–360 calculation. To resolve the radial distribution of C_M, at least two complex harmonics are required, as visible from bottom four subplots in Fig. 6.43. An excellent agreement between HB–2 and TD–360 is found for the C_M at all four azimuthal positions. From Fig. 6.43 it may also be concluded that the variation of the unsteady C_M is not significant with respect to the steady one.

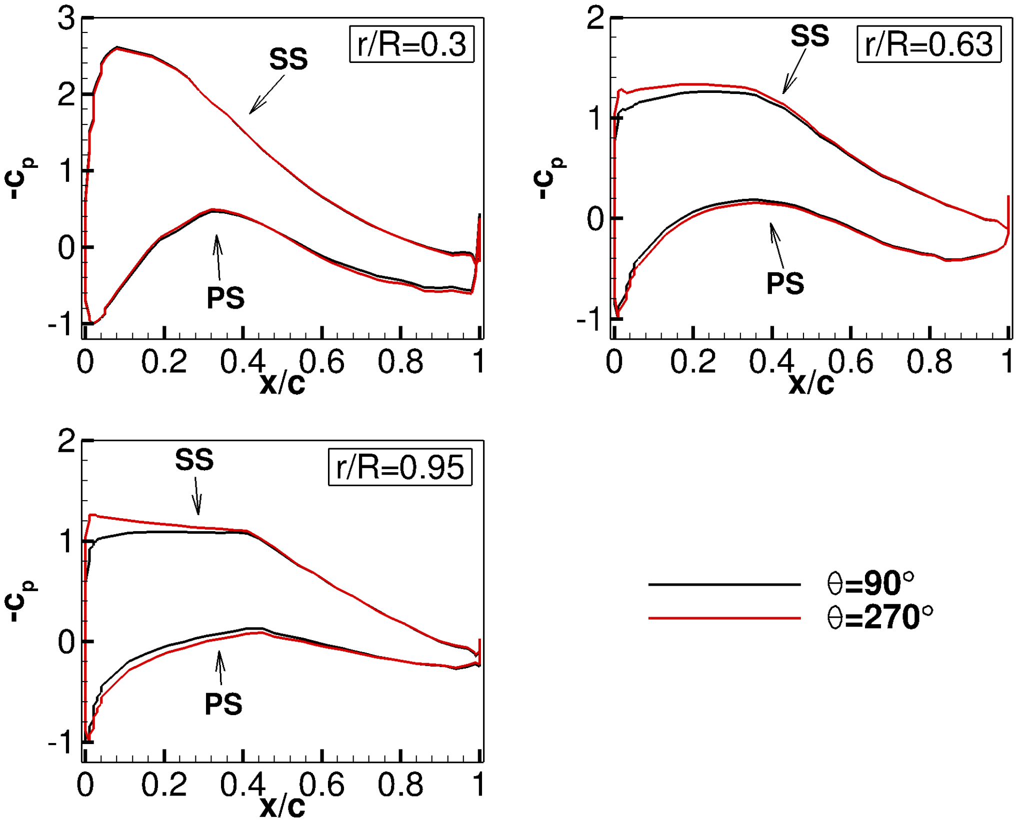


Figure 6.44: Pressure coefficient c_p distributions of the 5–MW baseline wind turbine blade for the azimuthal positions at $\theta = 90°$ (B) and $\theta = 270°$ (D), of the yawed wind periodic simulation.

Figure 6.44 examines the static pressure coefficient c_p profiles, for the three spanwise locations, $r/R = 0.3$, $r/R = 0.63$ and $r/R = 0.95$, at two selected azimuthal positions, $\theta = 90°$ (position B) and $\theta = 270°$ (position D), denoted in Fig. 6.37. c_p is defined with Eqn. (5.6), and its definition is based on the relative wind velocity. Figure 6.44 provides further insight into the main characteristics of the previously discussed hysteretic phenomena. At spanwise location $r/R = 0.3$, the two c_p profiles are nearly superimposed, small differences can be observed in the pressure side (PS) of the c_p profile. If the area between the suction side (SS) and PS portion of c_p is taken as a measure of the aerodynamic loading, then the aerodynamic loading slightly increases at position D. Looking at the spanwise location $r/R = 0.63$, it is evident that the aerodynamic loading at position D is now significantly higher than at position B, and the difference between position B and position D further increases at the spanwise location $r/R = 0.95$. The difference in the aerodynamic loading between the two positions is caused by the inflow gradient across the rotor blade, which is different at the two positions due to the yawed wind condition.

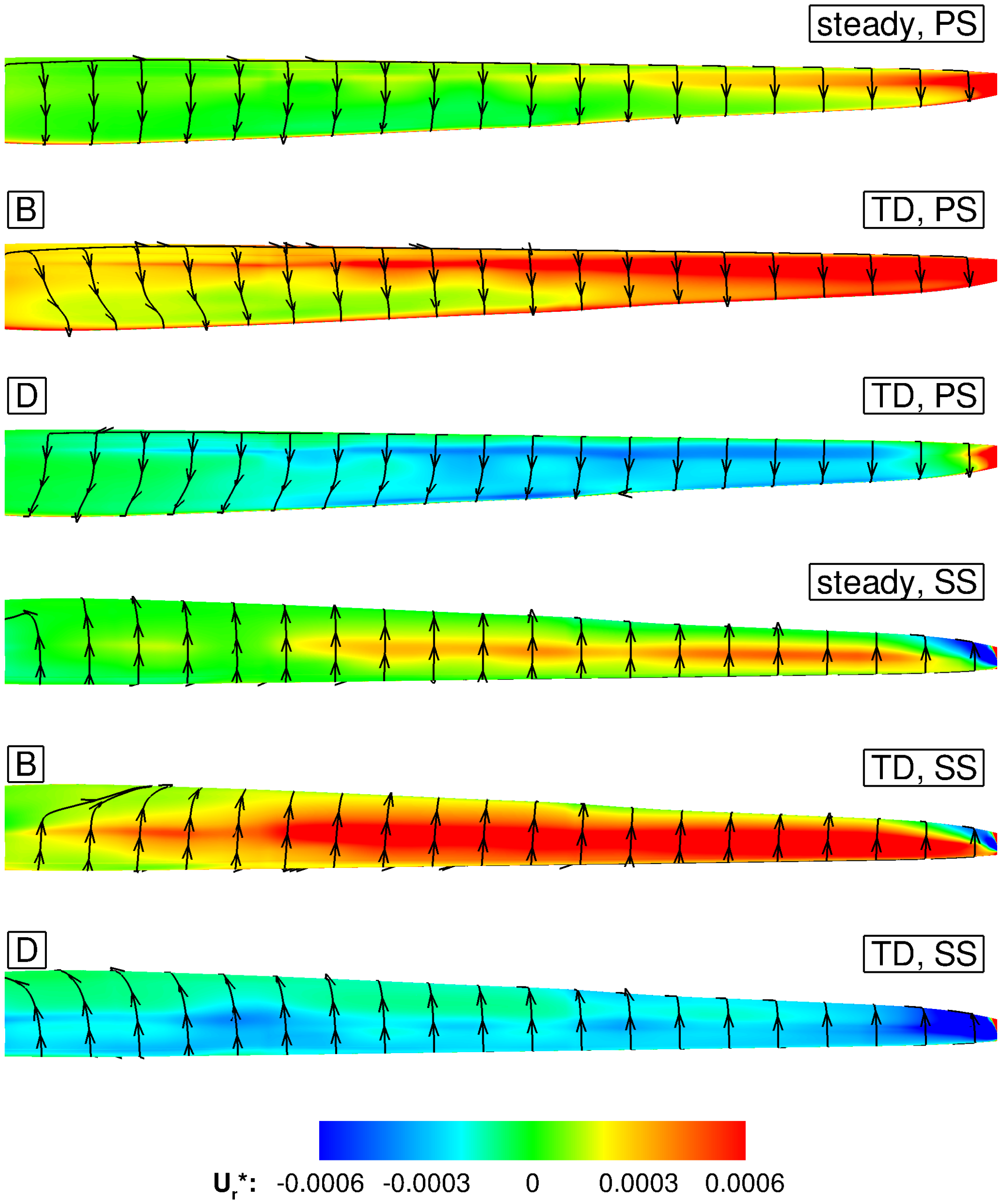


Figure 6.45: Skin friction lines and radial velocity component (U_r) on the pressure side (PS) and the suction side (SS) of the 5–MW baseline wind turbine blade for the steady simulation and the positions at $\theta = 90°$ (B) and $\theta = 270°$ (D) of the TD–360 yawed wind periodic simulation.

Figure 6.45 depicts the comparison of the skin friction lines and the radial velocity component on the first grid surface of the blade U_r^* of the steady simulation and the two azimuthal positions of TD–360 simulation, $\theta = 90°$ (position B) and $\theta = 270°$ (position D), denoted in Fig. 6.37. The skin friction lines are computed using the first grid surface off the blade, and are depicted on the pressure side (PS) and suction side (SS) of the blade. The top three subplots report the skin friction lines and U_r^* on the PS for steady simulation and the two TD–360 simulations at positions B and D, whereas the bottom three subplots represent the skin friction lines and U_r^* on the SS. Looking at the PS and SS subplots, one may notice that the skin friction lines of the steady simulation are well guided and kept aligned at all spanwise positions. However, for the TD simulation, the skin friction lines are deflected towards the tip at position B and towards the root at position D. This happens due to the impact of the U_r^*, which is different at positions B and D due to yawed wind regime. In zero–yaw wind case U_r^* is constant at all azimuthal positions, and it is always pointing radially outwards, as it can be noticed from the PS and SS steady simulation subplots. Whereas, at positions B and D, U_r^* is pointing radially outwards and radially inwards, respectively. It is also observed that in all TD subplots the skin friction lines deflections are more visible in the inboard part of the blade. The reason for this is the variation of the circumferential velocity along the blade span. Circumferential velocity is very small at the sections close to the root due to small angular velocity, and it significantly increases with the rotor radius, and therefore becomes dominant. Due to this reason these deflections on PS and SS are more visible close to the root. Moreover, as previously pointed out, the skin friction lines confirm that the dynamic stall on the SS of the blade is absent for this operating regime.

Looking at subplots of Fig. 6.45 one would expect that the U_r^* would be increasing and decreasing continuously. However, this does not happen in present case. One plausible reason for this anomaly is that the blade surface, which was generated with the loft tool inside grid generator, is not perfectly smooth. As the U_r^* is visualised on the first grid surface off the blade, any small anomalies in grid surface geometry can cause non–continuous behaviour of the U_r^*. Further away from the blade surface these anomalies disappear, as it may be seen in Fig. 6.46.

Figure 6.46 depicts the comparison of the two–dimensional streamlines and the radial velocity component U_r in the meridional plane of the steady simulation and the two azimuthal positions of TD–360 simulation, $\theta = 90°$ (position B) and $\theta = 270°$ (position D), denoted in Fig. 6.37. Streamlines of the three subplots in Fig. 6.46 highlight the effect of the velocity component parallel to the plane of rotation $\underline{\mathrm{u}}_\infty sin(\delta)$, which occurs due to the yawed flow operating regime. In the steady simulation, the streamlines are slightly inclined in the direction towards the tip, mainly due to the expansion of the streamtube, which occurs across the rotor. However, the inclination angle of the streamlines of the TD simulation is significantly higher than that of the steady simulation. At position B, when the radial velocity component due to rotation and $\underline{\mathrm{u}}_\infty sin(\delta)$ both point radially outwards, the streamlines are inclined towards the tip. At position D, when the radial velocity component due to rotation points radially outwards and $\underline{\mathrm{u}}_\infty sin(\delta)$ points radially inwards, the streamlines are inclined towards the root. Furthermore, close to the tip the air flowing over the pressure side acquires an additional outward velocity component due to the formation of the tip vortex, as the pressure on both sides of the blade will try to equalise. Due to the reasons above, the flow at position B is more three dimensional, and leads to a stronger induced downwash. Conversely, the flow at position D is more two–dimensional and features higher local AoAs. Combined effect of the tip vortex and the inflow gradient across the rotor blade due to the $\underline{\mathrm{u}}_\infty sin(\delta)$ causes larger hysteresis loops when moving towards

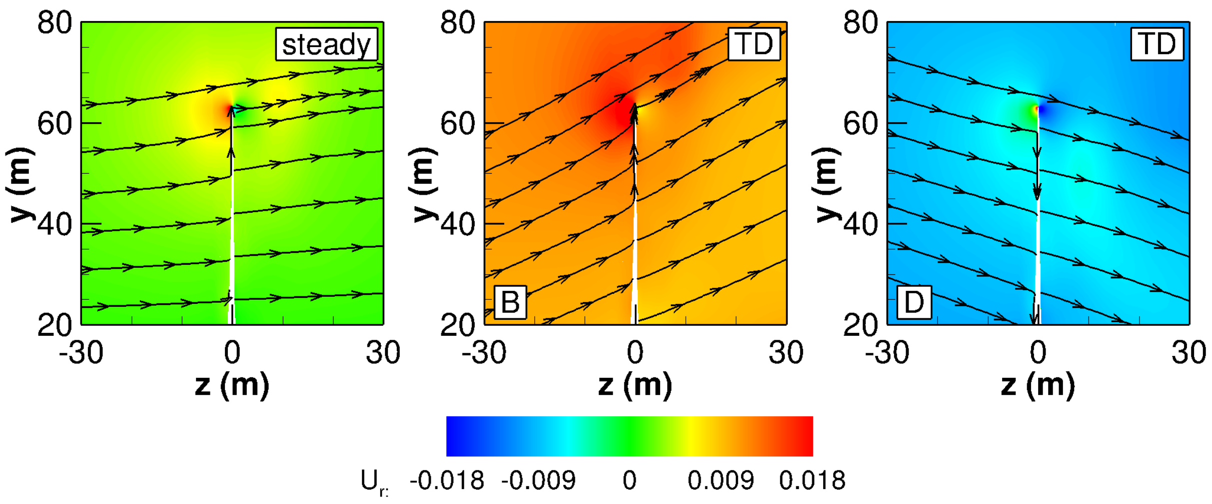


Figure 6.46: Contours of radial velocity in the meridional plane of the 5–MW baseline wind turbine blade for the steady and two TD simulations corresponding to azimuthal positions $\theta = 90°$ (B) and $\theta = 270°$ (D) of the yawed wind periodic simulation.

the tip.

Figure 6.47 depicts the comparison of the streamlines and the x–component of flow vorticity in the meridional plane of the steady simulation and the two azimuthal positions of TD–360 simulation, $\theta = 90°$ (position B) and $\theta = 270°$ (position D), denoted in Fig. 6.37. Three different slices are plotted for the steady and two TD simulations, at three different θ^* positions, where θ^* denotes the position of a slice in meridional plane of the blade. θ^* is negative, when the slice is rotated towards the trailing edge. At $\theta^* = 0°$, the inclination of the streamlines on the suction side of the blade is significantly different between steady simulation and positions B and D, due to the variation of U_r, as previously explained. The same observation holds also for other two θ^* positions. At $\theta^* = 0°$ and $\theta^* = -0.5°$ the formation of the tip vortex is nicely visible for all cases, whereas at $\theta^* = -1°$, the tip vortex is already fully formed.

The contours of the x–component of flow vorticity in the meridional plane in Fig. 6.47 are heavily distorted. The reason for this anomaly is that the mesh in the tip region is too coarse, and the stretching is too large. Unfortunately, due to limited computational resources it was impossible to use better refinement in the tip region of the mesh.

Figure 6.48 depicts the comparison of the static pressure coefficient c_p contours, in the meridional plane of the steady simulation and the two azimuthal positions of TD–360 simulation, $\theta = 90°$ (position B) and $\theta = 270°$ (position D), denoted in Fig. 6.37. The definition of c_p is provided below:

$$c_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty (u_\infty^2)}, \tag{6.21}$$

where p and p_∞ denote the local and freestream static pressure respectively. Such definition of the pressure coefficient is based on the freestream wind velocity u_∞, and thus, maximum value at the stagnation point will vary along the span. This definition is typically used when plotting the pressure coefficient on the entire surface of the HAWT blade, as it becomes too complicated to use the definition based on the relative wind velocity, defined by Eqn. (5.6).

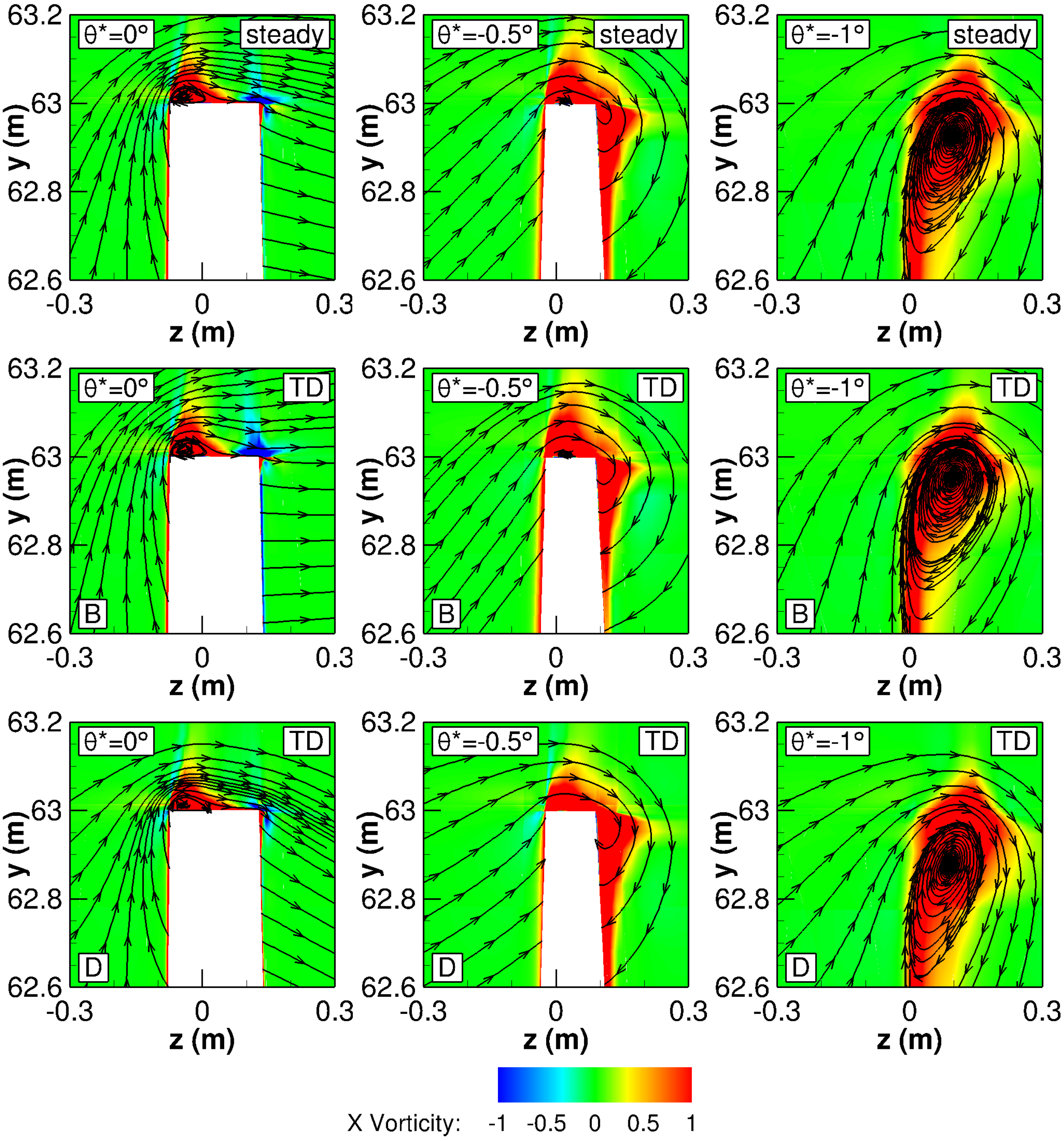


Figure 6.47: Contours of x–component of flow vorticity and 2D streamlines in the meridional plane of the 5–MW baseline wind turbine blade for the steady and two TD simulations corresponding to azimuthal positions $\theta = 90°$ (B) and $\theta = 270°$ (D) of the yawed wind periodic simulation.

The red colour in all nine images corresponds to the high–pressure region, and the blue colour corresponds to the low–pressure region. Looking at the three pictures at $\theta^* = 0°$, it is observed that the red colour at position D is much more intense close to the blade than for the steady simulation and position B. This means that the flow at position D is more two–dimensional, and the three–dimensional effects appear to be much larger at position B. At $\theta^* = -0.5°$ it is clearly visible that the pressure difference across PS and SS is the largest at position D, which suggests that the tip

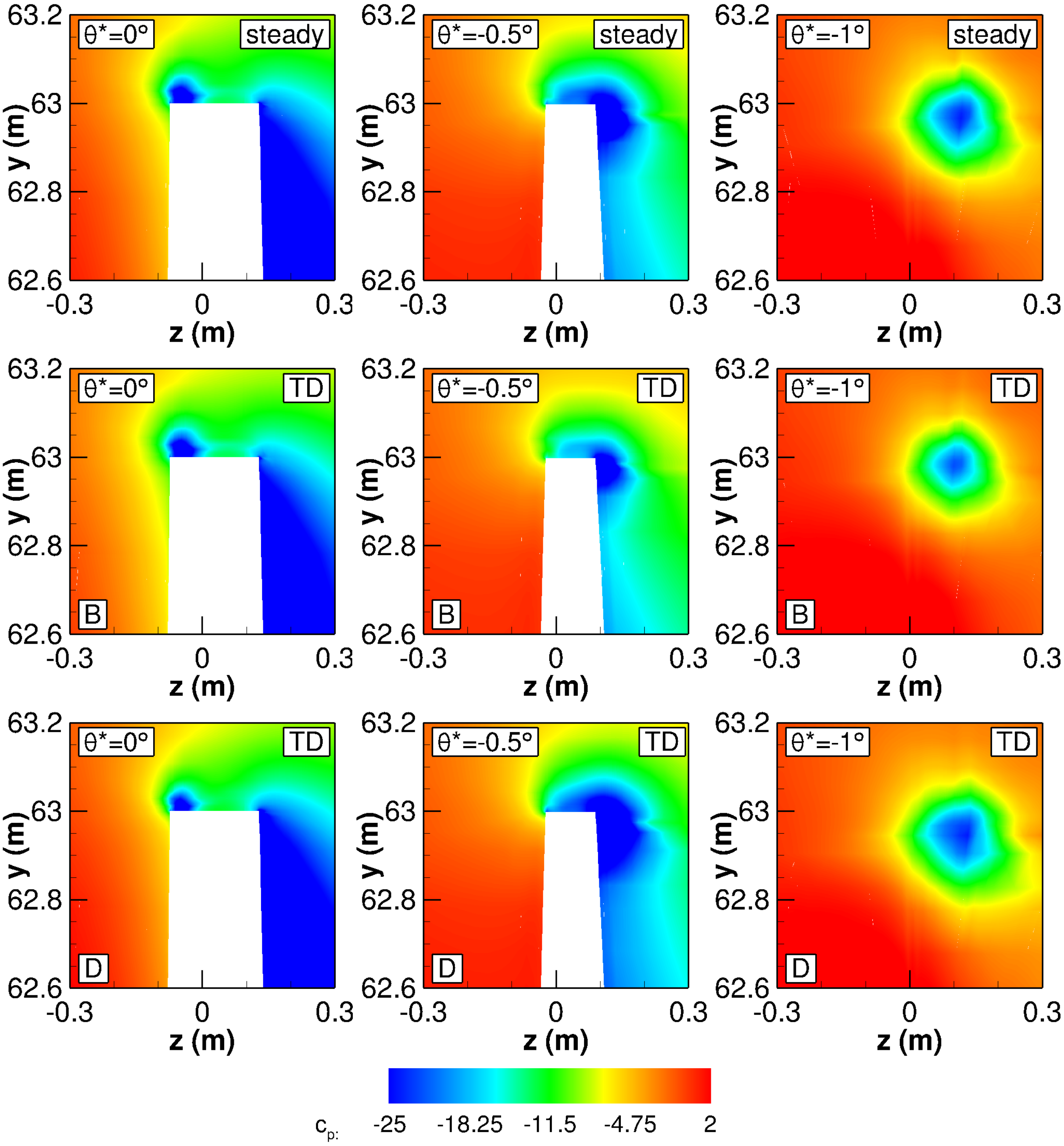


Figure 6.48: Contours of x–component of pressure coefficient c_p in the meridional plane of the 5–MW baseline wind turbine blade for the steady and two TD simulations corresponding to azimuthal positions $\theta = 90°$ (B) and $\theta = 270°$ (D) of the yawed wind periodic simulation.

vortex at position D is the strongest. Even though the tip vortex is stronger at position D than at position B, the aerodynamic loads close to the tip are higher at position D, as previously shown. This suggests, that the magnitude of the tip vortex does not significantly affect aerodynamic loads close to the tip.

Similarly than in Fig. 6.47, the contours of the c_p in the meridional plane in Fig. 6.48 are distorted due to insufficient grid refinement and too large stretching in the tip region.

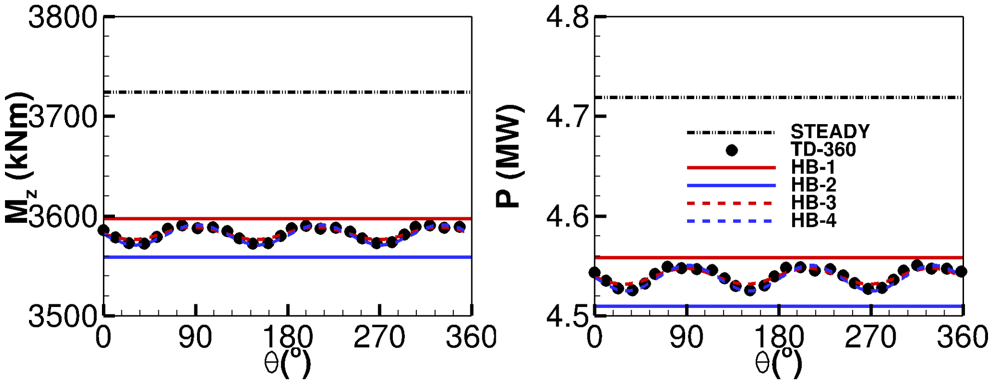


Figure 6.49: Torque and power profiles at 0° (steady) and 20° (HB 1–4 and TD–360) yaw angles, of the 5–MW baseline wind turbine blade.

Figure 6.49 reports the periodic variations of the rotor torque M_z and the rotor power P for the sum of all three blades. The rotor torque and power profiles over one rotor revolution are computed by the steady solver for the straight wind simulation and by the TD and HB solvers for the yaw angle 20° simulations, and are plotted against the azimuthal position θ of the reference blade. Looking at both subplots of Fig 6.49 one may notice there is a noticeable reduction of both M_z and P due to the yawed wind operating regime, leading to an apparent power reduction. Furthermore, although there is a significant variation of M_z on a single blade, the fluctuations of the main shaft are much smaller. Even though the power fluctuations on the main shaft are relatively small, it is important to accurately predict them, as they result in electrical power flickering. From both subplots of Fig 6.49 it is evident that HB–1 and HB–2 simulations are not able to predict any variations of the M_z and P. This happens due to the definition of the HDHB. When using one or two complex harmonics for the problem of the three–bladed turbine, an insufficient number of temporal points within the period is defined. Therefore, when the sum of the periodic profiles of M_z or P of all three blades is done, these variations cancel out. Furthermore, the contribution of the higher–frequency harmonics of the HB–3 and HB–4 analyses provide sufficient number of temporal points within the period, and are both able to predict variations of M_z and P well.

6.2.5 Computational performance of the HB solver

All four HB analyses have required $25,000$ iterations for the convergence of all harmonics of all the force components of these four HB analyses. Each physical time–step of the TD–360 analysis has instead required 700 iterations for the convergence of all force components. Twelve revolutions starting from a freestream initial condition had to be simulated, in order to reach the periodicity error threshold, defined in subsection 6.2.4, below 0.1 %. It has been noticed that the number of iterations required for the convergence of all harmonics of all the force components is fairly independent of N_H.

Figure 6.50 reports the residual convergence histories of the steady and four HB analyses over the first $5,000$ iterations, and the mean residual convergence history of the first period of the TD–360 simulation. The variable on the x–axis is the number of single–grid iterations. For the steady and HB analyses, the variable Δl_r on the

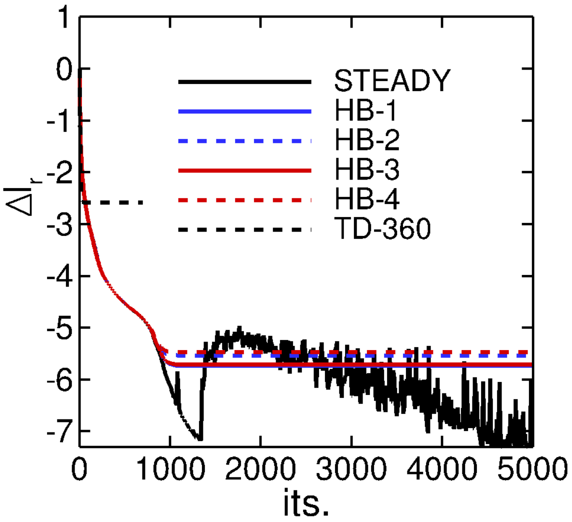


Figure 6.50: Residual convergence histories of steady, TD and HB solvers of the 5–MW baseline wind turbine blade.

y–axis is the logarithm in base 10 of the normalized RMS of all cell–residuals of the RANS and SST $k-\omega$ equations of the $2N_H+1$ snapshots. For the TD–360 analysis, the variable Δl_r on the y–axis is instead the logarithm in base 10 of the RMS of all cell–residuals of the the RANS and SST $k-\omega$ equations of the 360 physical times of the first period. For all steady, HB, and TD simulations, each residual history curve is normalized by the RMS value at the first MG cycle. Looking at four HB curves, one may notice an interesting feature, that the convergence histories of all HB analyses are fairly close to each other. This occurrence indicates that the periodic flow nonlinearity is dominated only by the first two harmonics. Furthermore, the contribution of the progressively smaller higher–frequency harmonics of the HB 3 and HB 4 analyses does not affect significantly the spectrum of the linearized operator associated with the integration of these HB set–ups with respect to that associated with the HB 2 set–up. The HB hysteretic force loops of the subplots in Fig. 6.38, also point to the fact that the first two harmonics are dominant in the Fourier reconstruction of this periodic flow. Inspecting the curves in Fig. 6.38 confirms that the largest differences among the HB results are those between the HB 1 simulation on one hand and the other three HB simulations on the other. This occurrence highlights a significant contribution of the second harmonic to the periodic flow, and rapidly decreasing contributions of the higher order harmonics. Figure 6.50 also reports the convergence history of the steady simulation of the zero–yaw wind flow. It is observed that the curve of the steady residual history is superimposed with HB curves for the first $1,000$ iterations. After this point the steady curve has more oscillatory behaviour and slightly better convergence.

When using the HB smoother provided by Eqn. (3.81) to solve the HB RANS and SST equations, the CPU–time of one HB iteration increases in a moderately superlinear fashion with N_H. This suggest that, for a given number of computer cores used for the simulation, the runtime of a HB N_H simulation with a given number of iterations is higher than $(2N_H+1)$ times the runtime of the steady simulation using the same number of iterations. This overhead is due to the calculation of the HB source term $\Omega V_H D \mathbf{Q}_H$ appearing in Eqn. (3.79), and is proportional to $(2N_H+1)^2$. Such an overhead can be quantified by taking the ratio of the measured CPU–time of one iteration of the

HB N_H analysis and that of one iteration of the steady analysis, and dividing such a ratio by $(2N_H + 1)$. The variable C_{IT} thus obtained is reported in the second row of Table 6.7. It is seen that the overhead for the calculation of the HB source term with the HB 4 analysis makes the average CPU–time of one HB iteration about 9 percent higher than that of one steady iteration.

Table 6.7: Overhead parameter C_{IT} of HB iteration with respect to steady iteration, and speed–up of HB analyses with respect to TD–360 analysis of the 5–MW baseline wind turbine blade.

	HB 1	HB 2	HB 3	HB 4	TD-360	steady
C_{it}	1.049	1.080	1.085	1.089	—	1.000
speed-up	115.3	67.2	47.8	37.0	1.0	—

The HB *speed–up* parameter, defined as the ratio of the runtime of the TD–360 simulation and the HB analysis for the four values of N_H, is reported in the third row of Table 6.7. It is seen that the HB 3 simulation, which yields an excellent estimate of the time–dependent loads, reduces the analysis runtime by a factor 48 with respect to the fully time–resolved TD–360 analysis, which is a remarkable benefit for practical applications.

Chapter 7

Conclusions

7.1 Summary and concluding remarks

A novel 3D steady, TD and HB compressible solvers of the RANS equations coupled to Menter's $k-\omega$ SST turbulence model have been developed, validated, demonstrated and assessed. For validation, several test cases have been used: the laminar delta wing, ONERA M6 wing, S809 airfoil, H–Darrieus vertical–axis wind turbine, oscillating wing and the upwind configuration of the NREL Phase VI rotor in straight and yawed flow conditions. The numerical results have been compared with the experimental data, and/or other widely used NS commercial and research codes. The capabilities of the newly developed solvers have been demonstrated using complex unsteady engineering problems. The work focused on the periodic unsteady flows of the oscillating wing devices and the wind turbine periodic flows in yawed wind. A detailed assessment of the actual benefits achievable by a HB RANS CFD code with the SST turbulence model for the analysis of wind turbine periodic aerodynamics has also been presented.

7.1.1 Oscillating wings

A detailed numerical investigation into the impact of flow three–dimensionality on the power generation efficiency of realistic oscillating wing configurations for marine renewable energy production has been presented. The study was based on the comparative performance assessment of an infinite wing and two aspect ratio 10 wings, one featuring endplates, the other featuring sharp tips. Two high efficiency extraction operating regimes were considered. In both cases, the oncoming stream had $Re = 1.5 \times 10^6$; the wing motion of case A was characterised by a fairly high power generation efficiency of the infinite wing at this Reynolds number and the occurrence of LEVS, whereas the the wing motion of case B was characterised by a high power generation efficiency of the infinite wing, without LEVS. The newly developed TD RANS solver featuring Menter's $k-\omega$ SST turbulence model was used to assess the differences of hydrodynamic performance of the three wing configurations (infinite wing and two aspect ratio 10 wings) and analyse the underlying flow patterns accounting for such differences. Computational grids with 30.7 million–cells were used for the 3D time–dependent simulations.

The mean overall power coefficient of the AR 10 wing with sharp tips and endplates is found to decrease respectively by about 17 % and 12 % with respect to that of the infinite wing for case A and by about 20 % and 15 % with respect to that of the infinite wing for case B. For case A, the finite wing losses are caused both by the reduction of the effective AoA at the near–tip sections induced by the downwash associated with the tip vortices, and the loss of synchronisation of LEVS and pitching motion of the wing. The latter phenomenon results in a lower efficiency of the finite span wing due to a significant loss of pitching power with respect to the ideal infinite wing. For case B, the finite wing losses are caused only by the reduction of the effective AoA at the near–tip sections induced by the downwash associated with the tip vortices.

The loss due to tip vortex–induced downwash depends on the wing tip geometry, and is smaller for the wing with endplates. The pitching power loss, however, does not depend on the wing tip geometry, and affects both AR 10 wings in a qualitatively and quantitatively similar fashion.

In the light of the efficiency reduction associated with the loss of favourable synchronisation between pitching motion and LEVS, when considering finite wing effects, it appears advisable to design these devices avoiding regimes characterised by 2D LEVS, as to minimise losses due to finite wing effects. Alternatively one would have to perform the design optimisation of the wing kinematic parameters making use of costly 3D flow simulations, since the results of 2D optimisation appear to be unsuitable to yield optimal efficiency of the 3D oscillating wing. The loses in case B are bigger than those of case A, however, the efficiency of case A wing with endplates and sharp tips is respectively 34.4 % and 32.6 %, whereas in case B is respectively 38.1 % and 36.0 %. Case B, therefore, yields much higher efficiency than case A. The higher losses for case B are associated with significantly higher peak heaving velocity compared to case A, which is nearly twice as big.

7.1.2 Horizontal–axis wind turbines

Detailed aerodynamic analyses and assessments of the benefits of using a HB RANS CFD code with the $k-\omega$ SST turbulence model for the analysis of the NREL 5–MW baseline wind turbine periodic aerodynamics have been presented. The assessments were based on the analysis of the periodic flow field past the NREL 5–MW blade section in a 20^o 11.4 m/s yawed wind.

Significant hysteresis cycles of the overall forces and moments acting on the 5–MW baseline horizontal–axis with turbine blade were observed, with variations of the tangential and axial force components of about ±5 % and ±6 %, respectively, of their mean values, and variations of the out–of–plane bending moment, the torsional moment and the in–plane bending moment of about ±7 %, ±13 %, and 11 %, respectively, of their mean value. Inspection of the flow solution showed there is no dynamic stall present on the blade, and that the velocity component parallel to the plane of rotation associated with yawed wind flow is responsible for such hysteretic behaviour of aerodynamic loads. This velocity component induces an inflow gradient across the rotor blade, which strength depends on the azimuthal position of the blade, resulting in the variation of the local AoA. Significant variations of the blade loads caused by the yawed flow contributes considerably to material fatigue, and must be accurately predicted. The hysteresis loops have been closely examined at four spanwise positions, $r/R = 0.3$, $r/R = 0.5$, $r/R = 0.7$ and $r/R = 0.95$, to investigate the variation of the hysteresis loops along the blade span. It was observed that the size of the hysteresis curves variations is progressively increasing from the blade root towards the tip. This is caused by the combined effect of the tip vortex and the inflow gradient across the rotor blade.

The HB analysis using 3 complex harmonics reproduced the solution of the fully time–resolved TD–360 analysis nearly 50 times more rapidly than the TD analysis. This points to the fact that HB RANS method has a strong potential of improving utility–scale HAWT design since it reduces the computational burden associated with computing periodic solutions using the conventional time–domain form of the RANS and $k-\omega$ SST equations. These equations are able to predict fatigue–inducing and power–reducing loads with increased reliability with respect to the low–fidelity analysis methods. Recent advances in computer technology, the high computational efficiency of the HB technology, and the fact that HB RANS is able to accurately account for complex unsteady flow features, offers the possibility of using the HB technology for the design optimisation of HAWT rotors, possibly with the initial support of reliable reduced order modelling.

7.2 Future work

On the algorithmic side, more work is needed to improve the stability and the convergence rate of the explicit MG algorithm of the developed CFD framework in the case of realistic high Reynolds number flows, since for the flow problems of this type reported in the thesis the MG solver either had a convergence rate similar to that of the SG iteration, or could not be used without lowering excessively CFD numbers. The difficulties encountered in solving high–Reynolds number flows are caused both by the very high aspect ratios (often between 1,000 and 10,000) of the grids required to resolve very thin boundary layers at high Reynolds number (the performance of MG decreases with growing levels of grid anisotropy), and the numerical stiffness of the SST equations. These stability issues may be reduced by adopting numerical stabilisation procedures such as those reported in [148] and [149]. Additional possibilities for improving the MG stability and convergence rate include using turbulence models with lower numerical stiffness and using different smoothers for the RANS equations and the turbulence model equations. It should also be noted, that the explicit MG stability issues at high Reynolds number affect to the same extent the steady, TD and HB solvers, and therefore, improvements of the explicit MG integration technology is unlikely to alter significantly the conclusions of this work regarding the computational benefits (lower runtimes) of using the HB rather than the TD formulation for the solution of wind turbine periodic flows. Further extensions of this work include the implementation of the transitional turbulence model, the development of an incompressible solver for hydrodynamic and low–speed aerodynamic applications, the implementation of LES and DES modelling capabilities, sliding mesh algorithm, and the implementation of the overset grid approach. The implementation of the dynamic load balancer is already in progress in order to simplify and accelerate the grid generation phase, which is presently constrained by the requirement of all blocks having the equal size.

Further work on the validation side includes the comparison of the numerical results of high–amplitude heaving wings and various pitching wings against the available experimental data. This data will further confirm whether the RANS CFD could be used with a confidence as a high–fidelity design tool for oscillating wing devices. Further validation on yawed HAWT flows is also required, as only one operating regime at two different yaw angles has been considered herein. More operating points of NREL PHASE VI experiment [9] in yawed wind flows should be considered, as well as the MEXICO experiment [29] should be addressed for further thorough validation.

Further work on the application side includes improved HAWT flow predictions making use of a laminar–to–turbulent transition model, and also the use of the newly developed HB solver for other HAWT unsteady flows such as sheared wind regimes and rotor aeroelasticity. It is also believed that the developed technology may be used for general open rotor design optimisation incorporating aerodynamic and aeroacoustic functionals.

Bibliography

[1] W. McKinney and J. DeLaurier. The Wingmill: An Oscillating-Wing Windmill. *Journal of Energy*, 5(1):109–115, 1981.

[2] K.D. Jones, K. Lindsey, and M.F. Platzer. An Investigation of the Fluid-Structure Interaction in an Oscillating-Wing Micro-Hydropower Generator. In Chakrabarti, Brebbia, Almozza, and Gonzalez-Palma, editors, *Fluid Structure Interaction 2*, pages 73–82, Southampton, United Kingdom, 2003. WIT Press.

[3] Engineering Business Ltd 2005. Stingray assembly. Available at: http://www.esru.strath.ac.uk/EandE/Web_sites/05-06/marine_renewables/technology/oschydro.htm [accessed 14/05/2016].

[4] ITPEnergised. Pulse tidal demonstrators. http://www.itpenergised.com/wp-content/uploads/2017/02/Pulse-Case-Study-.pdf [accessed 09/03/2017].

[5] Renewable Energy Focus. Pulse tidal plans array at isle of skye. http://www.renewableenergyfocus.com/view/9780/pulse-tidal-plans-array-at-isle-of-skye/ [accessed 27/07/2016].

[6] T. Kinsey, G. Dumas, G. Lalande, J. Ruel, A. Mehut, P. Viarogue, J. Lemay, , and Y. Jean. Prototype testing of a hydrokinetic turbine based on oscillating hydrofoils. *Renwable energy*, 36:1710–1718, 2011.

[7] E. Strohmaier, J. Dongarra, S. Horst, and M. Meuer. Top500 supercomputer sites. http://www.top500.org [accessed 14/05/2016].

[8] J.W. Slater. Nparc alliance verification and validation archive, January 2015. available at: `http://www.grc.nasa.gov/WWW/wind/valid/archive.html`.

[9] M. M. Hand, D. A. Simms, L. J. Fingersh, D. W. Jager, J. R. Cotrell, S. Schreck, and S. M. Larwood. Unsteady aerodynamics experiment phase vi: Wind tunnel test configurations and available data campaigns. Technical Report NREL/TP-500-29955, NREL, Golden, CO, USA, Dec 2001.

[10] Ian S. F. Jones. *Engineering Strategies for Greenhouse Gas Mitigation.* Cambridge University Press, 2011. Cambridge Books Online.

[11] M.O.L. Hansen, J.N. Sorensen, S. Voutsinas, N. Sorensen, and H.A. Madsen. State of the art in wind turbine aerodynamics and aeroelasticity. *Progress in Aerospace Sciences*, 42:285–330, 2006.

[12] P. Jain. *Wind energy engineering.* Mcgraw-Hill, New York, NY, USA, 2011.

[13] C. Bak. Aerodynamic design of wind turbine rotors. In Wolfgang Gentzsch and Uwe Harms, editors, *Advances in wind turbine blade design and materials*, volume 47 of *Energy*, pages 59–108. Woodhead Publishing, Cambridge, UK, 2013.

[14] K. Jones and M. Platzer. Numerical computation of flapping-wing propulsion and power extraction. AIAA paper 97-0826, January 1997. 35th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada.

[15] K. Jones, S. Davids, and M. Platzer. Oscillating-wing power generation. FEDSM paper 99-7050, July 1999. Joint Fluids Engineering Conference, San Francisco, California.

[16] Thomas Kinsey. *Analysis, Optimization and Demonstration of a New Concept of Hydroki-netic Turbine Based on Oscillating Hydrofoils.* PhD thesis, Faculty of Science and Engineering, Laval University, United Kingdom, 2011.

[17] J. Young, J.C.S. Lai, and M.F. Platzer. A review of progress and challenges in flapping foil power generation. *Progress in Aerospace Sciences*, 67:2–28, 2014.

[18] Q. Xiao and Q. Zhu. A review of flow energy harvesters based on flapping foils. *Journal of Fluids and Structures*, 46:174–191, 2014.

[19] Q. Xiao, W. Liao, S. Yang, and Y. Peng. How motion trajectory affects energy extraction performance of a biomimic energy generator with an oscillating foil? *Renewable Energy*, 37:67–75, 2012.

[20] I. Fenercioglu, Zaloglu B., J. Young, M.A. Ashraf, J.C.S. Lai, and M.F. Platzer. Flow structures around an oscillating-wing power generator, March 2015.

[21] T. Kinsey and G. Dumas. Parametric Study of an Oscillating Airfoil in a Power-Extraction Regime. *AIAA Journal*, 46(6):1318–1330, June 2008.

[22] M.S. Campobasso and J. Drofelnik. Compressible Navier-Stokes analysis of an oscillating wing in a power-extraction regime using efficient low-speed preconditioning. *Computers and Fluids*, 67:26–40, August 2012.

[23] T. Kinsey and G. Dumas. Computational Fluid Dynamics Analysis of a Hydrokinetic Turbine Based on Oscillating Hydrofoils. *Journal of Fluids Engineering*, 134:021104.1–021104.16, February 2012.

[24] P.R. Spalart and S.R. Allmaras. A one-equation turbulence model for aerodynamic flows. *La Recherche Aerospatiale*, 1:5–21, 1994.

[25] T. Kinsey and G. Dumas. Three-Dimensional Effects on an Oscillating-Foil Hydrokinetic Turbine. *Journal of Fluids Engineering*, 134:071105.1–071105.11, July 2012.

[26] M.S. Campobasso, A. Piskopakis, J. Drofelnik, and A. Jackson. Turbulent Navier-Stokes Analysis of an Oscillating Wing in a Power-Extraction Regime Using the Shear Stress Transport Turbulence Model. *Computers and Fluids*, 88:136–155, December 2013.

[27] T. Kinsey and G. Dumas. Optimal Operating Parameters for an Oscillating Foil Turbine at Reynolds Number 500,000. *AIAA Journal*, 52(9):1885–1895, September 2014.

[28] J. Manwell, J. McGowan, and A. Rogers. *Wind Energy Explained. Theory, Design and Application Second Edition.* John Wiley and Sons Ltd., 2009.

[29] H Snel, J G Schepers, and B Montgomerie. The mexico project (model experiments in controlled conditions): The database and first results of data processing and interpretation. *Journal of Physics: Conference Series*, 75(1):012014, 2007.

[30] S. Schreck. The nrel full-scale wind tunnel experiment introduction to the special issue. *Wind Energy*, 5(2-3):77–84, 2002.

[31] N. N. Sø rensen, J. A. Michelsen, and S. Schreck. Navier-stokes predictions of the nrel phase vi rotor in the nasa ames 80 ft Œ 120 ft wind tunnel. *Wind Energy*, 5(2-3):151–169, 2002.

[32] E.N. Duque, M.D. Burklund, and W. Johnson. Navier-stokes and comprehensive analysis performance predictions of the nrel phase vi experiment. *ASME. J. Sol. Energy Eng.*, 125(4):457–467, 2003.

[33] A. Le Pape and J. Lecanu. 3D Navier-Stokes computations of a stall-regulated wind turbine. *Wind Energy*, 7:309–324, 2004.

[34] A. Le Pape and V. Gleize. Improved Navier-Stokes Computations of a Stall-regulated Wind Turbine Using Low Mach Number Preconditioning. AIAA paper 2006-1502, January 2006. 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada.

[35] Gómez-Iradi S., Steijl R., and Barakos G.N. Development and validation of a cfd technique for the aerodynamic analysis of hawt. *ASME. J. Sol. Energy Eng.*, 3(131):031009–031009–13, 2009.

[36] G. Yu, X. Shen, M. Zhu, and Z. Du. An insight into the separate flow and stall delay for HAWT. *Renewable Energy*, 36:69–76, 2011.

[37] M. Moshfeghi, Y.J. Song, and Xie Y.H. Effects of near-wall grid spacing on sst-k-£ model using {NREL} phase {VI} horizontal axis wind turbine. *Journal of Wind Engineering and Industrial Aerodynamics*, 107£108:94 – 105, 2012.

[38] D.O. Yu, J.Y. You, and O.J. Kwon. Numerical investigation of unsteady aerodynamics of a horizontal-axis wind turbine under yawed flow conditions. *Wind Energy*, 16(5):711–727, 2013.

[39] J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-MW Reference Wind Turbine for Offshore System Development. Technical Report NREL/TP-500-38060, NREL, Golden, CO, USA, 2009.

[40] N. Sø rensen and J. Johansen. Upwind: aerodynamics and aero-elasticity rotor aerodynamics in atmospheric shear flow. Proceedings of The European Wind Energy Conference & Exhibition, EWEC 2007, May 2007. European Wind Energy Conference & Exhibition, EWEC 2007.

[41] R. Chow and C. van Dam. Inboard stall and separation mitigation techniques on wind turbine rotors. AIAA paper 2011-152, January 2011. 49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition Orlando, Florida.

[42] R. Chow and C.P. van Dam. Verification of computational simulations of the nrel 5 mw rotor with a focus on inboard flow separation. *Wind Energy*, 15(8):967–981, 2012.

[43] R. Chow and C.P. van Dam. Computational investigations of blunt trailing-edge and twist modifications to the inboard region of the nrel 5£mw rotor. *Wind Energy*, 16(3):445–458, 2013.

[44] N. Troldborg, F. Zahle, P. Rethore, and N. Sorensen. Comparison of the wake of different types of wind turbine cfd models. AIAA paper 2012-0237, January 2012. 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition Nashville, Tennessee.

[45] M. Imiela and F. Wienke. Towards multidisciplinary wind turbine design using high-fidelity methods. AIAA paper 2015-1462, January 2015. 33rd Wind Energy Symposium Kissimmee, Florida.

[46] K.C. Hall, J.P. Thomas, and W.S. Clark. Computations of Unsteady Nonlinear Flows in Cascades Using a Harmonic Balance Technique. *AIAA Journal*, 40(5):879–886, May 2002.

[47] E. van der Weide, A.K. Gopinath, and A. Jameson. Turbomachinery Applications with the Time Spectral Method. AIAA paper 2005-4905, June 2005. 17th AIAA Computational Fluid Dynamics Conference, Torontp, Ontario, Canada.

[48] X. Su and X. Yuan. Implicit Solution of Time-Spectral Method for Periodic Unsteady Flows. *International Journal for Numerical Methods in Fluids*, 63(7):860–876, 2010.

[49] A. Da Ronch, A.J. McCracken, K.J. Badcock, , M. Widhalm, and M.S. Campobasso. Linear Frequency Domain and Harmonic Balance Predictions of Dynamic Derivatives. *Journal of Aircraft*, 50(3):694–707, 2013.

[50] F. Sicot, G. Puigt, and M. Montagnac. Block-Jacobi Implicit Algorithms for the Time Spectral Method. *AIAA Journal*, 46(12):3080–3089, December 2008.

[51] M. A. Woodgate and K. J. Badcock. Implicit Harmonic Balance Solver for Transonic Flows with Forced Motions. *AIAA Journal*, 47(4):893–901, April 2009.

[52] M.S. Campobasso and M.H. Baba-Ahmadi. Analysis of Unsteady Flows Past Horizontal Axis Wind Turbine Airfoils Based on Harmonic Balance Compressible Navier-Stokes Equations with Low-Speed Preconditioning. *Journal of Turbomachinery*, 134(6), November 2012.

[53] J. Howison and K. Ekici. Unsteady Analysis of Wind Turbine Flows Using the Harmonic Balance Method. AIAA paper 2013-1107, January 2013. 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Grapewine (Dallas/Ft. Worth Region), Texas.

[54] J. Howison and K. Ekici. Dynamic stall analysis using harmonic balance and correlation based $\gamma - re_\theta$ transition models for wind turbine applications. *Wind Energy*, 18:2047–2063, 2015.

[55] M.S. Campobasso, F. Gigante, and J. Drofelnik. Turbulent Unsteady Flow Analysis of Horizontal Axis Wind Turbine Airfoil Aerodynamics Based on the Harmonic Balance Reynolds-Averaged Navier-Stokes Equations. ASME paper GT2014-25559, June 2014.

[56] M.S. Campobasso, J. Drofelnik, and F. Gigante. Comparative assessment of the harmonic balance Navier-Stokes technology for horizontal and vertical axis wind turbine aerodynamics. *Computers & Fluids*, 136:354 – 370, 2016.

[57] J.C. Howison. *Aeroelastic Analysis of a Wind Turbine Blade Using the Harmonic Balance Method*. PhD thesis, University of Tennessee, Knoxville, United States, 2015.

[58] S.G Horcas, F. Debrabandere, B. Tartinville, C. Hirsch, and Grégory Coussement. Rotor-tower interactions of dtu 10mw reference wind turbine with a non-linear harmonic method. *Wind Energy*, 20(4):619–636, 2017.

[59] L. He and W. Ning. Efficient approach for analysis of unsteady viscous flows in turbomachines. *AIAA Journal*, 36:11:2005–2012, 1998.

[60] M.O.L. Hansen. *Aerodynamics of wind turbines, Second edition*. Earthscan, London, 2008.

[61] S. Tsalicoglou, S. Barber, N. Cholani, and R.S. Abhari. Effect of Flow Inclination on Wind Turbine Performance. *Journal of Engineering for Gas Turbine and Power*, 134(12):122601.1–122601–8, December 2012.

[62] H. Himmelskamp. Profile investigations of a rotating airscrew. In *PhD Dissertation Goettingen 1945, Report and Translation 832*. MAP Voelkenrode, 1947.

[63] H.A. Madsen, N.N. Sorensen, and S. Schreck. Yaw Aerodynamics Analyzed with Three Codes in Comparison with Experiments. AIAA paper 2003-0519, January 2003. 41st AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada.

[64] N. Sezer-Uzol, A. Gupta, and L.N. Long. 3D Time-Accurate Inviscid and Viscous cfd Simulations of Wind Turbine Rotor Flow Fields. In I.H. Tuncer, U. Guelcat, D.R. Emerson, and K. Matsuno (Eds.), editors, *Parallel Computational Fluid Dynamics 2007*, volume 67 of *Lecture Notes in Computational Science and Engineering*, pages 457–464. Springer-Verlag, 2007.

[65] C. Tsalicoglou, S. Jafari, S. Chokani, and R.S. Abhari. Rans computations of wind turbine near-wake aerodynamics in uniform and yawed inflow. ASME paper GT2013-96034, June 2013.

[66] J. Blazek. *Computational Fluid Dynamics: Principles and Applications, Third Edition*. Elsevier, Oxford, 2015.

[67] J.F. Wendt. *Computational Fluid Dynamics An Introduction*. Springer-Verlag Berlin Heidelberg, Berlin, Germany, 2009.

[68] C. Hirsch. *Numerical Computation of Internal and External Flows*. John Wiley & Sons, Ltd, Burlington, MA, USA, 2007.

[69] F.R. Menter. Two-equation Turbulence-models for Engineering Applications. *AIAA Journal*, 32(8):1598–1605, August 1994.

[70] F.R. Menter. Zonal Two Equation K-ω Turbulence Models for Aerodynamic Flows. AIAA paper 93-2906, July 1993. 24th AIAA Fluid Dynamics Conference, Orlando, Florida.

[71] Osborne Reynolds. On the Dynamical Theory of Incompressible Viscous Fluids and the Determination of the Criterion. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 186:123–164, 1895.

[72] D.C. Wilcox. *Turbulence Modeling for CFD.* DCW Industries, Inc., California, USA, 2006.

[73] W.P. Jones and B.E. Launder. The Calculation of Low-Reynolds-Number Phenomena with a Two-Equation Model for Turbulence. *International Journal of Heat and Mass Transfer*, 16:1119–1130, 1973.

[74] D.C. Wilcox. Reassessment of the Scale-Determining Equation for Advanced Turbulence Models. *AIAA Journal*, 26(11):1299–1310, November 1988.

[75] J.E. Bardina, P.G. Huang, and T.J. Coakley. Turbulence Modeling Validation, Testing, and Development. Technical Report Technical Memorandum 110446, NASA, 1997.

[76] C.W Hirt, A.A Amsden, and J.L Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14(3):227 – 253, 1974.

[77] J. Donea, S. Giuliani, and J.P. Halleux. An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 33(1):689 – 723, 1982.

[78] J. J. Adamczyk. Model equation for simulating flows in multistage turbomachinery. Technical Report NASA-TM-86869, Glenn Research Center, Cleveland, OH, USA, November 1984.

[79] M.S. Campobasso. *Effects of Flow Instabilities on the Linear Harmonic Analysis of Unsteady Flow in Turbomachinery.* PhD thesis, University of Oxford, United Kingdom, August 2004.

[80] D.G. Holmes and S.S. Tong. A three-dimensional euler solver for turbomachinery blade rows. *ASME. J. Eng. Gas Turbines Power*, 107(2):258–264, 1985.

[81] R. K. AGARWAL and J. E. DEESE. Euler calculations for flowfield of a helicopter rotor in hover. *Journal of Aircraft*, 24:4:231–238, 1987.

[82] Amrit Raj Ghosh. *Solutions to Three-Dimensional Thin-Layer Navier-Stokes Equations in Rotating Coordinates for Flow Through Turbomachinery.* PhD thesis, Mississippi State Univ., Dept. of Aerospace Engineering, Mississippi State, MS United States, 1996.

[83] J. Chen, A Ghosh, K. Sreenivas, and D. Whitfield. Comparison of computations using navier-stokes equations in rotating and fixed coordinates for flow through turbomachinery. 35th Aerospace Sciences Meeting and Exhibit, January 1997. Reno, NV, U.S.A.

[84] C. B. Allen. Multigrid convergence of inviscid fixed- and rotary-wing flows. *International Journal for Numerical Methods in Fluids*, 39(2):121–140, 2002.

[85] Obayashi S. Freestream capturing for moving coordinates in three dimensions. *AIAA Journal*, 30 (4):1125 – 1128, 1992.

[86] L. Liu, J.P. Thomas, E.W. Dowell, P. Attar, and K.C. Hall. A comparison of classical and high dimensional harmonic balance approaches for a Duffing oscillator. *Journal of Computational Physics*, 215(1):298–320, 2006.

[87] M.S. McMullen and A. Jameson. The computational efficiency of non-linear frequency domain methods. *Journal of Computational Physics*, 212(2):637–661, 2006.

[88] J. H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics*. Springer-Verlag Berlin Heidelberg, Berlin, Germany, 2002.

[89] T.H. Pulliam and D.W. Zingg. *Fundamental Algorithms in Computational Fluid Dynamics*. Springer International Publishing, 2014.

[90] Antony Jameson. Solution of the Euler equations for two dimensional transonic flow by a multigrid method. *Applied Mathematics and Computation*, 13(3):327 – 355, 1983.

[91] J.L. Thomas and M.D Salas. Far-field boundary conditions for transonic lifting solutions to theeuler equations. *AIAA Journal*, 24(7):1074–1080, 1986.

[92] Minghan Yan. *Time-domain and harmonic balance turbulent Navier-Stokes analysis of wind turbine aerodynamics using a fully coupled low-speed preconditioned multigrid solver*. PhD thesis, Lancaster University, United Kingdom, 2015.

[93] E. Ferrer and X. Munduate. CFD predictions of transition and distributed roughness over a wind turbine airfoil. AIAA paper 2009-269, January 2009. 47th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, Orlando, Florida.

[94] Andreas Piskopakis. *Time-domain and harmonic balance turbulent Navier-Stokes analysis of oscillating foil aerodynamics*. PhD thesis, University of Glasgow, United Kingdom, 2014.

[95] P.L. Roe. Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes. *Journal of Computational Physics*, 43:357–372, 1981.

[96] Bram van Leer. Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method. *Journal of Computational Physics*, 32(1):101 – 136, 1979.

[97] S.K Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Jour Mat. Sb.*, 47(89)(3):271–306, 1595.

[98] Stanley Osher. Numerical solution of singular perturbation problems and hyperbolic systems of conservation laws. In L.S. Frank O. Axelsson and A. Van Der Sluis, editors, *Analytical and Numerical Approaches to Asymptotic Problems in AnalysisProceedings ofthe Conference on Analytical and Numerical Approachesto Asymptotic Problems*, volume 47 of *North-Holland Mathematics Studies*, pages 179 – 204. North-Holland, 1981.

[99] S. Osher and F. Solomon. Upwind difference schemes for hyperbolic systems of conservation laws. *Math. Comp.*, 38 (158):339 – 374, 1982.

[100] G.D. van Albada, B. van Leer, and W.W. Roberts Jr. A Comparative Study of Computational Methods in Cosmic Gas Dynamics. *Astronomy and Astrophysics*, 108:76–84, 1982.

[101] A. Harten, P.D. Lax, and B. van Leer. On upstream differencing and godunov-type schemes for hyperbolic conservation laws. *SIAM Review*, 25(1):35–61, 1983.

[102] H.C. Yee. Upwind and symmetric shock-capturing schemes. NASA Technical Memorandum 89464, NASA Aimes Research Center, Moffett Field, CA, USA, May 1987.

[103] H.C. Yee. A class of high-resolution explicit and implicit shock-capturing methods. NASA Technical Memorandum 101088, NASA Aimes Research Center, Moffett Field, CA, USA, February 1989.

[104] Fletcher C. A. *Computational techniques for fluid dynamics 2.* Springer-Verlag New York, NY, USA, 1988.

[105] M. Vinokur. An analysis of finite-difference and finite-volume formulations of conservation laws. *J. Comput. Phys.*, 81(1):1–52, 1989.

[106] A. Rizzi, P. Eliasson, I. Lindblad, C. Hirsch, C. Lacor, and J. Haeuser. The engineering of multiblock/multigrid software for navier-stokes flows on structured meshes. *Computers & Fluids*, 22(2):341 – 367, 1993.

[107] Kivanc Ekici, Kenneth C. Hall, and Earl H. Dowell. Computationally fast harmonic balance methods for unsteady aerodynamic predictions of helicopter rotors. *Journal of Computational Physics*, 227(12):6206 – 6225, 2008.

[108] A. Jameson, W. Schmidt, and E. Turkel. Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes. AIAA paper 81-1259, June 1981. 14th AIAA Fluid and Plasma Dynamic Conference, Palo Alto, California.

[109] F. Liu and X. Zheng. A Strongly Coupled Time-Marching Method for Solving the Navier-Stokes and $K - \omega$ Turbulence Model Equations with Multigrid. *Journal of Computational Physics*, 128(2):289–300, 1996.

[110] F.B. Lin and F. Sotiropoulos. Strongly-Coupled Multigrid Method for 3-D Incompressible Flows Using Near-Wall Turbulence Closures. *Journal of Fluids Engineering*, 119:314–324, June 1997.

[111] P. Eliasson and S. Wallin. A positive multigrid scheme for computations with two-equation turbulence models. In *European Congress on Computational Methods in Applied Sciences and Engineering*, Barcellona, Spain, September 2000.

[112] A. Jameson. Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings. AIAA paper 91-1596, June 1991. 10th AIAA Computational Fluid Dynamics Conference, Honolulu, Hawaii.

[113] A. Arnone, M.-S. Liou, and L.A. Povinelli. Multigrid Time-Accurate Integration of Navier-Stokes Equations. Technical Memorandum TM 106373, ICOMP-93-37, NASA Lewis Research Center, Cleveland, OH, USA, November 1993.

[114] N.D. Melson, D. Sanetrik, and H.L. Atkins. Time-accurate Navier-Stokes Calculations with Multigrid Acceleration. *Proc.* 6^{th} *Copper Mountain Conference on Multigrid Methods*, pages II423–II437, 1993.

[115] In Shamoon Jamshed, editor, *Using {HPC} for Computational Fluid Dynamics.* Academic Press, Oxford, 2015.

[116] Victor Eijkhout. *Introduction to High Performance Scientific Computing.* Lulu.com, 2012.

[117] Michael J. Flynn. Some computer organizations and their effectiveness. *IEEE Trans. Comput.*, 21(9):948–960, 1972.

[118] Georg Hager and Gerhard Wellein. *Introduction to High Performance Computing for Scientists and Engineers.* CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 2010.

[119] John T. Feo, David C. Cann, and Rodney R. Oldehoeft. Data-flow processing a report on the sisal language project. *Journal of Parallel and Distributed Computing*, 10(4):349 – 366, 1990.

[120] Ian Foster, Robert Olson, and Steven Tuecke. Productive parallel programming: The pcn approach. *Scientific Programming*, 1:51–66, 1992.

[121] MPI Forum. Openmp application programming interface, version 4.5. available from: `http://www.openmp.org` [accessed 14/05/2016].

[122] MPI Forum. MPI: A message-passing interface standard. available at: `http://www.mpi-forum.org` [accessed 14/05/2016].

[123] Rajkumar Buyya. *High Performance Cluster Computing: Programming and Applications.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 1999.

[124] A. Jackson and M.S. Campobasso. Shared-memory, Distributed-memory and Mixed-mode Parallelization of a CFD Simulation Code. *Computer Science Research and Development*, 26(3-4):187–195, 2011.

[125] A. Jackson, M.S. Campobasso, and M.H. Baba-Ahmadi. On the Parallelization of a Harmonic Balance Compressible Navier-Stokes Solver for Wind Turbine Aerodynamics. ASME paper GT2011-45306, 2011.

[126] Marc Snir, Steve Otto, Steven Huss-Lederman, David Walker, and Jack Dongarra. *MPI-The Complete Reference, Volume 1: The MPI Core.* MIT Press, Cambridge, MA, USA, 2nd. (revised) edition, 1998.

[127] W. Gropp, , E. Lusk, and A. Skjellum. *Using MPI (3rd Ed.): Portable Parallel Programming with the Message-passing Interface.* MIT Press, Cambridge, MA, USA, 2014.

[128] W. Gropp, T. Hoefler, R. Thakur, and Ewing Lusk. *Using Advanced MPI: Modern Features of the Message-Passing Interface.* MIT Press, Cambridge, MA, USA, 2014.

[129] The UK National Supercomputing Service ARCHER. Archer user guide. available at: `http://archer.ac.uk` [accessed 19/05/2016].

[130] CFD General Notation System. Cgns version 3.3.0. available at: `http://cgns.github.io` [accessed 18/05/2016].

[131] The UK National Supercomputing Service ARCHER. Archer phase 2. available at: `http://archer.ac.uk` [accessed 19/05/2016].

[132] M.S. Campobasso and M.H. Baba-Ahmadi. Ad-hoc Boundary Conditions for CFD Analyses of Turbomachinery Problems with Strong Flow Gradients at Farfield Boundaries. *Journal of Turbomachinery*, 133(4), October 2011.

[133] C.L. Rumsey. Test/validation cases. available at: `http://cfl3d.larc.nasa.gov/` [accessed 20/05/2016].

[134] B. Eisfeld. Reynolds stress modelling for complex aerodynamic flows. ECCOMAS CFD, June 2010.

[135] J. Mayeur, A. Dumont, D. Destarac, and V. Gleize. Rans simulations on tmr test cases and m6 wing with the onera elsa flow solver. AIAA paper 2015-1745, January 215. 53rd AIAA Aerospace Sciences Meeting, Kissimmee, Florida.

[136] R.G. Silva, J.L.F Azevedo, and E Basso. Simulation of onera m6 wing flows for assessment of turbulence modeling capabilities. AIAA paper 2016-0549, January 2016. 54th AIAA Aerospace Sciences Meeting, San Diego, California, USA.

[137] V. Schmitt and F. Charpin. Pressure distributions on the ONERA-M6-wing at transonic mach numbers. Technical report, Office National d'Etudes et Recherches Aerospatiales, 92320, Chatillon, France, 1979.

[138] NUMECA INTERNATIONAL. Fine/turbo. available at: `http://www.numeca.com` [accessed 06/06/2016].

[139] D.M. Somers. Design and Experimental Results for the S809 Airfoil. Technical Report NREL/SR-440-6918, NREL, Golden, CO, USA, 1997.

[140] V.K. Lakshminarayan. *Computational investigation of micro-scale coaxial rotor aerodynamics in hover*. PhD thesis, University of Maryland, MD, United States, 2009.

[141] A.C. Aranake, V.K. Lakshminarayan, and K. Duraysami. Computational analysis of shrouded wind turbine configurations using a 3-dimensional RANS solver. *Renewable Energy*, 75:818–832, 2015.

[142] M. Raciti-Castelli, A. Englaro, and E. Benini. The Darrieus wind turbine: proposal for a new performance prediction model based on cfd. *Energy*, 36:4919–4934, 2011.

[143] M. Raciti-Castelli, G. Pavesi, L. Battisti, E. Benini, and G. Ardizzon. Modeling strategy and numerical validation for a Darrieus vertical axis micro-wind turbine. ASME paper IMECE-2010-39548, November 2010. ASME 2010 International Mechanical Engineering Congress and Exposition, Vancouver, Canada.

[144] F. Balduzzi, A. Bianchini, F.A. Gigante, G. Ferrara, M.S. Campobasso, and L. Ferrari. Parametric and Comparative Assessment of Navier-Stokes CFD Methodologies for Darrieus Wind Turbine Performance Analysis. ASME paper GT2015-42663, June 2015. ASME Turbo Expo 2015, Gas turbine Technical Conference and Exposition, Montreal, Canada.

[145] Ansys. Fluent theory guide. Technical Report release 14.5, 2013.

[146] Sugoi Gomez-Iradi. *CFD for Horizontal Axis Wind Turbines.* PhD thesis, University of Liverpool, United Kingdom, 2009.

[147] A. Betz. Das Maximum der Teoretisch Möglichen Ausnützung des Windes Durch Windmotoren. *Zeitschrift für das Gesamte Turbinenwesen*, 20:307–309, September 1920.

[148] K.C. Hall, J.P. Thomas, and E.H. Dowell. Proper orthogonal decomposition technique for transonic unsteady aerodynamic flows. *AIAA Journal*, 38(10):1853–1862, October 2000.

[149] M. S. Campobasso and M. B. Giles. Stabilization of Linear Flow Solver for Turbomachinery Aeroelasticity Using Recursive Projection Method. *AIAA Journal*, 42(9):1765–1774, September 2004.

[150] J. JEong and F. Hussain. On the identification of a vortex, 1995.

[151] M.S. Campobasso, A. Mattheiss, U. Wenger, A. Arnone, and P. Boncinelli. Complementary Use of CFD and Experimental Measurements to Assess the Impact of Shrouded and Cantilevered Stators in Axial Compressors. ASME paper 99-GT-208, June 1999.

[152] C. Lindenburg, E. Bot, and H.B. Hendriks. Ecn-c-00-077, 2001. Petten.

[153] C Lindenburg. Aeroelastic modelling of the lmh64-5 blade., 2002. DOWEC-02-KL-083/0, ECN, Petten.

[154] H.B. Hendriks and M.B. Zaaijer. Dowec: exective summary of the public research activities, 2004. ECN, Petten, 2004.

[155] Peter J. Schubel and Richard J. Crossley. Wind turbine blade design. *Energies*, 5(9), 2012.

[156] H. von Renouard and E. Hau. *Wind Turbines: Fundamentals, Technologies, Application, Economics.* Springer Berlin Heidelberg, 2013.

[157] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen differenzengleichungen der mathematischen physik. *Mathematische Annalen*, 100(1):32–74.

[158] L. Martinelli. *Calculations of viscous flows with a multigrid method.* PhD thesis, Princeton Univ., NJ., 1987.

[159] A. Jameson and T.J. Baker. Solution of the Euler Equations for Complex Configurations. AIAA paper 83-1929, July 1983. 6th AIAA Computational Fluid Dynamics Conference, Danvers, Massachusetts.

[160] L. Martinelli and A. Jameson. Validation of a multigrid method for the reynolds averaged equations. 26th Aerospace Sciences Meeting Reno, NV, U.S.A., 1988.

[161] R. C. Swanson, E. Turkel, and J. A. White. An effective multigrid method for high-speed flows. *Communications in Applied Numerical Methods*, 8(9):671–681, 1992.

[162] R.P Fedorenko. A relaxation method for solving elliptic differential equations. *USSR Comput. Math. and Math. Phys.,*, (1):1092£1096, 1962.

[163] Achi Brandt. *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics: Vol. I General Lectures. Fundamental Numerical Techniques July 3–7, 1972 Universities of Paris VI and XI*, chapter Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems, pages 82–89. Springer Berlin Heidelberg, Berlin, Heidelberg, 1973.

[164] Achi Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, 1977.

[165] W. Schröder and D. Hänel. An unfactored implicit scheme with multigrid acceleration for the solution of the navier-stokes equations. *Computers & Fluids*, 15(3):313 – 336, 1987.

[166] Barry Koren. Multigrid and defect correction for the steady navier-stokes equations. *Journal of Computational Physics*, 87(1):25 – 46, 1990.

[167] J.W. Ruge and K. Stüben. Algebraic multigrid. multigrid methods. *SIAM frontiers in applied mathematics*, pages 73–131, 1987.

Appendix A

Numerical dissipation

In this section the derivation of numerical dissipation has been described first. Furthermore, the expression of numerical dissipation has been written in such form that maximal computational efficiency of the numerical implementation is assured.

Let us rewrite the expression of numerical dissipation given by Eqn. (3.12) for convenience:

$$\delta\mathbf{\Phi} = |K_U|\,\delta\mathbf{U} = P|\Lambda|P^{-1}\delta\mathbf{U} = P|\Lambda|\delta\mathbf{W}. \tag{A.1}$$

P is the matrix of right eigenvectors of K_U and the inverse matrix P^{-1} is the matrix of left eigenvectors of K_U. Λ is the diagonal matrix of eigenvalues of K_U, and $\delta\mathbf{W}$ represent the characteristic variables, defined by $\delta\mathbf{W} = P^{-1}\delta\mathbf{U}$. Equation (A.1) emphasises that the construction of the numerical dissipation requires the calculation of the eigenvalues and the eigenvectors of K_U. The Jacobian matrix K_U can be written as the sum of the three flux Jacobians of the convective fluxes A, B and C, defined by Eqn. (A.2):

$$K_U = An_x + Bn_y + Cn_z. \tag{A.2}$$

Its expression is the following:

$$K_U = \begin{vmatrix} 0 & n_x & n_y & n_z & 0 & 0 & 0 \\ \frac{(\gamma-1)}{2}q^2 n_x - uU_n & U_n - n_x(\gamma-2)u & n_y u - n_x(\gamma-1)v & n_z u - n_x(\gamma-1)w & n_x(\gamma-1) & -n_x(\gamma-1) & 0 \\ \frac{(\gamma-1)}{2}q^2 n_y - vU_n & n_x v - n_y(\gamma-1)u & U_n - n_y(\gamma-2)v & n_z v - n_y(\gamma-1)w & n_y(\gamma-1) & -n_y(\gamma-1) & 0 \\ \frac{(\gamma-1)}{2}q^2 n_z - wU_n & n_x w - n_z(\gamma-1)u & n_y w - n_z(\gamma-1)v & U_n - n_z(\gamma-2)w & n_z(\gamma-1) & -n_z(\gamma-1) & 0 \\ U_n\frac{(\gamma-1)}{2}q^2 - U_nH & n_xH - (\gamma-1)uU_n & n_yH - (\gamma-1)vU_n & n_zH - (\gamma-1)wU_n & \gamma U_n & -(\gamma-1)U_n & 0 \\ -U_nK & n_xK & n_yK & n_zK & 0 & U_n & 0 \\ -U_n\omega & n_x\omega & n_y\omega & n_z\omega & 0 & 0 & U_n \end{vmatrix}, \tag{A.3}$$

where $q^2 = u^2 + v^2 + w^2$ and, U_n is the flow velocity component along the outward face normal $\underline{n}$, expressed as:

$$U_n = un_x + vn_y + wn_z. \tag{A.4}$$

The calculation of the eigenvalues of K_U is a complex process when these are written in conservative form, due to the dense structure of K_U. It is easier to work when the system of governing equations are written in non–conservative form, as a function of the primitive variables. Primitive variables $\mathbf{V}$ are defined as:

$$\mathbf{V} = \begin{bmatrix} \rho \\ u \\ v \\ w \\ p \\ k \\ \omega \end{bmatrix}. \tag{A.5}$$

Once the governing equations are transformed in nonconservative form, nonconservative Jacobian K_V is expressed with respect to the variables $\mathbf{V}$:

$$K_V = \tilde{A}\frac{\partial \mathbf{V}}{\partial x} + \tilde{B}\frac{\partial \mathbf{V}}{\partial y} + \tilde{C}\frac{\partial \mathbf{V}}{\partial z} = \tilde{A}n_x + \tilde{B}n_y + \tilde{C}n_z, \tag{A.6}$$

where $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$ denote the components of the flux Jacobians of the convective fluxes in nonconservative form K_V. Their expressions are the following:

$$\tilde{A} = \begin{vmatrix} u & \rho & 0 & 0 & 0 & 0 & 0 \\ 0 & u & 0 & 0 & \frac{1}{\rho} & 0 & 0 \\ 0 & 0 & u & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & u & 0 & 0 & 0 \\ 0 & \rho c^2 & 0 & 0 & u & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & u & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & u \end{vmatrix}, \tilde{B} = \begin{vmatrix} v & 0 & \rho & 0 & 0 & 0 & 0 \\ 0 & v & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & v & 0 & \frac{1}{\rho} & 0 & 0 \\ 0 & 0 & 0 & v & 0 & 0 & 0 \\ 0 & 0 & \rho c^2 & 0 & v & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & v & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & v \end{vmatrix}, \tilde{C} = \begin{vmatrix} w & 0 & 0 & \rho & 0 & 0 & 0 \\ 0 & w & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & w & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & w & \frac{1}{\rho} & 0 & 0 \\ 0 & 0 & 0 & \rho c^2 & w & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & w & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & w \end{vmatrix}. \tag{A.7}$$

The symbol c denotes the speed of the sound, and is related to the static temperature through the relation:

$$c = \sqrt{\gamma R T}, \tag{A.8}$$

where γ represents the ratio of specific heats, R is the gas constant and T is the absolute velocity. The expression for flux Jacobian of the convective fluxes in nonconservative form reads:

$$K_V = \begin{vmatrix} U_n & \rho n_x & \rho n_y & \rho n_z & 0 & 0 & 0 \\ 0 & U_n & 0 & 0 & \frac{n_x}{\rho} & 0 & 0 \\ 0 & 0 & U_n & 0 & \frac{n_y}{\rho} & 0 & 0 \\ 0 & 0 & 0 & U_n & \frac{n_z}{\rho} & 0 & 0 \\ 0 & \rho c^2 n_x & \rho c^2 n_y & \rho c^2 n_z & U_n & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & U_n & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & U_n \end{vmatrix}. \tag{A.9}$$

The relations between the conservative and the non–conservative jacobians K_U and K_V can be expressed through a similarity transformation with matrix M. The relation is given by:

$$K_U = M K_V M^{-1} \text{ or } K_V = M^{-1} K_U M. \tag{A.10}$$

M^{-1} is inverse matrix of M. Their expressions are respectively:

$$M = \frac{\partial \mathbf{U}}{\partial \mathbf{V}} = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ u & \rho & 0 & 0 & 0 & 0 & 0 \\ v & 0 & \rho & 0 & 0 & 0 & 0 \\ w & 0 & 0 & \rho & 0 & 0 & 0 \\ q^2/2 + K & \rho u & \rho v & \rho w & \frac{1}{\gamma - 1} & \rho & 0 \\ K & 0 & 0 & 0 & 0 & \rho & 0 \\ \omega & 0 & 0 & 0 & 0 & 0 & \rho \end{vmatrix}, \tag{A.11}$$

$$M^{-1} = \frac{\partial \mathbf{V}}{\partial \mathbf{U}} = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -u/\rho & 1/\rho & 0 & 0 & 0 & 0 & 0 \\ -v/\rho & 0 & 1/\rho & 0 & 0 & 0 & 0 \\ -w/\rho & 0 & 0 & 1/\rho & 0 & 0 & 0 \\ (\gamma-1)q^2/2 & -u(\gamma-1) & -v(\gamma-1) & -w(\gamma-1) & \gamma-1 & -(\gamma-1) & 0 \\ -K/\rho & 0 & 0 & 0 & 0 & 1/\rho & 0 \\ -\omega/\rho & 0 & 0 & 0 & 0 & 0 & 1/\rho \end{vmatrix}. \qquad \text{(A.12)}$$

Equation. (A.10) defines a similarity transformation from the conservative to nonconservative variables (K_U to K_V) and vice versa. Both Jacobians K_U and K_V feature the same eigenvalues due to their similarity, and their eigenvectors are related through the Jacobian matrix of the transformation from the conservative to the non–conservative variables M as shown in the following expression. Inserting Eqn. (A.10) into Eqn. (A.1), following relations are established:

$$\delta \mathbf{\Phi} = |K_U|\,\delta \mathbf{U} = M\,|K_V|\,M^{-1}\delta \mathbf{U} = ML|\Lambda|L^{-1}M^{-1}\delta \mathbf{U} = ML|\Lambda|L^{-1}\delta \mathbf{V}. \qquad \text{(A.13)}$$

The symbol L denotes the matrix of right eigenvectors of K_V, and its inverse L^{-1} are the left eigenvectors of K_V. Comparing the two expressions of numerical dissipation, Eqn. (A.1) and Eqn. (A.13), the following relations can be established:

$$P = ML \qquad \text{(A.14)}$$

and

$$\delta \mathbf{W} = L^{-1}\delta \mathbf{V}. \qquad \text{(A.15)}$$

As the calculation of eigenvalues of nonconservative Jacobian K_V is much simpler than that of conservative Jacobian K_U, the matrix of right eigenvectors P and the variation $\delta \mathbf{W}$ of the characteristic variables appearing in Eqn. (A.1), are determined by the expressions above. Both expressions require the calculation of the matrix of left eigenvalues L^{-1} and right eigenvectors L of K_V.

The Jacobian matrices K_V and K_U have seven eigenvalues:

$$\begin{aligned} &\lambda_1 = \lambda_2 = \lambda_3 = \lambda_6 = \lambda_7 = U_n, \\ &\lambda_4 = U_n + c, \\ &\lambda_5 = U_n - c, \end{aligned} \qquad \text{(A.16)}$$

where:

$$U_n = \begin{bmatrix} k_1 & (k_2 + k_3) & -k_2 n_x/n_y & -k_3 n_x/n_z & -k_1/c^2 & k_4 & k_5 \end{bmatrix}. \qquad \text{(A.17)}$$

The constants $k_i,\ i = 1, 5$ may be selected arbitrarily. Using a typical choice for these constants, the following 5 left eigenvectors is obtained:

$$\begin{aligned} \lambda_1 &= \begin{bmatrix} 1 & 0 & 0 & 0 & -1/c^2 & 0 & 0 \end{bmatrix}, \\ \lambda_2 &= \begin{bmatrix} 0 & \rho n_y & -\rho n_x & 0 & 0 & 0 & 0 \end{bmatrix}, \\ \lambda_3 &= \begin{bmatrix} 0 & -\rho n_z & 0 & \rho n_x & 0 & 0 & 0 \end{bmatrix}, \\ \lambda_6 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \rho & 0 \end{bmatrix}, \\ \lambda_7 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \rho \end{bmatrix}. \end{aligned} \qquad \text{(A.18)}$$

The eigenvalues $U_n + c$ and $U_n - c$ yield respectively the eigenvectors:

$$\begin{aligned} \lambda_4 &= \begin{bmatrix} 0 & \dfrac{\rho n_x}{2c} & \dfrac{\rho n_y}{2c} & \dfrac{\rho n_z}{2c} & \dfrac{1}{2c^2} & 0 & 0 \end{bmatrix}, \\ \lambda_5 &= \begin{bmatrix} 0 & -\dfrac{\rho n_x}{2c} & -\dfrac{\rho n_y}{2c} & -\dfrac{\rho n_z}{2c} & \dfrac{1}{2c^2} & 0 & 0 \end{bmatrix}. \end{aligned} \qquad \text{(A.19)}$$

This leads to the expression of matrix of left eigenvectors:

$$L^{-1} = \begin{vmatrix} 1 & 0 & 0 & 0 & -\frac{1}{c^2} & 0 & 0 \\ 0 & \rho n_y & -\rho n_x & 0 & 0 & 0 & 0 \\ 0 & -\rho n_z & 0 & \rho n_x & 0 & 0 & 0 \\ 0 & \frac{\rho n_x}{2c} & \frac{\rho n_y}{2c} & \frac{\rho n_z}{2c} & \frac{1}{2c^2} & 0 & 0 \\ 0 & -\frac{\rho n_x}{2c} & -\frac{\rho n_y}{2c} & -\frac{\rho n_z}{2c} & \frac{1}{2c^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \rho & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \rho \end{vmatrix}, \tag{A.20}$$

and the expression for its inverse:

$$L = \begin{vmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & \frac{n_y}{\rho} & -\frac{n_z}{\rho} & \frac{n_x c}{\rho} & -\frac{n_x c}{\rho} & 0 & 0 \\ 0 & -\frac{{n_x}^2+{n_z}^2}{\rho n_x} & -\frac{n_y n_z}{\rho n_x} & \frac{n_y c}{\rho} & -\frac{n_y c}{\rho} & 0 & 0 \\ 0 & \frac{n_y n_z}{\rho n_x} & \frac{{n_x}^2+{n_y}^2}{\rho n_x} & \frac{n_z c}{\rho} & -\frac{n_z c}{\rho} & 0 & 0 \\ 0 & 0 & 0 & c^2 & c^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} \end{vmatrix}. \tag{A.21}$$

Using the expression provided by Eqn. (A.14), the matrix P can be obtained:

$$P = ML = \begin{vmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ u & n_y & -n_z & u + cn_x & u - cn_x & 0 & 0 \\ v & -\frac{n_x^2+n_z^2}{n_x} & -\frac{n_y n_z}{n_x} & v + cn_y & v - cn_y & 0 & 0 \\ w & \frac{n_y n_z}{n_x} & \frac{n_x^2+n_y^2}{n_x} & w + cn_z & w - cn_z & 0 & 0 \\ q^2/2 + K & U_t & V_t & H + U_n c & H - U_n c & 1 & 0 \\ K & 0 & 0 & K & K & 1 & 0 \\ \omega & 0 & 0 & \omega & \omega & 0 & 1 \end{vmatrix}. \tag{A.22}$$

where:

$$U_t = un_y - v\frac{n_x^2 + n_z^2}{n_x} + w\frac{n_y n_z}{n_x},$$
$$V_t = -un_z - v\frac{n_y n_z}{n_x} + w\frac{n_x^2 + n_y^2}{n_x}.$$

The expressions of characteristic variables $\delta\mathbf{W}$, which can be computed by means of Eqn. (A.15) are the following:

$$\begin{aligned}
\delta W_1 &= \delta\rho - \frac{1}{c^2}\delta p, \\
\delta W_2 &= \rho n_y \delta u - \rho n_x \delta v && = \rho(n_y \delta u - n_x \delta v), \\
\delta W_3 &= -\rho n_z \delta u + \rho n_x \delta w && = \rho(-n_z \delta u + n_x \delta w), \\
\delta W_4 &= \frac{\rho n_x}{2c}\delta u + \frac{\rho n_y}{2c}\delta v + \frac{\rho n_z}{2c}\delta w + \frac{1}{2c^2}\delta p && = \frac{\delta p}{2c^2} + \frac{\rho \delta U_n}{2c}, \\
\delta W_5 &= -\frac{\rho n_x}{2c}\delta u - \frac{\rho n_y}{2c}\delta v - \frac{\rho n_z}{2c}\delta w + \frac{1}{2c^2}\delta p && = \frac{\delta p}{2c^2} - \frac{\rho \delta U_n}{2c}, \\
\delta W_6 &= \rho \delta K, \\
\delta W_7 &= \rho \delta \omega.
\end{aligned} \tag{A.23}$$

The expression of numerical dissipation given by Eqn. (A.1) may also be rewritten in the following form:

$$\delta\mathbf{\Phi} = P|\Lambda|\delta\mathbf{W} = \sum_{k=1}^{7} |\lambda_k| \delta W_k \mathbf{r}_k, \tag{A.24}$$

where δW_k denotes the k^{th} component of characteristic variable $\delta \mathbf{W}$, and r_k represents the k^{th} right eigenvector of K_U. Therefore, the sought flux differences are the following:

$$\begin{aligned}\delta\mathbf{\Phi} = |\lambda_1|\delta W_1 \begin{vmatrix} 1 \\ u \\ v \\ w \\ \frac{q^2}{2}+K \\ K \\ \omega \end{vmatrix} + |\lambda_2|\delta W_2 \begin{vmatrix} 0 \\ n_y \\ -\frac{n_x^2+n_z^2}{n_x} \\ \frac{n_y n_z}{n_x} \\ U_t \\ 0 \\ 0 \end{vmatrix} + |\lambda_3|\delta W_3 \begin{vmatrix} 0 \\ -n_z \\ -\frac{n_y n_z}{n_x} \\ \frac{n_x^2+n_y^2}{n_x} \\ V_t \\ 0 \\ 0 \end{vmatrix} \\ +|\lambda_4|\delta W_4 \begin{vmatrix} 1 \\ u+cn_x \\ v+cn_y \\ w+cn_z \\ H+U_n c \\ K \\ \omega \end{vmatrix} + |\lambda_5|\delta W_5 \begin{vmatrix} 1 \\ u-cn_x \\ v-cn_y \\ w-cn_z \\ H-U_n c \\ K \\ \omega \end{vmatrix} + |\lambda_6|\delta W_6 \begin{vmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{vmatrix} + |\lambda_7|\delta W_7 \begin{vmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{vmatrix},\end{aligned} \tag{A.25}$$

and hence:

$$\begin{aligned}\delta\mathbf{\Phi} = U_n \left\{ \left(\delta\rho - \frac{\delta p}{c^2}\right) \begin{vmatrix} 1 \\ u \\ v \\ w \\ \frac{q^2}{2}+K \\ K \\ \omega \end{vmatrix} + \rho \begin{vmatrix} 0 \\ du - n_x\delta U_n \\ dv - n_y\delta U_n \\ dw - n_z\delta U_n \\ u\delta u + v\delta v + w\delta w - U_n\delta U_n + \delta K \\ \delta K \\ \delta\omega \end{vmatrix} \right\} \\ +(U_n+c)\left(\frac{\delta p}{2c^2} + \frac{\rho\delta U_n}{2c}\right) \begin{vmatrix} 1 \\ u+cn_x \\ v+cn_y \\ w+cn_z \\ H+U_n c \\ K \\ \omega \end{vmatrix} + (U_n-c)\left(\frac{\delta p}{2c^2} - \frac{\rho\delta U_n}{2c}\right) \begin{vmatrix} 1 \\ u-cn_x \\ v-cn_y \\ w-cn_z \\ H-U_n c \\ K \\ \omega \end{vmatrix}.\end{aligned} \tag{A.26}$$

Numerical implementation of numerical dissipation provided by Eqn. (A.24), can be written in more compact way. This can be done in such way, that a set of intermediate variables are first introduced:

$$\begin{aligned} \alpha_1 &= |\lambda_1|\left(\delta\rho - \frac{\delta p}{c^2}\right), \\ \alpha_2 &= |\lambda_2|\rho = |\lambda_3|\rho = |\lambda_6|\rho = |\lambda_7|\rho, \\ \alpha_3 &= |\lambda_4|\left(\frac{\delta p}{c^2} + \frac{\rho\delta U_n}{c}\right)/2, \\ \alpha_4 &= |\lambda_5|\left(\frac{\delta p}{c^2} - \frac{\rho\delta U_n}{c}\right)/2. \end{aligned} \tag{A.27}$$

Furthermore, using such intermediate variables, as provided by Eqn. (A.27), the components of $\delta\mathbf{\Phi}$ may then be computed in the following compact way:

$$\begin{aligned}
\delta\Phi_1 =&\alpha_1 + \alpha_3 + \alpha_4,\\
\delta\Phi_2 =&\alpha_1 u + \alpha_2(\delta u - n_x\delta U_n) + \alpha_3(u + cn_x) + \alpha_4(u - cn_x),\\
\delta\Phi_3 =&\alpha_1 v + \alpha_2(\delta v - n_y\delta U_n) + \alpha_3(v + cn_y) + \alpha_4(v - cn_y),\\
\delta\Phi_4 =&\alpha_1 w + \alpha_2(\delta w - n_z\delta U_n) + \alpha_3(w + cn_z) + \alpha_4(w - cn_z),\\
\delta\Phi_5 =&\alpha_1(\frac{q^2}{2} + K) + \alpha_2(u\delta u + v\delta v + w\delta w - U_n\delta U_n + \delta K)\\
&+ \alpha_3(H + cU_n) + \alpha_4(H - cU_n),\\
\delta\Phi_6 =&\alpha_1 K + \alpha_3 K + \alpha_4 K + \alpha_2\delta K,\\
\delta\Phi_7 =&\alpha_1\omega + \alpha_3\omega + \alpha_4\omega + \alpha_2\delta\omega.
\end{aligned} \tag{A.28}$$

Appendix B

Convergence acceleration techniques

The convergence of the iterative solution process of the governing equations may be greatly enhanced by using various methodologies to accelerate the solution to steady–state. The acceleration techniques are applicable to the solution of the steady equations as well as to the solution of the TD and HB equations. This is because the calculation of the flow solution at each step of the TD problem is obtained by using the same approach adopted for the solution of the steady equations. Similarly, the discrete form of the HB equations can be viewed as a system of $2N_H + 1$ steady problems, and the overall system is solved with the same numerical method used to solve a single steady problem. One of the acceleration methods used in this work, is the local time stepping. It is based purely on the modification of the system of ordinary differential equations. The other two acceleration methods used herein, multigrid and implicit residual smoothing, are improvements of the solution process [66, 68, 89].

B.0.1 Local time stepping

Local time stepping is a technique, where the maximum allowable pseudo time step τ for each control volume is calculated. This time step is then used in the integration process of the discretized governing equations, described in previous sections. In comparison to global time stepping, where the time step is the same for all control volumes, local time stepping approach greatly accelerates the convergence to steady state.

Maximum time step needs to be determined in such way that the numerical scheme remains stable. For the numerical scheme to be stable, it must fulfil the Courant–Friedrichs–Lewy (CFL) condition [157], which states that the numerical domain of dependence of any point in space and time must also include the domain of dependence of the partial differential equation. For the basic one dimensional explicit scheme it means that the time step should be equal or smaller than the time required to transport the information across the stencil of the spatial discretization scheme.

The maximum time step for non–linear governing equations in multiple dimensions can be estimated only approximately. There are more ways to calculate the time step τ. Herein we adopt the approach presented in [66, 113, 158]. The expression to determine the maximum allowable pseudo time step is the following:

$$\Delta\tau = CFL\frac{V}{\Lambda_i^c + \Lambda_j^c + \Lambda_k^c + C(\Lambda_i^d + \Lambda_j^d + \Lambda_k^d)}, \tag{B.1}$$

where matrix V represents the cell volumes, Λ_i^c, Λ_j^c and Λ_k^c are the spectral radii of the convective flux Jacobians, their expressions are the following:

$$\begin{aligned}\Lambda_i^c =& (|\underline{\mathrm{u}} \cdot \underline{\mathrm{n}}_i| + c)\Delta S_i,\\ \Lambda_j^c =& (|\underline{\mathrm{u}} \cdot \underline{\mathrm{n}}_j| + c)\Delta S_j,\\ \Lambda_k^c =& (|\underline{\mathrm{u}} \cdot \underline{\mathrm{n}}_k| + c)\Delta S_k.\end{aligned} \tag{B.2}$$

The normal vectors $\underline{\mathrm{n}}_i$, $\underline{\mathrm{n}}_j$ and $\underline{\mathrm{n}}_k$ and face areas ΔS_i, ΔS_j and ΔS_k are obtained by averaging the values from the two opposite sides of the control volume for each direction. Furthermore, Λ_i^d, Λ_j^d and Λ_k^d are the spectral radii of the diffusive flux Jacobians, and may be written as:

$$\begin{aligned}\Lambda_i^d =& \frac{\gamma}{\rho}\left(\frac{\mu}{Pr} + \frac{\mu_T}{\mathrm{Pr}_T}\right)\frac{\Delta S_i^2}{V},\\ \Lambda_j^d =& \frac{\gamma}{\rho}\left(\frac{\mu}{Pr} + \frac{\mu_T}{\mathrm{Pr}_T}\right)\frac{\Delta S_j^2}{V},\\ \Lambda_k^d =& \frac{\gamma}{\rho}\left(\frac{\mu}{Pr} + \frac{\mu_T}{\mathrm{Pr}_T}\right)\frac{\Delta S_k^2}{V}.\end{aligned} \tag{B.3}$$

μ and μ_T denote the laminar and the turbulent dynamic viscosity, respectively. Moreover, Pr and Pr_T are the laminar and the turbulent Prandtl numbers. The constant C, which multiplies the viscous spectral radii was set in between $1 \leq C \leq 4$, depending on the test case.

B.0.2 Implicit residual smoothing

Another very useful convergence acceleration technique is called implicit residual smoothing, where the local stability range and robustness of time–stepping scheme may be significantly extended. The main purpose of this technique is to give an explicit scheme the implicit character. General idea is to replace the residual at a certain point in the flow field by a weighted average of the residuals at the neighbouring points. This enables a substantial increase in the maximum allowable CFL number. Its further purpose is also a better damping of the residual high–frequency error components, which has a particular importance when it is used alongside multigrid. Implicit residual smoothing was first implemented on Runge–Kutta stepping scheme in [159], using the constant–coefficient approach. The approach was further investigated by many studies [158, 160, 161, 113] mostly using variable coefficients for implicit residual smoothing. For this work, we have used the variable coefficient approach proposed by [160].

The expression for the residual smoothing in factorised form is the following:

$$(1 - \beta_i\delta_i)(1 - \beta_j\delta_j)(1 - \beta_k\delta_k)\tilde{\mathbf{R}}_\phi = \mathbf{R}_\phi, \tag{B.4}$$

where the symbol δ represents a second order central difference operators in the three computational directions i, j and k. $\mathbf{R}_\phi$ is the original residual, and $\tilde{\mathbf{R}}_\phi$ is the residual after a sequence of smoothing in the three directions. The symbols β_i, β_j and β_k are locally varying smoothing coefficients, and can be calculated by the following expressions:

$$\begin{aligned}
\beta_i =& MAX\left\{0, \frac{1}{4}\left[\left(\frac{CFL\ \Lambda_i^c}{CFL^*\ (\Lambda_i^c+\Lambda_j^c+\Lambda_k^c)}\Phi_i\right)^2-1\right]\right\},\\
\beta_j =& MAX\left\{0, \frac{1}{4}\left[\left(\frac{CFL\ \Lambda_j^c}{CFL^*\ (\Lambda_i^c+\Lambda_j^c+\Lambda_k^c)}\Phi_j\right)^2-1\right]\right\},\\
\beta_k =& MAX\left\{0, \frac{1}{4}\left[\left(\frac{CFL\ \Lambda_k^c}{CFL^*\ (\Lambda_i^c+\Lambda_j^c+\Lambda_k^c)}\Phi_k\right)^2-1\right]\right\}.
\end{aligned} \tag{B.5}$$

CFL and $CFL*$ are the Courant–Friedrichs–Lewy numbers of smoothed and unsmoothed scheme respectively. The maximum value of the ratio of smoothed and unsmoothed CFL numbers depends on the type of spatial discretization scheme and value of the smoothing coefficient. In practice, the value $CFL/CFL^* \approx 2$ is used. Λ_i^c, Λ_j^c and Λ_k^c are convective spectral radii defined by Eqn. (B.3). And the coefficients Φ_i, Φ_j and Φ_k have the following definition:

$$\begin{aligned}
\Phi_i =& 1+\left(\frac{\Lambda_j^c}{\Lambda_i^c}\right)^\sigma+\left(\frac{\Lambda_k^c}{\Lambda_i^c}\right)^\sigma,\\
\Phi_j =& 1+\left(\frac{\Lambda_i^c}{\Lambda_j^c}\right)^\sigma+\left(\frac{\Lambda_k^c}{\Lambda_j^c}\right)^\sigma,\\
\Phi_k =& 1+\left(\frac{\Lambda_i^c}{\Lambda_k^c}\right)^\sigma+\left(\frac{\Lambda_j^c}{\Lambda_k^c}\right)^\sigma.
\end{aligned} \tag{B.6}$$

The exponent σ is usually defined in the region $0 < \sigma \leq 1$, for this work the value $\sigma = 0.4$ was used. The expression, given by Eqn. (B.4), is a convenient extension from one dimensional to three dimensional case, since only the inversion of tridiagonal matrix is required for the solution of smoothed residuals $\widetilde{R}_\phi$. The system of equations is solved using the efficient Thomas algorithm for the inversion of tridiagonal matrices, a simplified form of Gaussian elimination.

B.0.3 Multigrid

The multigrid method is another acceleration method, which has been used in this work. Generally speaking, the multigrid method systematically uses a sequence of computational grids, as shown in Fig B.1, to accelerate the convergence of the iterative schemes. More precisely, a faster convergence rate to steady–state on a fine grid is achieved by approximating the fine grid problem on successively coarser grids in the sequence. Coarser grids permit larger time steps, which is a consequence of having larger control volume, together with a reduced numerical effort. Moreover, multigrid also helps to dampen the low frequency components of the solution error, which slows down the convergence of single–grid schemes. As the low–frequency components on the finest grid become high–frequency components on the coarser grids, the error may be much more rapidly reduced when using multigrid, making the multigrid method a successful convergence acceleration method. The multigrid technique was initially developed for elliptic partial differential equations [162, 163, 164], and was further applied to an Euler equations [90, 159], and later to a NS equations [158, 165, 166].

When the generation of coarser grid levels takes place on geometric level, as depicted in Fig B.1, the method is called geometric multigrid or full approximation storage (FAS) scheme. An alternative to this method is algebraic multigrid (AMG) [167], where the

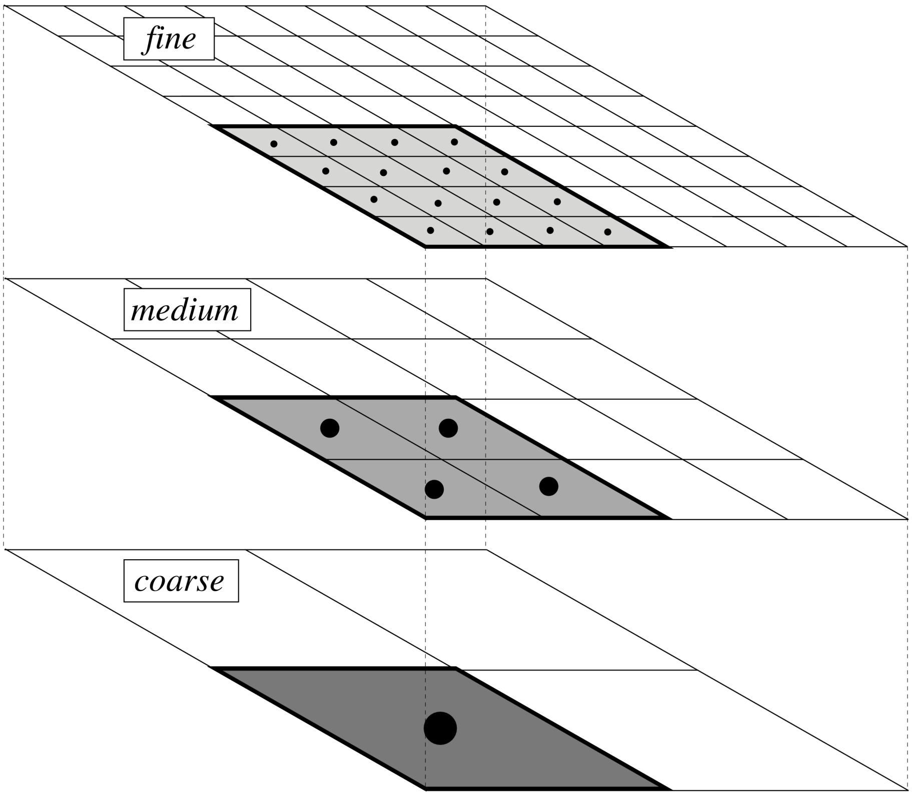


Figure B.1: Two dimensional representation of three grid levels, fine, medium and coarse.

grid topology on coarser grid levels is not required. AMG method was developed specifically for implicit numerical schemes, and it uses coarsening matrix in order to reduce the dimension of the implicit operator. Such reduced system then represents a coarse multigrid level, where a new solution is obtained, and then it is used for the correction of the fine grid level. The advantage of such technique is that it does not require the construction and storage of coarse grid topology.

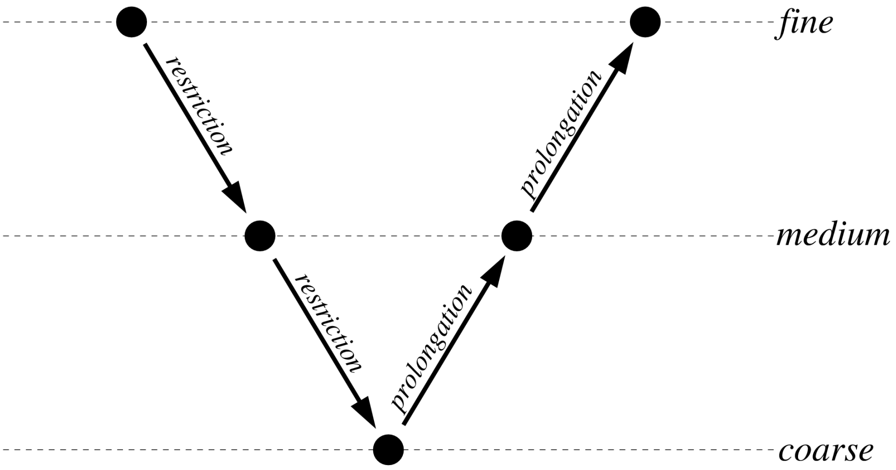


Figure B.2: FAS multigrid V–cycle.

Before applying the FAS scheme, the coarse grids have to be generated. For structured grids, the coarse grids are easily obtained by deleting every second grid line in the respective coordinate direction. Two typical FAS multigrid cycles exist, first is called V–cycle, and second is called W–cycle. For this work, we have used V–cycle, depicted in Fig. B.2. Figure B.2 represents V–cycle with the three multigrid levels, fine, medium and coarse. The FAS V–cycle described below, refers to that figure. Note, there are no other than geometrical restrictions (e.g. number of cells must be even) for the number of multigrid levels one may use in multigrid cycle. However, in practice not more than three multigrid levels are usually used. Let us first rewrite Eqn. (3.70) in the form, where we denote fine multigrid level with the symbol h:

$$\frac{\partial \mathbf{Q}_h}{\partial \tau} = -\frac{1}{V_h}\mathbf{R}_h(\mathbf{Q}_h). \tag{B.7}$$

The FAS V–cycle starts by applying one or more sweeps of Runge–Kutta integration, given by Eqn. (3.74). This is done at a finest grid level, starting from a known solution $\mathbf{Q}_h^l$, to obtain a new fine grid level solution $\mathbf{Q}_h^{l+1}$. The new residual $\mathbf{R}_h^{l+1}$ is also evaluated using the solution $\mathbf{Q}_h^{l+1}$. The new solution and residual are then transferred to medium grid level using the restriction operator. The restricted solution for medium grid may be written as:

$$\mathbf{Q}_{2h}^{(0)} = I_h^{2h}\mathbf{Q}_h^{l+1}, \tag{B.8}$$

where the symbol $2h$ denotes the medium grid level, and I_h^{2h} is the restriction operator when passing from fine to medium grid. To transfer the residual $\mathbf{R}_h^{l+1}$ to medium grid level, we must instead use the conservative restriction operator. For this reason a source term called multigrid forcing function must be introduced. Multigrid forcing function represents the difference between the transferred fine grid residual $\mathbf{R}_h^{l+1}$ and the residual computed on the medium grid using the initial solution $\mathbf{R}_{2h}^{(0)}$, and may be written as:

$$(\mathbf{f}_{MG})_{2h} = I_h^{2h}\mathbf{R}_h^{l+1} - \mathbf{R}_{2h}^{(0)}. \tag{B.9}$$

The solution on medium grid level is smoothed in the same way as on the fine grid level, therefore, the new medium grid level solution $\mathbf{Q}_{2h}^{l+1}$ is obtained. For the evaluation of the residual on medium grid, the multigrid forcing function is added to the residual of medium grid:

$$\mathbf{R}_{2hf}^{l+1} = \mathbf{R}_{2h}^{l+1} + (\mathbf{f}_{MG})_{2h}. \tag{B.10}$$

After one or more iterations on medium grid level are done, exactly the same procedure to restrict the solution on coarse grid level as previously is used. Let us denote the coarse grid level with the symbol $4h$. Hence, the restricted solution for coarse grid reads:

$$\mathbf{Q}_{4h}^{(0)} = I_{2h}^{4h}\mathbf{Q}_{2h}^{l+1}, \tag{B.11}$$

where the symbol I_{2h}^{4h} denotes the restriction operator when passing from medium to coarse grid level. The multigrid forcing function on the coarse grid is instead formed from the restricted residual of medium grid $\mathbf{R}_{2hf}^{l+1}$, and may be written as:

$$(\mathbf{f}_{MG})_{4h} = I_{2h}^{4h}[\mathbf{R}_{2h}^{l+1} + (\mathbf{f}_{MG})_{2h}] - \mathbf{R}_{4h}^{(0)}. \tag{B.12}$$

In this way the accuracy of the solution on coarse grid is controlled by the fine grid residual. It is also important to point out that the multigrid forcing function is computed only after the restriction operator is applied, and only for the first iteration of the first Runge–Kutta stage. Furthermore, as the spatial discretization of numerical scheme on coarse grids does not influence the accuracy of the fine grid level, first order scheme may be selected for coarser grid level. This leads to increased robustness of the numerical scheme, better damping properties, and lower computational cost in comparison to higher order schemes.

After one or several iterations are carried out on the coarse grid level, the coarse grid correction with respect to initial solution must be computed:

$$\Delta\mathbf{Q}_{4h} = \mathbf{Q}_{4h}^{l+1} - \mathbf{Q}_{4h}^{0}. \tag{B.13}$$

Moreover, the coarse–grid correction is interpolated to the medium grid level, to improve the solution on that level. The following expression may be written:

$$\mathbf{Q}_{2h}^{+} = \mathbf{Q}_{2h}^{l+1} + I_{4h}^{2h}\Delta\mathbf{Q}_{4h}, \tag{B.14}$$

where the term I_{4h}^{2h} denotes the prolongation operator from coarse to medium grid level. After the solution on medium grid has been obtained, one or several iterations are performed on the medium grid level, and the medium grid correction with respect to initial solution is computed:

$$\Delta\mathbf{Q}_{2h} = \mathbf{Q}_{2h}^{l+1} - \mathbf{Q}_{2h}^{0}. \tag{B.15}$$

Finally, the medium–grid correction is interpolated to the fine grid level, and the new solution on the fine grid reads:

$$\mathbf{Q}_{h}^{+} = \mathbf{Q}_{h}^{l+1} + I_{2h}^{h}\Delta\mathbf{Q}_{2h}, \tag{B.16}$$

where I_{2h}^{h} represents the prolongation operator from medium to fine grid level. The optimum number of iterations before the restriction and after the prolongation depends on the time–stepping scheme type and the physics of the problem we are trying to solve.

Restriction operator

The solution from fine to coarser grid level is restricted by a volume weighted interpolation, which enhances the robustness of the strongly coupled multigrid iteration. Equation (B.8) may be rewritten as:

$$\begin{aligned}
(\mathbf{Q}_{2h}^{(0)})_{i,j,k} = \frac{1}{(V_{2h})_{i,j,k}} \Big[& (\mathbf{Q}_{h}^{l+1})_{i,j,k}(V_h)_{i,j,k} \\
& + (\mathbf{Q}_{h}^{l+1})_{i+1,j,k}(V_h)_{i+1,j,k} \\
& + (\mathbf{Q}_{h}^{l+1})_{i,j+1,k}(V_h)_{i,j+1,k} \\
& + (\mathbf{Q}_{h}^{l+1})_{i+1,j+1,k}(V_h)_{i+1,j+1,k} \\
& + (\mathbf{Q}_{h}^{l+1})_{i,j,k+1}(V_h)_{i,j,k+1} \\
& + (\mathbf{Q}_{h}^{l+1})_{i+1,j,k+1}(V_h)_{i+1,j,k+1} \\
& + (\mathbf{Q}_{h}^{l+1})_{i,j+1,k+1}(V_h)_{i,j+1,k+1} \\
& +(\mathbf{Q}_{h}^{l+1})_{i+1,j+1,k+1}(V_h)_{i+1,j+1,k+1} \Big].
\end{aligned} \tag{B.17}$$

To restrict the residuals, the restriction operator is usually defined as a sum of the residuals from all cells which form one coarse–grid level control volume. However, we have used high order restriction operator based on weights, to increase the stability of multigrid scheme. The restriction of the residual term of Eqn. (B.9) may be rewritten as:

$$\begin{aligned}
(I_h^{2h}\mathbf{R}_h^{l+1})_{i,j,k} = \frac{1}{64} \Big[& 27(\mathbf{R}_h^{l+1})_{i,j,k} + 9(\mathbf{R}_h^{l+1})_{i+1,j,k} \\
& + 9(\mathbf{R}_h^{l+1})_{i,j+1,k} + 9(\mathbf{R}_h^{l+1})_{i,j,k+1} \\
& + 3(\mathbf{R}_h^{l+1})_{i+1,j+1,k} + 3(\mathbf{R}_h^{l+1})_{i+1,j,k+1} \\
& +3(\mathbf{R}_h^{l+1})_{i,j+1,k+1} + (\mathbf{R}_h^{l+1})_{i+1,j+1,k+1} \Big].
\end{aligned} \tag{B.18}$$

Prolongation operator

The prolongation of the medium–grid correction in Eqn. (B.16) is solved in two steps. First $\Delta\mathbf{Q}_{2h}$ is interpolated to the grid nodes, and therefore, the nodal values are averaged to obtain the value in the center of the fine–grid cell. The following relationship may be defined:

$$\begin{aligned}(I_{2h}^{h}\Delta\mathbf{Q}_{2h})_{i,j,k} = \frac{1}{64}\,&[27(\Delta\mathbf{Q}_{2h})_{i,j,k} + 9(\Delta\mathbf{Q}_{2h})_{i+1,j,k} \\ &+ 9(\Delta\mathbf{Q}_{2h})_{i,j+1,k} + 9(\Delta\mathbf{Q}_{2h})_{i,j,k+1} \\ &+ 3(\Delta\mathbf{Q}_{2h})_{i+1,j+1,k} + 3(\Delta\mathbf{Q}_{2h})_{i+1,j,k+1} \\ &+3(\Delta\mathbf{Q}_{2h})_{i,j+1,k+1} + (\Delta\mathbf{Q}_{2h})_{i+1,j+1,k+1}]\,.\end{aligned} \tag{B.19}$$

Appendix C

NREL Phase VI

Figures in this section compare the skin friction lines on the suction side of the NREL Phase VI blade for thee different CFD simulations, the medium and coarse grid COSA simulations and the coarse grid NUMECA simulation. Six wind velocities, corresponding to the analyses of section 5.5, have been considered: $7 m/s$, $10 m/s$, $13 m/s$, $15 m/s$, $20\, m/s$ and $25\, m/s$. Detailed description of these operating conditions is provided in Table 5.4. Looking at Figs. C.1- C.6, it can be noticed that the highest differences in the pattern of the skin friction lines among the three simulations occur at the lowest considered wind velocities. At $7\, m/s$ (Fig. C.1), the separation region of the COSA coarse grid is much larger than for the COSA medium. Also the NUMECA coarse grid predicts slightly more separation than the COSA medium, but less than the COSA coarse. The same trend is also observed for the velocities $10 m/s$ (Fig. C.2) and $13 m/s$ (Fig. C.3). For the rest of the operating conditions, some visual differences may still be observed looking at the three CFD simulations, however, these differences appears to be smaller.

The observed differences in the pattern of the skin friction lines among the COSA coarse and NUMECA coarse grid simulations, depicted in Figs. C.1- C.6 could be due to the usage of different numerical schemes in the two codes. Another reason could be the usage of different low–speed preconditioners in the two codes. COSA has used Merkle's low–speed preconditioner, whereas NUMECA has used Hakimi's preconditioner. For cases, where the flow is heavily stalled, the differences between COSA and NUMECA could also be due to the small differences in the implementation of the $k - \omega$ SST model.

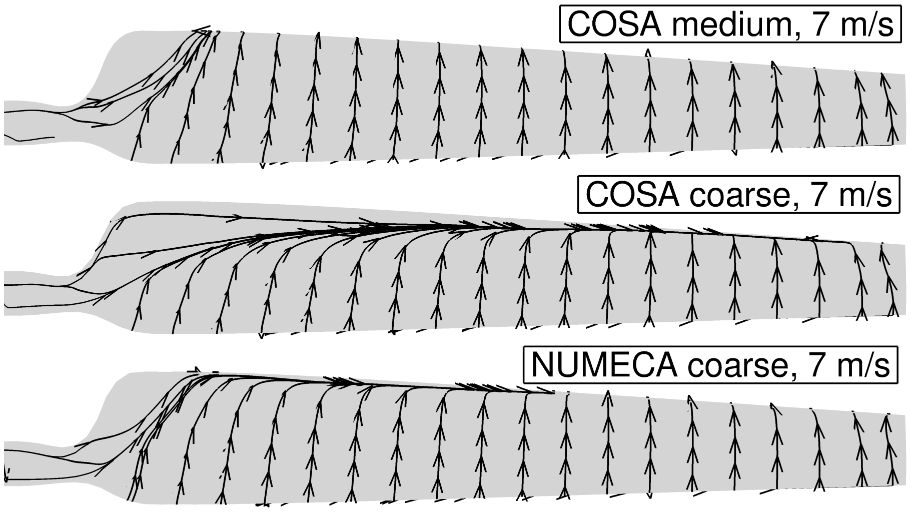


Figure C.1: Skin friction lines on the suction side of the NREL Phase VI blade, computed with the COSA medium grid, COSA coarse grid and NUMECA coarse grid for the wind velocity 7 m/s.

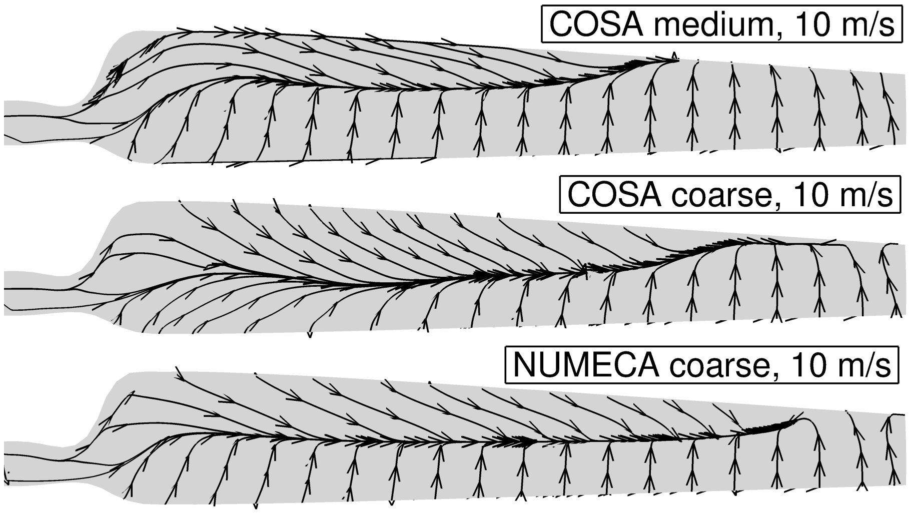


Figure C.2: Skin friction lines on the suction side of the NREL Phase VI blade, computed with the COSA medium grid, COSA coarse grid and NUMECA coarse grid for the wind velocity 10 m/s.

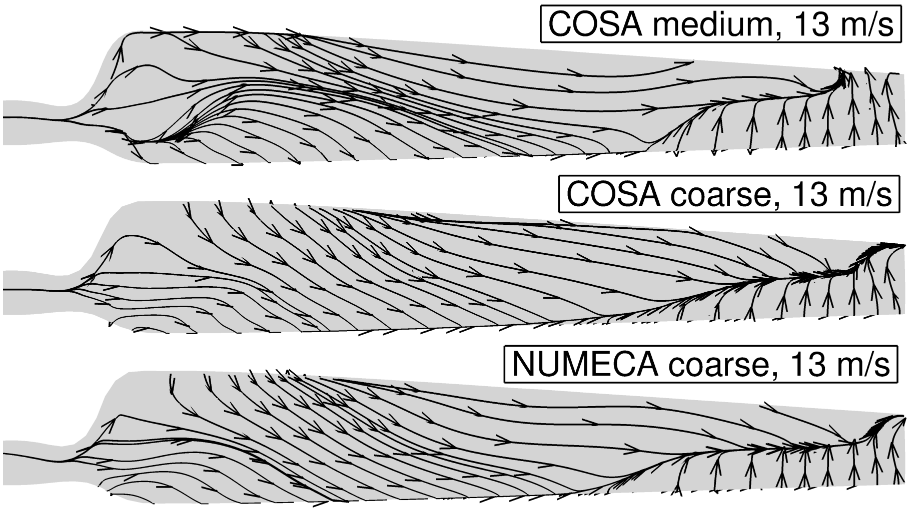


Figure C.3: Skin friction lines on the suction side of the NREL Phase VI blade, computed with the COSA medium grid, COSA coarse grid and NUMECA coarse grid for the wind velocity 13 m/s.

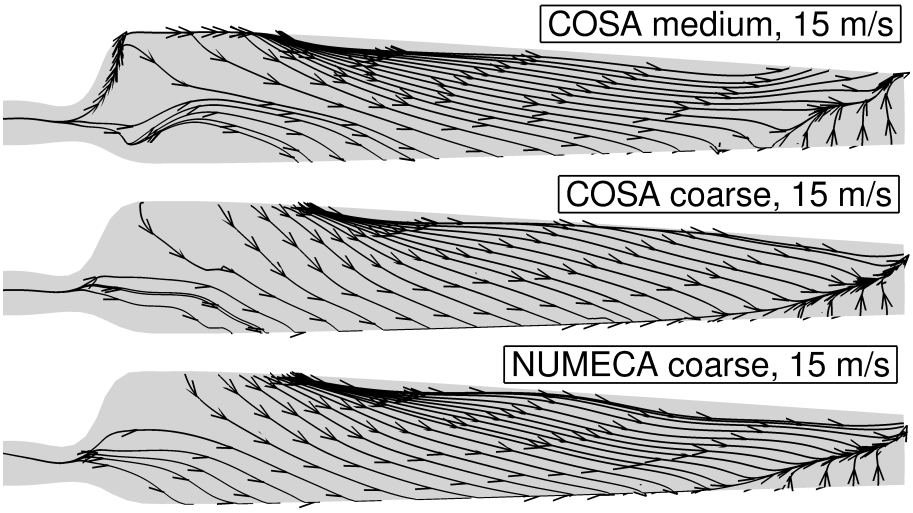


Figure C.4: Skin friction lines on the suction side of the NREL Phase VI blade, computed with the COSA medium grid, COSA coarse grid and NUMECA coarse grid for the wind velocity 15 m/s.

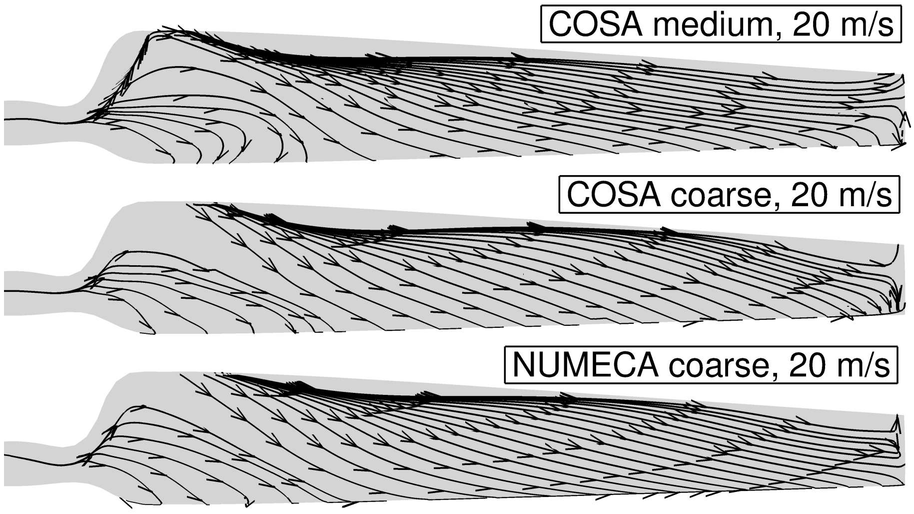


Figure C.5: Skin friction lines on the suction side of the NREL Phase VI blade, computed with the COSA medium grid, COSA coarse grid and NUMECA coarse grid for the wind velocity 20 m/s.

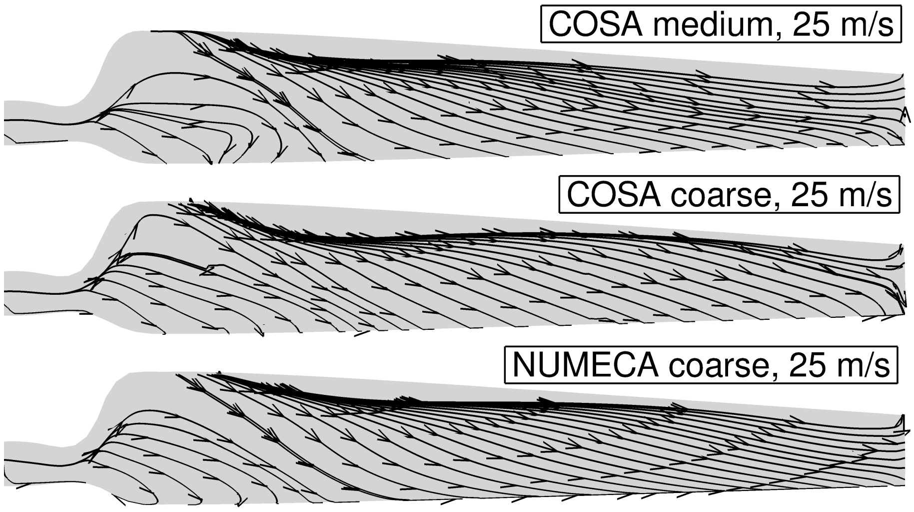


Figure C.6: Skin friction lines on the suction side of the NREL Phase VI blade, computed with the COSA medium grid, COSA coarse grid and NUMECA coarse grid for the wind velocity 25 m/s.