



University
of Glasgow

Piskopakis, Andreas (2014) Time-domain and harmonic balance turbulent Navier-Stokes analysis of oscillating foil aerodynamics. PhD thesis.

<http://theses.gla.ac.uk/5604/>

Copyright and moral rights for this thesis are retained by the author

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the Author

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the Author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.



University
of Glasgow

Time-Domain and Harmonic Balance Turbulent Navier-Stokes Analysis of Oscillating Foil Aerodynamics

By

Andreas Piskopakis

Thesis submitted for the degree of Doctor of Philosophy
at the University of Glasgow

School of Engineering
University of Glasgow
James Watt Building South
University Avenue, G12 8QQ, Glasgow, UK

**Submitted
May 2014**

Abstract

The underlying thread of the research work presented in this thesis is the development of a robust, accurate and computationally efficient general-purpose Reynolds-Averaged Navier-Stokes code for the analysis of complex turbulent flow unsteady aerodynamics, ranging from low-speed applications such as hydrokinetic and wind turbine flows to high-speed applications such as vibrating transonic wings. The main novel algorithmic contribution of this work is the successful development of a fully-coupled multigrid solution method of the Reynolds-Averaged Navier-Stokes equations and the two-equation shear stress transport turbulence model of Menter. The new approach, which also includes the implementation of a high-order restriction operator and an effective limiter of the prolonged corrections, is implemented and successfully demonstrated in the existing steady, time-domain and harmonic balance solvers of a compressible Navier-Stokes research code. The harmonic balance solution of the Navier-Stokes equations is a fairly new technology which can substantially reduce the run-time required to compute nonlinear periodic flow fields with respect to the conventional time-domain approach. The thesis also features the investigation of one modelling and one numerical aspect often overlooked or not comprehensively analysed in turbulent computational fluid dynamics simulations of the type discussed in the thesis. The modelling aspect is the sensitivity of the turbulent flow solution to the, to a certain extent, arbitrary value of the scaling factor appearing in the solid wall boundary condition of the second turbulent variable of the Shear Stress Transport turbulence model. The results reported herein highlight that the solution variability associated with the typical choices of such a scaling factor can be similar or higher than the solution variability caused by the choices of different turbulence models. The numerical aspect is the sensitivity of the turbulent flow solution to the order of the discretisation of the turbulence model equations. The results reported herein highlight that the existence of significant solution differences between first and second order space-discretisation of the turbulence equations vary with the flow regime (e.g. fully subsonic or transonic), operating conditions that may or may not result in flow separation (e.g. angle of attack), and also the grid refinement.

The newly developed turbulent flow capabilities are validated by considering a wide range of test cases with flow regime varying from low-speed subsonic to transonic. The solutions of the research code are compared with experimental data, theoretical solutions and also numerical solutions obtained with a state-of-the-art time-domain commercial

code. The main computational results of this research regard a low-speed renewable energy application and an aeronautical engineering application. The former application is a thorough comparative analysis of a hydrokinetic turbine working in a low-speed laminar and a high-Reynolds number turbulent regime. The time-domain results obtained with the newly developed turbulent code are used to analyse and discuss in great detail the unsteady aerodynamic phenomena occurring in both regimes. The main motivation for analysing this problem is both to highlight the predictive capabilities and the numerical robustness of the developed turbulent time-domain flow solver for complex realistic problems, and to shed more light on the complex physics of this emerging renewable energy device. The latter application is the time-domain and harmonic balance turbulent flow analysis of a transonic wing section animated by pitching motion. The main motivation of these analyses is to assess the computational benefits achievable by using the harmonic balance solution of the Reynolds-Averaged Navier-Stokes and Shear Stress Transport equations rather than the conventional time-domain solution, and also to further demonstrate the predictive capabilities of the developed Computational Fluid Dynamics system. To this aim, the numerical solutions of this research code are compared to both available experimental data, and the time-domain results computed by a state-of-the-art commercial package regularly used by the industry and the Academia worldwide.

Keywords: compressible Reynolds-Average Navier-Stokes equations, shear stress transport turbulence model, harmonic balance, finite volume discretisation, multigrid, hydrokinetic turbines, transonic wings.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr M. Sergio Campobasso for his support and guidance during these amazing three and a half years of my Ph.D. He is a remarkable person with amazing knowledge of the field. He has the ability to transmit his passion and meticulousness to others and he was undoubtedly a great example for me and my colleagues. In our meetings he always surprised me with his brilliant advice. I would also like to acknowledge Dr Edmondo Minisci for his assistance and his availability to help me solve the issues I faced when I was trying to learn the software NUMECA during the first year of my studies. I would also like to acknowledge my examiners Dr Marco Vezza, Dr Tomas Scanlon (University of Strathclyde) and convener Dr Eric Gillies, for reading my thesis and providing advice which improved the quality of my work.

My warmest and deepest sense of gratitude goes to my fiancée Maria Tsakiri, who gave me support, encouragement, endless love and understanding during the difficult periods of my PhD. I would like also to acknowledge my parents who always support my decisions since I was born. Last but not least, I would like to thank my lovely sister who spent a great deal of time on my thesis cover.

Declarations

Part of the work presented in this thesis has been published in the following articles:

M. S. Campobasso, A. Piskopakis, M. Yan, “Analysis of an Oscillating Wing in a Power-Extraction Regime Based on the Compressible Reynolds-Averaged Navier-Stokes Equations and the $K - \omega$ SST Turbulence Model,” in *ASME Turbo Expo*, San Antonio, Texas, USA, 2013. [DOI:10.1115/GT2013-94531](https://doi.org/10.1115/GT2013-94531)

M. S. Campobasso, A. Piskopakis, J. Drofelnik, A. Jackson, “Turbulent Navier–Stokes Analysis of an Oscillating Wing in a Power-Extraction Regime using the Shear Stress Transport Turbulent Model,” *Computers & Fluids*, vol. 88, pp. 136-155, 2013. [DOI: 10.1016/j.compfluid.2013.08.016](https://doi.org/10.1016/j.compfluid.2013.08.016)

M. S. Campobasso, M. Yan, J. Drofelnik, A. Piskopakis, M. Caboni, “Compressible Reynolds-Average Navier-Stokes Analysis of Wind Turbine Turbulent Flows using a Fully-Coupled Low-Speed Preconditioned Multigrid Solver,” in *ASME Turbo Expo*, Dusseldorf, Germany, 2014.

I declare that this thesis is my own work and it has not been submitted for any other degree at the University of Glasgow or any other institution.

Andreas Piskopakis
Glasgow, May 2014

Contents

<i>Abstract</i>	2
<i>Acknowledgements</i>	4
<i>Declarations</i>	5
<i>Contents</i>	6
<i>List of Figures</i>	9
<i>Nomenclature</i>	12
1 INTRODUCTION	15
1.1 Computational fluid dynamics	15
1.2 Oscillating wings.....	19
1.2.1 Hydrokinetic turbines.....	19
1.2.2 Wing aerodynamics.....	22
1.3 COSA solver	23
1.4 Objectives, novelty and overview of the thesis.....	25
2 REYNOLDS AVERAGED NAVIER-STOKES EQUATIONS AND TURBULENCE CLOSURE	30
2.1 Integral form of the Navier-Stokes equations	31
2.1.1 Arbitrary Lagrangian Eulerian formulation	35
2.2 Reynolds-Favre averaging	36
2.3 Boussinesq approximation	37
2.4 Wilcox's $K - \omega$ model	38
2.5 Menter's shear stress transport model.....	41
2.6 Boundary conditions	44
2.6.1 Far field	44
2.6.2 Solid wall	45
2.6.3 Comparison of Wilcox and Menter wall boundary conditions	47
2.7 System of URANS and SST equations	49
2.7.1 Divergence form.....	50
2.7.2 Integral form	51
3 HARMONIC BALANCE FORMULATION OF THE GOVERNING EQUATIONS	53
3.1 Harmonic balance.....	54
3.2 Harmonic balance integration: ODE example	56

3.2.1	Classical harmonic balance	57
3.2.2	High-dimensional harmonic balance	61
3.3	HDHB formulation of RANS and SST equations	65
4	SPACE DISCRETISATION.....	68
4.1	Meshes	69
4.2	Discretisation approaches and the finite volume method	72
4.2.1	The finite volume method	74
4.3	Convective fluxes.....	76
4.4	Entropy fix	81
4.5	Diffusive fluxes.....	83
5	NUMERICAL INTEGRATION.....	86
5.1	Explicit and implicit integration.....	87
5.2	Runge-Kutta time-marching	91
5.2.1	Turbulent steady problems.....	93
5.2.2	Turbulent time-domain problems.....	100
5.2.3	Turbulent harmonic balance problems.....	105
5.3	Local time-stepping.....	108
5.4	Implicit residual smoothing.....	109
5.5	CFL ramping	111
6	MULTIGRID ACCELERATION.....	113
6.1	Introduction.....	114
6.2	The multigrid cycle	115
6.3	Restriction operator	118
6.4	Prolongation operator.....	124
6.5	Multigrid for nonlinear problems.....	126
6.6	Technicalities on the application of multigrid to RANS and SST equations.....	130
7	VALIDATION.....	133
7.1	Steady turbulent flat plate	134
7.2	Steady NACA4412	139
7.3	Steady RAE2822.....	149
8	RESULTS	157
8.1	Hydrokinetic turbines.....	158
8.1.1	Fundamentals of hydrokinetic turbines.....	158
8.1.2	Physical and numerical problem set-up.....	161
8.1.3	Aerodynamic analysis	167

8.2 Oscillating transonic wings	177
8.2.1 Fundamentals of oscillating wings	178
8.2.2 Physical and numerical problem set-up.....	179
8.2.3 Time-domain analysis	183
8.2.4 Frequency-domain analysis	190
8.2.5 Computational efficiency	199
8.3 Oscillating subsonic wings.....	200
8.3.1 Physical and numerical problem set-up.....	200
8.3.2 Time-domain and Frequency-domain analysis	204
8.3.3 Computational efficiency	213
9 CONCLUSIONS	215
9.1 Summary and concluding remarks.....	215
9.1.1 Algorithmic conclusions	215
9.1.2 Fluid dynamics conclusions.....	217
9.2 Future work	219
<i>Appendices</i>	220
A) Derivation of Reynolds-Average Navier-Stokes equations	220
B) Closure approximations	224
C) Correlations	226
D) Non-dimensionalisation	226
E) Compact divergence form of the URANS and SST equations	229
F) Quasi-linear form of the URANS and SST equations	231
G) Harmonic balance solution of an ODE example.....	231
H) Numerical dissipation.....	234
I) Diffusive fluxes	242
J) Hydrokinetic turbine: turbulent and laminar Cf comparison	248
<i>Bibliography</i>	252

List of Figures

2.1.	<i>Control Volume</i>	31
4.1.	<i>Structured grid</i>	70
4.2.	<i>Structured multi-block grid</i>	71
4.3.	Left: <i>cell-centred scheme</i> , Right: <i>cell-vertex scheme</i>	75
4.4.	<i>Flux-difference splitting</i>	77
4.5.	<i>Mapping function between Cartesian and Curvilinear coordinates</i>	85
5.1.	<i>CFL ramping</i>	112
6.1.	<i>Multigrid</i>	114
6.2.	<i>Multigrid cycles</i>	118
6.3.	<i>1-d simple injection</i>	119
6.4.	<i>1-d Full weight restriction</i>	119
6.5.	<i>Full weighting restriction operator for two-dimensional cell-vertex schemes</i>	120
6.6.	<i>Weighted restriction operator for two-dimensional cell centred schemes</i>	121
6.7.	<i>Residual injection operator</i>	122
6.8.	<i>2-d residual high-weighted restriction operator</i>	123
6.9.	<i>Full-weighting prolongation for one-dimensional problems</i>	125
6.10.	<i>Full-weighting prolongation for two-dimensional cell-vertex problems</i>	125
6.11.	<i>Bilinear prolongation for two-dimensional cell-centred problems</i>	126
7.1.	<i>Turbulent flat plate: grid</i>	134
7.2.	<i>Turbulent flat plate: velocity profiles and skin friction coefficient</i>	136
7.3.	<i>Turbulent flat plate: convergence histories</i>	138
7.4.	<i>NACA4412 airfoil: grid</i>	140
7.5.	<i>NACA4412 airfoil: static pressure coefficient</i>	141
7.6.	<i>NACA4412 airfoil: Wilcox's and Menter's wall BC velocity profiles</i>	142
7.7.	<i>NACA4412 airfoil: first and second order velocity profiles</i>	144
7.8.	<i>NACA4412 airfoil: first and second order skin friction coefficients</i>	147
7.9.	<i>NACA4412 airfoil: convergence histories</i>	148
7.10.	<i>RAE2822 airfoil: grid</i>	150
7.11.	<i>RAE2822 airfoil: static pressure coefficient</i>	151
7.12.	<i>RAE2822 airfoil: skin friction coefficient</i>	153
7.13.	<i>RAE2822 airfoil: mach contours</i>	154
7.14.	<i>RAE2822 airfoil: convergence histories</i>	155

8.1.	<i>NACA0015 airfoil: prescribed motion of oscillating wing</i>	159
8.2.	<i>NACA0015 airfoil: medium-refinement grid</i>	162
8.3.	<i>NACA0015 airfoil: operating condition A - varying time-refinement</i>	164
8.4.	<i>NACA0015 airfoil: operating condition A - varying space-refinement</i>	165
8.5.	<i>NACA0015 airfoil: operating condition A - turbulent vs laminar analysis</i>	168
8.6.	<i>NACA0015 airfoil: operating condition A - Streamlines & vorticity contours</i>	171
8.7.	<i>NACA0015 airfoil: operating condition A - vorticity contours</i>	172
8.8.	<i>NACA0015 airfoil: operating condition B - turbulent vs laminar analysis</i>	174
8.9.	<i>NACA0015 airfoil: operating condition A - convergence histories</i>	176
8.10.	<i>NACA0012 airfoil: prescribed motion of oscillating wing</i>	178
8.11.	<i>NACA0012 airfoil: transonic – grid</i>	180
8.12.	<i>NACA0012 airfoil: transonic – varying time-refinement</i>	182
8.13.	<i>NACA0012 airfoil: transonic – TD hysteresis cycle of lift coefficient</i>	184
8.14.	<i>NACA0012 airfoil: transonic – TD hysteresis cycle of moment coefficient</i>	184
8.15.	<i>NACA0012 airfoil: transonic – TD hysteresis cycle of drag coefficient</i>	185
8.16.	<i>NACA0012 airfoil: transonic – TD instantaneous pressure coefficient</i>	187
8.17.	<i>NACA0012 airfoil: transonic – TD convergence histories using Menter’s BC</i>	188
8.18.	<i>NACA0012 airfoil: transonic – TD convergence histories using Wilcox’s BC</i>	189
8.19.	<i>NACA0012 airfoil: transonic – HB hysteresis cycle of lift coefficient</i>	191
8.20.	<i>NACA0012 airfoil: transonic – HB hysteresis cycle of moment coefficient</i>	192
8.21.	<i>NACA0012 airfoil: transonic – HB hysteresis cycle of drag coefficient</i>	193
8.22.	<i>NACA0012 airfoil: transonic – HB zeroth surface pressure coefficients</i>	194
8.23.	<i>NACA0012 airfoil: transonic – HB first surface pressure coefficients</i>	195
8.24.	<i>NACA0012 airfoil: transonic – HB second surface pressure coefficients</i>	195
8.25.	<i>NACA0012 airfoil: transonic – HB instantaneous pressure coefficient</i>	197
8.26.	<i>NACA0012 airfoil: transonic – HB convergence histories using Wilcox’s BC</i>	198
8.27.	<i>NACA0015 airfoil: subsonic – grid</i>	202
8.28.	<i>NACA0015 airfoil: subsonic – varying time-refinement</i>	203
8.29.	<i>NACA0012 airfoil: subsonic – TD & HB hysteresis cycle of lift coefficient</i>	205
8.30.	<i>NACA0012 airfoil: subsonic – TD & HB hysteresis cycle of moment coeff</i>	205
8.31.	<i>NACA0012 airfoil: subsonic – TD & HB hysteresis cycle of drag coeff</i>	206
8.32.	<i>NACA0012 airfoil: subsonic – TD & HB zeroth surface pressure coeff</i>	207
8.33.	<i>NACA0012 airfoil: subsonic – TD & HB first surface pressure coeff</i>	208
8.34.	<i>NACA0012 airfoil: subsonic – TD & HB second surface pressure coeff</i>	208
8.35.	<i>NACA0012 airfoil: subsonic – TD & HB instantaneous pressure coeff</i>	210

8.36.	<i>NACA0012 airfoil: subsonic – TD convergence histories using Wilcox’s BC</i>	.211
8.37.	<i>NACA0012 airfoil: subsonic – HB convergence histories using Wilcox’s BC</i>	.212
G.1.	<i>Duffing’s oscillator solution</i>	233
J.1.	<i>Skin friction coefficients for wing positions labelled 1-3</i>	249
J.2.	<i>Skin friction coefficients for wing positions labelled 4-6</i>	250

Nomenclature

a_k	Runge-Kutta coefficients
c	Speed of sound, airfoil chord
c_v	Specific heat at constant volume
c_p	Specific heat at constant pressure
C_L	Lift coefficient
C_D	Drag coefficient
C_M	Momentum coefficient
C_P	Power coefficient
C_Y	Vertical force coefficient
C_X	Horizontal force coefficient
D_K	Destruction of K
D_ω	Destruction of ω
d	Damping coefficient of Duffing's oscillator
E	Total energy per unit mass
\hat{E}_c	x -components of $\hat{\Phi}_c$
\hat{E}_d	x -components of $\hat{\Phi}_d$
\hat{F}_d	y -components of $\hat{\Phi}_d$
\hat{F}_c	y -components of $\hat{\Phi}_c$
F	Harmonic forces of Duffing's oscillator
f_{MG}	Multigrid forcing function
f	Oscillation frequency
H	Total enthalpy per unit mass
k_L	Thermal conductivity
k_T	Turbulent thermal conductivity
k	Stiffness of Duffing's oscillator
L_{IRS}	Implicit residual smoothing operator
Δl_r	Root mean square of the cell residuals
M	Mach number
\mathbf{n}	Unit vector
N_H	Number of complex harmonics
p	Static pressure

P_r	Prandtl number
P_{r_T}	Turbulent Prandtl number
P_d	Production term
P_K	Production of K
P_ω	Production of ω
\mathbf{q}	Heat flux vector
$q_{i(j)}$	Cartesian components of the heat flux vector \mathbf{q}
\mathbf{q}_T	Turbulent heat flux vector
q_{Tj}	Cartesian components of the turbulent heat flux vector \mathbf{q}_T
\hat{Q}	Array of unknowns
Re	Reynolds number
\hat{R}	Cell residuals
\underline{S}	Strain rate tensor
S_R	Surface roughness parameter
S	Surface
dS	Surface element
t	Time
T	Static temperature
\hat{U}	Conservative flow variables
u_∞	Free-stream velocity
\mathbf{u}	Velocity vector
$u_{i(j)}$	Cartesian components of the flow velocity vector \mathbf{u}
\mathbf{u}_τ	Friction velocity vector
u^+	Non-dimensionalized velocity component parallel to the wall
v_e	Effective velocity
v_y	Heaving velocity
\mathbf{v}_b	Velocity vector of the boundary
V	Volume
dV	Volume element
\mathbf{x}	Position vector
$x_{i(j)}$	Cartesian components of the position vector
y^+	Nondimensionalized wall distance

a	Coefficient of Duffing's oscillator, effective angle of attack
β^*	Turbulent coefficient
β	Turbulent coefficient
γ	Ratio of specific heat, turbulent coefficient
γ^*	Turbulent coefficient
δ_{ij}	Kronecker Delta Function
ζ	Nondimensionalized damping coefficient of Duffing's oscillator
η	Efficiency
θ	Pitching angle
K	Turbulent kinetic energy
λ	Eigenvalues, reduce frequency
μ	Molecular dynamic viscosity
μ_T	Eddy viscosity
μ_t	Total dynamic viscosity
ρ	Density
σ_K	Turbulent coefficient
σ_ω	Turbulent coefficient
$\underline{\tau}$	Molecular stress tensor
τ_{ij}^R	Components of the Reynolds stress tensor
τ_{ij}^F	Components of the Reynolds-Favre stress tensor
$\Delta\tau$	Local pseudo-time-step
$\widehat{\Phi}_c$	Generalized convective flux vectors
$\widehat{\Phi}_d$	Generalized diffusive flux vectors
ω	Specific dissipation rate, excitation frequency
ω_0	Natural frequency of Duffing's oscillator
Ω	Pitching velocity

Chapter 1

INTRODUCTION

1.1 Computational fluid dynamics

The growth of digital computer technology in the last two decades has dramatically reduced the time needed for engineers to accurately analyse and solve very complex flow problems. This has led not only to significant improvements in the design of a wide range of products, ranging from cars to aircraft and aircraft engines, but also to the exploration of new design solutions which were previously not considered due to the complexity of the fluid dynamics of such solutions, and to the high level of uncertainty associated with such complex fluid dynamics. Due to the abovesaid rapid growth of computing power and the availability of revolutionary new and powerful computing hardware such as *General Purpose Graphics Processing Units* (GPU) [1], designers and researchers have developed new aerodynamic and aeroelastic analysis and design technologies to solve complex fluid problems for modern engineering tasks. The development of these aerodynamic and aeroelastic computational tools applicable to many areas, including mechanical, aeronautical, marine and civil engineering is the subject of *Computational Fluid Dynamics* (CFD). The underlying principle of CFD is the numerical solution of the fundamental conservation laws of fluid mechanics at the discrete points of a computational grid, discretizing the physical domain of interest [2, 3, 4]. For both the academic and the industrial sectors, a variety of new numerical methods has been implemented in complex computer programs with great success both in terms of physical accuracy and computational efficiency. Nowadays, CFD is used by both industry and academia for a wide variety of engineering applications, including design and analysis of wind [5], hydraulic [6] and gas [7] turbines, aircraft [8] and rotorcraft [9] components or whole aircraft [10], car shape [11] and many other diverse products. On the other hand in high complexity physical model applications, experimental results may be more accurate than those of CFD. For this reason experiments are still the subject of research and development and their results are an invaluable source of data for validating new CFD

methods. The rapidly increasing popularity of CFD in the industry in the past few decades has enabled the reduction of product development cost since expensive experimental campaigns could be reduced by cheaper CFD simulations. Although the final development stages of complex products such as aircraft and aircraft engines still rely on costly experimental measurements campaigns, the use of CFD is allowing substantially more innovative design to be introduced, such as the highly three-dimensional fan blades of modern ultra-high-bypass-ratio turbofan engines [12].

The fundamental tool of CFD is a computer code developed to solve numerically the physical conservation laws governing the fluid problem at hand. There exist several levels of fluid flow models of widely varying complexity. Such models range from the steady incompressible inviscid irrotational potential flow model that can be solved (numerically in general, and analytically only in special cases) by considering a Laplacian operator, to the time-dependent compressible *Navier-Stokes* (NS) viscous flow model [13], that can be solved by considering a system of parabolic (with respect to time-variable) *partial differential equations* (PDEs). The Navier-Stokes flow model is one of the most general ones, but unfortunately, in the vast majority of flow problems of practical interest it is impossible to determine the analytical solution to these equations. This is the reason why the development of novel numerical approaches required by Navier-Stokes CFD codes to solve engineering flow problems of practical interest is a crucially important problem. Historically, the expression Navier-Stokes equations denoted only the equation describing the conservation of the linear momentum of the flow field under consideration. More recently, however, the expression has been used to name all three conservation laws required to solve a time-dependent compressible fluid flow problem, namely the conservation of mass, the conservation of linear momentum (which is a vectorial equation with 3 or 2 components depending on the problem dimensionality), and the conservation of energy. Many real flow fields past stationary or moving objects (e.g. wings, blades, buses, aircraft or ships) are turbulent. This means that the flow field is stochastic and chaotic, and eddies of random size can appear in the flow field.

Using the Navier-Stokes equations to solve directly all temporal and spatial scales of turbulent high Reynolds number flow requires formidably high temporal and spatial resolutions due to the very wide range of temporal and spatial scales present in such flows. This approach, called *Direct Numerical Simulation* (DNS) [14] is not affordable for most problems of engineering interest due to the present lack of sufficiently large computational resources. Fortunately, there are several other approaches to accounting for turbulent flow effects when using the Navier-Stokes equations to solve complex

turbulent fluid flow problems, and such approaches differ from each other both because of their level of approximations of the turbulent flow features, and their computational cost. In General, the models with fewer approximations are the computationally more expensive ones, as expected. One of the most widespread, simpler and computationally cheap approaches is that associated with the use of the so-called *Reynolds-Average Navier-Stokes* (RANS) equations. In this method the NS equations are time-averaged on the time-scales of turbulence. By doing so, one obtains the RANS equations, which is a set of 3 partial differential equations formally very similar to the NS equations. Formally, the RANS equations differ from the NS equations for a small number of additional terms resulting from the time-averaging, taking into account the effects of turbulence in a mean sense. Such additional terms introduce new unknowns, and, in order to *close* the system, i.e. have an equal number of PDEs and unknowns, one has to introduce a turbulence model [15]. Turbulence model can consist of a single algebraic equation resulting from the use of a semi-empirical model, like the Baldwin-Lomax model [16], one or more additional partial differential equations, like the one-equation Spalart-Allmaras model [17], and the two equation $K - \varepsilon$ [18] and $K - \omega$ [19] models, and also the seven-equation Reynolds stress model [20, 21]. From a conceptual viewpoint, the main differences between the NS and RANS equations is that in the former case the flow variables represent instantaneous values of the time-dependent highly fluctuating turbulent flow, whereas in the latter case the flow variables represent mean values (i.e. time-independent), and the only information on the turbulent fluctuations is contained in an averaged form in the additional terms of the RANS equations. The Reynolds averaging approach requiring the use of one or just a few additional partial differential equations has the fundamental benefit that the temporal and spatial refinement of the computational grids required to tackle realistic engineering problems is such that the problem can be analysed using relatively small computational resources. This is because one does not need a particularly high refinement since there is no need to solve temporally and spatially the turbulence scales, neither the very tiny nor the large ones. The effects of turbulence are accounted for in an averaged fashion. The use of RANS models yields a lower fidelity than that achievable by DNS, and this is because the turbulence model required for the system closure depends partly on empirical coefficients determined from a small number of experimental results. The turbulence models featuring such constants are then applied to a wide variety of problems in which the assumptions of the experiments used to determine part of the turbulence model coefficients, often do not hold. Nevertheless, in flow problems characterised by a low to medium degree of separation near solid wall

boundaries and where the main objective of the analysis is the calculation of the forces acting on the body of interest rather than the far-field evolution of wakes and shed vorticity, it is found that the fidelity of RANS models is adequate for solving a wide class of practical engineering problems. It should be mentioned that other turbulence modelling approaches for the NS equations featuring fidelity higher than those of the RANS models and lower than that of DNS exist. The most notable method of this class is *Large Eddy Simulation* (LES) [22]. The LES approach to turbulence modelling is conceptually similar to that of the RANS approach. The main difference between LES and RANS models is that the former approach simulates directly (both temporally and spatially) the larger temporal and spatial turbulent scales, whereas it models (similarly to what the differential turbulence models used for the turbulence closure of the RANS equations do) the smaller temporal and spatial scales of turbulence. The modelling fidelity of LES is higher than that of RANS particularly for predicting the far-field temporal and spatial evolution of wakes and regions away from the body where such rotational flows originate. However, it is often found that, despite this feature, the LES and RANS prediction of the force acting on aerodynamic bodies such as aircraft wings and wind turbine blades are very close [23]. For high-Reynolds number wall-bound flows, however, the grid refinement required for accurate LES CFD simulations makes these analyses barely affordable on the largest supercomputers available today, and therefore the use of LES for practical engineering design remains still very limited. This is also the case for hybrid RANS/LES methods known as *Detached Eddy Simulations* (DES) [24, 25, 26] and *Delayed Detached Eddy Simulation* (DDES) [27, 28]. In DES, the RANS model is used close to solid wall boundaries, and LES is used in the rest of the domain. The use of RANS in the wall proximity, enables the use of substantially coarser grids than LES would require, and this reduces the computational cost of DES with respect to that of LES, though the computational cost of DES remains significantly higher than that of RANS approach. The DES approach is a promising compromise between computational cost and solution accuracy, particularly for high Reynolds number flows featuring significant vorticity production and propagation. Two of the main difficulties of a reliable DES CFD code is the selection and the implementation of an adequate criterion that identifies the domain portions where the use of either the RANS or the LES approach is applicable, and the implementation of a smooth transition from one model to the other [29, 30] .

1.2 Oscillating wings

Three complex and realistic time-dependent fluid flow problems that are of particular interest to this thesis, as they will also serve the purpose of highlighting the predictive capabilities and the computational efficiency of the computational methods developed in this research, are the unsteady fluid dynamics of hydrokinetic turbines, the transonic aircraft wing sections and subsonic oscillating wing. These three problems share the existence of an oscillating wing animated by a harmonic motion, and such a motion is the root cause of the observed flow unsteadiness. An additional reason for selecting the hydrokinetic turbine problem is that the computational technologies developed in this research have also been used to investigate the complex flow physics of this device and obtain novel information on this emerging renewable energy device. The three problems are briefly summarised in the following two subsections, along with a brief literature survey of the work carried out in these areas.

1.2.1 Hydrokinetic turbines

Large-scale electricity production from the wind is primarily based on the use of multi-megawatt horizontal axis turbines. Recently, the interest in the exploitation of tidal and river flows is also rapidly increasing. In most cases, the machine designs that are being considered and starting to be installed for the exploitation of tidal energy are derived from wind turbine technology, due to knowledge and measured data already available in the wind energy case. Therefore, even for tidal flow applications, one of the most popular machine layouts that is being considered is that of the horizontal axis wind turbine [31], though the vertical axis concept is also receiving some attention [32]. However, the use of significantly different devices for renewable energy production based on the conversion of the kinetic energy of fluid streams into mechanical and ultimately, electrical energy is receiving increasing attention from the scientific and industrial communities. A promising concept relies on the use of oscillating wings simultaneously heaving and pitching as power extraction devices. This concept was initially proposed by McKinney and DeLaurier [33] in 1981, and was further investigated by Jones et al. [34] more

recently. A thorough parametric CFD investigation into the effects of kinematic parameters (heaving amplitude, pitching amplitude and frequency) and geometric parameters (airfoil shape and location of pitching axis) on the efficiency of the power extraction achievable by the oscillating wing has been performed by Kinsey and Dumas [35]. The simulations reported in that study have been performed using the incompressible solver of the commercial CFD package FLUENT, and the majority of these analyses refer to a laminar flow regime with a Reynolds number based on the airfoil chord and the freestream velocity of 1100. The main conclusions of the article [35] are that the oscillating wing can extract energy from an oncoming fluid stream with efficiencies as high as 34%, and the main aerodynamic feature responsible for such a relatively high efficiency is the unsteady leading edge vortex shedding associated with dynamic stall. These conclusions were also confirmed by a later independent study performed by Campobasso and Drofelnik using the compressible research code with a low-speed preconditioner optimized for time-dependent flows [36].

A prototype of the oscillating wing for extracting energy from an oncoming water stream has been recently tested in water at Lac-Beauport near Quebec City. The measured data have confirmed fairly high values of the energy conversion efficiency of this device. Based on the data reported in [37], the Reynolds number of the considered operating condition was 0.48 million. In this condition, the flow regime is predominantly turbulent, and CFD analyses aiming to provide accurate and quantitative data for investigating the fluid dynamics of the oscillating wing and optimizing its design ought to include the effects of turbulence. In a recent article of Kinsey and Dumas [38], the hydrodynamics of the oscillating wing for power production at a Reynolds number of 0.5 million is investigated by means of two-dimensional (2D) and three-dimensional turbulent incompressible FLUENT flow simulations using the *Spalart-Allmaras* (SA) turbulence model [17]. Although the article [38] does not report a comparative study of the hydrodynamics of the oscillating wing in the laminar and turbulent regimes, cross-comparison of the laminar and turbulent flow simulations reported in the articles [35, 38] points to the fact that the efficiency of the energy conversion appears to increase significantly as the Reynolds number increases from low values, typical of laminar regimes, to fairly high values, at which a predominantly turbulent regime is expected. This cross-comparison analysis between laminar regime at a Reynolds number of 1100 and turbulent regime at a Reynolds number of 1.5 million has been performed by Campobasso and Piskopakis et al [39] using a compressible research code with the two-equation turbulence model of Menter [40] for closure. The main conclusion of article [39]

is that the oscillating wing can extract energy from an oncoming fluid stream with efficiency as high as 40% but due to turbulent flow effects, the optimal kinematic parameters obtained for the laminar regime have been lost. A new parametric CFD investigation on the effects of kinematic parameters (heaving amplitude, pitching amplitude and frequency) to achieve a maximum efficiency of the power extraction for the turbulent regime at a Reynolds number of 0.5 million has been performed by Kinsey and Dumas [41]. The main conclusions of the article [41] are that the oscillating wing can extract energy from an oncoming fluid stream with efficiencies as high as 43%, and the main aerodynamic feature responsible for such a relatively high efficiency is no longer related to the unsteady leading edge vortex shedding associated with dynamic stall as one of the optimal cases was found to be without a leading edge vortex shedding.

To date, there appears to exist more experimental and prototype-based studies regarding the use of the oscillating wing device to extract energy from oncoming water rather than air streams. Nevertheless, overall feasibility studies and detailed aerodynamic analyses aiming at developing oscillating wing devices to extract energy from the wind are ongoing. In 2008 Platzer et al. [42] have proposed the use of the oscillating wing technology for the development of ‘flying flapping-wing power generators for the purpose of tapping into the abundant energy available in the global jet streams’. They have also provided preliminary multidisciplinary assessments of the technical viability of this concept, which has been patented by Bradley and Platzer in 2009 [43]. The interest in the use of the oscillating wing device to extract energy from air streams is also highlighted by the increasing number of computational studies in this area [44, 45, 46]. A review of progress and challenges in flapping foil power generation can be found here [47].

One of the objectives of this thesis is to thoroughly investigate the effects of flow turbulence on the detailed aerodynamic features accounting for the energy extraction of the oscillating wing by performing a comparative aerodynamic analysis of the device at the laminar flow condition with a Reynolds number of 1100 considered in [35, 36], and a turbulent regime with a Reynolds number of 1.5 million [39]. The presented analyses will highlight that the different aerodynamics of the laminar and turbulent regimes result in significantly different levels of power conversion efficiency. Although the turbulent flow analysis of the oscillating wing operating at a Reynolds number of 0.5 million has been reported in [38], it is instructive to perform comparative turbulent/laminar CFD analyses of this device because the Reynolds number of real installations is likely to vary due to both site-dependent design specifications and off-design operation. Hence, detailed

knowledge of how the fluid dynamics of oscillating wing devices varies over a wide range of the Reynolds number for given geometric and kinematic characteristics is needed to accurately assess the energy yield of this device. From this perspective, the present thesis is complementary to the other published studies in this area.

1.2.2 Wing aerodynamics

Wing aerodynamics has been a subject of research for many years. In 1911 Bryan [48] introduced the first aerodynamic models that could be applied on flight dynamics, and since then many types of experiments have been performed for test and design purposes. Recently McDaniel et al [49] compared computational fluid dynamics solutions for a fighter aircraft, with realistic flight test data, the so called aircraft flight testing [50], which is a costly process as an aircraft has to be developed. An alternative experiment, cheaper than aircraft flight testing, is the Wind-Tunnel testing of scaled models [51], where the object under consideration is positioned inside a closed tubular passage where the wind is produced by a powerful fan. Constraints such as scaling, blockage effects and the fact that it is difficult to achieve real-flight conditions, limit the effectiveness of Wind-Tunnel testing. Nowadays the most cost-efficient approach, used by industry and academia worldwide, to design, predict and optimize nonlinear flow physics is Computational Fluid Dynamics [52].

In CFD applications, periodic oscillating wings have recently been a subject of research by aeronautical engineers, which include flutter oscillations for aircraft aeroelastic stability [53] or windmills [33], hydroelectric generators [37] and horizontal [31] or vertical [5, 54] axis wind turbines, to extract energy from an oncoming fluid stream [55, 56]. The aim of these applications is to numerically compute the sought periodic solution of an oscillating wing using CFD tools. The sought periodic solution obtained from time-domain CFD solvers requires significant computational effort to bypass transient effects, which, in many cases of engineering interest, means that several periods have to be simulated [57, 58]. An alternative and computationally cheaper method with respect to the time-domain approaches has recently been developed, the so-called harmonic balance method, which will be introduced in the following subsection.

One of the objectives of this thesis is to assess the computational efficiency of the turbulent RANS SST harmonic balance solver by performing time-domain and harmonic

balance turbulent analyses of a transonic oscillating airfoil animated by pitching motion, representative of the cross section of a transonic aircraft wing, and comparing the computational resources required by either approach. The comparison of the run-time associated with the COSA time-domain and harmonic balance analyses will highlight that the latter approach reduces at least one order of magnitude the run-time required to determine the periodic flow solution with respect to the run-time required by the time-domain analysis. Similar comparison analyses will be performed for a subsonic oscillating airfoil animated by pitching motion where even higher run-time reduction of the harmonic balance with respect to the run-time of the time domain will be highlighted.

1.3 COSA solver

The computational technologies developed in this research and discussed in this thesis have been implemented and used to analyse the fluid dynamics problems discussed above in a compressible time-dependent RANS research code. The *CFD Optimised Structured multi-block Algorithms* (COSA) RANS code [39] solves the steady and time-dependent compressible RANS equations space-discretized on structured multi-block grids using an efficient explicit multigrid solver. The turbulent COSA solver solves the two systems of algebraic equations resulting from the time- and space-discretization of the RANS (or mean flow) equations and the two-equation Shear Stress Transport [40] turbulence model equations by means of an explicit multigrid algorithm based on a four-stage Runge-Kutta smoother. The two systems are solved using a strongly coupled approach [59, 60, 61], whereby the mean flow and the turbulence equations are solved simultaneously in the iterative process. This integration approach has been shown to lead to significantly faster convergence rates than the loosely coupled method [62, 63], whereby the mean flow and turbulence equations are solved separately and often with different methods. It is also possible to use a 'hybrid' integration approach, whereby multigrid is applied simultaneously to the two systems, and time-marching on each grid level is decoupled [64], but this approach has not been adopted in the COSA code, which instead features the standard strongly coupled approach: the multigrid solver is applied simultaneously to the mean flow and turbulence equations, and the two systems are time-marched simultaneously on each grid level. The turbulent COSA code adopts the strongly coupled

integration method also for computing time-dependent problems, whereby the explicit multigrid integration is used to solve the *unsteady RANS* (URANS) equations coupled to the SST turbulence model. For such time-dependent problems, the turbulent multigrid solver also features a point-implicit treatment of certain terms arising from the discretization of the physical time-derivatives. This approach is an extension of the stabilization process reported by Melson et al. [65], and it enables the use of fairly high *Courant Friedrichs Lewy* (CFL) numbers, thus significantly reducing the number of multigrid cycles required to achieve a user-given reduction of the flow residuals.

COSA features two different approaches to the solution of the URANS equations. One approach, applicable to general unsteady flows (e.g. transient states) is the classical time-domain solution, based on the *dual time stepping* (DTS) approach where by one uses a backward finite difference discretisation to implicitly time-march the solution from one physical time to the next one with a user-given time step, and the multigrid solver to compute iteratively the solution at each physical time [66]. The other approach, applicable to nonlinear periodic flows, is the nonlinear *frequency-domain* (FD) *harmonic balance* (HB) integration. The harmonic balance is a perturbation method for the rapid solution of nonlinear problems described by one or more *ordinary differential equations* (ODE). The sought solution is represented as a truncated Fourier series retaining a user given number of complex harmonics, and the given time-domain problem is reformulated and solved in the frequency-domain using the solution approximation provided by the truncated Fourier series. The method is particularly effective when only a relatively small number of complex harmonics is required to represent accurately the time-domain solution. In the case of periodic turbulent flow, the harmonic balance solution enables a substantial reduction of the computational time with respect to the time-domain case with negligible accuracy penalties. Like all frequency-domain methods, it aims at computing directly the sought periodic solution, whereas time-domain methods need to solve also the flow transient preceding the establishment of the sought periodic state. This is one of the main reasons why the harmonic balance solver is substantially faster than the time-domain solver. Another important reason for the effectiveness of the harmonic balance RANS technology is that many periodic flows of engineering interest, though significantly nonlinear, can be accurately represented by using a few harmonics. As shown later in the thesis, the computational advantage of using the harmonic balance rather than the time-domain solution is particularly high when the number of retained complex flow harmonics is low. The harmonic balance solution of the RANS equations was first introduced by Hall et al. [67], who first showed that, for the case of turbomachinery flows,

the technology can reduce the computational time to solution by at least one order of magnitude compared to the time-domain approach.

The development of COSA started within an EPSRC project [68] in 2008. By the end of the project, the code featured novel advanced far-field *boundary conditions* (BCs) [69], a novel unsteady low-speed preconditioner (the so-called *mixed-preconditioning* algorithm) [70] to enable the CFD analysis of very low-speed flows as well as flows featuring both high- and low-speed regions, the harmonic balance solver for the rapid analysis of unsteady periodic flows like those associated with the yawed wind regime of horizontal axis wind turbines [70], and also a first implementation of the $K - \omega$ turbulence model of Wilcox [19], a two-equation eddy viscosity model. The first implementation of $K - \omega$ model, however, was not robust. As a consequence, neither the advanced low-speed modelling capabilities enabled by the mixed-preconditioning technology nor the substantial reductions of the run-time of unsteady periodic flow analyses enabled by the harmonic balance solver could be proved in the case of realistic high-Reynolds number turbulent flows. Part of the research work of this thesis consisted of investigating and successfully solving the robustness issues of the $K - \omega$ turbulence model, and successfully implementing a new, more robust and accurate turbulence model, namely *Shear Stress Transport* (SST) model of Menter [40]. The first published demonstration of the new SST capability is reported in [39].

1.4 Objectives, novelty and overview of the thesis

The main drive of the research work reported in this thesis was two-fold: on one hand, the research aimed to develop, assess and validate novel algorithmic and modelling technologies to improve the robustness and the computational efficiency of the RANS model-based analysis of general unsteady and periodic flow fields; on the other hand, the work aimed to demonstrate the accuracy and the effectiveness of the developed technologies by using the new RANS framework to investigate three challenging turbulent high-Reynolds number unsteady flows, namely the unsteady aerodynamics of a hydrokinetic turbine, the periodic transonic flow field past a transonic oscillating wing section, and a periodic subsonic oscillating wing. More specifically, the main objectives associated with the algorithmic work of this research were to:

- develop a numerically robust i.e. stable, multigrid-based solution framework for the fully-coupled integration of the steady, time-domain and harmonic balance RANS and SST turbulence model equations, and
- assess the computational efficiency of the turbulent RANS SST harmonic balance solver by performing time-domain and harmonic balance turbulent analyses of a transonic wing section animated by pitching motion and comparing the computational resources required by each approach.

The main objective associated with the applied fluid dynamics part of this work was instead to investigate the flow mechanisms accounting for the high energy extraction efficiency of a hydrokinetic turbine both at a low-Reynolds number laminar flow condition, and a high-Reynolds number fully turbulent flow condition.

The thesis presents several elements of novelty, on the algorithmic side, on the turbulence modelling side and also on the fluid mechanics of hydrokinetic turbines. The developed multigrid fully coupled integration of the steady and time-domain RANS and SST equations is partly an extension of the approach proposed by Liu and Zheng [59] for steady equations and Yao et al [71] for the time-domain equations. However, the adaptation of this approach to the SST turbulence model and the detailed theoretical and numerical analyses carried out to optimise the algorithmic design of this procedure are a novel feature reported for the first time in this thesis. Here, this approach is also extended to the fully coupled integration of the turbulent harmonic balance RANS and SST equations, and this is also a feature, which, to the best of the writer's knowledge, had never been implemented and used before. On the modelling side, the thesis reports, for the first time, an important parametric analysis on the numerical set-up of the turbulent wall boundary condition of the second turbulent variable ω . This study highlights that the variations of the solution associated with the, to a certain extent, arbitrary choice of the scaling factor for ω at the wall using a given turbulence model, can be of the same order of magnitude or even larger than the variations observed using a given scaling factor and different turbulence models. This result highlights an additional, typically overlooked source of uncertainty in the RANS-DES based turbulent flow analyses. Another important algorithmic aspect of the thesis is a comprehensive assessment of the solution sensitivity to the use of first or second order schemes for the turbulence equations for both subsonic and transonic flow problems. As for the fluid dynamics of hydrokinetic turbines, the

reported study highlights that the flow mechanisms accounting for the high energy extraction efficiency of oscillating wings in laminar and turbulent flow conditions are different. This new finding has recently been confirmed in a paper published in 2014 [41].

The thesis is organised as follows. Chapter 2 reports in great detail the time-domain formulation of the compressible RANS and SST turbulence model equations, the Boussinesq approximation yielding the assumption of isotropic turbulence and enabling the use of eddy viscosity turbulence model, and the far-field and solid wall boundary conditions for the system of RANS and SST equations.

Chapter 3 presents the classical formulation of the harmonic balance perturbation method for finding the periodic solution of systems of ordinary differential equations, and also the so-called high-dimensional formulation of the harmonic balance method, which is a variant of the former formulation better suited to the numerical solution of the system of the RANS and SST equations. The two formulations are highlighted with a simple ODE example, and the chapter is concluded with the harmonic balance formulation of the RANS and SST equations. Chapter 4 focuses on the cell-centred finite volume space discretisation of COSA. It briefly reports the methods used for the space discretisation of the convective fluxes, the diffusive fluxes and the source terms of the SST turbulence model. The chapter highlights how the space discretisation of the RANS and SST partial differential equations yields a large set of nonlinear algebraic equations, the numerical solution of which yields the sought CFD solution.

Chapter 5 focuses on the underlying iterative solver (smoother) used to solve iteratively the steady, time-domain and harmonic balance RANS and SST equations, namely a four-stage Runge-Kutta scheme. The smoother is modified to enable a numerically robust fully-coupled integration of the flow and turbulence equations. Two versions of such a fully coupled approach are presented and discussed, an exact variant and an approximate variant yielding a reduction of computation costs. The chapter also briefly describes several convergence acceleration techniques used for the integration, namely the local time stepping method, a variable coefficient centred implicit residual smoothing algorithm and two CFL ramping schemes. The COSA code also uses a full approximation scheme multigrid algorithm for further accelerating the residual convergence. A detailed description of the turbulent multigrid algorithm used by COSA is reported in chapter 6. This includes new variants of the restriction and prolongation operators implemented in this research aiming at further improving the robustness of the integration of turbulent flow problems.

Chapter 7 presents the solution of three turbulent flow problems, which are used to

validate the turbulent flow predictive capabilities of COSA. Firstly a steady flat plate boundary layer for which theoretical results are available is considered. The characteristic boundary layer profiles computed by COSA and the available theoretical results are compared. The second test case is the NACA4412 airfoil in a low-speed freestream featuring a significant amount of flow reversal on the rear part of the suction side. The third test case is a RAE2822 airfoil operating in a transonic flow regime. All three test cases are analysed not only to validate the predictive flow capabilities of COSA, but also to:

- assess the computed solution variability (i.e. uncertainty level) associated with the typical values selected for the scaling factor of the ω turbulent flow variable at solid wall boundaries,
- demonstrate the high robustness of the turbulent solver resulting from the fully coupled integration and the turbulent multigrid enhancements developed in this research, and
- assess the impact of using either a first or a second order discretisation of the computed turbulent flow equations for both subsonic and transonic problems and also for varying degrees of mesh refinement.

The importance of this last analysis stems from the fact that published literature on this aspect provides a somewhat patchy set of conclusions, as some studies conclude that the use of second order discretisation for turbulence models is essential for high accuracy solution, and some other studies claim the opposite and prefer the use of first order discretisation, presumably because this may lead to more stable numerical solution procedures. Moreover, none of the existing studies on this topic addresses exhaustively the dependence of this issue on the flow regime and the grid refinement level.

Chapter 8 provides the main computational results of this research, consisting of a low-speed renewable energy application and two aeronautical engineering applications. The first application is a thorough comparative analysis of a hydrokinetic turbine working in a low-speed laminar and a high-Reynolds number turbulent regime, and it is also analyses and discusses in great detail the unsteady aerodynamics phenomena occurring in both regimes. The main motivation for the analysing this problem is to highlight the predictive capabilities and the numerical robustness of the developed turbulent time-

domain flow solver for complex realistic problems, and also to shed more light on the complex physics of this emerging renewable energy device. The second application is the time-domain and harmonic balance turbulent flow analysis of a transonic wing section animated by pitching motion. The main motivation of these analyses is to assess the computational benefits achievable by using the harmonic balance solution of the RANS and SST equations rather than the conventional time-domain solution, and also to further demonstrate the predictive capabilities of the developed CFD system. To this aim, the numerical solutions of COSA are compared to both available experimental data, and the time-domain results computed by a state-of-the-art commercial package regularly used by the industry and the academia worldwide. The harmonic balance and time domain comparison analysis is then repeated for a third application but in a subsonic regime. The conclusions of the thesis and future work are provided in chapter 9.

Chapter 2

REYNOLDS AVERAGED NAVIER-STOKES EQUATIONS AND TURBULENCE CLOSURE

The underlying principle of Computational Fluid Dynamics is the numerical solution of the governing equations of fluid mechanics at the discrete points of a computational grid, discretizing the physical domain of interest. This chapter outlines the derivation of the Reynolds-Averaged Navier-Stokes equations. This derivation is a multi-stage process. First the time-dependent equations are averaged on the turbulence time-scales. This operation yields a new system, which differs from the original time-dependent equations for two reasons: a) the new system does no longer have time-derivatives, so it can be used for the numerical solution of steady turbulent flows, and b) the new system has additional unknown terms which form the components of a second order tensor, called Reynolds stress tensor. In order to determine this tensor, use of the Boussinesq approximation and two-equation turbulence models is made. The solution of steady turbulent problems is obtained by solving the system of *Partial Differential Equations* (PDEs) made up of the Navier-Stokes equations averaged on the turbulence time-scales (i.e. Reynolds Averaged Navier-Stokes equations) and the two PDEs associated with the turbulence model. Time-dependent turbulent flows can instead be solved by adding to each RANS PDE and to each PDE of the turbulent model a new physical time-derivative, which considers time-variations on the time-scales of the engineering problem at hand (e.g. period of oscillation of vibrating blade or period of vortex shedding from blunt objects in fluid flows). The *Arbitrary Lagrangian Eulerian* (ALE) formulation of the unsteady RANS equations, needed for solving moving body problems, is also presented.

2.1 Integral form of the Navier-Stokes equations

The fundamental equations of fluid dynamics are based on the three universal laws of conservation: conservation of mass, conservation of momentum and conservation of energy. The conservation of mass states that the total mass of any system remains constant over time and mass is neither created nor destroyed. This system is closed from any external transfers of matter, energy or any kind of external sources (e.g. combustion). This equation is also known as continuity equation. The principle of the momentum conservation derives from Newton's second law, which states that the variation of momentum is caused by the overall force acting on a mass element. In two dimensional problems, the momentum conservation results in two scalar equations, one for each direction. In fluid mechanics these equations are known as momentum equations. The conservation of energy, known as energy equation, corresponds to the first law of thermodynamics which states that the total energy remains constant over time. In fluid dynamics the momentum equations are by definition the so-called *Navier-Stokes* (NS) equations. However, in recent years the expression ‘NS equations’ refers to all four equations mentioned above.

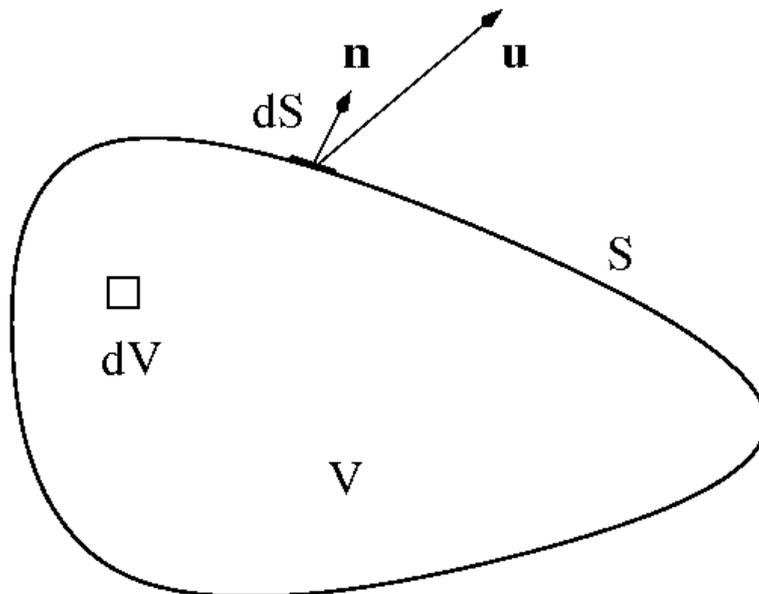


Figure 2.1: Control Volume

The NS equations can be written in the differential form, applicable at a point or in the integral form, applicable to an extended region. This region can be identified from a

quantity of matter called *control mass* (CM). This technique is commonly used in solid problems however this is not the case in fluid flows. It is very difficult to investigate the motion of a specific particle of matter through the physical domain. For this reason a spatial region is used, called *control Volume* (CV). Figure 2.1 highlights a volume V bounded by a surface S . The figure also depicts a surface element dS and its normal unit vector \mathbf{n} . This volume is the so-called *Control Volume*. In two dimensional problems V is a surface and S is a curve. The mass element dm is enclosed by an infinitesimal volume dV and it is equal to ρdV where ρ is the density. The expression of the integral form of the NS equations depends on whether this CV is fixed or not. In case of a fixed CV we say that the "Eulerian approach" is used. The following paragraphs explain briefly the terms of each equation as they are well established in the literature [2, 3, 4].

The conservation of mass yields:

$$\frac{\partial}{\partial t} \int_V \rho dV + \oint_S \rho (\mathbf{u} \cdot \mathbf{n}) dS = 0 \quad (2.1.1)$$

where the symbols t and \mathbf{u} are the time and the flow velocity vector, respectively. The first term of the continuity equation represents the variation of mass in time over a CV and the second term represents the transfer of mass through the boundary of the control volume.

The conservation of momentum, or Newton's second law yields:

$$\begin{aligned} \frac{\partial}{\partial t} \int_V \rho \mathbf{u} dV + \oint_S \rho \mathbf{u} (\mathbf{u} \cdot \mathbf{n}) dS \\ = - \oint_S p \mathbf{n} dS + \oint_S \underline{\boldsymbol{\tau}} \cdot \mathbf{n} dS. \end{aligned} \quad (2.1.2)$$

The symbol p is the static pressure and $\underline{\boldsymbol{\tau}}$ denotes the molecular stress tensor. This tensor depends on the divergence of the flow velocity vector \mathbf{u} and the strain rate tensor $\underline{\mathbf{S}}$. For a Newtonian fluid, the expression of stress tensor is:

$$\tau_{ij} = 2\mu \left[S_{ij} - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right], \quad S_{ij} = \frac{1}{2} \left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right] \quad (2.1.3)$$

where μ is the molecular dynamic viscosity, $u_{i(j)}$ are the Cartesian components of the flow velocity vector \mathbf{u} , δ_{ij} is the Kronecker Delta Function and $x_{i(j)}$ are the components of the position vector \mathbf{x} . Sutherland's Law is used to compute the dynamic viscosity μ :

$$\mu = \mu_{ref} \left(\frac{T}{T_{ref}} \right)^{3/2} \frac{T_{ref} + S_{\mu}}{T + S_{\mu}} \quad (2.1.4)$$

where T is the static temperature, $\mu_{ref} = 1.716 \cdot 10^{-5} \text{ kgm}^{-1} \text{ s}^{-1}$, $T_{ref} = 273.15 \text{ K}$ and $S_{\mu} = 110.4 \text{ K}$. The first term of equation (2.1.2) represents the variation of momentum in time over the CV and the second term represents the transfer of momentum through the control boundary. The first term of the right hand side of the equation (2.1.2) represents the pressure forces acting on the control volume, whereas the second term corresponds to the shear stresses acting on the control volume, respectively.

The last conservation law, which is an expression of the first law of thermodynamics, provides the conservation of energy. Its expression is:

$$\begin{aligned} \frac{\partial}{\partial t} \int_V \rho E dV + \oint_S \rho E (\mathbf{u} \cdot \mathbf{n}) dS &= \oint_S (\underline{\boldsymbol{\tau}} \cdot \mathbf{u}) \cdot \mathbf{n} dS \\ &- \oint_S p (\mathbf{u} \cdot \mathbf{n}) dS - \oint_S \mathbf{q} \cdot \mathbf{n} dS \end{aligned} \quad (2.1.5)$$

where the symbol E denotes the total energy per unit mass and \mathbf{q} represents the heat flux vector. The first term represents the variation of energy in time over the control volume and the second term denotes the transport of energy through the boundary of the control volume. The first two terms of the right hand side of the same equation correspond to the rate of work done on control volume. The last term of equation (2.1.5) represents the thermal heat fluxes \mathbf{q} , the rate of heat energy transfer through the control volume, and is given by the generalized Fick's gradient law, $\mathbf{q} = -k_L \nabla T$ where k_L is the thermal conductivity. The definition of the total energy E is:

$$E = e + \frac{u_i u_i}{2}, \quad e = c_v T \quad (2.1.6)$$

where the symbol e denotes the internal energy per unit mass and c_v is the specific heat at constant volume. For a calorically perfect flow, the specific heat coefficients are

constant. As a result, it is possible to relate temperature to the internal energy and static enthalpy and rewrite the equation (2.1.5) as:

$$\begin{aligned} \frac{\partial}{\partial t} \int_V \rho E dV + \oint_S \rho H (\mathbf{u} \cdot \mathbf{n}) dS &= \oint_S (\underline{\boldsymbol{\tau}} \cdot \mathbf{u}) \cdot \mathbf{n} dS \\ &- \oint_S \mathbf{q} \cdot \mathbf{n} dS \end{aligned} \quad (2.1.7)$$

where H is the total enthalpy per unit mass. Its definition is:

$$H = h + \frac{u_i u_i}{2}, \quad h = c_p T, \quad H = E + \frac{p}{\rho} \quad (2.1.8)$$

where the symbol h is static enthalpy per unit mass and c_p is the specific heat at constant pressure.

Finally we can write the Eulerian formulation as:

$$\frac{\partial}{\partial t} \left(\int_V \hat{\mathbf{U}} dV \right) + \oint_S (\hat{\boldsymbol{\Phi}}_c - \hat{\boldsymbol{\Phi}}_d) dS = 0 \quad (2.1.9)$$

where the symbol $\hat{}$ denotes an array and $\hat{\mathbf{U}}$ is the array of the conservative flow variables. The symbols $\hat{\boldsymbol{\Phi}}_c$ and $\hat{\boldsymbol{\Phi}}_d$ are the generalized convective and diffusive flux vectors, respectively. The array $\hat{\mathbf{U}}$ of conservative flow variables is defined as:

$$\hat{\mathbf{U}} = [\rho \quad \rho u_x \quad \rho u_y \quad \rho E]' \quad (2.1.10)$$

where the superscript $'$ donates the transpose operator. The generalized convective flux vector $\hat{\boldsymbol{\Phi}}_c$ is:

$$\hat{\boldsymbol{\Phi}}_c = \hat{E}_c \mathbf{n}_x + \hat{F}_c \mathbf{n}_y \quad (2.1.11)$$

where \hat{E}_c and \hat{F}_c are respectively the x - and y -components of $\hat{\boldsymbol{\Phi}}_c$ and are given by:

$$\hat{E}_c = [\rho u_x \quad \rho u_x^2 + p \quad \rho u_x u_y \quad \rho u_x H]' \quad (2.1.12)$$

$$\hat{F}_c = [\rho u_y \quad \rho u_y u_x \quad \rho u_y^2 + p \quad \rho u_y H]'. \quad (2.1.13)$$

The generalized diffusive flux vector $\hat{\Phi}_d$ is:

$$\hat{\Phi}_d = \hat{E}_d \mathbf{n}_x + \hat{F}_d \mathbf{n}_y \quad (2.1.14)$$

where \hat{E}_d and \hat{F}_d are respectively the x- and y- components of $\hat{\Phi}_d$ and are given by:

$$\hat{E}_d = [0 \quad \tau_{xx} \quad \tau_{xy} \quad u_x \tau_{xx} + u_y \tau_{xy} - q_x]' \quad (2.1.15)$$

$$\hat{F}_d = [0 \quad \tau_{xy} \quad \tau_{yy} \quad u_x \tau_{xy} + u_y \tau_{yy} - q_y]'. \quad (2.1.16)$$

The scalars q_x and q_y are the Cartesian components of the heat flux vector \mathbf{q} . The system of PDEs defined so far and used to determine the solution of two-dimensional problems, has 5 unknowns, namely 2 velocity components and 3 thermodynamic variables. The relationship required to close the system is the perfect gas equation of state. That can be written as:

$$p = (\gamma - 1) \left[\rho E - \frac{1}{2} \rho (u_i u_i) \right]. \quad (2.1.17)$$

Where the symbol γ is the ratio of specific heat and is taken as 1.4.

2.1.1 Arbitrary Lagrangian Eulerian formulation

The formulation just presented is adequate for the solution of fluid dynamic problems with motionless bodies, including unsteady problems, where the unsteadiness originates from phenomena such as vortex shedding. In this study, however, we are also interested

in solving problems with moving bodies. For such problems, the Eulerian approach has to be generalized to include body motion. For this reason the so-called *Arbitrary Lagrangian Eulerian* (ALE) formulation of the NS equations has to be adopted [13]. In the ALE formulation the CV is allowed to move and deform during the considered time interval. Given a time-varying control volume $V(t)$ with boundary $S(t)$, it can be shown that the ALE integral form of the system of the 2D time- dependent NS equations is:

$$\frac{\partial}{\partial t} \left(\int_{V(t)} \hat{U} dV \right) + \oint_{S(t)} (\hat{\Phi}_c - \hat{\Phi}_d) dS = 0. \quad (2.1.18)$$

The generalized convective flux vector $\hat{\Phi}_c$ is:

$$\hat{\Phi}_c = \hat{E}_c \mathbf{n}_x + \hat{F}_c \mathbf{n}_y - \mathbf{v}_b \hat{U}. \quad (2.1.19)$$

The vector \mathbf{v}_b is the velocity of the boundary $S(t)$, and the flux term $-\mathbf{v}_b \hat{U}$ is the boundary velocity contribution to the overall flux balance. When the velocity \mathbf{v}_b is equal to the velocity of the fluid, it can be easily proved that the second term of the mass conservation (2.1.1) becomes zero. When that happens, the mass remains constant and the CV is also a CM. In this case we have the Lagrangian mass conservation equation, $\partial m / \partial t = 0$. The generalized diffusive flux vector $\hat{\Phi}_d$ using the ALE formulation is unchanged from the Eulerian approach.

2.2 Reynolds-Favre averaging

In the case of turbulent flows, the effects of turbulence are often taken into account by averaging the NS equations on the time-scales of turbulence. In 1895 Reynolds proposed the first approximation of turbulent flows [72]. The idea is based on the decomposition of the flow variables into a mean and a fluctuating part. In order to calculate the mean part of the flow variables, there are three well known forms of averaging [2, 19]. The first method is the so-called *time averaging*, which is suitable for stationary turbulence flows. This type of turbulence flow, on average, does not change in time. The second method is

the so called *spatial averaging*, used in turbulent homogeneous flows which are uniform in all directions. The last method is the so called *ensemble averaging*, which depends on experimental data. In this study, time averaging method is used, as the flow is not homogeneous. The derivation of Reynolds-Average Navier-Stokes equations along with the differences between Reynolds averaging and Favre averaging are reported in Appendix A.

2.3 Boussinesq approximation

The Boussinesq approximation [2] forms one of the most important assumptions in turbulence modelling. It arises from the observation that in the turbulent flow, the momentum transfer is dominated by the large energetic turbulent eddies. The Boussinesq hypothesis assumes that the turbulent shear stress is linearly proportional to the mean strain rate, as in the laminar flow. In this way one can write the Favre stress tensor as:

$$\tau_{ij}^F = - \overline{\rho u_i'' u_j''} = 2 \mu_T \left(\tilde{S}_{ij} - \frac{1}{3} \frac{\partial \tilde{u}_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \bar{\rho} K \delta_{ij} \quad (2.3.1)$$

$$\tilde{S}_{ij} = \frac{1}{2} \left[\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right] \quad (2.3.2)$$

where $\tilde{\mathbf{S}}$ is the Favre-averaged strain-rate tensor, the formal definition of which is the same as that of the molecular strain rate tensor. In addition it is found that the total stress tensor $\hat{\tau}_{ij}$ is the sum of the laminar stress tensor τ_{ij} and the so-called *Favre-Reynolds stress tensor* τ_{ij}^F

$$\hat{\tau}_{ij} = \tau_{ij} + \tau_{ij}^F. \quad (2.3.3)$$

The formal definition of the molecular stress tensor τ_{ij} is given by equation (2.1.3). The symbol K in the equation (2.3.1) is the turbulent kinetic energy and the symbol μ_T represents an eddy viscosity parameter, which does not represent a physical characteristic

of the fluid, but is a function of the local flow conditions and the flow history. The total dynamic viscosity μ_t is the sum of a laminar and turbulent component

$$\mu_t = \mu + \mu_T . \quad (2.3.4)$$

Other approximations relative to the Reynolds-Averaged Navier-Stokes equations can be found in *Appendix C*.

2.4 Wilcox's K – ω model

One sees that the RANS equations contain two additional variables with respect to the non-averaged NS equations, namely the eddy viscosity μ_T and the turbulent kinetic energy K . These two variables establish a strong coupling with the 2 PDEs associated with the $K - \omega$ turbulence model. The $K - \omega$ turbulence model consists of two transport equations: one PDE describing the convection, diffusion, creation and destruction of the turbulent kinetic energy, and one PDE describing the evolution of the specific dissipation rate, the second turbulent variable of the $K - \omega$ model. The turbulent kinetic energy K has dimensions $m^2 s^{-2}$ and the specific dissipation rates ω has dimension s^{-1} . In the present work, the version of the $K - \omega$ turbulence model reported by Wilcox [19] is used:

turbulent mixing energy,

$$\begin{aligned} \frac{\partial}{\partial t}(\rho K) + \frac{\partial}{\partial x_j}(\rho u_j K) \\ = \tau_{ij}^F \frac{\partial u_i}{\partial x_j} - \beta^* \rho \omega K + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_K \mu_T) \frac{\partial K}{\partial x_j} \right] \end{aligned} \quad (2.4.1)$$

specific dissipation rate,

$$\begin{aligned} \frac{\partial}{\partial t}(\rho \omega) + \frac{\partial}{\partial x_j}(\rho u_j \omega) \\ = \frac{\gamma \rho}{\mu_T} \tau_{ij}^F \frac{\partial u_i}{\partial x_j} - \beta \rho \omega^2 + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_\omega \mu_T) \frac{\partial \omega}{\partial x_j} \right]. \end{aligned} \quad (2.4.2)$$

The eddy viscosity μ_T at any position in the computational domain is determined by using the current values of K and ω

$$\mu_T = \gamma^* \frac{\rho K}{\omega}. \quad (2.4.3)$$

In Equation (2.4.1), the term $\tau_{ij}^F \frac{\partial u_i}{\partial x_j}$ leads to production of turbulent kinetic energy, whereas the term $\beta^* \rho \omega K$ leads to destruction of the same variable and these are the so called *source terms* \hat{S} . Diffusion of the turbulent kinetic energy is instead enforced by the term $\frac{\partial}{\partial x_j} \left[(\mu + \sigma_K \mu_T) \frac{\partial K}{\partial x_j} \right]$. Following Wilcox [15], it is appropriate to notice that the Pressure Diffusion and Pressure Dilatation terms are not included in equations (2.4.2) and (2.4.3) as they are simply ignored because of the density fluctuations. Summarizing the source terms of all equations one can write the following equation:

$$P_K = \tau_{ij}^F \frac{\partial u_i}{\partial x_j}, \quad P_\omega = \frac{\gamma \rho}{\mu_T} P_K \quad (2.4.4)$$

$$D_K = \beta^* \rho \omega K, \quad D_\omega = \beta \rho \omega^2. \quad (2.4.5)$$

The production terms P_K and P_ω can also be written as:

$$P_K = \mu_T P_d - \frac{2}{3} (\nabla \cdot \mathbf{u}) \rho K \quad (2.4.6)$$

$$P_\omega = \gamma \rho P_d - \frac{\gamma \rho}{\mu_T} \frac{2}{3} (\nabla \cdot \mathbf{u}) \rho K \quad (2.4.7)$$

$$P_d = 2 \left(\underline{\mathbf{S}} - \frac{1}{3} \nabla \cdot \mathbf{u} \right) \nabla \mathbf{u}. \quad (2.4.8)$$

It can be shown that the production term P_d is always positive. Thus the source terms of the turbulent kinetic energy equation include a term which is always positive (production term proportional to P_d), a term which is always negative (destruction term D_K) and a term which is positive or negative depending on the sign of $\nabla \cdot \mathbf{u}$. Similarly to the K equation case, the source terms of the ω equation also include a term which is always positive (production term proportional to P_d), a term which is always negative (destruction term D_ω), and a term which is positive or negative depending on the sign of $\nabla \cdot \mathbf{u}$. As highlighted in past studies [60, 59] on the numerical integration of the $K - \omega$ turbulence model equations, the identification of positive and negative source terms is of crucial importance. When one uses convergence acceleration methods such as explicit multigrid methods, the adoption of different numerical treatments for the positive and negative source terms improves the convergence rate of the solution process. More detail can be found in the chapter 5: *numerical integration*.

Finally in order to close the system of equations the turbulent coefficients have to be specified. The $K - \omega$ model has six closure coefficients. Following Wilcox [19], these coefficients can be derived by the following relations. Firstly, in order to be consistent with the experiments that have taken place by Townsend [73] it was found that $\frac{\beta}{\beta^*} = \frac{5}{6}$. Secondly, in order to be consistent with the law of the wall [74] $\gamma = \frac{\beta}{\beta^*} - \frac{\sigma_\omega \kappa^2}{\sqrt{\beta^*}}$ for the inner layer of a constant-pressure boundary layer and in order to reproduce a correct solution (with $\kappa = 0.41$ the Von Karman constant), we conclude that β^* and σ_ω are equal to 0.09 and 0.5 respectively. The values of the constants defining the $K - \omega$ turbulence model are:

$$[\beta \quad \beta^* \quad \gamma \quad \gamma^* \quad \sigma_K \quad \sigma_\omega] = \left[\frac{3}{40} \quad \frac{9}{100} \quad \frac{5}{9} \quad 1.0 \quad 0.5 \quad 0.5 \right]. \quad (2.4.9)$$

More details about the derivation of the turbulent coefficients or the complete derivation of the $K - \omega$ turbulence model as well as the approximations adopted in this process can be found in the book of Wilcox [15].

2.5 Menter's shear stress transport model

Menter [40] introduced two new variants of the $K - \omega$ turbulence model, the so called *Baseline* (BSL) model and the so called *Shear Stress Transport* (SST) model. The design of the BSL model is based on the original $K - \omega$ model of Wilcox and the standard $K - \varepsilon$ model [18, 75] rewritten using the ω rather than the ε variable. In the near wall region the model uses the robust and fairly accurate $K - \omega$ model. Outside the boundary layer, the BSL changes gradually to the standard $K - \varepsilon$ model. The advantage of using the $K - \varepsilon$ model outside shear layers is that such a model is substantially less sensitive to the freestream turbulence data than the standard $K - \omega$ model. Conversely, the $K - \omega$ model is more accurate than the $K - \varepsilon$ model in the prediction of boundary layers. For free shear layers the new model is virtually similar to the $K - \varepsilon$ model as well. Using the same turbulent transport equations of the BSL model, Menter introduced a second variant of the $K - \omega$ model named SST model. Such a model has improved predictive capabilities of separated flows in adverse pressure gradient with respect to the BSL model. The SST model is based on Bradshaw's assumption that the principal shear-stress is linearly proportional to the turbulent kinetic energy. The two transport equations of Menter [40] for the turbulent mixing energy K and the specific dissipation rate ω are respectively:

$$\begin{aligned} \frac{\partial}{\partial t}(\rho K) + \frac{\partial}{\partial x_j}(\rho u_j K) \\ = \tau_{ij}^F \frac{\partial u_i}{\partial x_j} - \beta^* \rho \omega K + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_K \mu_T) \frac{\partial K}{\partial x_j} \right] \end{aligned} \quad (2.5.1)$$

$$\begin{aligned} \frac{\partial}{\partial t}(\rho \omega) + \frac{\partial}{\partial x_j}(\rho u_j \omega) \\ = \frac{\gamma \rho}{\mu_T} \tau_{ij}^F \frac{\partial u_i}{\partial x_j} - \beta \rho \omega^2 + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_\omega \mu_T) \frac{\partial \omega}{\partial x_j} \right] \\ + CD_{term} \end{aligned} \quad (2.5.2)$$

where

$$CD_{term} = 2 \rho (1 - F_1) \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial K}{\partial x_j} \frac{\partial \omega}{\partial x_j}. \quad (2.5.3)$$

The new definition of eddy viscosity is:

$$\mu_T = \frac{\alpha_1 \rho K}{\max(\alpha_1 \omega, \Omega F_2)} \quad (2.5.4)$$

where α_1 and Ω are a constant and the modulus of the vorticity respectively. The variables F_1 and F_2 are blending functions used to combine the $K - \varepsilon$ and $K - \omega$ models. Their definitions are:

$$F_1 = \tanh(\arg_1^4) \quad (2.5.5)$$

$$\arg_1 = \min \left[\max \left(\frac{\sqrt{K}}{\beta^* \omega d}, \frac{500 \mu}{\rho \omega d^2} \right), \frac{4 \rho \sigma_{\omega 2} K}{CD_{K\omega} d^2} \right] \quad (2.5.6)$$

$$CD_{K\omega} = \max \left(2 \rho \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial K}{\partial x_j} \frac{\partial \omega}{\partial x_j}, 10^{-20} \right) \quad (2.5.7)$$

$$F_2 = \tanh(\arg_2^2) \quad (2.5.8)$$

$$\arg_2 = \max \left(\frac{2 \sqrt{K}}{\beta^* \omega d}, \frac{500 \mu}{\rho \omega d^2} \right) \quad (2.5.9)$$

where d is the distance to the nearest wall. Equation (2.5.4) selects the minimum eddy viscosity between the value of the standard $K - \omega$ model and the Bradshaw's assumption (the principal shear stress is proportional to the turbulent kinetic energy). The blending function F_2 limits Bradshaw's assumption within the boundary layer region, whereas the rest of the computational domain makes use of the original definition of the turbulent viscosity. The source term of the ω -equation features the additional cross-diffusion term $CD_{K\omega}$ compared with the corresponding source term in the original $K - \omega$ model. This cross-diffusion term can be positive or negative.

Finally in order to complete the definition of the SST turbulence model, the turbulent coefficients have to be specified. In the SST model, there are two sets of coefficients, which are combined using the blending function Φ . The constants for set 1 are $\beta_1, \sigma_{K1}, \sigma_{\omega 1}$ and γ_1 , and for set 2 are $\beta_2, \sigma_{K2}, \sigma_{\omega 2}$ and γ_2 which are reported in equations (2.5.12) and (2.5.13) below. Other coefficients used by the model are $\beta^* = 0.09$, $\kappa = 0.41$ and $\alpha_1 = 0.31$. The equation used to calculate the coefficients γ_1 and γ_2 is:

$$\gamma_1 = \frac{\beta_1}{\beta^*} - \frac{\sigma_{\omega 1} \kappa^2}{\sqrt{\beta^*}} \quad \gamma_2 = \frac{\beta_2}{\beta^*} - \frac{\sigma_{\omega 2} \kappa^2}{\sqrt{\beta^*}}. \quad (2.5.10)$$

Using the blending function Φ defined by equation (2.5.11), one can compute the coefficients β , σ_K , σ_ω and γ for any area of the computational domain

$$\Phi = F_1 \Phi_1 + (1 - F_1) \Phi_2 \quad (2.5.11)$$

$$\text{set 1: } [\beta_1 \quad \sigma_{K1} \quad \sigma_{\omega 1} \quad \gamma_1] = [0.0750 \quad 0.85 \quad 0.500 \quad 0.55317] \quad (2.5.12)$$

$$\text{set 2: } [\beta_2 \quad \sigma_{K2} \quad \sigma_{\omega 2} \quad \gamma_2] = [0.0828 \quad 1.00 \quad 0.856 \quad 0.44035]. \quad (2.5.13)$$

As mentioned above, both models, the $K - \omega$ model of Wilcox and the SST model of Menter are two-equation eddy viscosity turbulence models, which require the solution of two additional transport equations, one for the turbulent kinetic energy, and one for the specific turbulence dissipation rate. The SST model is an extension of the original $K - \omega$ model of Wilcox, developed to

- greatly reduce the sensitivity of the original $K - \omega$ model to the somewhat arbitrary value of the specific dissipation rate enforced at the farfield boundaries of the computational domain, and
- enhance the solution accuracy of turbulent flows by improving the capability of the $K - \omega$ model to predict the onset and amount of separation in adverse pressure flows.

The numerical results of [40] and later comparative analyses performed for internal [76] and external [77] turbulent flows highlight that the SST model achieves both objectives. Other extensions of the original $K - \omega$ model aiming to achieve the same objectives have also been developed by Wilcox [78]. One-equation eddy viscosity turbulence models [17, 79] require the solution of only one transport equation. Historically, the development of one-equation models has followed that of two-equation models, and its main motivation has been to reduce the computational cost associated with two-equation

models while limiting the accuracy loss with respect to flow simulations based on two-equation models. Several comparative analyses of realistic turbulent flow problems using both one- and two-equation turbulence modelling highlight that, though the results of modern two-equation models often appear to be closer to experimental data, the solution differences between one- and two-equation models are indeed often small. A wider review of turbulence modelling is beyond the scope of this study, and the interested reader is referred to the review article [80] for a wider overview of the present state, challenges and needs of turbulence modelling for engineering applications, and long term projections for the progress in this area.

2.6 Boundary conditions

Appropriate *boundary conditions* (BCs) must be imposed at the far field and solid wall boundaries. The results of turbulent flow simulations strongly depend on the models used to define the boundary data (e.g. calculation of ω at wall boundaries), and, in some cases, directly on user-given data (e.g. value of ω at far field boundaries).

2.6.1 Far field

In the case of external flow problems, the far-field BCs for the mean flow equations are based on suitable combinations of one dimensional (1D) Riemann invariants and user-given free-stream data, namely sound speed, entropy and velocity components. Using all the above free stream values one can compute the conservative variables (ρ , ρu_i , ρu_j , ρE) at the far field boundary. The way in which the user-given free stream boundary data are combined depends on whether the flow is subsonic or supersonic and whether the flow is entering (inflow boundary) or leaving (outflow boundary) the domain. The complete formulation of the characteristic far field boundary conditions can be found in the paper of Jameson [81].

In the case of internal flow problems, the subsonic inflow BCs for the mean flow equations are constructed by extrapolating the outgoing 1D characteristic variable and enforcing user-given values of total temperature, total pressure and flow direction. For supersonic inflow conditions, all thermodynamic and kinematic variables are imposed. The subsonic outflow conditions are determined by enforcing a user-given value of the static pressure and extrapolating the remaining variables from the interior of the domain with suitable procedures, whereas the supersonic outflow conditions are obtained by extrapolating all flow variables from the interior. Several variants of the far-field BCs for internal problems, including one which preserves the nominal second order of the numerical scheme for problems with strong flow gradients on the far-field boundaries, are reported in the article [69].

On far field boundaries at which an inflow regime occurs, the two equations of the $K - \omega$ model require the prescription of the free-stream values of K and ω . Making use of equation (2.5.4), one can then calculate the turbulent viscosity. It is also possible to specify at the boundary the values of K and μ_T and then use the equation (2.5.4) to calculate the free stream value of ω . Only if the flow is entering the domain then the free stream values of the two selected turbulent variables are enforced at the farfield boundary. On the other hand if the flow is leaving the domain, then the values of K and ω are extrapolated from the interior.

2.6.2 Solid wall

At solid wall boundaries, the static pressure is extrapolated from the interior and the density is computed from static pressure using the adiabatic condition of the wall. Both velocity components are set equal to the wall velocity, which is zero only in the case of problems with motionless grids. The two equations of the $K - \omega$ or SST turbulence model require two wall conditions. One is that the turbulent kinetic energy K must be zero at the wall boundary and the other condition requires specifying the value of ω at the wall. Two models are implemented to determine the value of ω at wall boundaries.

The first one was proposed by Wilcox [15] and allows roughness effects to be taken into account. According to this model the dissipation rate ω at the wall is:

$$\omega_w = \frac{\mathbf{u}_\tau^2}{\nu_w} S_R = S_R \left(\frac{\partial \mathbf{u}_{||}}{\partial \mathbf{n}} \right)_w \quad (2.6.1)$$

where $u_{||}$ denotes the flow velocity component parallel to the wall boundary. The symbol \mathbf{n} denotes the local coordinate normal to the wall and ν_w is the kinematic viscosity at the wall. The symbol \mathbf{u}_τ is the friction velocity and its definition is:

$$\mathbf{u}_\tau = \sqrt{\frac{\tau_w}{\rho}}. \quad (2.6.2)$$

The symbol S_R denote a surface roughness parameter, varying between $0 \leq S_R \leq \infty$. The value $S_R = 0$ corresponds to a rough wall, and $S_R \rightarrow \infty$ for a smooth wall. The parameter S_R depends on the average height of the sand-grain roughness elements k_R . Note that k_R has the dimension of a length, and it is customary to use k_R^+ , the non-dimensional counterpart of k_R obtained by using the reference length ν_w / u_τ . For flow over very rough surfaces [15] the following relationship was found:

$$k_R^+ = \frac{u_\tau k_R}{\nu_w} = \frac{k_R \sqrt{\rho_w \tau_w}}{\mu_w} \quad (2.6.3)$$

where τ_w is the shear stress at the wall given:

$$\tau_w = \mu_w \left(\frac{\partial u_{||}}{\partial \mathbf{n}} \right)_w \quad (2.6.4)$$

where μ_w is the dynamic viscosity at the wall. Analytically, for a smooth wall one has $k_R^+ = k_R = 0$. From the above definitions and following Wilcox [15], it is possible to determine a value for S_R using:

$$S_R = \left(\frac{50}{k_R^+} \right)^2 \quad k_R^+ < 25 \quad (2.6.5)$$

$$S_R = \left(\frac{100}{k_R^+} \right) \quad k_R^+ \geq 25. \quad (2.6.6)$$

Note that equation (2.6.5) corresponds to the case of smooth walls, and also that, numerically, one cannot use the analytical condition of smooth wall $k_R^+ = 0$, since this would yield overflow when computing S_R . Indeed, the solver used in this study sets $k_R^+ = 5$ in the case of perfectly smooth walls ($k_R = 0$). More recent studies, however, propose $S_R = 2500$ for a smooth wall [82]. The impact of the choice of S_R on the computed viscous drag is analysed in chapter 7.

The second formulation to determine the value of ω at a smooth wall was proposed by Menter [40]. According to this author, the value of ω at a smooth wall can be obtained using the equation:

$$\omega_w = 60 \frac{\mu_w}{\rho_w \beta (\Delta y)^2} \quad (2.6.7)$$

where Δy is the distance to the first point off the wall from the wall itself and β is a turbulent coefficient. Menter reports that equation (2.6.7) is equivalent to equation (2.6.1) written for a smooth wall provided that the nondimensionalised wall distance from the wall y^+ is equal to or smaller than 3. The two different options that we have considered are examined in detail in the next subsection as they have a strong impact on the numerical solution.

2.6.3 Comparison of Wilcox and Menter wall boundary conditions

In the previous subsection two different wall boundary conditions for the calculation of the specific dissipation rate ω have been presented. In this subsection, the two BCs are compared with an emphasis on the impact of either model on the accuracy of the numerical solution and the stability of the numerical integration. Starting from Wilcox wall boundary condition (2.6.1) for smooth walls ($k_R^+ = 5$), it follows that:

$$\omega_w = \frac{100 u_\tau^2}{\nu_w}. \quad (2.6.8)$$

Inserting the expression of the Law of the Wall [74] provided by equation (2.6.9) into the wall boundary condition proposed by Menter, i.e equation (2.6.7), and using the given value $\beta = 3/40$, yields equation (2.6.10)

$$y^+ = \frac{u_\tau y \rho_w}{\mu_w} \quad (2.6.9)$$

$$\omega_w = \frac{800 u_\tau^2}{\nu_w (\Delta y^+)^2}. \quad (2.6.10)$$

Inserting $\Delta y^+ \approx 3$ into equation (2.6.10) one finds that Wilcox's smooth wall BC and Menter's wall BC yield indeed comparable values of ω , as stated by Menter. In real CFD simulations, however, the value of y^+ at the wall varies significantly along the wall. Such a variation is not accounted for by Menter's wall BC, which assumes a value of y^+ constant and equal to 3. Numerical experiments performed within the work of this thesis have shown that when the mean value of y^+ is close to 3, the computed solutions obtained by using either Wilcox's or Menter's wall BCs are similar, despite the local variations of y^+ . When the mean value of y^+ differs significantly from 3, however, the solutions obtained with either wall BCs differ significantly. Numerical experiments reported in chapter 7 also show that Menter's wall BC for problems featuring values of y^+ between 1 and 3 close to viscous wall boundaries yields computed solutions which are in better agreement with available experimental and theoretical data than computed solutions obtained using Wilcox's BC. This highlights the necessity of modifying the constant 100 appearing in Wilcox's condition (equation 2.6.8). Wilcox's wall BC features a higher level of flexibility with respect to Menter's BC both because the former condition depends on the local value of the viscous stress at the wall (which is more realistic than depending from the distance of the first point off the wall), and because Wilcox's wall can also take into account wall roughness [19] and variation of this parameters [82].

From a numerical viewpoint, the use of Wilcox's wall BC structure is also preferable. This is because Menter's value of ω at the wall depends on the minimum distance from the wall, a parameter affected by the local grid refinement. Moreover, even when the grid is sufficiently refined, Menter's value of ω at the wall keeps changing when the minimum wall distance is further reduced. This feature may prevent a grid independent solution from being attained due to the dependence of the value of ω at the wall on the wall distance. Wilcox's wall BC structure, conversely, does not have this drawback, because

the value of ω at the wall depends on the wall viscous stress, and such parameter does no longer vary with further grid refinement once a grid independent solution has been obtained. To prevent the ‘uncontrolled’ grid dependence of numerical solutions obtained using Menter’s wall BC, some authors (e.g. [83]) have adopted grid- and flow- dependent bounds of the wall values of ω provided by equation (2.6.7). Conversely, the use of equation (2.6.1) yields values of ω which, above a minimum level of grid refinement, are grid independent without requiring additional constraints.

The use of Wilcox’s wall BC also has another possible advantage over Menter’s wall BC when using the multigrid algorithm (chapter 6) for convergence acceleration. The use of such a condition, in fact, would lead to values of ω at the wall, which are similar on all grid levels, provided that also the coarser grids have sufficient spatial refinement. Using Menter’s BC, conversely, would always yield significantly values of ω at the wall on all grids, since the wall distance is doubled every time the solver moves from a grid level to coarser level. These strong variations of ω at the wall on the various grid levels may spoil the convergence of the multigrid solver, and the problem may be alleviated only by introducing additional constraints, thus increasing the complexity of the computer code

2.7 System of URANS and SST equations

In section 2.2, the RANS equations have been obtained by averaging the time-dependent Navier-Stokes equations on the turbulence time-scales. For steady problems the averaged time derivative is zero. For time-dependent problems the RANS equations are augmented with new time-derivatives accounting for time-variations on the characteristic time-scales associated with the engineering problem at hand. Thus, for time-dependent turbulent problems we solve the so-called *Unsteady Reynolds Average Navier Stokes* (URANS) equations. This chapter is concluded by the presentation of the non-dimensional URANS and SST equations in divergence and integral form. The derivation of the non-dimensional form of the RANS equations can be found in *Appendix D* whereas a compact divergence form and a quasi-linear form of the equations are reported in *Appendix E* and *F* respectively.

2.7.1 Divergence form

The divergence form of the 2D time-dependent RANS equations coupled to the two transport equations of the *SST* turbulence model in non-dimensional form are:

mass conservation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j} (\rho u_j) = 0, \quad (2.7.1)$$

momentum conservation:

$$\frac{\partial}{\partial t} (\rho u_i) + \frac{\partial}{\partial x_j} (\rho u_j u_i) = - \frac{\partial p}{\partial x_j} + \frac{M_\infty}{Re} \frac{\partial \hat{\tau}_{ij}}{\partial x_j}, \quad i = 1, 2, \quad (2.7.2)$$

mean energy conservation:

$$\begin{aligned} \frac{\partial}{\partial t} (\rho E) + \frac{\partial}{\partial x_j} (\rho u_j H) \\ = \frac{M_\infty}{Re} \frac{\partial}{\partial x_j} \left[u_i \hat{\tau}_{ij} + (\mu + \sigma_K \mu_T) \frac{\partial K}{\partial x_j} - \hat{q}_j \right], \end{aligned} \quad (2.7.3)$$

turbulent kinetic energy conservation:

$$\begin{aligned} \frac{\partial}{\partial t} (\rho K) + \frac{\partial}{\partial x_j} (\rho u_j K) \\ = \frac{M_\infty}{Re} \tau_{ij}^F \frac{\partial u_i}{\partial x_j} - \beta^* \rho \omega K \\ + \frac{M_\infty}{Re} \frac{\partial}{\partial x_j} \left[(\mu + \sigma_K \mu_T) \frac{\partial K}{\partial x_j} \right], \end{aligned} \quad (2.7.4)$$

dissipation rate ω conservation:

$$\begin{aligned} \frac{\partial}{\partial t}(\rho \omega) + \frac{\partial}{\partial x_j}(\rho u_j \omega) &= \frac{\gamma \rho}{\mu_T} \tau_{ij}^F \frac{\partial u_i}{\partial x_j} - \beta \rho \omega^2 \\ &+ \frac{M_\infty}{Re} \frac{\partial}{\partial x_j} \left[(\mu + \sigma_\omega \mu_T) \frac{\partial \omega}{\partial x_j} \right] + CD_{term} \end{aligned} \quad (2.7.5)$$

2.7.2 Integral form

Given a control volume $V(t)$ with boundary $S(t)$, the integral form of the 2D time-dependent RANS equations coupled to the two transport equations of the *SST* turbulence model in non-dimensional form are:

$$\frac{\partial}{\partial t} \int_{V(t)} \hat{U} dV + \oint_{S(t)} (\hat{\Phi}_c - \hat{\Phi}_d) dS = \int_{V(t)} \hat{S} dV \quad (2.7.6)$$

where analytically the equations can be written separately as:

mass conservation

$$\frac{\partial}{\partial t} \int_{V(t)} \rho dV + \oint_{S(t)} \rho (\mathbf{u} \cdot \mathbf{n}) dS = 0, \quad (2.7.7)$$

momentum conservation

$$\begin{aligned} \frac{\partial}{\partial t} \int_{V(t)} \rho \mathbf{u} dV \\ + \oint_{S(t)} \rho \mathbf{u} (\mathbf{u} \cdot \mathbf{n}) dS \\ = - \oint_{S(t)} p \mathbf{n} dS + \frac{M_\infty}{Re} \oint_S \hat{\boldsymbol{\tau}} \cdot \mathbf{n} dS, \end{aligned} \quad (2.7.8)$$

mean energy conservation

$$\begin{aligned}
& \frac{\partial}{\partial t} \int_{V(t)} \rho E dV \\
& + \oint_{S(t)} \rho H (\mathbf{u} \cdot \mathbf{n}) dS = \frac{M_\infty}{Re} \oint_{S(t)} (\hat{\mathbf{t}} \cdot \mathbf{u} - \hat{\mathbf{q}}) \cdot \mathbf{n} dS \quad (2.7.9) \\
& + \frac{M_\infty}{Re} \int_{V(t)} (\mu + \sigma_K \mu_T) \nabla K dV,
\end{aligned}$$

turbulent kinetic energy conservation:

$$\frac{\partial}{\partial t} \int_{V(t)} \rho K dV + \oint_{S(t)} \rho K (\mathbf{u} \cdot \mathbf{n}) dS = \int_{V(t)} (P_K - D_K) dV, \quad (2.7.10)$$

dissipation rate ω conservation:

$$\begin{aligned}
& \frac{\partial}{\partial t} \int_{V(t)} \rho \omega dV \\
& + \oint_{S(t)} \rho \omega (\mathbf{u} \cdot \mathbf{n}) dS = \int_{V(t)} (P_\omega - D_\omega + CD_{term}) dV. \quad (2.7.11)
\end{aligned}$$

Chapter 3

HARMONIC BALANCE FORMULATION OF THE GOVERNING EQUATIONS

At present the most common method to solve numerically the URANS and the turbulence model equations is the time-domain approaches, whereby the time-dependent solution is time-marched from an initial state until a user-given physical time. In the case of periodic flows, this method often requires a long wall clock time due to the fact that several periods have to be simulated before a periodic solution is achieved. A computationally cheaper alternative is to solve the URANS and the turbulence model equations by using a Frequency-Domain formulation. As discussed in chapter 1, the use of Frequency-Domain methods for computing unsteady periodic flows has been shown to reduce the run-time of the CFD simulation by one or more orders of magnitude with respect to the case in which the conventional time-marching solution is used. The *harmonic balance* (HB) method is an efficient nonlinear frequency-domain method for the solution of nonlinear periodic problems defined by a single *ordinary differential equation* (ODE) or a system of ODEs. After providing the general definition of the HB solution process, this chapter presents the HB solution of a single ODE, the Duffing oscillator. This example is discussed to further clarify the general HB solution process and also highlight the differences between two mathematical implementations of the method, the classical harmonic balance method and the *high dimensional harmonic balance* (HDHB) method. A comparison of these two methods is provided to explain why HDHB route is adopted for solving the system of ODEs resulting from the space-discretisation of the time-dependent CFD equations. Finally the formulation of the HDHB formulation of the URANS and the turbulence model equations is presented.

3.1 Harmonic balance

At present the most common method to solve numerically the URANS and the turbulence model equations is to solve this system of equations in the time-domain (TD). The solution process starts from a user-given initial time and the solution is time-marched until a user-given final time. In general this method often requires long wall-clock times. In the case of time-periodic flows, such as oscillating wings or vertical axis wind turbines, one is often interested only in the periodic solution, rather than the whole transient from an arbitrary initial solution to the sought periodic state. In these circumstances, the TD solution approach is inefficient because several periods have to be simulated before a fully developed periodic solution is achieved. When multiple simulations of this type are required, like in the case of design optimisation, the numerical burden associated with the unnecessary solution of the transient phase becomes enormous. In order to achieve this periodic solution within a reasonable time, a new technology has been recently developed which exploits the periodic character of the sought flow solution by solving the URANS equations in the *Frequency-Domain* (FD). In this way one can define a harmonic form of the unsteady flow problem and solve the unsteady problem as a steady system.

Firstly in 1999 Verdon et al. [84] and later in 2000 Clark et al. [85] developed a time-linearization technique whereby the unsteady flow field is decomposed into two parts, a steady nonlinear mean flow and a small harmonic perturbation. Using this modification, the computational cost needed to complete the calculations is reduced by more than one order of magnitude, compared to conventional time-marching algorithms. The disadvantage of this technology is that it cannot capture strong nonlinear flow features like shock discontinuities. Another FD technique was developed at the same time by Ning and He [86] named time-averaged nonlinear harmonic technique which can compute correctly flow shock discontinuities. Ning and He [86] followed the same procedure of Verdon et al. [84] which splits the unsteady flow field into two parts but instead of using a steady flow field they used a time-averaged flow field as the basis of the harmonic perturbations.

In 2002, Hall, Thomas and Clark [67] generalised the approach of Ning and He, and developed a new nonlinear frequency domain technique capable of solving arbitrarily strong nonlinear periodic flows. This method, called *harmonic balance* (HB), uses truncated Fourier series for representing the nonlinear periodic variation of each

conservation variable. The first term of the Fourier series represents the time-averaged flow-field solution over one period, while all the other terms of the series represent the harmonic components of the unsteady flow under investigation. The frequency used in each term of the Fourier series is a constant user-given input and this is the fundamental excitation frequency of the engineering problem at hand. Hall et al. [67] found that making use of several terms of the Fourier series (harmonics) improves the solution of the aerodynamic forces. Inserting the Fourier representations into the original equations will yield a system of coupled partial differential equations where the unknowns are the Fourier coefficients of the series. The process of matching or balancing harmonics of the same order yields a system, the size of which equals to the number of the starting nonlinear PDEs, times the number of retained Fourier harmonics. The same authors [67] introduced the *High-Dimensional Harmonic Balance* (HDHB) method, which is an implementation of the harmonic balance technology more amenable to the solution of the turbulent NS equations. The HDHB approach will be explained in detail in the following sections. This system resulting from the space-discretization of the HDHB turbulent RANS equations can be solved using standard pseudo time-marching CFD methods. Maple et al. [87] following the work of Hall et al. [67] developed a similar HB solver to solve the Euler equations with the ability to change the number of harmonics at each grid point depending on the local flow conditions. In the last two years researchers started using the HB method to solve the RANS equations and the turbulence model equations for closure. L. He [88] used the one-equation Spalart-Allmaras turbulence model to calculate the flow around turbo-machinery blades with separation, while Corral and Crespo [89] used the $K - \omega$ turbulence model of Wilcox [19]. In the last ten years many other studies have been done using the HB approach [90, 91, 92, 93, 94, 95, 96].

The computational cost of the HB method depends on the size of the grid and the number of harmonics to be used. As one will see later in chapter 8, for a given turbulent flow field without discontinuities, a small number of harmonics is needed to compute accurately the aerodynamic forces and it was found that the HB method was at least 8 times faster compared to standard time-marching techniques. On the other hand when the flow field has strong discontinuities more harmonics are required to achieve the desired accuracy. Many researchers have developed the HB method and they have found a significant reduction of the computational cost [97, 98, 99, 100].

3.2 Harmonic balance integration: ODE example

Duffing's oscillator is a nonlinear ODE which differs from the linear equation defining the forced response of mass-dashpot-spring system because of a nonlinear term proportional to the third power of the mass displacement x . The equation that describes the time-dependent forced motion of this nonlinear oscillator is:

$$m \ddot{x} + d \dot{x} + k x + a x^3 = F \sin(\omega t) \quad (3.2.1)$$

where m is the mass, d is the coefficient of the linear damping term, k is the stiffness of the oscillator, and a is the constant coefficient of the nonlinear cubic term. The symbol F denotes the amplitude of the harmonic forces applied to the system, ω is the excitation frequency and t is the time. The natural frequency ω_0 and the nondimensionalized damping coefficient ζ of the linear oscillator obtained by neglecting the nonlinear term are given respectively by:

$$\omega_0 = \sqrt{\frac{k}{m}} \quad (3.2.2)$$

$$\zeta = \frac{d}{2m\omega_0}. \quad (3.2.3)$$

In order to simplify the notation in the solution process presented below, it is convenient to nondimensionalize equation (3.1.1) using the following expressions:

$$t = \frac{\tau}{\omega_0} \quad (3.2.4)$$

$$\omega = \tilde{\omega} \omega_0 \quad (3.2.5)$$

$$x = \tilde{x} h \quad (3.2.6)$$

where the τ , $\tilde{\omega}$ and \tilde{x} are the non-dimensional time, frequency and unknown function, respectively. The expression of h is:

$$h = \sqrt{\frac{k}{a}}. \quad (3.2.7)$$

Using the expression above, the non-dimensional form of equation (3.2.1) is found to be:

$$\tilde{x}'' + 2 \zeta \tilde{x}' + \tilde{x} + \tilde{x}^3 = \tilde{F} \sin(\tilde{\omega} \tau) \quad (3.2.8)$$

where the expression of the nondimensionalized external force amplitude is:

$$\tilde{F} = \frac{F}{k h} = \frac{F}{k} \sqrt{\frac{a}{k}} \quad (3.2.9)$$

3.2.1 Classical harmonic balance

Using the same nomenclature introduced in [101], the standard implementation of the HB method is defined as classical HB here too. This method was successfully implemented by Virgin [102] to solve the Duffing's oscillator problem using only *one harmonic* (HB1). Moreover Donescu [103] implemented the classical HB method to solve the same problem using more harmonics. In this subsection the solution steps of the same ODE using the classical method with one harmonic are reported in great detail.

In general, the HB calculation of the periodic solution of a nonlinear ODE like that associated with the Duffing's oscillator starts by expressing the sought periodic solution as a truncated Fourier series retaining N_H complex harmonics. The form of such a truncated Fourier series is:

$$x(t) = \hat{x}_0 + \sum_{n=1}^{N_H} (\hat{x}_{2n-1} \cos(n \omega t) + \hat{x}_{2n} \sin(n \omega t)) \quad (3.2.10)$$

where ω is the known fundamental frequency. The term \hat{x}_0 is the mean value of the sought periodic solution, whereas the symbols \hat{x}_{2n-1} and \hat{x}_{2n} , for $n = 1, 2, \dots, N_H$ contain the real and imaginary parts of the N_H complex harmonics retained in the presentation of the

sought periodic solution. One can also write the expressions of the time-derivatives appearing in the ODE at hand as truncated Fourier series. In the case of the Duffing's oscillator only the first and second time-derivatives of the unknown solution is needed. The Fourier series of the first and second derivatives are respectively:

$$\dot{x}(t) = \sum_{n=1}^{N_H} (-n \omega \hat{x}_{2n-1} \sin(n \omega t) + n \omega \hat{x}_{2n} \cos(n \omega t)) \quad (3.2.11)$$

$$\begin{aligned} \ddot{x}(t) = \sum_{n=1}^{N_H} & (-n^2 \omega^2 \hat{x}_{2n-1} \cos(n \omega t) \\ & - n^2 \omega^2 \hat{x}_{2n} \sin(n \omega t)). \end{aligned} \quad (3.2.12)$$

The crucially important point is that all nonlinear terms appearing in the ODE at hand must also be expressed as truncated Fourier series. In the case of the Duffing's oscillator, the Fourier representation of the nonlinear cubic term is:

$$x^3(t) = \hat{r}_0 + \sum_{n=1}^{N_H} (\hat{r}_{2n-1} \cos(n \omega t) + \hat{r}_{2n} \sin(n \omega t)) \quad (3.2.13)$$

where \hat{r}_0 is the mean value of the cubic term, \hat{r}_{2n-1} is the real part of the n^{th} complex harmonic of the nonlinear cubic term, and \hat{r}_{2n} is the imaginary part of the n^{th} complex harmonic of the nonlinear cubic term. Their expressions can be defined as:

$$\begin{aligned} \hat{r}_0 = \frac{\omega}{2\pi} \int_0^{\frac{2\pi}{\omega}} & \left(\hat{x}_0 \right. \\ & + \sum_{n=1}^{N_H} (\hat{x}_{2n-1} \cos(n \omega t) \\ & \left. + \hat{x}_{2n} \sin(n \omega t)) \right)^3 dt \end{aligned} \quad (3.2.14)$$

$$\begin{aligned} \hat{r}_{2n-1} = & \frac{\omega}{\pi} \int_0^{\frac{2\pi}{\omega}} \left(\hat{x}_0 \right. \\ & + \sum_{n=1}^{N_H} \left(\hat{x}_{2n-1} \cos(n \omega t) \right. \\ & \left. \left. + \hat{x}_{2n} \sin(n \omega t) \right) \right)^3 \cos(n \omega t) dt \end{aligned} \quad (3.2.15)$$

$$\begin{aligned} \hat{r}_{2n} = & \frac{\omega}{\pi} \int_0^{\frac{2\pi}{\omega}} \left(\hat{x}_0 \right. \\ & + \sum_{n=1}^{N_H} \left(\hat{x}_{2n-1} \cos(n \omega t) \right. \\ & \left. \left. + \hat{x}_{2n} \sin(n \omega t) \right) \right)^3 \sin(n \omega t) dt . \end{aligned} \quad (3.2.16)$$

It is noted that the coefficients defined by equations (3.2.14), (3.2.15) and (3.2.16) are functions of the unknown coefficients \hat{x}_n . The process of matching or balancing harmonics of the same order yields a system of $2N_H + 1$ equations for the unknown $2N_H + 1$ harmonic coefficients \hat{x}_n . The $2N_H + 1$ algebraic equations of the above system can be written in a vector form as:

$$(A^2 + 2\zeta A + I) \hat{Q}_x + \hat{R}_x = \tilde{F} \hat{H} \quad (3.2.17)$$

where

$$\hat{Q}_x = \begin{bmatrix} \hat{x}_0 \\ \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_{2N_H} \end{bmatrix}, \quad \hat{R}_x = \begin{bmatrix} \hat{r}_0 \\ \hat{r}_1 \\ \hat{r}_2 \\ \vdots \\ \hat{r}_{2N_H} \end{bmatrix}, \quad \hat{H} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \quad (3.2.18)$$

and

$$A = \begin{bmatrix} 0 & & & & \\ & J_1 & & & \\ & & J_2 & & \\ & & & \dots & \\ & & & & J_{N_H} \end{bmatrix}, \quad (3.2.19)$$

$$J_n = n \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad n = 1, 2, \dots, N_H$$

The arrays \hat{Q}_x , \hat{R}_x and \hat{H} have length $2N_H + 1$, and the matrix A has a block structure where only the diagonal blocks contain non-zero entries. The block (1,1) is a scalar 0, whereas all the other diagonal blocks are 2x2. The overall size of A is $(2N_H + 1) \times (2N_H + 1)$. Solving the harmonic integrals appearing in equations (3.2.14), (3.2.15) and (3.2.16) using $N_H = 1$, one finds the following expressions of the \hat{r}_n coefficients:

$$\hat{r}_0 = \hat{x}_0^3 + \frac{3}{2} \hat{x}_0 \hat{x}_1^2 + \frac{3}{2} \hat{x}_0 \hat{x}_2^2 \quad (3.2.20)$$

$$\hat{r}_1 = 3 \hat{x}_0^2 \hat{x}_1 + \frac{3}{4} (\hat{x}_1^3 + \hat{x}_1 \hat{x}_2^2) \quad (3.2.21)$$

$$\hat{r}_2 = 3 \hat{x}_0^2 \hat{x}_2 + \frac{3}{4} (\hat{x}_2^3 + \hat{x}_2 \hat{x}_1^2) \quad (3.2.22)$$

The above expressions of the coefficients \hat{r}_n complete the definition of equation (3.2.17). The comparison of the HB solution obtained by solving system (3.2.17) and the numerical solution obtained by solving the ODE (3.2.8) with a MATLAB ODE solver is reported in *Appendix G*.

Equation (3.2.17) represents a system of algebraic quadratic equations. It is thus seen how the HB method simplifies the calculation of periodic solution: a given ODE (and more generally, a system of ODEs) is transformed into a system of algebraic equations. Each equation yields one harmonic component of the truncated Fourier series representation of the sought periodic solution. The coupling of these equations is caused by the nonlinear term of the given ODE, and it manifests itself in the dependence of the \hat{r}_n on the harmonic components of the sought periodic flow. The calculation of the \hat{r}_n terms for ODEs and PDEs featuring nonlinear terms having a very complicated expression becomes extremely difficult, and this is the reason why, in such cases, the classical HB solution procedure presented above cannot be easily used. This problem is discussed by Hall et al. in [67] and more recently by Liu et al. in [101]. It was encountered when proposing the HB procedure for calculating the periodic solution of the RANS and

turbulence model equations. This problem can be circumvented by using the so-called *high-dimensional HB* method, presented in the next subsection.

3.2.2 High-dimensional harmonic balance

In the case of NS equations and, even more, the RANS equations augmented with the PDEs of differential turbulence models, the derivation of the \hat{r}_n terms is so complex that the authors who first proposed this approach in RANS-based CFD (Hall et al. [67]) gave up this route. The same authors [67] developed a new variant of the HB method that was later called the *High-Dimensional Harmonic Balance* (HDHB) method [101]. The main rationale for using HDHB approach is to avoid the derivation of the \hat{r}_n terms, a process that even for simple problems such as the Duffing's oscillator is quite involved, as shown by the complexity of equations (3.2.14), (3.2.15) and (3.2.16). The fundamental simplification introduced by the HDHB approach is to replace the $2N_H + 1$ unknowns corresponding to the mean value and the real and imaginary parts of the retained complex N_H Fourier harmonics of the sought periodic flow solution with $2N_H + 1$ equally spaced time-domain snapshots of the same sought periodic flow solution. This modification enormously simplifies the derivation of the HB equations to be solved, and it is also results in the form of the equations to be solved taking an extremely simple form. The $2N_H + 1$ harmonic balance Fourier coefficient solution variables \hat{Q}_x are related to $2N_H + 1$ equally spaced solution snapshots \tilde{Q}_x over a period of oscillation via a constant Fourier transform matrix E^{-1} . This dependence can be expressed as:

$$\tilde{Q}_x = E^{-1} \hat{Q}_x \quad (3.2.23)$$

where

$$\tilde{Q}_x = [x(t_0) \quad x(t_1) \quad x(t_2) \quad \dots \quad x(t_{2N_H-1}) \quad x(t_{2N_H})]^T \quad (3.2.24)$$

$$\hat{Q}_x = [\hat{x}_0 \quad \hat{x}_1 \quad \hat{x}_2 \quad \dots \quad \hat{x}_{2N_H-1} \quad \hat{x}_{2N_H}]^T \quad (3.2.25)$$

and

$$t_i = \frac{i}{2 N_H + 1} \frac{2 \pi}{\omega}. \quad (3.2.26)$$

The expression of the Fourier matrix E^{-1} is:

$$E^{-1} = \begin{bmatrix} 1 & \cos(\omega t_0) & \sin(\omega t_0) & \cdots & \cos(\omega N_H t_0) & \sin(\omega N_H t_0) \\ 1 & \cos(\omega t_1) & \sin(\omega t_1) & \cdots & \cos(\omega N_H t_1) & \sin(\omega N_H t_1) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \cos(\omega t_{2N_H-1}) & \sin(\omega t_{2N_H-1}) & \cdots & \cos(\omega N_H t_{2N_H-1}) & \sin(\omega N_H t_{2N_H-1}) \\ 1 & \cos(\omega t_{2N_H}) & \sin(\omega t_{2N_H}) & \cdots & \cos(\omega N_H t_{2N_H}) & \sin(\omega N_H t_{2N_H}) \end{bmatrix} \quad (3.2.27)$$

The relationship between the solution snapshots \tilde{Q}_x and the Fourier coefficients of the solution is obtained by inverting equation (3.2.23), and is given by:

$$\hat{Q}_x = E \tilde{Q}_x \quad (3.2.28)$$

where

$$E = \frac{2}{2 N_H + 1} \begin{bmatrix} 1/2 & 1/2 & \cdots & 1/2 \\ \cos(\omega t_0) & \cos(\omega t_1) & \cdots & \cos(\omega t_{2N_H}) \\ \sin(\omega t_0) & \sin(\omega t_1) & \cdots & \sin(\omega t_{2N_H}) \\ \cos(2 \omega t_0) & \cos(2 \omega t_1) & \cdots & \cos(2 \omega t_{2N_H}) \\ \sin(2 \omega t_0) & \sin(2 \omega t_1) & \cdots & \sin(2 \omega t_{2N_H}) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(N_H \omega t_0) & \cos(N_H \omega t_1) & \cdots & \cos(N_H \omega t_{2N_H}) \\ \sin(N_H \omega t_0) & \sin(N_H \omega t_1) & \cdots & \sin(N_H \omega t_{2N_H}) \end{bmatrix}. \quad (3.2.29)$$

Similarly

$$\hat{R}_x = E \tilde{R}_x \quad (3.2.30)$$

$$\hat{H} = E \tilde{H} \quad (3.2.31)$$

where

$$\tilde{R}_x = [x(t_0)^3 \quad x(t_1)^3 \quad x(t_2)^3 \quad \dots \quad x(t_{2N_H-1})^3 \quad x(t_{2N_H})^3]^T \quad (3.2.32)$$

$$\tilde{H} = [\text{sint}_0 \quad \text{sint}_1 \quad \text{sint}_2 \quad \dots \quad \text{sint}_{2N_H-1} \quad \text{sint}_{2N_H}]^T. \quad (3.2.33)$$

Inserting equations (3.2.28), (3.2.30), and (3.2.31) into the classical HB formulation of Duffing's oscillator equation (3.2.17) yields:

$$(A^2 + 2 \zeta A + I) E \tilde{Q}_x + E \tilde{R}_x = \tilde{F} E \tilde{H}. \quad (3.2.34)$$

Multiplying both sides of the above equation by E^{-1} gives:

$$(D^2 + 2 \zeta D + I) \tilde{Q}_x + \tilde{R}_x = F \tilde{H} \quad (3.2.35)$$

where

$$D = E^{-1} A E, \quad D^2 = E^{-1} A^2 E. \quad (3.2.36)$$

Equation (3.2.35) is the HDHB formulation of Duffing's oscillator. Both equations (3.2.17) and (3.2.35) represent a nonlinear system of $2N_H + 1$ equations. Unlike equation (3.2.17), the assembly of equation (3.2.35) does not require complicated analytical transformations such as those needed for the construction of \hat{r}_n terms of equation (3.2.17). This simplicity feature of the HDHB approach is crucially important when applying the HB technology for solving sophisticated system of equations such as those arising from the discretisation of the RANS equations and the PDEs of differential turbulence model equations. It should also be noted that the coupling of the flow snapshots in equation (3.2.35) occurs through the non-diagonal matrices D^2 and D , which are the spectral counterparts of the second and first time-derivatives in the time-domain. These operators link the unknown snapshots in the system of equations. As previously stated, the coupling of the flow harmonics in equation (3.2.17) occurs instead through the terms \hat{r}_n , which depend on the unknown coefficients of the retained Fourier harmonics. In the case $N_H = 1$, one has $t_0 = 0$, $t_1 = \frac{2\pi}{3\omega}$ and $t_2 = \frac{4\pi}{3\omega}$. Therefore, the matrix E and E^{-1} become respectively:

$$E = \frac{2}{3} \begin{bmatrix} 1/2 & 1/2 & 1/2 \\ 1 & \cos\left(\frac{2\pi}{3}\right) & \cos\left(\frac{4\pi}{3}\right) \\ 0 & \sin\left(\frac{2\pi}{3}\right) & \sin\left(\frac{4\pi}{3}\right) \end{bmatrix} \quad (3.2.37)$$

$$E^{-1} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & \cos\left(\frac{2\pi}{3}\right) & \sin\left(\frac{2\pi}{3}\right) \\ 1 & \cos\left(\frac{4\pi}{3}\right) & \sin\left(\frac{4\pi}{3}\right) \end{bmatrix}. \quad (3.2.38)$$

From the equation (3.2.19) one can compute the arrays A and A^2

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \quad (3.2.39)$$

$$A^2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (3.2.40)$$

and through those arrays one can calculate the new arrays D and D^2 needed for the HDHB method:

$$D = \begin{bmatrix} 0 & \frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} \\ -\frac{\sqrt{3}}{3} & 0 & \frac{\sqrt{3}}{3} \\ \frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} & 0 \end{bmatrix} \quad (3.2.41)$$

$$D^2 = \frac{2}{3} \begin{bmatrix} -1 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -1 \end{bmatrix}. \quad (3.2.42)$$

3.3 HDHB formulation of RANS and SST equations

As shown later in this thesis, space-discretising the system of time-dependent RANS and the turbulence equations (*chapter 4*), but retaining the continuing physical time-derivatives, results in a very large system of nonlinear ODEs. The number of those ODEs is equal to the number of discrete points or cells used to discretise the physical domain of interest and the number of unknowns (i.e. PDEs) per point or cell. Therefore, the calculation of time-dependent solutions of the RANS and the turbulence model equations is accomplished by time-marching such large system of ODEs. In the case of periodic flow fields featuring strong nonlinearity levels, the solution of the aforementioned time-dependent CFD equations can be achieved by applying the HB technology to the large system of ODEs resulting from the space-discretisation. The application of the HB approach to this system yields a system of nonlinear algebraic equations the size of which equals that of the starting system of nonlinear ODEs times the number of harmonics to be used. In view of the higher simplicity of obtaining the HB equations using the HDHB approach rather than the classical HB approach, the former technique has been used for obtaining the governing HB RANS and SST equations solved by the CFD solver used in this research. Recently Ekici and Hall developed an HDHB approach to solve the RANS equations and the Spalart-Allmaras turbulent equation [104, 105]. Denoting by $\hat{\mathbf{u}}$ and $\hat{\mathbf{h}}$ respectively the volume and surface integral of the RANS and turbulence equations (2.7.6), one can approximate both variables by means of the following truncated Fourier series:

$$\hat{\mathbf{u}}(t) = \hat{\mathbf{u}}_0 + \sum_{n=1}^{N_H} (\hat{\mathbf{u}}_{a_n} \cos(\omega n t) + \hat{\mathbf{u}}_{b_n} \sin(\omega n t)) \quad (3.3.1)$$

$$\hat{\mathbf{h}}(t) = \hat{\mathbf{h}}_0 + \sum_{n=1}^{N_H} (\hat{\mathbf{h}}_{a_n} \cos(\omega n t) + \hat{\mathbf{h}}_{b_n} \sin(\omega n t)) \quad (3.3.2)$$

The time-derivative of $\hat{\mathbf{u}}$ is instead approximated by:

$$\frac{\partial \hat{\mathbf{u}}(t)}{\partial t} = \sum_{n=1}^{N_H} n \omega (-\hat{\mathbf{u}}_{a_n} \sin(\omega n t) + \hat{\mathbf{u}}_{b_n} \cos(\omega n t)) \quad (3.3.3)$$

Note all arrays appearing in equations (3.3.1), (3.3.2) and (3.3.3) have length N_{PDE} , where N_{PDE} is the number of considered conservation laws. In the case of 2D RANS equation coupled to the SST model N_{PDE} are equal to 6. Inserting the expressions (3.3.2) and (3.3.3) into equation (2.7.6) and 'balancing' or matching harmonics of the same order results in a system of $N_T = N_{PDE} \times (2N_H + 1)$ equations in N_T unknowns, which can be expressed as

$$\omega A \hat{\mathbf{u}} + \hat{\mathbf{h}} = 0. \quad (3.3.4)$$

The symbol A denotes a $N_T \times N_T$ matrix containing the entries of equation (3.2.19), while the arrays $\hat{\mathbf{u}}_n$ and $\hat{\mathbf{h}}_n$ have length N_{PDE} and represent respectively the real and imaginary parts of the complex harmonics of the volume and surface integrals of the system of RANS and SST equations.

As highlighted with the example of the Duffing's oscillator presented above, the difficulty with solving equations (3.3.2) is deriving analytically the expressions for $\hat{\mathbf{h}}_n$ as functions of $\hat{\mathbf{u}}_n$. To avoid this problem, the HDHB method is applied. By doing this the unknowns become the $2N_H + 1$ equally spaced snapshots of the sought periodic flow field with period $T = \frac{2\pi}{\omega}$. The array $\tilde{\mathbf{u}}$ containing the snapshots of the volume integral and the array $\tilde{\mathbf{h}}$ containing the snapshots of the surface integral are given respectively by:

$$\tilde{\mathbf{u}} = \begin{Bmatrix} \mathbf{u}(t_0 + \Delta t) \\ \mathbf{u}(t_0 + 2\Delta t) \\ \cdot \\ \cdot \\ \mathbf{u}(t_0 + T) \end{Bmatrix}, \quad \tilde{\mathbf{h}} = \begin{Bmatrix} \mathbf{h}(t_0 + \Delta t) \\ \mathbf{h}(t_0 + 2\Delta t) \\ \cdot \\ \cdot \\ \mathbf{h}(t_0 + T) \end{Bmatrix} \quad (3.3.5)$$

where $\Delta t = \frac{2\pi}{(N_T \omega)}$. As in the example of Duffing's oscillator, the Fourier harmonics are related to the snapshots by means of a Fourier matrix E^{-1} through the following transformation:

$$\tilde{\mathbf{u}} = E^{-1} \hat{\mathbf{u}} \quad (3.2.6)$$

$$\tilde{\mathbf{h}} = E^{-1} \hat{\mathbf{h}}. \quad (3.3.7)$$

Inserting these expressions into equation (3.3.4) gives:

$$\omega A E \tilde{\mathbf{u}} + E \tilde{\mathbf{h}} = 0. \quad (3.3.8)$$

Multiplying all terms by the transformation matrix E^{-1}

$$\omega E^{-1} A E \tilde{\mathbf{u}} + E^{-1} E \tilde{\mathbf{h}} = 0 \quad (3.3.9)$$

And

$$\omega D \tilde{\mathbf{u}} + \tilde{\mathbf{h}} = 0. \quad (3.3.10)$$

Inserting the integral definitions of $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{h}}$ into equation (3.3.10) leads to the so called *high-dimensional harmonic balance* formulation [95] of the RANS and turbulence equations:

$$\omega D \left(\frac{\partial}{\partial t} \int_{V(t)} \hat{\mathbf{U}}_H dV \right) + \oint_{S(t)} (\hat{\Phi}_{cH} - \hat{\Phi}_{dH}) dS = \int_{V(t)} \hat{S}_H dV \quad (3.3.11)$$

Where

$$\hat{\mathbf{U}}_H = [\hat{\mathbf{U}}(t_0) \quad \hat{\mathbf{U}}(t_1) \quad \hat{\mathbf{U}}(t_2) \quad \dots \quad \hat{\mathbf{U}}(t_{2N_H-1}) \quad \hat{\mathbf{U}}(t_{2N_H})]^T \quad (3.3.12)$$

$$\begin{aligned} & \hat{\Phi}_{cH} \\ &= [\hat{\Phi}_{cH}(t_0) \quad \hat{\Phi}_{cH}(t_1) \quad \hat{\Phi}_{cH}(t_2) \quad \dots \quad \hat{\Phi}_{cH}(t_{2N_H-1}) \quad \hat{\Phi}_{cH}(t_{2N_H})]^T. \end{aligned} \quad (3.3.13)$$

Similar expressions hold for $\hat{\Phi}_{dH}$ and \hat{S}_H . As one can see the number of unknowns of the system has been increased from N_{pde} to $N_{pde} \times (2N_H + 1)$. Despite the fact that the number of PDE's to be solved has increased, the HB approach allows one to compute unsteady periodic flows at a lower computational cost with respect to the time-domain approach. This will be shown in chapter 8 of this thesis.

Chapter 4

SPACE DISCRETISATION

Chapter 2 highlighted the physical interpretation of the PDEs of the RANS system, and also the two PDEs of the shear stress transport equation used for the turbulence closure, and outlined the main steps taken in the derivation of these equations. Chapter 3 introduced the Harmonic Balance form of the RANS and SST equations, an alternative form to the time-domain formulation of these conservation laws, well suited for the rapid calculation of unsteady periodic flow fields. The presented system of PDEs, however, can be very rarely solved analytically. In general, these equations have to be solved numerically. The first stage of the numerical solution consists of space-discretising the system of RANS and the SST equations. This chapter outlines the space-discretisation used to solve these governing equations in the framework of this research. Firstly the physical domain has to be space-discretised and this lead to the construction of the computational domain to which one can associate a mesh. Once the generation of the mesh is completed, the equations have to be discretised choosing a particular discretisation approach (finite differences, finite volumes, finite elements or other methods). This step yields a large set of nonlinear algebraic equations. In this research, a cell-centred finite volume discretisation strategy has been selected. The form of the governing equations best suited to the finite volume discretisation is the integral form of the conservation laws. As customary in compressible finite volume methods a different discretization type has to be used for the convective fluxes, the diffusive fluxes and the source terms, even though a single general discretization approach is selected to move from the continuous to the discrete representation of the governing equations. The discretisation of the convective fluxes is performed using Van Leer's Monotone Upstream centred extrapolations and Roe's flux-difference splitting method, and flux limiters are applied to avoid non-physical values of the entropy. The diffusive fluxes are computed by using the generalized curvilinear coordinates associated with the grid line system, and central finite differencing. All source terms of a particular cell are evaluated at the cell centre, and therefore, unlike the convective and the diffusive fluxes, the discretization of the source terms of a cell, does not couple the unknown flow field of that cell to that of neighbouring cells.

4.1 Meshes

The RANS and SST equations described in *chapter 2* have to be solved in a bounded physical domain. For this reason the physical domain of interest is discretised by means of a computational mesh or grid, which is divided into a number of geometrical elements, called *cells*. This process, known as *grid generation*, yields a computational grid or mesh. Meshes can be unstructured or structured. At the conceptual level, unstructured meshes are obtained by considering a cloud of points enclosed by the boundaries of the computational domain and joining these points according to certain rules. In 2D problems, unstructured meshes are usually (not always) made up of triangles, whereas in 3D problems, they are made up of tetrahedral. Because of their construction, unstructured meshes lack the inherent topology required to easily identify the neighbours of a given grid node, the edges sharing a common vertex and the volumes sharing a common vertex. These characteristics increase the complexity of certain aspects of unstructured CFD codes, such as the calculation of the convective fluxes. The typical motivation for accepting this kind of additional complexity of unstructured CFD codes, is the ease by which unstructured grids can handle geometric complexities, i.e. the grid generation process required to model the flow past very complex geometries.

A simple structured mesh is defined by 2 (in two dimensional problems) and 3 (in three dimensional problems) families of grid lines intersecting each other and filling the space defined by the physical domain. The vertices of the intersections define the grid vertices. Since the intersections can be easily numbered using a progressive sequence in each direction, the neighbours of each vertex, the edges sharing a common vertex and the volumes sharing a common vertex can be immediately defined, given the intrinsic topology of the structured meshes. In 2D problems the elements of structured meshes are rectangles, whereas in 3D problems the elements are hexahedra. An example of a simple structured mesh required for the calculation of the flow field past a NACA4412 airfoil is provided in figure 4.1.

The structured grid generator, namely a computer program that takes as input the geometry of the boundaries of the physical domain, and other grid-related information such as number of grid lines in the 2 directions and the stretching factors and/or minimum distances from selected boundaries, gives as output the coordinates of the vertices of the required computational grid. When using structured meshes, necessary conditions to guarantee a good solution quality are that the grid lines be as smooth as possible (i.e. their

first derivatives should not have discontinuities) and the linear dimension of the grid elements in each direction vary as smoothly as possible. The family lines (2 in 2D problems and 3 in 3D problems) should intersect locally at an angle that should be as close as possible to 90 degrees, i.e. the grid should be as close as possible to the local orthogonally condition. In addition no holes should exist between the grid cells and the lines that connect the vertices should not overlap.

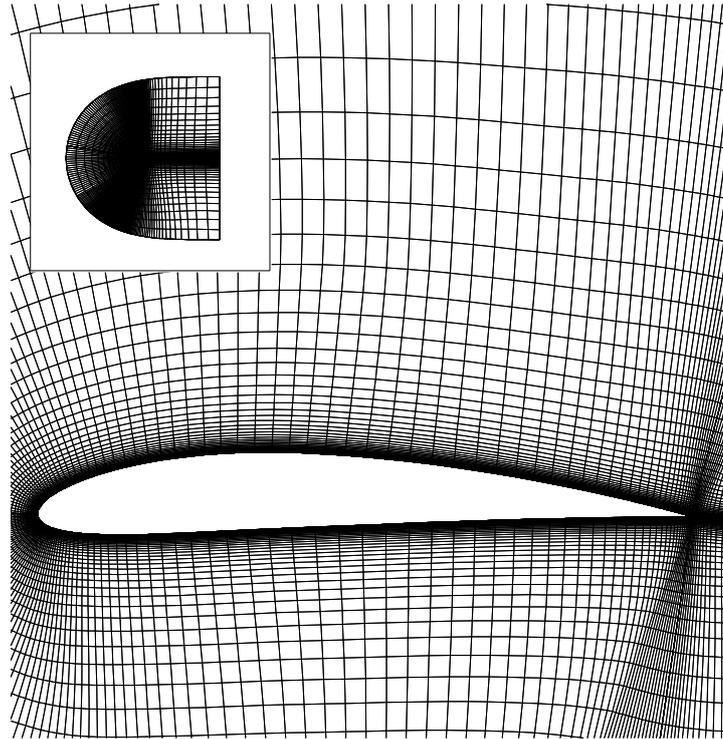


Figure 4.1:structured grid

The main advantage of using structured meshes is that certain features of structured CFD solvers, such as the calculation of the convective fluxes, are simpler than in the case of unstructured CFD solvers. A potential disadvantage of the structured meshes comes from the fact that it is hard to generate good quality meshes past very complex geometries using a single block, like that reported in figure 4.1. This problem, however, is easily overcome by using multi-block grids, i.e. composite structured grids resulting from the union of simple structured blocks. An example of a structured multi-block grid is illustrated in figure 4.2.

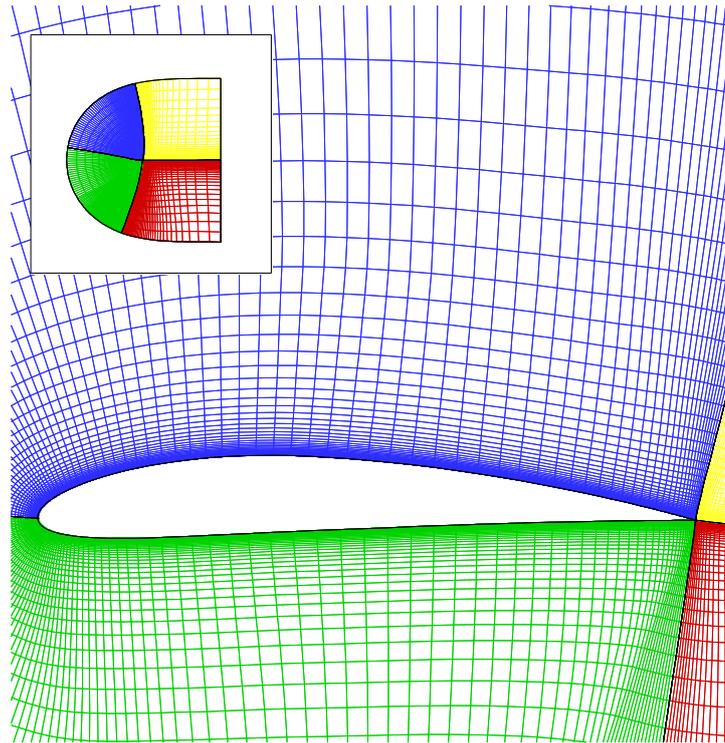


Figure 4.2: structured multi-block grid

The structured multi-block capability allows geometric complexity to be effectively handled also with structured meshes. The multi-blocking feature, however, introduces additional complexity in the architecture of the structured CFD solver, particularly in the code sections associated with the data information among blocks. Hence, it is seen that the overall level of added complexity of structured and unstructured CFD solvers when dealing with fluid problems with complex geometries ends up being fairly similar. Detailed information on the theory behind structured and unstructured grid generation, and good practice in both processes can be found in [106] whereas details about the generation of structured and unstructured grids can be found in the books of Thompson et al [107, 108].

For complex turbulent unsteady problems, computational run-times can be very large, particularly when using meshes with large numbers of cells. It is therefore essential to use a high level of spatial refinement only where strictly necessary, that is in the flow region characterised by large flow gradients. The necessity of maintaining the overall grid size within the bounds that make the simulation affordable with the available computational resources, introduces grid generation constraints that make the grid generation process

more involved. An additional, complementary, means of reducing run-times is to make use of parallel computing. A typical effective way of parallelising large CFD simulations is to use distributed parallel computing based on the *Message Passing Interface* (MPI) libraries. In the case of structured multi-block codes, each MPI handles a subset of grid blocks, and periodically, flow information is exchanged among the MPI processes through the interfaces of the grid blocks. More detail on the MPI parallelisation on the CFD solver used in this research is available in [109, 110, 111].

4.2 Discretisation approaches and the finite volume method

Once the generation of the mesh is completed, the equations can be discretized leading to a large set of nonlinear algebraic equations. There are three kinds of discretization approaches for the numerical solution of a PDE or a system thereof: the *finite difference* (FD) method, the *finite element* (FE) method, and the *finite volume* (FV) method.

The finite difference method [2, 3, 4, 13, 112] was proposed by Euler around 1768 and was the first method applied to the numerical solution of differential equations. Using this approach, the conservation laws are solved in differential form at each grid point. The derivatives of the flow variables are replaced by finite differences obtained by suitably combining truncated Taylor series. For a one-dimensional structure mesh with N discrete points x_i , $i = 1, \dots, N$ one can approximate the first derivative of a function $u(x)$ at the point x_i , using a Taylor series as:

$$\frac{\partial u_i}{\partial x} = \frac{u_{i+1} - u_i}{\Delta x} + O(\Delta x) \quad (4.2.1)$$

$$\frac{\partial u_i}{\partial x} = \frac{u_i - u_{i-1}}{\Delta x} + O(\Delta x) \quad (4.2.2)$$

where Δx is the distance between the discrete points and $O(\Delta x)$ is the truncation error. Both equations (4.2.1) and (4.2.2) are first order approximations of the first derivative of u whereas the former equation is the so called *forward difference* and the latter equation is the so called *backward difference*. In addition one can use the truncated Taylor series

to obtain a second order approximation for the first derivative of u , the so-called central difference, written as:

$$\frac{\partial u_i}{\partial x} = \frac{u_{i+1} - u_{i-1}}{2\Delta x} + O(\Delta x^2). \quad (4.2.3)$$

Equation (4.2.3) can be used to calculate the derivative of the mid-point between x_i and x_{i+1} , or between x_{i-1} and x_i . In this way the forward difference or backward difference can be considered as second order approximations of the derivatives $\frac{\partial u_{i+1/2}}{\partial x}$ and $\frac{\partial u_{i-1/2}}{\partial x}$ respectively.

$$\frac{\partial u_{i+1/2}}{\partial x} = \frac{u_{i+1} - u_i}{\Delta x} + O(\Delta x^2). \quad (4.2.4)$$

$$\frac{\partial u_{i-1/2}}{\partial x} = \frac{u_i - u_{i-1}}{\Delta x} + O(\Delta x^2). \quad (4.2.5)$$

As a result each grid point yields one algebraic equation for each PDE to be solved, and the number of unknowns of each equation is equal to the number of nodes used to form all finite differences used to discretise each PDE at a given grid vertex. More details about the development of the FD formulas and their accuracy can be found in the book of Hildebrand [113]. The main advantages of the FD method are its simplicity and the potentially high-order that can be achieved by usable suitable finite differences. However, one of the general drawbacks of the FD method is that it can be used only for structured meshes. In the case of compressible CFD applications, the use of the FD method does not yield shock-capturing schemes, a property of crucial importance for the solution of transonic flow problems. For this type of applications, a compressible CFD solver based on the FD method would require the use of a shock fitting method [114, 115] and this would significantly complicate the implementation of the CFD code.

The finite element (FE) method was first introduced by Turner [116] in 1956 and later by Clough [117] in 1960. Using this approach, the method discretizes the physical domain into a number of geometrical “elements” which have arbitrary size and shape. Usually, the shape of these elements is triangular and quadrilateral and their only constraint is to fulfil the physical domain and not overlap with each other. Inside each element or on its boundaries, one can define a certain number of points where the values of the PDEs and their derivatives have to be solved. The total number of unknowns for each element is

called *degrees of freedom* of the numerical problem. In addition the so-called *shape functions* represent the variation of the solution inside each element and they have to be defined. Usually shape functions are linear, defined in such a way that their value outside of the corresponding element is zero. The nature of the FE discretisation method makes it more suitable for unstructured grids and for this reason the method can be used to solve fluid flows past very complex geometries. More detail about the FE method can be found in the book of Taylor et al [118], Chung [119] and Thomasset [120]. More advanced applications of finite elements to fluid flows can be found in Baker [121].

The most popular method to discretise the PDEs is the finite volume method, which was first proposed and implemented by McDonald [122] in 1971 for the simulation of two dimensional inviscid flows. The idea is to use the mesh to subdivide the computational domain into a number of arbitrary CVs, and apply directly the integral formulation of the governing conservation laws. The method is very flexible as it can be used for any type of grid and a large number of options are available for defining the CV. The approximation of the surface integrals derives from the sum of the fluxes on the faces of the CV. The accuracy of this approximation depends on the scheme used to calculate the fluxes. One of the main advantages of the FV method is that it has the ability to compute weak solutions on the governing equations. This means that it is a shock capturing method while, on the other hand, the FD method does not have this ability. Its main drawback is that it's difficult to obtain high order schemes compared to FD and FE methods. Since the study reported in this thesis has been performed using a compressible CFD code based on the finite volume method, this approach is described in more detail in the following subsection.

4.2.1 The finite volume method

The FV method solves directly the integral form of RANS and turbulence equations. The most common schemes of FV are: the cell-centred scheme and the cell-vertex scheme. In the cell-centred scheme, the unknown flow field is taken to be referred to the cell centres of the grid cell or finite volumes, and the CVs over which the conservation laws are enforced are the grid-cells themselves. A grid cell is highlighted with a solid red line in the left subplot of figure 4.3. In the cell-vertex scheme, on the other hand, the unknown

flow field is taken to be referred to cell-vertices of the grid, which are the nodes of the given computational grid. In this case, the CVs over which the conservation laws are enforced are the cells on the dual grid obtained by joining the cell centres of the input data. A finite volume of the grid for a cell-vertex CFD code is reported in the right subplot of figure 4.3. The CFD code used for the research presented in this thesis uses a cell-centred finite volume scheme.

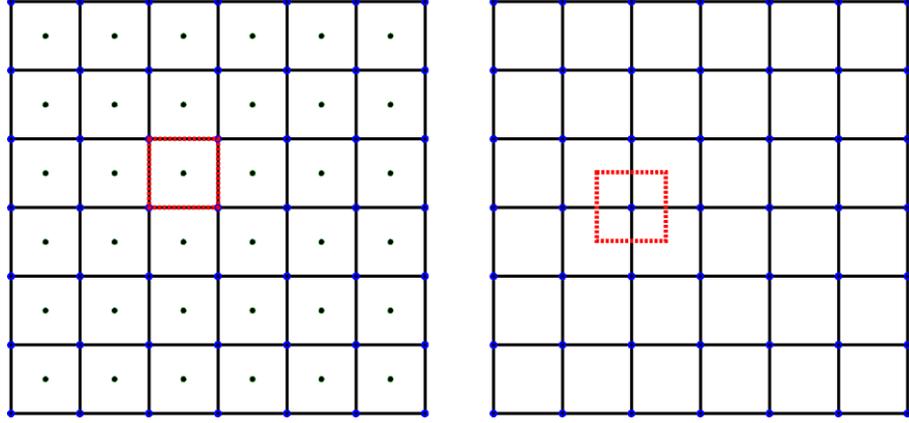


Figure 4.3: **Left:** cell-centred scheme, **Right:** cell-vertex scheme

When using the FV to solve the RANS and the turbulence equations, the general integral from the governing equations given by equation (2.7.6) and reported below for clarity

$$\frac{\partial}{\partial t} \int_V \hat{U} dV + \oint_S (\hat{\Phi}_c - \hat{\Phi}_d) dS = \int_V \hat{S} dV \quad (4.2.6)$$

is applied to each of the finite volumes defined by the given grid. The entries of the unknown flow field state \hat{U} are taken to be averaged values over the CV, and such an average is taken to be defined at the cell centre. The first term on the left hand side of equation (4.2.6) and the term on the right hand side of the same equation require integration over the CV. The simplest approximation to compute these terms numerically is to replace the volume integral by the product of the mean value of the integrand and the volume of the cell. The second term of the above equation represents the net flux through the CV boundary and it can be replaced by the sum of the fluxes over all the faces of the CV, the number of which is equal to 4 in two-dimensional problems, and 6 in three-dimensional problems. Applying the FV method to the continuous form of the conservation law provided by equation (4.2.6) yields the following space-discretised form

of the governing equations:

$$\frac{\partial}{\partial t}(\hat{U}_c V_c) + \sum_{sides} (\hat{\Phi}_c - \hat{\Phi}_d)_c S = \hat{S}_c V_c \quad (4.2.7)$$

where the subscript c denotes a particular CV or cell of the computational domain and V_c is the volume of the cell. Each CV produces one algebraic equation and the sum of all these equations yields equation (4.2.6), which can thus be viewed as the set of conservation laws applied to the control volume corresponding to the entire physical domain, as previously discussed. The first term of equation (4.2.7) represents the time variation of \hat{U}_c . In order to compute this term (which, as shown in the following chapters, is the key term for solving the set of discrete equations by means of the explicit pseudo-time-marching process), the calculation of the volumes V_c is required. In the cell-centred scheme, which is used in the present study, the calculation of V_c simply requires the coordinates of the mesh vertices, which are the points defining all CVs. The net flux of each cell is calculated independently of that of all other cells of the domain, but the calculation of each face flux requires multiple cell centres, as discussed in the following sections. The coupling of the flow field at all cell centres is caused by this dependence of each face flux on multiple cell centres. As shown in section 4.3, the calculation of the convective fluxes is accomplished by Van Leer's *Monotone Upstream-centred Schemes for Conservation Laws* (MUSCL) [123] and Roe's flux-difference splitting [124]. The use of this approach requires knowledge of both the value of the flow field at each cell face, and the discontinuity or jump of the flow field at the same interface. Finally, the diffusive fluxes are computed by using the system of generalised coordinates associated with the family lines of the structured grid, and applying central-differencing. This is explained in more detail in section 4.5.

4.3 Convective fluxes

In this section the approximation of the convective fluxes will be explained in detail. There are many methods that one can use to calculate the values of the fluxes using the

FV method. The method that is used in this study is the *Flux-Difference Splitting* (FDS) [125]. In this method, each entry of the convective flux vector (i.e. mass, momentum and energy flux) is written as the sum of the average of a left and right flux (left and right with respect to the considered cell face), and a flux difference across the considered face. Each flux difference can be split into two components according to the sign of the speeds of the characteristic variables of the Euler equations (the characteristic variables can be viewed as the dependent variables of a system of wave equations equivalent to the Euler equations [4]). Each flux difference is written as a sum of wave or characteristic contributions taking into account the direction of propagation of each wave. This, along with the use of suitable averages [126], guarantees good shock capturing features of the FV scheme thus constructed. In this way a better resolution of shocks can be achieved.

Before presenting the main description of Roe's FDS and Van Leer's MUSCL extrapolations, the meaning of the designations of left (L) and right (R) has to be introduced. Figure 4.4 reports a 1-D mesh with several points. Each point belongs to one CV with two boundaries, and each face can be approached from two different sides, from the left side of the face or from the right side of the face. As shown later, using the *upwind method*, the calculation of the left flux through the right face of cell i requires the flow states of cell i , and cell $i - 1$, and the calculation of the right flux through the right face of the same cell i requires the flow states of cells $i + 1$, and $i + 2$.

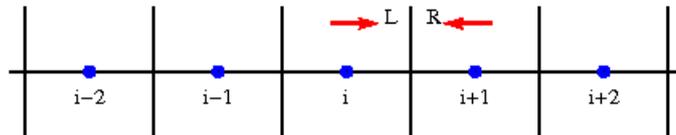


Figure 4.4: Flux-difference splitting

In 2D problems, starting from equation (2.7.6) the analytical expression of the convective fluxes through a cell face of area dS and unit normal \mathbf{n} is given by:

$$\hat{\Phi}_c = (\hat{\Phi}_{c,f} \cdot \mathbf{n}) dS = (\hat{E}_c \mathbf{n}_x + \hat{F}_c \mathbf{n}_y) dS \quad (4.3.1)$$

where \mathbf{n}_x and \mathbf{n}_y are respectively the x- and y- components of \mathbf{n} . In order to calculate the fluxes $\hat{\Phi}_c$, the flow state \hat{U} must be extrapolated from the cell centres to the sides of the CV faces under consideration. Following [127], if we consider a flow state \hat{U}_i of a cell i (figure 4.4) and calculate the first derivative of \hat{U}_i obtained by suitably combining

truncated Taylor series, one obtains the expression below:

$$\frac{\partial \hat{U}_i}{\partial x} = \frac{3 \hat{U}_i - 4 \hat{U}_{i-1} + \hat{U}_{i-2}}{2 \Delta x} + O(\Delta x^2). \quad (4.3.2)$$

Equation (4.3.2) is a *second order backward finite difference* approximation and can be expressed as the sum of first order backward difference (4.2.2) and a correction for second order accuracy:

$$\frac{\partial \hat{U}_i}{\partial x} = \frac{\hat{U}_i - \hat{U}_{i-1}}{\Delta x} + \frac{\hat{U}_i - \hat{U}_{i-1}}{2 \Delta x} - \frac{\hat{U}_{i-1} - \hat{U}_{i-2}}{2 \Delta x}. \quad (4.3.3)$$

Equation (4.3.3) can be written as:

$$\frac{\partial \hat{U}_i}{\partial x} = \frac{\hat{U}_{i+\frac{1}{2}}^- - \hat{U}_{i-\frac{1}{2}}^-}{\Delta x} \quad (4.3.4)$$

where

$$\hat{U}_{i+\frac{1}{2}}^- = \hat{U}_i + \frac{\hat{U}_i - \hat{U}_{i-1}}{2} \quad (4.3.5)$$

$$\hat{U}_{i-\frac{1}{2}}^- = \hat{U}_{i-1} + \frac{\hat{U}_{i-1} - \hat{U}_{i-2}}{2}. \quad (4.3.6)$$

The superscript symbol " - " denotes the backward finite difference. Similarly, calculating the first derivative of \hat{U}_i with a *second order forward finite difference*, obtained by suitably combining truncated Taylor series, one obtains the expression below:

$$\frac{\partial \hat{U}_i}{\partial x} = \frac{-\hat{U}_{i+2} + 4 \hat{U}_{i+1} - 3 \hat{U}_i}{2 \Delta x} + O(\Delta x^2). \quad (4.3.7)$$

Following the same procedure but using this time the second order forward finite difference approximation, one can obtain the expression below:

$$\frac{\partial \hat{U}_i}{\partial x} = \frac{\hat{U}_{i+\frac{1}{2}}^+ - \hat{U}_{i-\frac{1}{2}}^+}{\Delta x} \quad (4.3.8)$$

where

$$\hat{U}_{i+\frac{1}{2}}^+ = \hat{U}_{i+1} - \frac{\hat{U}_{i+2} - \hat{U}_{i+1}}{2} \quad (4.3.9)$$

$$\hat{U}_{i-\frac{1}{2}}^+ = \hat{U}_i - \frac{\hat{U}_{i+1} - \hat{U}_i}{2}. \quad (4.3.10)$$

The superscript symbol " + " denotes the forward finite difference. Using the expressions (4.3.5) and (4.3.9) one can approximate a convective flux from the left and from the right, respectively, using only information from one side of the flux. This is the so-called *monotone upstream-centred scheme for conservation laws* (MUSCL). Finally one can write equations (4.3.5) and (4.3.9) as:

$$\hat{U}_{i+\frac{1}{2}}^L = \hat{U}_i + \frac{\hat{U}_i - \hat{U}_{i-1}}{2} \quad (4.3.11)$$

$$\hat{U}_{i+\frac{1}{2}}^R = \hat{U}_{i+1} - \frac{\hat{U}_{i+2} - \hat{U}_{i+1}}{2}. \quad (4.3.12)$$

Equations (4.3.11) and (4.3.12) provide second order fully one side left and right approximations of the flow state on the interface $i + \frac{1}{2}$ and such approximation form the basis of the *linear upwind method*. Limiters have to be applied in equation (4.3.11) and (4.3.12) to avoid any oscillations around shocks and other discontinuities or sharp changes in the solution domain. The Van Albada limiter [128] has been implemented in this study. Using this limiter equations (4.3.11) and (4.3.12) can be written as:

$$\hat{U}_{i+\frac{1}{2}}^L = \hat{U}_i + \varphi_L(r) \frac{\hat{U}_i - \hat{U}_{i-1}}{2} \quad (4.3.13)$$

$$\hat{U}_{i+\frac{1}{2}}^R = \hat{U}_{i+1} - \varphi_R(r) \frac{\hat{U}_{i+2} - \hat{U}_{i+1}}{2} \quad (4.3.14)$$

where

$$\varphi_R = \frac{r(r+1)}{r^2+1}, \quad r = \frac{\widehat{U}_{i+1} - \widehat{U}_i}{\widehat{U}_{i+2} - \widehat{U}_{i+1}} \quad (4.3.15)$$

$$\varphi_L = \frac{r(r+1)}{r^2+1}, \quad r = \frac{\widehat{U}_{i+1} - \widehat{U}_i}{\widehat{U}_i - \widehat{U}_{i-1}} \quad (4.3.16)$$

After calculating the flow state \widehat{U} at the cell faces, the numerical representation of the continuous convective fluxes described by equation (4.3.1) at the cell faces can be obtained by using the flux difference splitting method. The adoption of this technique yields:

$$\widehat{\Phi}_c^* = \frac{1}{2} [\widehat{\Phi}_{c,f}(\widehat{U}_L) + \widehat{\Phi}_{c,f}(\widehat{U}_R) - |\widetilde{K}_U| \delta\widehat{U}]. \quad (4.3.17)$$

Here the superscript "*" denotes a numerical approximation. The subscripts f , L and R denote respectively face fluxes, and flow states extrapolated from the left and from the right side of the cell face under consideration. The numerical dissipation term $|\widetilde{K}_U| \delta\widehat{U}$ depends on the flow state discontinuity across the cell face, defined by $\delta\widehat{U} = \widehat{U}_R - \widehat{U}_L$, and the generalized flux Jacobian \widetilde{K}_U evaluated at the face under consideration using suitably defined mean values of the flow state. The expression of this flux Jacobian is:

$$\widetilde{K}_U = \frac{\partial \widehat{\Phi}_c}{\partial \widehat{U}} = \left(\frac{\partial \widehat{E}_c}{\partial \widehat{U}} \mathbf{n}_x + \frac{\partial \widehat{F}_c}{\partial \widehat{U}} \mathbf{n}_y \right) = (\widetilde{A} \mathbf{n}_x + \widetilde{B} \mathbf{n}_y) \quad (4.3.18)$$

where the symbols \widetilde{A} and \widetilde{B} denote respectively the flux Jacobian of the convective fluxes in the x- and y- direction. One can rewrite the last term of equation (4.3.17) as:

$$\delta\widehat{\Phi} = |\widetilde{K}_U| \delta\widehat{U} = \widetilde{\mathbf{P}} |\widetilde{\Lambda}| \widetilde{\mathbf{P}}^{-1} \delta\widehat{U} = \widetilde{\mathbf{P}} |\widetilde{\Lambda}| \delta\widehat{\mathbf{W}} \quad (4.3.19)$$

where $\widetilde{\mathbf{P}}$ is the matrix of the right eigenvectors of \widetilde{K}_U (more specifically the columns of $\widetilde{\mathbf{P}}$ are the right eigenvectors of \widetilde{K}_U), $\widetilde{\Lambda}$ is the matrix of the eigenvalues of \widetilde{K}_U (more specifically $\widetilde{\Lambda}$ is diagonal and its nonzero entries are the eigenvalues of \widetilde{K}_U), and the symbol $\delta\widehat{\mathbf{W}}$ denotes the discontinuity of the characteristic variables at the cell interface, defined by $\delta\widehat{\mathbf{W}} = \widetilde{\mathbf{P}}^{-1} \delta\widehat{U}$. The symbol $\widetilde{\mathbf{P}}^{-1}$ denotes the matrix of the left eigenvectors

of \tilde{K}_U : its rows are the left eigenvectors of \tilde{K}_U . Equation (4.3.19) highlights that the construction of the numerical dissipation requires the calculation of the eigenvalues and the eigenvectors of \tilde{K}_U . More details on the calculation of the numerical dissipation term of equation (4.3.19) can be found in *Appendix H*.

There are other methods that one can use to calculate the numerical fluxes at the interfaces of the cells when using the FV method. Such methods include: *Central differencing* [112], *Flux-vector splitting* [124, 129] and *Total variation diminishing* (TVD) [130, 131]. The simplest method is central differencing where the approximation of the convective fluxes is based on the values of the two adjacent cells (equation (4.2.4) and (4.2.5)). It is a second order method but is unbounded. This means that, in problems characterised by strong shocks, the method produces non-physical oscillations. For this reason, the introduction of numerical dissipation is required. When using the so called *matrix-dissipation method* [132] the properties of the central differencing method can be made similar to those of the flux-difference splitting approach discussed above. The main idea underlying the second method, Flux-vector splitting, is that the flux can be split into two components according to the sign of certain characteristic variables. In this way a better resolution of shocks and boundary layers can be achieved. This method is more suitable for the Euler equations than for the RANS equations. For the latter flow model, in fact it has been found that the method leads to inaccurate stagnation and wall temperature. The main idea underlying the last method, Total variation diminishing, is that the total variation of any physically admissible solution does not increase in time. This usually is based on non-linear limiters where the variation of the numerical solution is controlled in a non-linear way. More details on the particular implementation of these methods can be found in the books of Blazek [2], Peric [13] or other relative books in computational fluid mechanics.

4.4 Entropy fix

It has been shown that Roe's method for the calculation of convective fluxes at the cell interfaces is not consistent with an entropy inequality (second law of thermodynamics) [133, 134, 135, 136]. As a consequence, the scheme might converge to non-physical solutions such as unphysical expansion shocks, characterised by unphysical entropy

reductions. Woodward and Colella [135] used a variety of different methods to treat the discontinuities of the flow and they highlighted the importance of this problem. To fix this problem a slight modification can be introduced in the construction of the numerical dissipation (flux differences). This modification is known as entropy fix. The entropy fix prevents the occurrence of unphysical features like expansion shocks, which violate the second law of thermodynamics. The entropy fix used in this study is that proposed by Yee in 1989 [134]. In his report, Yee has recommended the following relationship to modify the eigenvalues used to construct the numerical dissipation term of equation (4.3.19):

$$\lambda = \begin{cases} |\lambda| & \lambda \geq \delta \\ (\lambda^2 + \delta^2)/2\delta & \lambda < \delta \end{cases} \quad (4.4.1)$$

where δ is a user given small constant. $\delta = 0$ corresponds to the use of no entropy fix. The larger the value of δ , the more dissipative the scheme becomes. Yee [134] found that for subsonic to low supersonic steady state NACA0012 airfoil simulations, the resolution of the shock waves is good and fairly insensitive to the values of δ in the interval $0.1 \leq \delta \leq 0.125$. Thus, for this type of flows, the use of a constant value of δ in the entropy fix seems to be sufficient. However for hypersonic flows, especially for blunt body flows, the use of a constant δ was found to be inadequate. On the other hand, an entropy fix based on variable value of δ , that is a value depending on the local spectral radius of the flux Jacobian is very beneficial both in terms of solution accuracy and stability and convergence rate of the solver. A proper choice of the entropy parameter δ for higher Mach number flows not only helps in preventing non-physical solutions but can smooth the convergence rate and improve the resolution (i.e. sharpness) of the shocks. However, δ cannot be arbitrarily large. In this reference [134] it is reported that for the blunt body steady state calculations, the solution accuracy can be significantly improved by using the following expression for δ :

$$\delta = \tilde{\delta} (|u_x| + |u_y| + c) \quad (4.4.2)$$

with $0.05 \leq \tilde{\delta} \leq 0.25$. More recently Madrane and Tadmor [137] proposed a new entropy condition combining the work of Harten [136] and Tadmor [138] in order to achieve a better resolution at stagnation point. They recommend the following relation

for λ :

$$\lambda = \begin{cases} |\lambda| & \lambda \geq \delta/2 \\ (2\lambda^2 + \delta^2/2)/2\delta & \lambda < \delta/2 \end{cases} \quad (4.4.3)$$

4.5 Diffusive fluxes

The discretization at the cell faces of the derivatives (i.e. $\frac{\partial u_x}{\partial x}$, $\frac{\partial u_x}{\partial y}$, $\frac{\partial u_x}{\partial y}$, $\frac{\partial u_y}{\partial y}$) appearing in the components of the diffusive fluxes given by equations (B.5) and (B.6) are based on second order finite-differencing, whereas all the face values of the flow variables (density ρ , static pressure p , turbulent kinetic energy K and static temperature T required to calculate the molecular viscosity μ) are by means of simple arithmetic averages of the form:

$$\psi_{i+1/2} = \frac{1}{2} (\psi_i + \psi_{i+1}) \quad (4.5.1)$$

where ψ is any of the above flow variables, ψ_i and ψ_{i+1} are the values at the midpoint of cells i and $i + 1$, respectively, and $\psi_{i+1/2}$ is the sought value at the interface of cell i and $i + 1$.

To illustrate the discretisation of the diffusive fluxes, let us consider the net flux balance of the viscous stresses over a control volume of area S . Such overall flux is given by the integral of the viscous stress over S , namely by:

$$\oint_S \underline{\hat{\tau}} \cdot \mathbf{n} dS \quad (4.5.2)$$

The expression above has to be discretized on the faces of each cell of the computational domain. Hence, the discretised representation of the net flux of the viscous stress on the boundary of cell i, j is given by the sum of the fluxes parallel to the i family of grid lines and parallel to the j family of grid lines. The x-component of the net flux of the viscous stress is thus given by:

$$\begin{aligned}
& [(\hat{t}_{xx}n_x + \hat{t}_{xy}n_y)\Delta S]_{i+1/2,j} + [(\hat{t}_{xx}n_x + \hat{t}_{xy}n_y)\Delta S]_{i,j+1/2} \\
& \quad + [(\hat{t}_{xx}n_x + \hat{t}_{xy}n_y)\Delta S]_{i-1/2,j} \\
& \quad + [(\hat{t}_{xx}n_x + \hat{t}_{xy}n_y)\Delta S]_{i,j-1/2}
\end{aligned} \tag{4.5.3}$$

Whereas the y-component of the net flux is given by:

$$\begin{aligned}
& [(\hat{t}_{xy}n_x + \hat{t}_{yy}n_y)\Delta S]_{i+1/2,j} + [(\hat{t}_{xy}n_x + \hat{t}_{yy}n_y)\Delta S]_{i,j+1/2} \\
& \quad + [(\hat{t}_{xy}n_x + \hat{t}_{yy}n_y)\Delta S]_{i-1/2,j} \\
& \quad + [(\hat{t}_{xy}n_x + \hat{t}_{yy}n_y)\Delta S]_{i,j-1/2}
\end{aligned} \tag{4.5.4}$$

The components τ_{xx} , τ_{xy} and τ_{yy} , of the stress tensor need to be computed on the four faces of each control volume. From the definition of the stress tensor (2.1.3), it follows that:

$$\hat{t}_{xx} = 2(\mu + \mu_T) \left[\frac{\partial u_x}{\partial x} - \frac{1}{3} \left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} \right) \right] - \frac{2}{3} \rho K \tag{4.5.5}$$

$$\hat{t}_{xy} = (\mu + \mu_T) \left[\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right] \tag{4.5.6}$$

$$\hat{t}_{yy} = 2(\mu + \mu_T) \left[\frac{\partial u_y}{\partial y} - \frac{1}{3} \left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} \right) \right] - \frac{2}{3} \rho K \tag{4.5.7}$$

The Cartesian velocity derivatives needed to calculate the components of the molecular stress tensor $\underline{\tau}$ and also those of the turbulent stress tensor $\underline{\tau}^F$ are determined by considering the local generalized curvilinear coordinates (ξ, η) associated with the grid lines, using the chain rule to relate the Cartesian derivatives of the velocity components to their derivatives in the (ξ, η) domain.

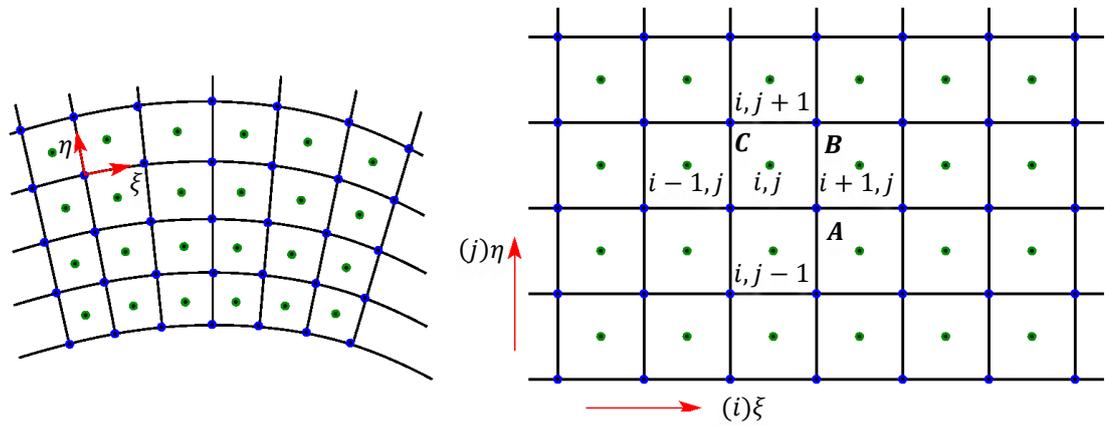


Figure 4.5: Mapping function between Cartesian and Curvilinear coordinates

We introduce a mapping function between the Cartesian and the curvilinear coordinates system (figure 4.5) as:

$$\xi = \xi(x, y) \quad (4.5.8)$$

$$\eta = \eta(x, y) \quad (4.5.9)$$

where the spacing $\Delta\xi$ and $\Delta\eta$ are considered to be equal to 1. Using this mapping function and the chain rule, one can compute all velocity derivatives appearing in the equations (4.5.5), (4.5.6) and (4.5.7) where their expressions are reported in *Appendix I*.

In the energy equation one has to discretise the two terms below:

$$\oint_S (\hat{\mathbf{t}} \cdot \mathbf{u}) \cdot \mathbf{n} dS \quad (4.5.10)$$

$$\oint_S \hat{\mathbf{q}} \cdot \mathbf{n} dS. \quad (4.5.11)$$

The procedure followed to discretize equations (4.5.10) and (4.5.11) is the same explained for the case of the two components of the momentum equation where their expressions can be found in *Appendix I*.

Chapter 5

NUMERICAL INTEGRATION

Numerical methods for solving the large system of algebraic equations resulting from the space-discretisation of the flow and turbulence model equations can be explicit or implicit. After briefly reviewing advantages and disadvantages of both categories, this chapter provides a detailed description of the adopted explicit integration strategy for solving the space-discretised RANS equations and the two equations of the SST turbulence model. The steady RANS equations and the two-turbulence equations are treated as a single set of strongly coupled equations and solved iteratively with the same multi-stage Runge-Kutta smoother. The detrimental effects of the system stiffness resulting from the presence of certain source terms in the turbulence equations are alleviated by treating implicitly such source terms within the Runge-Kutta integration. General time-dependent flow problems are solved using the so-called dual-time-stepping approach, whereby the physical time-derivatives of the space-discretised governing equations are discretised by means of backward finite-difference yielding a system of nonlinear equations for the flow field at each discrete physical time. Each of these systems is then solved using the same Runge-Kutta smoother used for the solution of steady problems. In the case of unsteady periodic flows, the high-dimensional harmonic balance RANS and SST equations are solved, as this results in substantial reductions of run-times with respect to the case in which the time-domain equations are solved. The harmonic balance RANS and SST equations are solved using the same numerical integration strategy used for steady problems. Local time-stepping and centred variable-coefficient implicit residual smoothing are adopted for accelerating the convergence rate of the Runge-Kutta smoother, and a brief description of both methods is also reported. An additional important convergence acceleration technique used by the CFD code of this study is a full approximation scheme multigrid solver, but the description of this feature is provided in the next chapter. The chapter is concluded by the description of two CFL ramping strategies which have been implemented in the CFD code used in this study, and have been found to greatly improve the stability and the convergence rate of complex turbulent problems.

5.1 Explicit and implicit integration

The space-discretisation of the convective and diffusive terms of the steady RANS and SST equations, and the space-discretisation of the source terms of the steady SST turbulence model equations yield a system of non-linear algebraic equations. Such a system can be solved by using an explicit or an implicit time-marching strategy. Time-marching methods rely on re-introducing the time-derivatives of the unknown flow variables at the cell centres into the system of algebraic equations resulting from the space-discretisation of the governing conservation laws. This operation yields a system of ODEs, the steady state of which can be obtained by time-marching, enabled by suitably time-discretising the aforementioned time-derivative. The time-marching starts from a convenient initial solution and is carried out until the sought steady solution is achieved. Denoting by $f(\hat{Q})$ the system of N equations obtained from the space discretisation, and \hat{Q} the array storing the unknowns Q_1, Q_2, \dots, Q_N , the solution of the steady equations can be obtained by time-marching the solution of the following system of ODEs:

$$\frac{\partial \hat{Q}}{\partial t} + f(\hat{Q}) = 0. \quad (5.1.1)$$

The sought steady solution is obtained when $f(\hat{Q})$, which represents the residuals of the steady equations, equals zero.

A simple explicit time-marching method can be implemented by evaluating the residuals term $f(\hat{Q})$ at time level n and using a first order finite-difference to approximate the time-derivative, obtaining:

$$\hat{Q}^{n+1} = \hat{Q}^n - h f(\hat{Q}^n) \quad (5.1.2)$$

where \hat{Q}^n is the value of \hat{Q} at time $t = n h$. Equation (5.1.2) can be used in an iterative fashion to update the stage \hat{Q}^{n+1} until the procedure converges and \hat{Q}^{n+1} becomes an accurate estimate of the steady solution. Each step of the explicit time-marching solution procedure does not require a large number of *floating points operations* (FLOPs), and, from this viewpoint, explicit time marching is quite convenient. Moreover, the approach does not require building and storing Jacobian matrices, and for this reason it is very

convenient in terms of memory usage. On the other hand, there exist bounds on the size of the marching step h . For numerical stability reasons, the value of h must be smaller than scheme-dependent and also problem-dependent thresholds. As a consequence, the number of iterations required to obtain the sought steady solution may become very large. Thus, the overall FLOP count of explicit methods (and therefore the time required to carry out the calculation) can become quite large since each iteration is computationally cheap, but a very large number of iterations is required. For this reason, explicit time-marching methods cannot be used in their simplest form, but require the concurrent use of acceleration techniques such as implicit residual smoothing [81, 139, 140, 141] and multigrid [142, 143].

One of the most widely adopted explicit time-marching strategies for solving the RANS and the turbulence model equations is the explicit multi-stage Runge-Kutta time-marching method, which was first introduced by Jameson to solve the Euler equations [144]. The method was later extended to the solution of the laminar NS equations [145] and the RANS equations augmented with algebraic turbulence models [146, 16] and differential turbulence model equations [60, 59, 147].

Implicit time-marching solution methods also rely on using the time-derivatives of the governing equations to time-march the solution until a steady state is achieved. However, the residual is evaluated at time level $n + 1$ and the time-derivatives are discretised by means of a backward finite-difference. With reference to the system of ODEs represented by equation (5.1.1), implicit approach yields the following system of algebraic equations:

$$\hat{Q}^{n+1} = \hat{Q}^n - h f(\hat{Q}^{n+1}). \quad (5.1.3)$$

It is thus seen that computing the solution of $f(\hat{Q}) = 0$ by applying iteratively equation (5.1.3) requires solving a system of equations at each time level $n + 1$. The fact that the solution update requires the solution of a large system of equations at each time-step is certainly a disadvantage with respect to the explicit time-marching method. The attractiveness of implicit time-marching approach, however, stems from the fact that numerical stability analysis shows that the maximum size of the time-step h usable with implicit time-marching is substantially higher than in the explicit time-marching case. Hence, the overall amount of FLOPs required computing a steady flow solution can be comparable or even smaller than with explicit methods augmented with convergence acceleration algorithms, because, though each implicit time-step is computationally

expensive, the overall number of required time-steps can be substantially smaller than with explicit methods. In other words, using implicit methods, the additional computational cost associated with solving a system of equations at each time level may be outweighed by a significant reduction of the number of time-steps required to achieve the sought steady state. In general, the overall number of FLOPs and/or the run-time to solution needed with explicit and implicit time-marching methods for a given amount of computational resources is quite case-dependent.

In order to illustrate in more detail the solution of steady problems using the most common implementations of implicit time-marching methods, one needs to linearize equation (5.1.3). Performing this operation yields:

$$\left(\frac{I}{h} + A\right) \Delta\hat{Q} = -f(\hat{Q}^n) \quad (5.1.4)$$

where

$$\Delta\hat{Q} = \hat{Q}^{n+1} - \hat{Q}^n \quad (5.1.5)$$

and the Jacobian matrix A is defined by:

$$A = \frac{\partial f(\hat{Q}^n)}{\partial \hat{Q}^n}. \quad (5.1.6)$$

One way of solving (5.1.4) consists of solving the system of linear equations arising at each time step $n + 1$ using an effective, often preconditioned, linear solver such as the generalized minimal residual algorithm [148], a Krylov subspace method. After each linear system is solved, the solution is updated using the relationship:

$$\hat{Q}^{n+1} = \hat{Q}^n + \Delta\hat{Q} \quad (5.1.7)$$

and the implicit time-marching procedure then moves to the next time-step. This type of approach is adopted in the CFD code described in [149, 66, 150].

It should be noted that using this approach and setting the time-step to very large value, thus neglecting the term proportion to $1/h$ in equation (5.1.4) yields Newton's method.

The convergence of this integration procedure is quadratic, and therefore extremely favourable. However, it is very difficult to develop a stable solution procedure based on Newton's method because this algorithm requires the initial solution to be very close to the sought steady solution. For this reason, CFD codes based on Newton's method start by using a relatively small time step h , and this often happens only towards the end of the implicit time-march. More detail on the use of this approach for solving the Navier-Stokes equations can be found in the articles of Zingg [151] and his researchers [152, 153].

Another very popular technique for solving the RANS and turbulence model equations represented by system (5.1.4) by means of implicit time-marching methods consists of approximating the matrix operator on LHS of equation (5.1.4) using the product of 2 (in two dimensions) or 3 (in three dimensions) suitably built simple matrices (e.g. tridiagonal matrices). Each factor or the product refers to a particular direction in space. System (5.1.4) can be solved iteratively, by solving 2 (or 3) simple systems at each iteration. At the end of each iteration the residuals represented by the RHS of equation (5.1.4) is updated using the value of \hat{Q}^{n+1} just calculated. This type of methods is the so called *Approximate Factorization Alternating Direction Implicit technique* (AF-ADI), and it has been used by many researchers [154, 155, 156, 157, 83].

The computational cost of using Newton's method to solve the steady Navier-Stokes equations in terms of run-time may vary significantly depending on the method used to solve the system of linear equations arising at each step of the nonlinear solution process. The memory storage also varies significantly on whether matrix-free or matrix-based implementations of the iterative linear solver at each step of Newton's method are used. When using linear iterative solvers at each Newton's step, the choice of the preconditioner plays a significant role on the convergence of the linear solver, which in turn affects the overall run-time to solution [153, 158]. The main advantage of Newton's method is its quadratic convergence, which allows the sought steady state to be achieved in a very small number of steps. Its main disadvantages include: a) the difficulty of starting the solution process when the initial solution is not sufficiently close to the sought solution, b) the very high memory storage when not using a matrix-free implementation of the linear solver, and c) the high computational cost required to solve each linear system. The convergence rate of AF-ADI methods is usually not as favourable as that of Newton solvers, but AF-ADI solvers have the significant advantage of being matrix-free, which substantially reduce the memory requirement of the simulation.

5.2 Runge-Kutta time-marching

One of the most popular explicit time-marching approaches to the solution of RANS and turbulence model equations is the multi-stage *Runge-Kutta* (RK) time-marching, the integration method used in this study. The use of multi-stage Runge-Kutta time-marching for solving the Euler equations was first introduced by Rizzi and Eriksson [159], who implemented a three stage RK scheme to solve a transonic flow around a wing and fuselage. Jameson and his researchers [144] expanded and optimised the RK method into a four-stage scheme. More recently other researchers like [160, 161, 162, 163] have implemented the RK time-marching algorithm with great success. The underlying idea of this method is that the time-derivative appearing in the Euler and Navier-Stokes equations can be used to time-march the flow field until a steady state is achieved asymptotically. Time-marching is accomplished by discretising the time derivative using optimally designed RK schemes [164, 165].

Once discretized in space, the governing equations are reduced to a set of ordinary differential equations of the same system (5.1.1). The multi-stage RK schemes introduced by Jameson compute the RHS of the system (5.1.2) at several values of \hat{Q} in the interval between $n \Delta t$ and $(n + 1) \Delta t$. Considering a k -stage RK cycle, the sequence of operations required to advance the solution from time level l to time level $l + 1$ is:

$$\begin{aligned}
 \hat{Q}^{(0)} &= \hat{Q}_l \\
 \hat{Q}^{(1)} &= \hat{Q}^{(0)} - a_1 \Delta t f(\hat{Q}^{(0)}) \\
 \hat{Q}^{(2)} &= \hat{Q}^{(0)} - a_2 \Delta t f(\hat{Q}^{(1)}) \\
 &\vdots \\
 &\vdots \\
 \hat{Q}^{(k)} &= \hat{Q}^{(0)} - a_k \Delta t f(\hat{Q}^{(k-1)}) \\
 \hat{Q}_{l+1} &= \hat{Q}^{(k)}
 \end{aligned} \tag{5.2.1}$$

where a_1 , a_2 and a_k are the RK coefficients. The CFD code used in this study uses a four-stage RK4 scheme, and its coefficients are:

$$a_1 = \frac{1}{4}, \quad a_2 = \frac{1}{3}, \quad a_3 = \frac{1}{2}, \quad a_4 = 1 \quad (5.2.2)$$

These coefficients are those proposed by Jameson [81] for use with space discretisation of the Euler equations based on central schemes. Van Leer and his researchers [164, 165] later introduced a method for optimising the RK scheme coefficients, thus providing a tool to maximise the convergence rate of the Euler equations also taking account the type of adopted space discretisation. The coefficients of the 4-stage RK scheme based on upwind space discretisation of Euler equations are:

$$a_1 = 0.1084, \quad a_2 = 0.2602, \quad a_3 = 0.5052, \quad a_4 = 1 \quad (5.2.3)$$

The comparative analysis of the convergence rate of the turbulent solver used in this research using either set of coefficients was beyond the scope of this research, but preliminary investigations have shown that the choice of either set does not significantly vary the convergence rate of the solver for complex turbulent flow problems.

The turbulent flow solver used in this study solves the two systems of algebraic equations resulting from the space discretization of the RANS equations and the turbulence model equations by means of an explicit multigrid algorithm based on explicit time-marching approach, accomplished by using a four-stage RK algorithm with stage coefficients provided by equation (5.2.2). The two systems are solved using a strongly coupled approach [60, 59, 61], whereby the mean flow and the turbulence equations are solved simultaneously in the iterative process. This integration approach has been shown to lead to significantly faster convergence rates than the loosely coupled method [62, 63], whereby the mean flow and the turbulence equations are solved separately and often with different integration methods [147, 152]. As reported later in this chapter, the turbulent CFD code used in this study adopts the strongly coupled integration method for solving the TD formulation of the RANS and the turbulence model equations [39], and the harmonic balance formulation of the two systems [95]. In TD case, the turbulent multigrid solver also features a point-implicit treatment of certain terms arising from the discretization of the physical time-derivatives (see below). This approach is an extension of the stabilization process reported by Melson et al. [65], and it enables the use of fairly high CFL numbers, thus significantly reducing the number of multigrid cycles required to achieve a user-given reduction of the flow residuals. A similar stabilisation procedure has also been successfully developed and applied to the HB case. In this circumstance,

the point-implicit treatment is applied to the source term which is the spectral counterpart of the time derivatives of the conservative variables in the TD case [70, 95].

5.2.1 Turbulent steady problems

For steady turbulent problems the time-derivative appearing in equation (2.7.6) vanishes. The space-discretisation of all remaining terms, leads to a system of non-linear algebraic equations of the form:

$$\hat{R}_\phi(\hat{Q}) = 0 \quad (5.2.4)$$

The entries of the array \hat{Q} are the unknown flow variables at the cells centres of the discretized physical domain. If the computational grid has n_{cell} cells, then the array \hat{Q} can be made up of n_{cell} sub-arrays, each of which stores the number of PDE (n_{pde}) flow unknowns at a particular cell centre. The length of \hat{Q} is therefore ($n_{pde} \times n_{cell}$). The array \hat{R}_ϕ stores the cell residuals, and its structure is the same as that of \hat{Q} . For each cell, the residual of each PDE is obtained by adding the convective fluxes $\hat{\Phi}_c$ and the diffusive fluxes $\hat{\Phi}_d$ through all its faces, and, for the K and ω residuals, by also adding the associated source terms \hat{S} evaluated at the cell centre, given by equations (E.6 - E.10). The unknown flow array \hat{Q} is computed by solving iteratively equation (5.1.1) on the user-given computational grid (the developed turbulent multigrid capability of the CFD code in this study is described in great detail in the next chapter). A fictitious time-derivative $\partial\hat{Q}/\partial\tau$ pre-multiplied by the cell volumes is added to this system, and this yields:

$$V \frac{\partial\hat{Q}}{\partial\tau} + \hat{R}_\phi(\hat{Q}) = 0. \quad (5.2.5)$$

The fictitious time derivative is then discretized with a four-stage RK scheme. The numerical solution is thus marched in pseudo-time until the steady state is achieved. The convergence rate is greatly enhanced by means of *local time-stepping* (LTS), variable-

coefficient central *implicit residual smoothing* (IRS) and *full-approximation scheme* (FAS) *multigrid* (MG) algorithm. The LTS method consists of using an optimal pseudo-time-marching step for each cell. The IRS technique results from applying a smoothing operator to the entire residual field, and this process can be viewed as the application of a Laplacian operator with homogeneous Dirichlet boundary conditions enforced at the boundaries of the computational domain. The MG algorithm is a technique for convergence acceleration resulting from solving the equations on multiple grid levels, including the user-given grid (fine grid), and a sequence of internally built coarse grids obtained by coarsening the fine grid recursively in each direction. The use of MG results in the appearance of a forcing term in the equations to be solved on the coarse grid levels and depending on the residuals of the fine grid. Each method will be explained in more detail in sections 5.3, 5.4 and chapter 6, respectively. Denoting by $\Delta\tau$ the local pseudo-time-step, l the RK cycle counter, k the RK stage index and a_k the k^{th} RK coefficient, the explicit RK iteration for solving the RANS and $K - \omega$ SST turbulent equations is:

$$\begin{aligned}\hat{Q}^{(0)} &= \hat{Q}_l \\ \hat{Q}^{(k)} &= \hat{Q}^{(0)} - a_k \Delta\tau V^{-1} L_{IRS} [\hat{R}_\phi(\hat{Q}^{(k-1)}) + f_{MG}] \\ \hat{Q}_{l+1} &= \hat{Q}^{(k)}\end{aligned}\tag{5.2.6}$$

where L_{IRS} denotes the IRS operator, and f_{MG} is the MG forcing function, which is zero only on the fine grid. The diagonal matrix V stores the volumes of the grid cells. It can 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.

The iteration (5.2.6) has a very poor convergence rate, due to the stiffness of the iterative operator caused by the large negative source terms of the turbulence model, such as $-D_K$ and $-D_\omega$. To alleviate this problem, a semi-implicit integration strategy can be adopted: the negative source terms of the turbulence equations are evaluated at the RK stage k rather than at the stage $k - 1$. Liu and Zheng [59] adopted this approach to develop an efficient strongly coupled MG iteration for the compressible RANS equations coupled with the standard $K - \omega$ model. Those authors treat implicitly also the negative source term proportional to $-\nabla \cdot \mathbf{u}$ when the velocity divergence is positive. The turbulent RK iteration implemented in this study uses an approach similar to that developed in [59] and later adopted by other researchers too [71], but also presents important differences,

highlighted below, with respect to the scheme of Liu and Zheng, due primarily to the modelling differences of the standard $K - \omega$ and the $K - \omega SST$ models.

The equations of Wilcox's $K - \omega$ turbulence model can be written as:

$$\frac{\partial(\rho K)}{\partial t} + R_K(\rho K, \rho \omega) = 0 \quad (5.2.7)$$

$$\frac{\partial(\rho \omega)}{\partial t} + R_\omega(\rho K, \rho \omega) = 0 \quad (5.2.8)$$

where

$$R_K(\rho K, \rho \omega) = \frac{1}{V} (\Phi_{c_K} - \Phi_{D_K}) - S_K \quad (5.2.9)$$

$$R_\omega(\rho K, \rho \omega) = \frac{1}{V} (\Phi_{c_\omega} - \Phi_{D_\omega}) - S_\omega \quad (5.2.10)$$

where Φ_{c_K} and Φ_{c_ω} are the discrete form of the convective flux of K and ω respectively. The symbols Φ_{D_K} and Φ_{D_ω} denote the discrete form of the diffusive flux of K and ω respectively. The symbols S_K and S_ω denote the cell values of the source terms of K and ω equations respectively. The expressions of the terms appearing on the RHS of equation (5.2.9) and (5.2.10) are:

$$\Phi_{c_K} = \sum_{is=1}^4 [(\rho K \mathbf{u}_n) \cdot \Delta \mathbf{S}]_{is} \quad (5.2.11)$$

$$\Phi_{c_\omega} = \sum_{is=1}^4 [(\rho \omega \mathbf{u}_n) \cdot \Delta \mathbf{S}]_{is} \quad (5.2.12)$$

$$\Phi_{D_K} = \sum_{is=1}^4 [(\mu + \sigma_K \mu_T) \nabla K \cdot \Delta \mathbf{S}]_{is} \quad (5.2.13)$$

$$\Phi_{D_\omega} = \sum_{is=1}^4 [(\mu + \sigma_\omega \mu_T) \nabla \omega \cdot \Delta \mathbf{S}]_{is} \quad (5.2.14)$$

$$S_K = \mu_T P_d - \frac{2}{3} (\nabla \cdot \mathbf{u}) \rho K - \frac{\beta^*}{\rho} (\rho K)(\rho \omega) \quad (5.2.15)$$

$$S_\omega = \gamma \rho P_d - \gamma \frac{2}{3} (\nabla \cdot \mathbf{u}) \rho \omega - \frac{\beta}{\rho} (\rho \omega)^2 \quad (5.2.16)$$

where the expression of P_d is given by equation (2.4.8). The turbulent viscosity that appears in the second term of the right hand side of equation (2.4.7) does no longer appear in equation (5.2.16) as it has been replaced from the equation (2.4.3). The $\mu_T P_d$ and $\gamma \rho P_d$ terms are the major contributors to the production of K and ω respectively. Those two source terms are always positive. The terms $-\frac{2}{3} (\nabla \cdot \mathbf{u}) \rho K$ and $-\gamma \frac{2}{3} (\nabla \cdot \mathbf{u}) \rho \omega$ provide minor contributions to the production of K and ω , but they can be either positive or negative. The terms $-\frac{\beta^*}{\rho} (\rho K)(\rho \omega)$ and $-\frac{\beta}{\rho} (\rho \omega)^2$ are destruction terms, and are always negative. Thus, they annihilate K and ω : the larger these terms, the faster K and ω decay, but the system also becomes stiffer due to the larger negative eigenvalues.

The explicit time-marching formula for the K and ω equations used at each stage of the multi-stage scheme can be modified to treat part of the source terms implicitly reducing the equation stiffness and thus improving the convergence rate of the turbulent equations. If we define

$$\Delta^+ = \max\left(0, \frac{2}{3} \nabla \cdot \mathbf{u}\right) \quad (5.2.17)$$

all negative source terms of the K and ω equations can be moved to the LHS of the equations (5.2.7) and (5.2.8) to form a semi-implicit time marching formula. Let us rewrite equation (5.2.7) as:

$$\begin{aligned} \frac{\partial}{\partial t}(\rho K) + \frac{1}{V} (\Phi_{c_k} - \Phi_{D_k}) - \mu_T P_d + \Delta^+ \rho K + \Delta^- \rho K \\ + \frac{\beta^*}{\rho} (\rho K)(\rho \omega) = 0 \end{aligned} \quad (5.18)$$

where

$$\Delta^- = \frac{2}{3} \nabla \cdot \mathbf{u} - \Delta^+ . \quad (5.2.19)$$

The general RK step with semi-implicit treatment can be written as

$$\begin{aligned}
& \frac{(\rho K)^k - (\rho K)^0}{a_k \Delta\tau} + \frac{1}{V} (\Phi_{cK} - \Phi_{DK})^{k-1} - \frac{(\mu_T P_d)^{k-1}}{V} \\
& + (\Delta^+ \rho K)^k + \underline{(\Delta^+ \rho K)^{k-1}} - (\Delta^+ \rho K)^{k-1} \\
& + \underline{(\Delta^- \rho K)^{k-1}} + \left(\frac{\beta^*}{\rho} (\rho K)(\rho \omega) \right)^k \\
& + \underline{\left(\frac{\beta^*}{\rho} (\rho K)(\rho \omega) \right)^{k-1}} - \left(\frac{\beta^*}{\rho} (\rho K)(\rho \omega) \right)^{k-1} = 0
\end{aligned} \tag{5.2.20}$$

where the superscript k denotes the RK stage and a_k is the RK stage coefficient. The underlined terms form the term R_K^{k-1} , which denotes the complete cell residual array of ρK . Therefore one obtains:

$$\begin{aligned}
& \left(\frac{1}{\alpha_k \Delta\tau} + \Delta^+ \right) (\rho K)^k - \frac{(\rho K)^0}{\alpha_k \Delta\tau} - \Delta^+ (\rho K)^{k-1} \\
& + \frac{\beta^*}{\rho} [(\rho K)^k (\rho \omega)^k - (\rho K)^{k-1} (\rho \omega)^{k-1}] \\
& = -R_K^{k-1}
\end{aligned} \tag{5.2.21}$$

in which the two following approximations have been used:

$$(\Delta^+)^k = (\Delta^+)^{k-1} \tag{5.2.22}$$

$$\frac{1}{\rho^k} = \frac{1}{\rho^{k-1}} \tag{5.2.23}$$

Following the same procedure for the ω equation one obtains the following semi-implicit iteration for updating ω :

$$\begin{aligned}
& \left(\frac{1}{\alpha_k \Delta\tau} + \gamma \Delta^+ \right) (\rho \omega)^k - \frac{(\rho \omega)^0}{\alpha_k \Delta\tau} - \gamma (\Delta^+ \rho \omega)^{k-1} \\
& + \frac{\beta}{\rho} [(\rho \omega)^k (\rho \omega)^k - (\rho \omega)^{k-1} (\rho \omega)^{k-1}] \\
& = -R_\omega^{k-1}
\end{aligned} \tag{5.2.24}$$

where R_ω^{k-1} denotes the complete cell residual array of $\rho\omega$. We now have to linearize equations (5.2.21) and (5.2.24). Introducing the following definitions:

$$\delta\rho K = (\rho K)^k - (\rho K)^{k-1} \quad (5.2.25)$$

$$\delta\rho\omega = (\rho\omega)^k - (\rho\omega)^{k-1} \quad (5.2.26)$$

the linearized form of the terms $(\rho K)^k(\rho\omega)^k$ and $(\rho\omega)^k(\rho\omega)^k$, are found to be respectively:

$$\begin{aligned} (\rho K)^k (\rho\omega)^k &= ((\rho K)^{k-1} + \delta\rho K) ((\rho\omega)^{k-1} + \delta\rho\omega) \\ &\cong (\rho K)^{k-1} (\rho\omega)^{k-1} + (\rho K)^{k-1} \delta\rho\omega \\ &\quad + (\rho\omega)^{k-1} \delta\rho K \end{aligned} \quad (5.2.27)$$

$$\begin{aligned} (\rho\omega)^k (\rho\omega)^k &= ((\rho\omega)^{k-1} + \delta\rho\omega) ((\rho\omega)^{k-1} + \delta\rho\omega) \\ &\cong (\rho\omega)^{k-1} (\rho\omega)^{k-1} + 2 (\rho\omega)^{k-1} \delta\rho\omega \end{aligned} \quad (5.2.28)$$

Inserting expression (5.2.27) into the equation (5.2.21), the final equation for updating ρK becomes:

$$\begin{aligned} [1 + a_k \Delta\tau (\Delta^+ + \beta^* \omega^{k-1})] (\rho K)^k + \alpha_k \Delta\tau \beta^* K^{k-1} (\rho\omega)^k \\ = (\rho K)^0 + \alpha_k \Delta\tau (\Delta^+ + 2 \beta^* \omega^{k-1}) (\rho K)^{k-1} \\ - \alpha_k \Delta\tau \{R_K^{k-1}\} \end{aligned} \quad (5.2.29)$$

Similarly, inserting equation (5.2.28) into equation (5.2.24), the equation for updating $\rho\omega$ becomes:

$$\begin{aligned} [1 + a_k \Delta\tau (\gamma \Delta^+ + 2 \beta \omega^{k-1})] (\rho\omega)^k \\ = (\rho\omega)^0 + \alpha_k \Delta\tau (\gamma \Delta^+ + 2 \beta \omega^{k-1}) (\rho\omega)^{k-1} \\ - \alpha_k \Delta\tau \{R_\omega^{k-1}\} \end{aligned} \quad (5.2.30)$$

It is thus seen that, using the semi-implicit integration of ρK and $\rho\omega$, the update of ρK depends on the new value of $\rho\omega$. Due to this partial coupling of the K and ω equations, one must update $\rho\omega$ first, and ρK thereafter. The fully coupled RK iteration for updating both the RANS and the $K - \omega$ variables incorporating IRS and MG can be written as:

$$\begin{aligned}
\hat{Q}^{(0)} &= \hat{Q}_l \\
(I + a_k \Delta\tau A) \hat{Q}^{(k)} &= \hat{Q}^{(0)} + a_k \Delta\tau A \hat{Q}^{(k-1)} \\
&\quad - a_k \Delta\tau V^{-1} L_{IRS} [\hat{R}_\phi(\hat{Q}^{(k-1)}) + f_{MG}] \\
\hat{Q}_{l+1} &= \hat{Q}^{(k)}
\end{aligned} \tag{5.2.31}$$

The matrix A is block-diagonal and it has size $(n_{cell} \times n_{cell})$. The only nonzero elements of each $(n_{pde} \times n_{pde})$ block on the diagonal of A are those of the bottom right (2×2) partition, and, due to the structure of these blocks, this occurrence results in the necessity of updating $\rho\omega$ before ρK . The definitions of the matrix A is:

$$A(5:6, 5:6) = A_{K-\omega} = \begin{bmatrix} \Delta^+ + \beta^* \omega & \beta^* K \\ 0 & \gamma \Delta^+ + 2 \beta \omega \end{bmatrix} \tag{5.2.32}$$

in which all variables are evaluated at the RK stage $k - 1$.

In the case of the SST model, the equation of the turbulent kinetic energy K is identical to that of Wilcox $K - \omega$ model, but for the equation of the specific dissipation rate ω has an additional cross-diffusion term CD_{term} . In addition the turbulent viscosity is given by the equation (2.5.4). The source term of the ω equation can be written as:

$$S_\omega = \gamma \rho P_d - \gamma \gamma^* \frac{2}{3} (\nabla \cdot \mathbf{u}) \frac{\rho K}{\nu_T} - \frac{\beta}{\rho} (\rho \omega)^2 + CD_{term} \tag{5.2.33}$$

The cross-diffusion term CD_{term} can be positive or negative depending on the local flow conditions, and therefore, when negative, it could be treated like Δ^+ in the semi-implicit integration. However, this approach would make the implementation substantially more complex and less efficient because the term $CD_{K\omega}$ depends on ∇K and $\nabla \omega$. The evaluation of these gradients at stage k would couple the update process of several cells, thus requiring the inversion of significantly larger systems. For this reason, it has been preferred to treat the term CD_{term} explicitly regardless of its sign. It should be noted that this term is absent in the standard $K - \omega$ model. Following a procedure similar to that adopted for the case of the standard $K - \omega$ process, the iteration for updating ρK and $\rho \omega$ in the SST case is found to be formally identical to equations (5.2.31). The expression of the matrix A , however, is different and is given respectively by:

$$A(5:6, 5:6) = A_{SST} = \begin{bmatrix} \Delta^+ + \beta^* \omega & \beta^* K \\ \frac{\gamma \Delta^+}{\nu_T} & 2 \beta \omega \end{bmatrix} \quad (5.2.34)$$

One important difference between the semi-implicit integration of the standard $K - \omega$ model reported in references, [60, 59] and that of the SST model is that, in the former case, $\rho\omega$ can be updated independently of ρK . This is however not possible in the SST case, since the element (2, 1) of A_{SST} is not zero. Due to this feature, a (2 x 2) matrix inversion is required at each grid cell to update ρK and $\rho\omega$. The different turbulent variables update of $K - \omega$ and SST models occurs because the expression of the turbulent viscosity of the former model is obtained by setting $F_2 = 0$ in equation (2.5.4). In general, when using the SST turbulence model, one would adopt equation (5.2.34) rather than equation (5.2.32). However, numerical experiments performed in this study revealed that the results computed with either approach present zero differences for low-speed flows and transonic flows for Mach number less than one. For this reason and due to lower computational cost associated with the use of equation (5.2.32), all low-speed flow analyses presented in chapter 8 are based on the use of this equation. The errors associated with the choice of equation (5.2.32) when using the SST model may be significant for high-speed problems (supersonic), due to the higher values of $\nabla \cdot \mathbf{u}$ caused by compressibility effects.

In the update process performed by the RK iteration, the new estimate of ω is prevented from assuming unphysical low values by limiting it with a minimum threshold based on the production term P_d of equation (2.4.8), following the guidelines of [59]. This limiter is given by the following expression:

$$(\rho \omega)_{min} = \gamma \gamma^* \rho \sqrt{P_d} \quad (5.2.35)$$

5.2.2 Turbulent time-domain problems

In order to compute the flow state at each physical time $n + 1$, the physical time-derivative of system (2.7.6) is discretized with a second order backward finite difference computed at the aforementioned time level $n + 1$, and all spatial terms are also evaluated

at the same physical level. The set of nonlinear algebraic equations resulting from this implicit time-discretization and the space-discretization of system (2.7.6) already discussed in chapter 4 is then solved using the same explicit time-marching procedure, and the same LTS, IRS and MG convergence acceleration techniques used for solving the steady equations. The discretization of the physical time-derivative of the unknown flow state by means of a second order backward finite difference and the introduction of the fictitious time-derivative $V \left(\frac{\partial \hat{Q}}{\partial \tau} \right)^{n+1}$ required to implement the explicit time-marching solution process yield the equation:

$$V \frac{\partial \hat{Q}^{n+1}}{\partial \tau} + \hat{R}_g(\hat{Q}^{n+1}) = 0 \quad (5.2.36)$$

where

$$\hat{R}_g(\hat{Q}^{n+1}) = \frac{3 \hat{Q}^{n+1} - 4 \hat{Q}^n + \hat{Q}^{n-1}}{2 \Delta t} V + \hat{R}_\phi(\hat{Q}^{n+1}). \quad (5.2.37)$$

The symbol \hat{R}_g denotes a residual vector which also includes the source terms associated with the discretization of the physical time-derivative $\frac{\partial \hat{U}}{\partial t}$ contained in equation (2.7.6). Note that, also for time-dependent problems with moving bodies, the matrix V is independent of the physical time-level (denoted by the superscripts $n + 1$, n and $n - 1$) because in this study only rigid-body grid motion is considered. The symbol Δt indicates the user-given physical time-step. Equation (5.2.36) can thus be viewed as a system of $(n_{pde} \times n_{cell})$ ordinary differential equations in which the unknown is \hat{Q}^{n+1} , the flow state at time level $n + 1$. The calculation of \hat{Q}^{n+1} is performed iteratively by discretizing the fictitious time-derivative $\left(\frac{\partial \hat{Q}}{\partial \tau} \right)^{n+1}$ of equation (5.2.36) with the same four-stage RK scheme, described in the previous subsection, and marching the equations in pseudo-time until a steady state is achieved. Such steady state is the flow solution at the physical time being considered. Similarly to the case of steady flow problems, the convergence rate is enhanced by means of LTS, variable-coefficient central IRS and MG which will be explained in the following sections. Once the flow solution at the physical time-level $n + 1$ has been calculated, the array \hat{Q}^n is moved to \hat{Q}^{n-1} , the array \hat{Q}^{n+1} is moved to \hat{Q}^n , and the calculation of a new time-level is started. This procedure is the so-called dual-time-stepping approach to the integration of time-dependent problems [66, 166, 39].

This solution procedure may become unstable when the physical time-step Δt is significantly smaller than the pseudo-time-step $\Delta \tau$. This instability was reported in [167], and thoroughly investigated by Melson et al. [65]. The latter study, considering the simulation of turbulent flow problems by means of the thin-layer Navier–Stokes and the algebraic Baldwin–Lomax turbulence model [16] elegantly solved the stability problem by treating implicitly the term \hat{Q}^{n+1} of the physical time-derivative within the RK integration process. This strategy has also been implemented in the turbulent TD flow solver used in this study. The residual R_g is split into the contribution depending on the \hat{Q}^{n+1} term of the physical time derivative, and a term R_d equal to the difference of R_g and the aforementioned \hat{Q}^{n+1} term:

$$\hat{R}_g(\hat{Q}^{n+1}) = \frac{V}{\Delta t} \left[\frac{3}{2} \hat{Q}^{n+1} + g(\hat{Q}^n, \hat{Q}^{n-1}) \right] + \hat{R}_\phi(\hat{Q}^{n+1}) \quad (5.2.38)$$

where

$$\hat{g}(\hat{Q}^n, \hat{Q}^{n-1}) = -2 \hat{Q}^n + 0.5 \hat{Q}^{n-1}. \quad (5.2.39)$$

Equation (5.2.38) can also be written as:

$$\hat{R}_g(\hat{Q}^{n+1}) = \hat{R}_d(\hat{Q}^{n+1}) + \frac{3V}{2\Delta t} \hat{Q}^{n+1} \quad (5.2.40)$$

with

$$\hat{R}_d(\hat{Q}^{n+1}) = \frac{V}{\Delta t} \left[-2 \hat{Q}^n + 0.5 \hat{Q}^{n-1} \right] + \hat{R}_\phi(\hat{Q}^{n+1}). \quad (5.2.41)$$

Including the additional terms resulting from the backward discretisation of the physical time-derivative into equation (5.2.21), the iteration used for updating ρK in the case of steady problems, and treating such terms as described above, yields the following iteration for updating the turbulent kinetic energy in case of time-dependent problems:

$$\begin{aligned}
 & \left(\frac{1}{\alpha_k \Delta \tau} + \frac{3}{2 \Delta t} + \Delta^+ \right) (\rho K)^k - \frac{(\rho K)^0}{\alpha_k \Delta \tau} - \Delta^+ (\rho K)^{k-1} \\
 & + \frac{\beta^*}{\rho} [(\rho K)^k (\rho \omega)^k - (\rho K)^{k-1} (\rho \omega)^{k-1}] \\
 & = -R_K^{k-1}.
 \end{aligned} \tag{5.2.42}$$

Starting from equation (5.2.24) and following the same steps, the iteration used for updating the specific dissipation rate in the case of time-dependent problems is found to be:

$$\begin{aligned}
 & \left(\frac{1}{\alpha_k \Delta \tau} + \frac{3}{2 \Delta t} + \gamma \Delta^+ \right) (\rho \omega)^k - \frac{(\rho \omega)^0}{\alpha_k \Delta \tau} - \gamma (\Delta^+ \rho \omega)^{k-1} \\
 & + \frac{\beta}{\rho} [(\rho \omega)^k (\rho \omega)^k - (\rho \omega)^{k-1} (\rho \omega)^{k-1}] \\
 & = -R_\omega^{k-1}
 \end{aligned} \tag{5.2.43}$$

Equations (5.2.42) and (5.2.43) can be linearized following the same procedure reported above for the steady case. After doing so, the RK cycle for updating all conservative flow variables at physical time level $n + 1$ in the case of turbulent time-dependent flows reads:

$$\begin{aligned}
 \hat{Q}^{(0)} &= \hat{Q}_l \\
 (I + a_k (\beta_{TD} I + \Delta \tau A)) \hat{Q}^{(k)} \\
 &= \hat{Q}^{(0)} + a_k \Delta \tau A \hat{Q}^{(k-1)} - a_k \Delta \tau V^{-1} \hat{R}_d(\hat{Q}^{(k-1)}) \\
 \hat{Q}_{l+1} &= \hat{Q}^{(k)}
 \end{aligned} \tag{2.44}$$

where \hat{Q}_l is shorthand for \hat{Q}_l^{n+1} and

$$\beta_{TD} = \frac{1.5 \Delta \tau}{\Delta t} \tag{5.2.45}$$

The stability analysis of [65] shows that the stability of algorithm (5.2.44) no longer depends on the ratio $\Delta \tau / \Delta t$. However, iteration (5.2.44) is still unsuitable when IRS and MG are also used, because both acceleration techniques have to be applied to a residual term that vanishes at convergence, and this is not the case of R_d . The solution is to

introduce the residual \hat{R}_g which does vanish at convergence. Using equation (5.2.40) one finds:

$$\Delta\tau \hat{R}_d(\hat{Q}^{(k-1)}) = -\beta_{TD} V \hat{Q}^{(k-1)} + \Delta\tau \hat{R}_g(\hat{Q}^{(k-1)}). \quad (5.2.46)$$

Inserting this expression into equation (5.2.44) and performing some algebraic transformation, one finds that the IRS-MG-tailored counterpart of algorithm (5.2.44) is:

$$\begin{aligned} \hat{Q}^{(0)} &= \hat{Q}^n \\ (I + a_k (\beta_{TD} I + \Delta\tau A)) \hat{Q}^{(k)} &= \hat{Q}^{(0)} + a_k \Delta\tau (\beta_{TD} I + \Delta\tau A) \hat{Q}^{(k-1)} \\ &\quad - a_k \Delta\tau V^{-1} L_{IRS} [\hat{R}_g(\hat{Q}^{(k-1)}) + f_{MG}] \\ \hat{Q}^{n+1} &= \hat{Q}^{(k)} \end{aligned} \quad (5.2.47)$$

Note that the matrix multiplying $\hat{Q}^{(k)}$ at the second line of algorithm (5.2.47) is block-diagonal with N_{cell} blocks. In each block the top left (4×4) partition is proportional to the identity matrix through the coefficient $(1 + a_k \beta_{TD})$, the bottom right (2×2) partition is given by the sum of the (2×2) identity matrix multiplied by $(1 + a_k \beta_{TD})$ and a non-diagonal (2×2) block given by equation (5.2.32) or (5.2.34), depending on whether the exact or approximate update of $\rho\omega$ is used, and all other entries are zero. Similarly to the case of the integration of the steady equations, this structure of the matrix pre-multiplying $\hat{Q}^{(k)}$ results in the coupling of the update process of the turbulent variables, whereas it still enables the four mean flow variables to be updated without any actual matrix inversion. Due to the fact that the \hat{Q}^{n+1} term arising from the backward finite-difference of the physical time-derivative is evaluated at stage k , algorithm (5.2.47) is said to be based a *point-implicit Runge-Kutta* (PIRK) integration of the time-dependent mean flow and turbulence equations. The standard *fully explicit Runge-Kutta* (FERK) integration method is retrieved by setting $\beta_{TD} = 0$ in this algorithm. The integration scheme of the steady equations is instead obtained by also replacing \hat{R}_g with \hat{R}_ϕ in algorithm (5.2.47). Several numerical tests, including the analyses of the oscillating wing presented in chapter 8 (*results*), have highlighted that the turbulent PIRK integration significantly improves the stability of the fully-coupled integration, enabling stable pseudo-time-marching with larger CFL numbers than with the standard FERK

integration. This yields significant reductions of runtimes, due to the reduction of the overall number of MG cycles required to achieve a user-given reduction of the flow residuals.

5.2.3 Turbulent harmonic balance problems

The only formal difference between the integral harmonic balance formulation of the system of RANS and SST equations (3.3.11) and the integral TD formulation (2.7.6) is that the time-derivative of the latter system is replaced by a volumetric source term proportional to ω and a combination of the flow snapshots resulting from the flow harmonics retained in the Fourier representation of the sought periodic flow. Such a volumetric source term corresponds to the spectral derivative representation of the high-dimensional harmonic balance formulation. The system of non-linear equations resulting from the space discretisation of the harmonic balance RANS and SST equations is solved using the same RK smoother used for steady problems and described in section 5.2.1. A fictitious time-derivative $d\hat{Q}/d\tau$ pre-multiplied by the cell volumes is added to this system, and this operation yields the following system of ordinary differential equations:

$$\frac{d\hat{Q}_H}{d\tau} + V^{-1} \hat{R}_{g_H}(\hat{Q}_H) = 0 \quad (5.2.48)$$

where

$$\hat{R}_{g_H}(\hat{Q}_H) = \omega V D \hat{Q}_H + \hat{R}_{\phi_H}(\hat{Q}_H) \quad (5.2.49)$$

The array \hat{Q}_H is made up of $2N_H + 1$ sub-arrays. Each sub-array represents the flow state or snapshot at one of the physical times defined by equation (3.2.26). Thus the array \hat{Q}_H can be written as:

$$\hat{Q}_H = [\hat{Q}(t_0) \quad \hat{Q}(t_1) \quad \hat{Q}(t_2) \quad \dots \quad \hat{Q}(t_{2N_H-1}) \quad \hat{Q}(t_{2N_H})]^T \quad (5.2.50)$$

where each sub-array $\hat{Q}(t_i)$ has length $(n_{pde} \times n_{cell})$ and denotes the unknown flow

variables at the cells centres at time t_i . The arrays \hat{R}_{ϕ_H} and \hat{R}_{g_H} have the same structure. The former stores the cell residuals associated with the convective and diffusive fluxes and the turbulent source terms. The latter is the sum of the residuals \hat{R}_{ϕ_H} and the source term $\omega VD\hat{Q}_H$. The matrix D is defined by equation (3.2.36) and the number of unknowns of the system (5.2.48) is equal to $(n_{pde} \times n_{cell} \times (2N_H + 1))$. The source term is treated implicitly for the stability reasons discussed in [70]. As a result discretising the fictitious time-derivative of equation (5.2.48) with the same multi-stage RK scheme used for solving the steady equations, and considering the source term of Equation (5.2.49) at stage k rather than at stage $(k - 1)$ yields the following RK algorithm:

$$\begin{aligned}
 \hat{Q}_H^{(0)} &= \hat{Q}_{H_l} \\
 (I + a_k \Delta\tau_H (A + \omega D)) \hat{Q}_H^{(k)} \\
 &= \hat{Q}_H^{(0)} + a_k \Delta\tau_H A \hat{Q}_H^{(k-1)} - a_k \Delta\tau_H V_H^{-1} \hat{R}_{\phi_H} (\hat{Q}_H^{(k-1)}) \quad 2.51 \\
 \hat{Q}_{H_{l+1}} &= \hat{Q}_H^{(k)}
 \end{aligned}$$

The symbol $\Delta\tau_H$ denotes an array made up of $2N_H + 1$ sub-arrays, one for each harmonic. These sub-arrays have length n_{cell} and they store the local time step of each cell. The matrix A has $2N_H + 1$ sub-arrays and each of them has size $(n_{cell} \times n_{cell})$. Equation (5.2.51) is still not suitable for using the MG and IRS acceleration techniques. This is because these methods have to be applied to a residual term that vanishes at convergence. Therefore, the residual array \hat{R}_{ϕ_H} has to be replaced by the residual \hat{R}_{g_H} by using the equation (5.2.49), because the residual \hat{R}_{g_H} becomes zero at convergence. By suitably combining equations (5.2.49) and (5.2.51), the MG and IRS enhanced RK iteration for solving the HB equations is found to be:

$$\begin{aligned}
 \hat{Q}_H^{(0)} &= \hat{Q}_{H_l} \\
 (I + a_k \Delta\tau_H (A + \omega D)) \hat{Q}_H^{(k)} \\
 &= \hat{Q}_H^{(0)} + a_k \Delta\tau_H (A + \omega D) \hat{Q}_H^{(k-1)} \\
 &\quad - a_k \Delta\tau_H V_H^{-1} L_{IRS} \left[\hat{R}_{g_H} (\hat{Q}_H^{(k-1)}) + f_{MGH} \right] \\
 \hat{Q}_{H_{l+1}} &= \hat{Q}_H^{(k)}
 \end{aligned} \quad (5.2.52)$$

where the array of the HB MG forcing term f_{MGH} has the same structure of the array \hat{Q}_H . The HB IRS operator has the same block structure of A . The use of the turbulent PIRK HB smoother (5.2.52) enables the use of significantly larger CFL numbers than the use of its FERK counterpart. Moreover, the higher stability of this PIRK relative to that of the FERK iteration increases significantly with N_H .

It should be noted that the computational cost reduction achieved by using equation (5.2.32) rather than equation (5.2.34) is even higher when using the SST turbulence model with the HB RANS and SST solver [70]. This is because, using the stabilised iteration (5.2.52) and the SST structure of the matrix A (equation (5.2.34)), the size of the matrix inversions at each grid cell required to update all N_H complex harmonic components of ρK and $\rho\omega$ is $[2(2N_H + 1) \times 2(2N_H + 1)]$. Conversely, the independence of the ω update on the new value of K obtained by using equation (5.2.32) requires only the inversion of block size $[(2N_H + 1) \times (2N_H + 1)]$, for updating the harmonic components of K and ω . For highly nonlinear periodic flows, the value of N_H needed for a satisfactory time-resolution often exceeds 5, and this implies that n_{cell} matrices of size $[11 \times 11]$ or more have to be inverted using the $K - \omega$ structure of (5.2.32), and n_{cell} matrices of size $[22 \times 22]$ or more have to be inverted using the SST structure of (5.2.34). Since these n_{cell} matrices are dense and unstructured, Gaussian elimination is used for their inversion, and the computational cost of such inversions is proportional to the third power of the system size. Therefore, the ratio of the computational cost for updating the harmonics of the SST turbulence variables using equation (5.2.34) and (5.2.32) is 8, and therefore the use of equation (5.2.32) rather than equation (5.2.34) for updating the harmonics of the SST turbulence variables yields a reduction of the computational cost of nearly one order of magnitude. Like in the TD case, the errors associated with the choice of equation (5.2.32) when using the SST model may be significant for high-speed problems for Mach number greater than 1, due to the higher values of $\nabla \cdot \mathbf{u}$ caused by compressibility effects. However, it should be noted that the primary cause of the possible inadequacy of equation (5.2.32) for updating the SST turbulence variables for high-speed flows, is not the non-negligible magnitude of $\nabla \cdot \mathbf{u}$, but rather the SST expression of μ_T , which in general prevents decoupling the semi-implicit update of ρK and $\rho\omega$. Other two-equation turbulence models, including the Baseline model of Menter [40], feature expressions of μ_T structurally similar or identical to that of Wilcox's $K - \omega$ model, and therefore enable an exact computationally cheaper update of the turbulence variables for both low- and high-speed flows.

5.3 Local time-stepping

The time step $\Delta\tau$ that appears in equations (5.2.31) and (5.2.47) needs to be defined. In explicit time-marching methods, the time-step is subject to an upper threshold dictated by numerical stability requirements [2, 13]. This upper threshold depends on the numerical scheme adopted to solve the problem at hand, on local flow field parameters such as the local flow and sound speeds, and on local characteristic length. The dependence of the time step on the scheme is represented by the *Courant-Friedrichs-Lewy* (CFL) number, a parameter representing the maximum time step suitably nondimensionalized using local flow variables and characteristic lengths. For a given numerical scheme, one could compute the local time step for each cell of the computational grid and take the minimum value to time-march the whole solution using a single time step. This is the so-called *Global time stepping* approach. However, it is extremely inefficient when using time-marching methods to solve steady problems. This is because the speed by which the solution proceeds towards convergence will be heavily reduced in all regions where the time step can be higher than in the area requiring the minimum time step determined with the stability analysis. Since the time-accuracy is not important when using time-marching methods to solve steady problems, a much more effective solution approach consists of using a *local time stepping* approach that is using the maximum possible time step dictated by stability analysis for each cell of the computational domain. By doing so, a significant acceleration of the convergence process is achieved. In the case of the multi-dimensional RANS and SST equations different expressions of the local time-step can be used [168, 169, 170]. The definition of the local time step adopted by the CFD code used in this study is:

$$\Delta\tau = CFL \frac{V}{\Lambda_i^c + \Lambda_j^c + \Lambda_i^v + \Lambda_j^v} \quad (5.3.1)$$

where the symbols Λ_i^c and Λ_j^c denote the absolute value of the maximum eigenvalues of the convective flux Jacobian based on the two contravariant velocity components and the sound speed. The expressions of Λ_i^c and Λ_j^c are

$$\Lambda_i^c = (|\mathbf{u} \cdot \mathbf{n}_i| + c) \Delta S_i \quad (5.3.2)$$

$$\Lambda_j^c = (|\mathbf{u} \cdot \mathbf{n}_j| + c) \Delta S_j \quad (5.3.3)$$

where symbols \mathbf{n} denotes the normal vector on x- and y- direction whereas symbol c denotes the speed of sound. The symbol ΔS_i denotes the mean size of the cell faces with normal in the x direction, and ΔS_j denotes the mean size of the cell faces with normal in the y direction. The symbol Λ_i^v and Λ_j^v in equation (5.3.1) denotes the viscous eigenvalues and its expression is:

$$\Lambda_i^v = \frac{4 \gamma}{\rho} \left(\frac{\mu}{P_r} + \frac{\mu_T}{P_{rT}} \right) \frac{\Delta S_i^2}{V} \quad (5.3.4)$$

$$\Lambda_j^v = \frac{4 \gamma}{\rho} \left(\frac{\mu}{P_r} + \frac{\mu_T}{P_{rT}} \right) \frac{\Delta S_j^2}{V} \quad (5.3.5)$$

where symbols μ and μ_T are the laminar and turbulent dynamic viscosity, respectively, whereas P_r and P_{rT} are the laminar and the turbulent Prandtl numbers. The derivation of the expression of the viscous eigenvalues can be found in [168].

5.4 Implicit residual smoothing

The implicit residual smoothing is a convergence acceleration technique which enables the use of significantly higher value of the CFL numbers with respect to those one can use with the unsmoothed scheme. The use of higher CFL numbers leads to larger local time steps, and this reduces the number of RK cycles required for the solution convergence. The method was thoroughly analysed and extensively used for solving both external and internal aerodynamics problems [81, 139, 141, 145, 161, 171, 172]. The general principle is to apply an implicit Laplacian operator with homogeneous Dirichlet boundary conditions at each step of the iterative solution process. To avoid the solution of large linear systems, the exact Laplacian operator is generally approximated with a factorised operator, namely as the product of one-dimensional operators (one of each direction) applying the smoothing in each direction separately. The coefficients of the linear system for each direction can be constant [81], or variable [172] constructed as

functions of the local flow field variables [139, 141]. These operators are applied to each PDE separately, but when using variable coefficient versions of the IRS method, coupling of smoothed residuals of all PDEs occurs through the dependence of the system coefficients on all local flow variables. In the CFD code used in this study, the factorised IRS operator is applied to smooth the residuals at the end of each RK stage where the residuals \hat{R}_ϕ are replaced by the smoothed residuals \hat{R}_ϕ^* before the solution \hat{Q} is updated. This implementation requires the solution of two sets of tridiagonal systems, one for each direction, at each RK stage. Denoting respectively by i_{cell} and j_{cell} the number of cells in i and j directions, the first set has $i_{cell} \times n_{pde}$ systems and each system has size j_{cell} , whereas the second set has $j_{cell} \times n_{pde}$ systems and each system has i_{cell} unknowns. The tridiagonal systems, one of each direction can be written as:

$$-\beta_\xi \hat{R}_{\phi_{i-1}}^* + (1 + 2\beta_\xi) \hat{R}_{\phi_i}^* - \beta_\xi \hat{R}_{\phi_{i+1}}^* = \hat{R}_{\phi_i} \quad (5.4.1)$$

$$-\beta_\eta \hat{R}_{\phi_{j-1}}^* + (1 + 2\beta_\eta) \hat{R}_{\phi_j}^* - \beta_\eta \hat{R}_{\phi_{j+1}}^* = \hat{R}_{\phi_j} \quad (5.4.2)$$

where the coefficients β_ξ and β_η are given by the following expressions:

$$\beta_\xi = \max \left[\frac{1}{4} \left(\frac{CFL^*}{CFL} \frac{\varphi_\xi}{1 + r_x} \right)^2 - 1, \quad 0 \right] \quad (5.4.3)$$

$$\beta_\eta = \max \left[\frac{1}{4} \left(\frac{CFL^*}{CFL} \frac{\varphi_\eta}{1 + 1/r_x} \right)^2 - 1, \quad 0 \right] \quad (5.4.4)$$

where CFL^* and CFL are user given parameters and their ratio is recommended to be equal to 2 [172]. The expressions of φ and r_x are given respectively as:

$$\varphi_\xi = 1 + r_x^\zeta \quad (5.4.5)$$

$$\varphi_\eta = 1 + r_x^{-\zeta} \quad (5.4.6)$$

$$r_x = \frac{\Lambda_j^c}{\Lambda_i^c} \quad (5.4.7)$$

$$\zeta = \frac{2}{3} \quad (5.4.8)$$

By applying the IRS technology, the residual variations are averaged and the stability of the solver is increased, making it possible to use higher CFL numbers. The method can be found analytically in a lot of CFD books such as Blazek [2] and Hirsch [4].

5.5 CFL ramping

In order to make the numerical algorithm stable and robust, a CFL ramping method has been used. In the early phase of turbulent flow simulations starting from a crude freestream flow initialisation, it is very hard to use high values of the CFL number and maintain the numerical stability of the solution process. These numerical stability problems are further exacerbated by the use of MG. On the basis of numerical experiments, this appears to be due to the fact that fairly high grid refinement is required to maintain the robustness of the time-marching integration of the coupled RANS and SST turbulence model equations, particularly when the flow state used to start up the simulation is quite far from the problem solution. When using a sufficiently refined user-given grid, the coarser grids created by the MG solver are often insufficiently refined to preserve the aforementioned robustness due to the poor quality of the initial solution. In order to maintain the numerical stability of the solution process, low CFL numbers have to be used in the initial stage of the simulation. As the calculation progresses, the flow state approaches the sought solution and higher CFL numbers can be therefore be used. This increment of the CFL number from the small value required in the first MG cycles to the final high value is called *CFL ramping*, and this phase usually requires only a relatively small fraction of the overall number of MG cycles needed to obtain a converged solution starting from the freestream initialisation. Two different schemes for CFL ramping have been implemented in the turbulent CFD solvers used in this study.

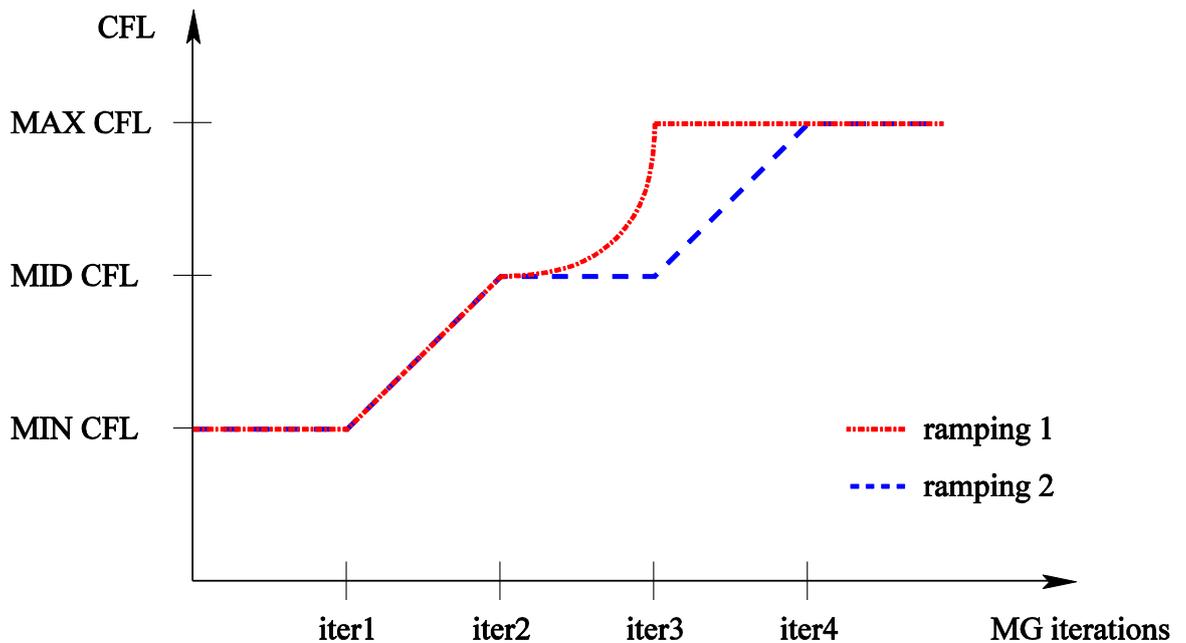


Figure 5.1: CFL ramping

The first method, called *ramping1*, has three stages. The first stage starts from the first MG cycle of the simulation and ends at the iteration labelled *iter1* in Figure 5.1. In the first stage, the CFL number is kept constant and equal to the minimum level denoted by MIN CFL in figure 5.1. In the second stage, the CFL number increases linearly from the value CFL MIN to an intermediated value, called CFL MID. This variation occurs from the MG cycle *iter1* to later MG cycle labelled *iter2*, as visible in the figure. In the third and last ramping stage, the CFL number increases in a cubic fashion from MG cycle *iter2* to a later MG cycle labelled *iter3*, at which the CFL number takes its final maximum value, denoted by MAX CFL. This value is then maintained until the conclusion of the simulation.

The second ramping stage, called *ramping2*, has four stages. The first two stages are the same as those of *ramping1*. In the third stage the CFL is kept constant until the MG cycle, labelled *iter3*. In the fourth and final stage, the CFL number increases again in a linear fashion until it reaches MAX CFL at the MG cycle labelled *iter4*. The CFL number then remains constant and equal to its maximum value until the conclusion of the simulation. The values *iter1*, *iter2*, *iter3*, *iter4* and the values MIN CFL, MID CFL and MAX CFL are all user-given parameters. The complete pattern of *ramping1* and *ramping2* are sketched in Figure 5.1.

Chapter 6

MULTIGRID ACCELERATION

This chapter outlines a convergence acceleration technique called *multigrid* and describes the implementation of this method in the CFD code used for the analyses of this thesis. The multigrid method was initially proposed for the solution of elliptic multi-dimensional problems [143, 173] on uniform (i.e. unstretched) grids. Multigrid has been shown to yield a dramatic reduction of the computational time required for solving iteratively these problems. Successively, several multigrid variants for reducing the computational work required for solving non-elliptic problems on highly stretched grids were developed, including multigrid algorithms for solving the RANS equations and those resulting from the discretization of the PDEs of differential turbulence models [39, 64, 174]. However, developing an efficient multigrid set-up for the computationally efficient solution of realistic turbulent flow problems is often still a challenging task, due to the extreme grid stretching required for adequately solving turbulent boundary layers, and resulting in very large cell aspect ratios, of up to 1 million. The chapter starts with a general introduction to the basics of multigrid. This is followed by a description of multigrid cycling, and the fundamental operators of the algorithm, namely the prolongation and the restriction operators, including the high-order restriction operator used for the solution of the turbulent flow equations of the adopted CFD code. Then, a section reporting how the multigrid solver can be used for solving nonlinear problems, like the discrete system of RANS and SST turbulence model equations is presented. The chapter is concluded by a section focusing on the modifications of the nonlinear multigrid algorithm required for achieving a numerically stable solution process of the RANS and SST equations.

6.1 Introduction

The multigrid method is the most powerful and efficient acceleration technique known today [4, 66, 142, 175]. The full potential of the multigrid approach was first defined and customized by Brandt [173]. It has recently been applied with great success to a variety of problems, such as Euler and Navier-Stokes equations, and it is still a subject for research and development. Multigrid takes its name from using multiple grids to approximate the solution to the original problem.

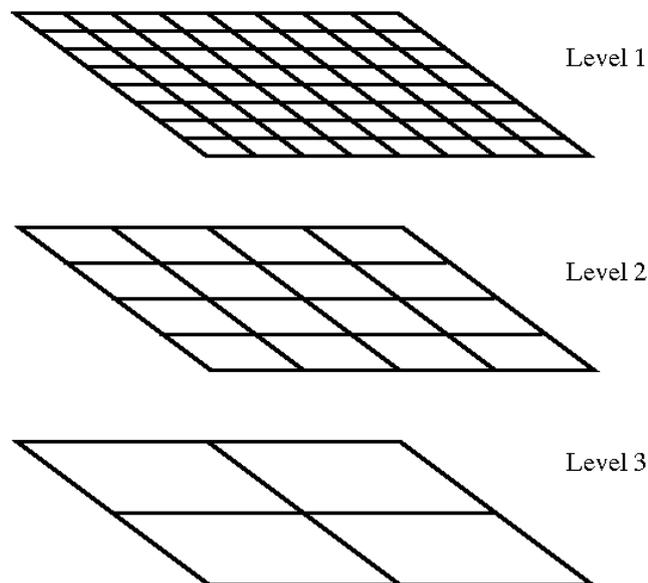


Figure 6.1: Multigrid

The accurate numerical solution of some fluid dynamics problems often requires computational grids to be very fine, either over the entire computational domain or in some critical areas, such as wall boundary layers, shock and/or flow separation regions. The iterative solution of the governing equations based on these kinds of grids can require very large amounts of computing time. One way to reduce this computational cost is to solve the equations using multiple grids where each grid will have a different number of cells. Finer grids provide more accurate solutions compared with coarser grids. However, if the coarser grid is not too coarse, information about the solution can be interpolated from it onto the finer grid to accelerate the solution process. The fundamental idea behind the multigrid method is to use coarser grids in order to smooth rapidly the numerical errors and drive faster the solution on the finest grid. We actually use a coarse-grid solution to correct a fine-grid solution. These numerical errors are made up of high- and

low-frequency components and they will be explained in the next section of this chapter.

In structured multigrid methods, one starts from a user-given fine grid, called level 1 grid. Coarser grids are then built by recursively coarsening the given fine grid. Denoting by h the mesh size of the fine grid, the level 2 coarse grid, with spacing $2h$, is obtained from the level 1 grid by removing every second grid line in each direction. The level 3 coarse grid is obtained in the same manner starting from the level 2 coarse grid. When using three multigrid levels, the level 1 grid is called fine grid, the level 2 grid is called medium refinement grid, and the level 3 grid is called coarse grid. A schematic view of the three grid levels is reported in figure 6.1. As a consequence of this coarsening process, each cell of grid level $i + 1$ will correspond to four grid level i cells sharing a common vertex. Consequently, grid level $i + 1$ will have one fourth of the cells of grid level i . Even in complex CFD problems, one could use more than 3 grid levels. In turbulent problems, however, the numerical difficulties associated with the solution of the turbulence model equations rapidly grow as the grid refinement of the coarser grids decrease. For this reason, it is seldom found that realistic turbulent flow simulations are carried out using more than 3 or 4 multigrid levels [64, 83, 162, 176, 177, 178, 179, 180].

6.2 The multigrid cycle

Iterative solvers achieve convergence by progressively reducing the numerical errors representing the difference between the sought solution and the current estimate of the sought solution. Such numerical errors are made up of high- and low-frequency components. Using the fine grid (i.e. level 1 grid), smoothers like the RK pseudo-time marching algorithm described in section 5.2 can rapidly reduce the high-frequency errors of the solution. On the other hand, the removal of the low-frequency error components using the same level 1 grid is significantly slower, and it thus requires many more iterations than the removal of the high-frequency error components. Since the solution estimate at all iterations contains, in general, both high- and low-frequency errors, the iterative solution based on the level 1 grid is very inefficient, due to the high number of iterations required to remove the low-frequency error components.

One of the key principles of multigrid is to accelerate the removal of the low-frequency components of the solution estimate throughout the iterative solution process. The

construction of the multigrid cycle starts by observing that the current solution error on the level 1 grid can be represented by a spatial truncated Fourier series. The first half of the components of such a truncated series has relatively low spatial frequency, whereas the second half of the components has high spatial frequency. Representing the level 1 solution error on the level 2 grid has as a result, the low-frequency error components of the level 1 solution to appear as high-frequency error on the level 2 grid. Thus applying a few RK iterations on both grid level 1 and grid level 2 results in an overall faster convergence to the sought solution because this process results in a faster removal of the low-frequency errors on the user-given grid [181]. This approach can be applied recursively using more than two grids at a time. For linear elliptic problems, one could coarsen the level 1 grid until a grid with a single cell is obtained (provided that the level 1 grid has the same number of points in all directions). Using this approach, linear elliptic problems can be solved with a single multigrid iteration. The solution of nonlinear problems, however, requires the recursive application of the multigrid cycle, and these problems cannot be solved with a single multigrid cycle. In strongly nonlinear non-elliptic problems like the Navier-Stokes equations, it turns out that the optimal choice of the number of grid levels is often between 3 and 5. This is because the addition of more levels increases the amount of floating-point operations per multigrid cycle but often does not increase by a large extent the convergence rate of the overall iterative process. As an example, using 10 rather 3 multigrid levels may reduce the number of required multigrid iterations to achieve a given convergence level by 10 percent but the cost of the 10-level multigrid iterations may be 30 percent higher than the cost of the 3-level multigrid iteration. In these circumstances, the run-time of the 10-level calculation would be higher than the runtime of the 3-level calculation, despite the fact that the required number of multigrid cycles of the former calculation is smaller than that of the latter one. The distinction between overall runtime and overall number of multigrid cycles is essential when assessing the performance of a newly developed multigrid algorithm, since consideration of the number of multigrid cycles alone can lead to misleading conclusions on the effectiveness of the method. Further detail on the computational cost of the multigrid cycle as a function of the selected number of grid levels can be found in [181].

Using the multigrid error handling strategy highlighted above, one needs a mathematical operator to represent the errors or the entire solution estimate of grid level i onto grid level $i + 1$, and also to perform the inverse step, i.e. report the solution error estimate or the entire solution obtained on grid level $i + 1$ back onto grid level i . The first operation is achieved by means of the restriction operator and the latter by the

prolongation operator. Both operators are described and discussed in the next sections of this chapter.

Before introducing all the steps of the multigrid cycle, it is necessary to give the definitions of the residuals and the numerical error. Let us consider a system of linear equations and denote it by $Au = f$, where u is the sought solution, A is a constant coefficient square matrix and f is the right-hand-side of the system, a constant vector. Denoting by v an approximation to u , we can compute the numerical error e and the residuals r as:

$$e = u - v \quad (6.2.1)$$

$$r = f - Av \quad (6.2.2)$$

respectively. One can combine the above equations and rewrite the given system as follows:

$$Ae = r. \quad (6.2.3)$$

Equation (6.2.3) is the residual equation and it plays a crucial role in the multigrid cycle, as it allows one to establish a relationship between the residual of the equations and the error affecting the solution at each iteration of the smoother. This information enables one to obtain the error estimates on each grid level required by the restriction and prolongation operators discussed below. Equation (6.2.3) refers to the case in which multigrid is used for the solution of linear problems. In the case of nonlinear problems, the residual equation has a different form, though the solution strategy of the nonlinear multigrid cycle is conceptually the same as that of the linear multigrid iteration. The nonlinear multigrid strategy will be explained in section 6.5 of this chapter.

The multigrid cycle for linear problems starts by applying one or more sweeps of an iterative method on the fine grid, where a new approximation solution v is computed and its new residual r is evaluated. Using a restriction operator, the new residual is then transferred to a coarse grid and after solving equation (6.2.3) an estimate error e of the solution is obtained. Finally this error is transferred back to the fine grid using a prolongation operator where it corrects the solution v using the equation (6.2.1).

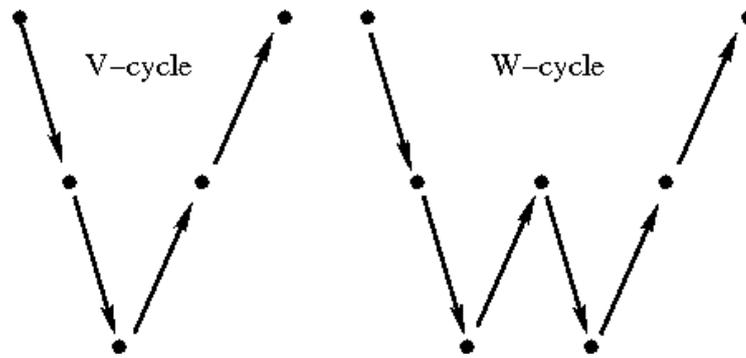


Figure 6.2: multigrid cycles

For both linear and nonlinear problems, two main multigrid cycles exist, namely the ‘V cycle’ and the ‘W cycle’. The ‘V cycle’ starts on the finest level and travels all the grids, applying the selected smoother on each level, until it reaches the coarsest grid. Then it travels back until it returns to the finest grid. This sequence of operations forms the skeleton of the ‘V cycle’. The ‘W cycle’ is similar to the ‘V cycle’ except for the fact that it does two coarsest grid corrections per multigrid iteration. In figure 6.2 one sees the two different approaches. The COSA CFD code uses the ‘V cycle’.

6.3 Restriction operator

The restriction operator, as mentioned above, transfers the solution from a fine to a coarse grid. In order to highlight how the restriction operator works, it is convenient to consider a one-dimensional problem, in which the unknowns are stored at the nodes of the user-given grid. In CFD, this approach is known as the cell-vertex scheme. Figure 6.3 depicts a fine grid and the corresponding coarse grid. The coarse grid is obtained by removing every second point of the fine grid. The figure also shows the initial relationship between the nodes of the coarse grid and their counterparts on the fine grid. The simplest restriction method is the injection operator. In this case, the only node values of the fine grid transferred to the coarse grid are those defined at fine nodes which also exist on the coarse grid. The transfer takes place by simply copying or injecting the fine nodal values to their corresponding positions on the coarse grid, as shown in figure 6.3.

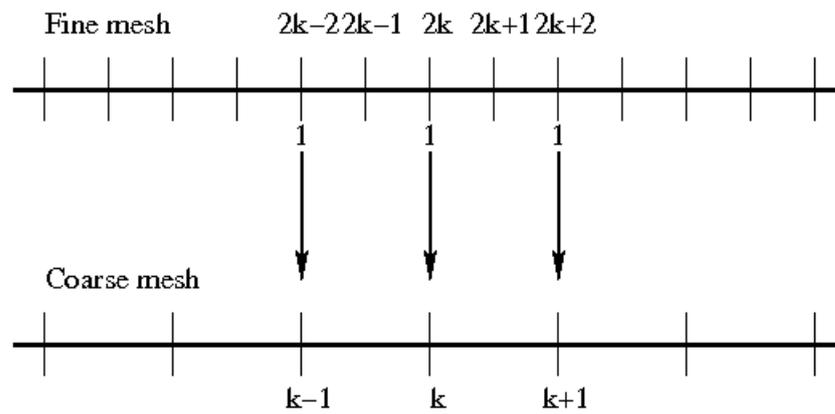


Figure 6.3: 1-d simple injection

An alternative restriction method is the *full weighting* operator. The information transferred from the fine to the coarse grid is reported in figure 6.4. With full-weighting, each coarse mesh point receives a contribution from its counterpart on the fine grid (like in the injection case), but also receive contributions from the two fine grid nodes adjacent to its fine grid counterpart. The three contributions are weighed: the contribution of the fine counterpart is $\frac{1}{2}$ and that of the two fine neighbours is $\frac{1}{4}$. For consistency, the sum of the weighting factors should always be equal to one.

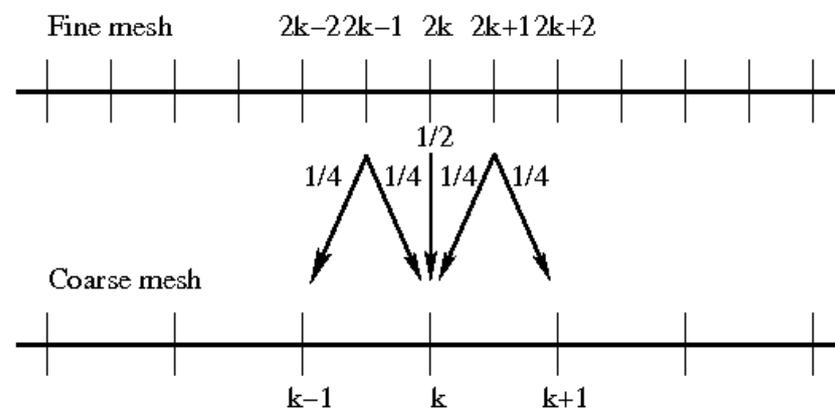


Figure 6.4: 1-d Full weight restriction

In the case of two-dimensional problems, the injection operator has the same structure of its one-dimensional counterpart: only the solution estimate of the fine grid point which also exists on the coarser grid is transferred to the corresponding coarse grid node. The principle behind the design of the two-dimensional full weighting restriction operator is the same as in the one-dimensional case, but the numerical stencil of two-dimensional implementation is expectedly more articulated. In the two-dimensional case information from 9 nodes of the fine grid is used to build the restricted solution estimate at each node of the coarse grid. The numerical stencil of the two-dimensional full weighting restriction

operator is shown in figure 6.5.

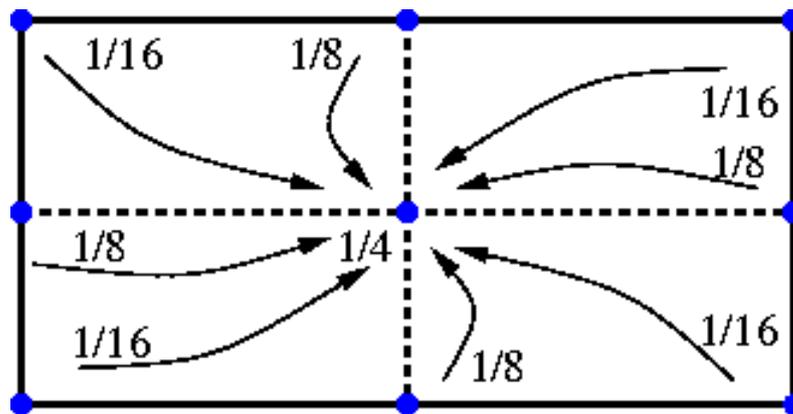


Figure 6.5: full weighting restriction operator for two-dimensional cell-vertex schemes

Here the fine grid is defined by both the solid and the dashed lines, whereas the represented portion of the underlying coarse grid is defined by the dashed lines only. In other words, all 9 blue circles are grid vertices of the fine grid, whereas only the central circle is the vertex of the underlying coarse grid. The sketch also reports the weights by which the 9 fine grid nodal values have to be scaled to build the coarse grid central node estimate of the solution. Such an estimate is therefore a linear combination of the 9 fine grid values. Note that also in the present two-dimensional case, the sum of the weights is one for consistency.

The sketch of figure 6.5 refers to the case in which the unknowns are defined at the vertices of the user-given grid. Such circumstance arises when the differential form of a set of governing equations is space-discretised using a finite-difference approach, or when the integral form of a set of governing equations is space-discretised using a cell-vertex finite volume approach. Alternatively, the integral form of a set of governing equations like the Navier-Stokes equations, can be space-discretised using a cell-centred finite volume approach. In this case, the unknowns are stored at the centres of the cells or finite volumes defined by the user-given grids. In the case of multigrid solvers applied to the solution of the Euler and Navier-Stokes equations space-discretised with a cell-centred finite volume scheme, the full-weighting restriction operator reported for the cell-vertex case is often replaced by a simpler weighted restriction operator, whereby only four fine grid cells are used.

In this case, the four fine grid cell volumes, to which the fine grid cell centre values used to build the coarse grid cell centre values refer, are the weights of this weighted

restriction operator. The structure of this weighted restriction is essentially that of the injection restriction operator. The numerical stencil of this weighting cell-centred restriction operator is sketched in figure 6.6, where the blue circles denote the cell centres of the four fine grid cells enclosed by the dash lines, and the red circle denotes the cell centre of the coarse grid cell enclosed by the solid lines.

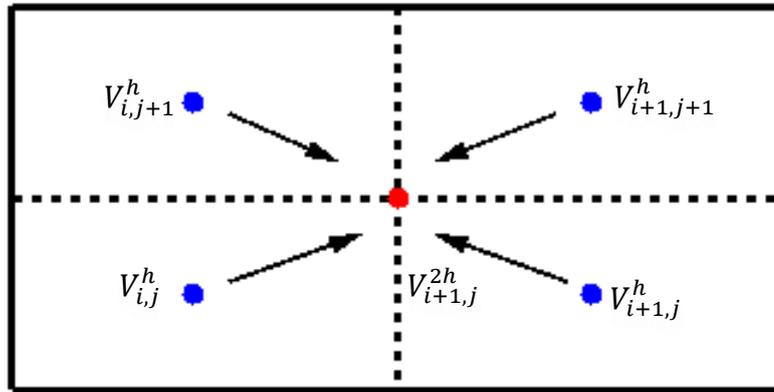


Figure 6.6: weighted restriction operator for two-dimensional cell centred schemes

Denoting by \hat{Q} solution estimates, V cell volumes, superscript h fine grid variables and superscript $2h$ coarse grid variables, the mathematical form of this weighted operator is:

$$\begin{aligned} \hat{Q}_{i,j}^{2h} &= \frac{V_{i,j}^h \hat{Q}_{i,j}^h + V_{i,j+1}^h \hat{Q}_{i,j+1}^h + V_{i+1,j}^h \hat{Q}_{i+1,j}^h + V_{i+1,j+1}^h \hat{Q}_{i+1,j+1}^h}{V_{i,j}^{2h}}. \end{aligned} \quad (6.3.1)$$

As discussed below, the use of multigrid for solving nonlinear problems like Euler and Navier-Stokes equations results in the necessity to transfer also fine grid residuals to the coarse grid, not only the fine grid solution estimate \hat{Q}_h . Two different ways for transferring fine grid residuals to the coarse grid have been developed and tested in this work. The first approach consists of summing the residuals of the four fine grid cells forming the cell of the coarser grid level to obtain the restricted residual on the coarse grid. This operation amounts to applying the injection restriction operator to obtain the coarse grid residuals from the fine grid residuals. The mathematical form of this operator is given by equation (6.3.2), where the symbol \hat{R} denotes residuals.

$$\hat{R}_{i,j}^{2h} = \hat{R}_{i,j}^h + \hat{R}_{i,j+1}^h + \hat{R}_{i+1,j}^h + \hat{R}_{i+1,j+1}^h. \quad (6.3.2)$$

A graphical representation of the set of performed operations is provided in figure 6.7.

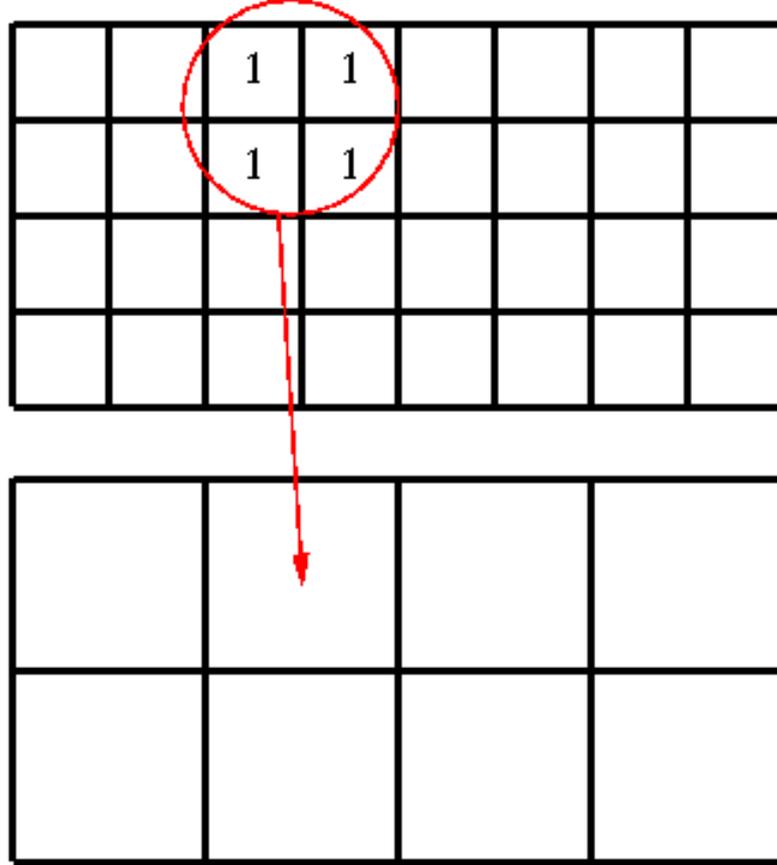


Figure 6.7: residual injection operator

The second residual restriction approach is a new high-ordered restriction operator [61], the stencil of which is reported in figure 6.8. Using this operator, the coarse grid residual is calculated using 12 fine grid residuals rather than four fine grid residuals. The coarse grid residual computed by the new higher order restriction operator is:

$$\begin{aligned} \hat{R}_{i,j}^{2h} = \frac{1}{16} & \left(9 \hat{R}_{i,j}^h + 9 \hat{R}_{i,j+1}^h + 9 \hat{R}_{i+1,j}^h + 9 \hat{R}_{i+1,j+1}^h \right. \\ & + 3 \hat{R}_{i-1,j}^h \\ & + 3 \hat{R}_{i-1,j+1}^h + 3 \hat{R}_{i,j-1}^h + 3 \hat{R}_{i,j+2}^h + 3 \hat{R}_{i+1,j-1}^h \\ & + 3 \hat{R}_{i+1,j+2}^h + 3 \hat{R}_{i+2,j}^h + 3 \hat{R}_{i+2,j+1}^h + \hat{R}_{i-1,j-1}^h \\ & \left. + \hat{R}_{i-1,j+2}^h + \hat{R}_{i+2,j-1}^h + \hat{R}_{i+2,j+2}^h \right) \end{aligned} \quad (6.3.3)$$

and a graphical representation of this operator is provided in figure 6.8. It is shown that, when the higher order restriction is used, residual boundary values of the fine mesh are required in order to compute the coarse grid residuals in the cells adjacent to the domain boundaries.

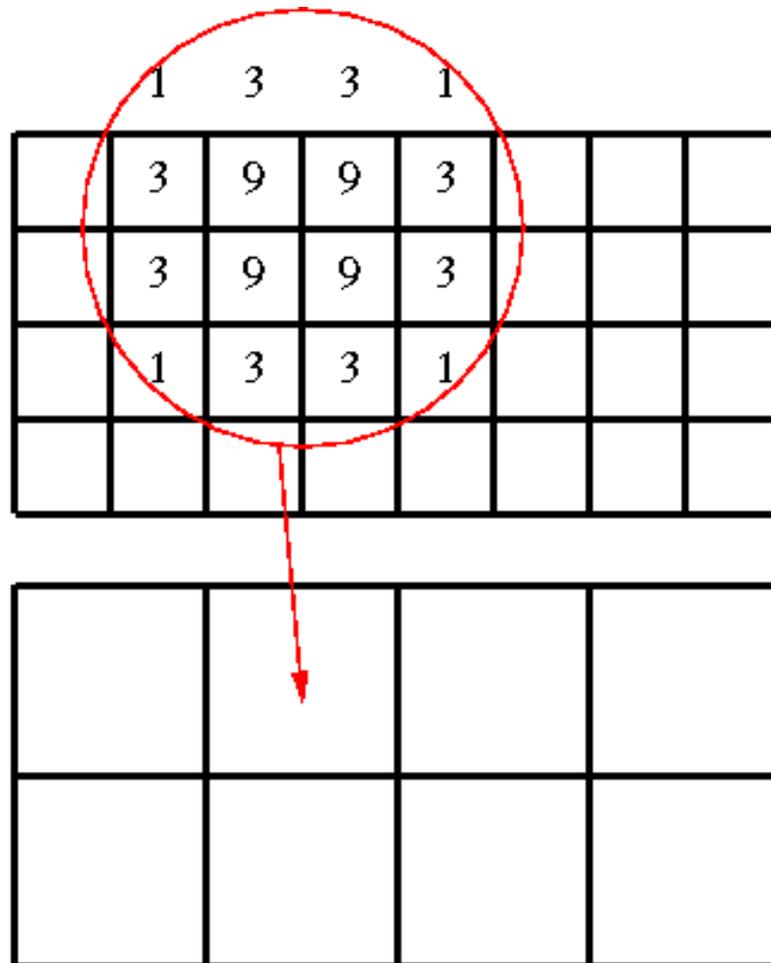


Figure 6.8: 2-d residual high-weighted restriction operator

Numerical experiments have shown that the choice of fine grid boundary cell residuals that gives the best numerical stability is:

$$\hat{R}_w = -\hat{R}_1 \quad (6.3.4)$$

where \hat{R}_1 is the residual of the interior cell adjacent to the wall. The use of equation (6.3.4) results in a reduction of the residuals in the wall proximity. It has been observed that the main effect accounting for the good convergence rate of the turbulent multigrid solver accomplished by using equation (6.3.4) is the reduction of the residual of ω . A sharp

gradient of ω and its residual near the wall is always observed near the wall. Equation (6.3.4) reduced the gradient of ω and its residuals on the coarser grid levels, and this has a beneficial effect on the convergence of the turbulent multigrid solver. On the other hand, since the accuracy of the final solution is unaffected by the approximations made on the coarser grid levels, the use of equation (6.3.4) does not spoil the accuracy of the computed solution. At fine grid far-field boundaries, the residual boundary condition adopted for the high-order residual restriction is instead:

$$\hat{R}_{far} = \hat{R}_1. \quad (6.3.5)$$

The reason for not reducing the fine grid boundary residual weight at far-field boundaries is that such residuals have a negligible impact on the rate of convergence. This happens because the residuals close to the far field and also their gradients are very small compared to the residuals close to the wall.

For the high Reynolds number flow analyses performed in this study and reported later in the thesis, it has been found that the use of the higher order restriction operator greatly improves numerical stability with respect to the simpler injection operator provided by equation (6.3.2). In most high-Reynolds number flow problems, the sought solution could be computed only by using the higher order restriction as the CFD code featuring the injection operator was unstable. Moreover, for cases in which both the injection and the higher order restriction worked, it was often found that the convergence rate of the code featuring the higher order restriction was higher than that of the code featuring the injection operator.

6.4 Prolongation operator

The prolongation operator serves the purpose of transferring or interpolating the solution from a coarse grid to a fine grid. Taking a one-dimensional cell-vertex example for simplicity, the method to transfer a coarse grid solution to the fine grid is a linear interpolation. Using this approach, the value at a fine grid node which also exists in the coarse grid is taken to be the coarse grid value. The value at a fine grid point which does

not exist in the coarse grid, conversely, is taken to be the arithmetic average of the two coarse grid points to the left and the right of the coarse grid point which exists also on the fine grid. The weights of this one-dimensional full-weighting prolongation are reported in figure 6.9.

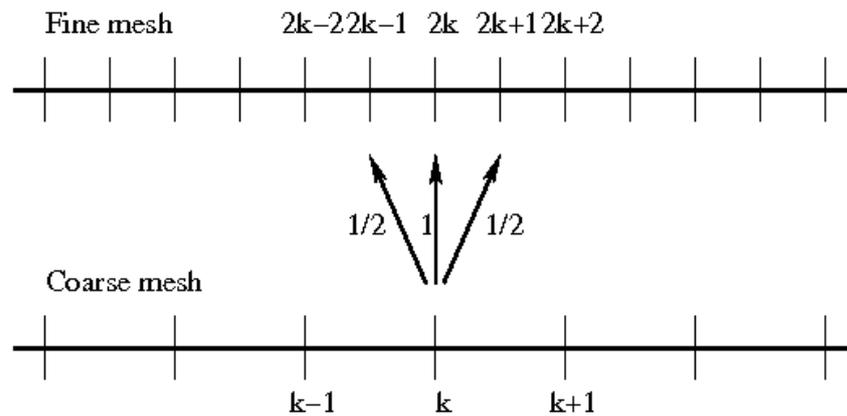


Figure 6.9: full-weighting prolongation for one-dimensional problems

For two dimensional cell-vertex problems the prolongation operator transfers information from the coarse grid point that exists also on the fine grid to its fine grid counterpart and to the 9 neighbouring points of its fine grid counterpart. The stencil of this operator is shown in figure 6.10.

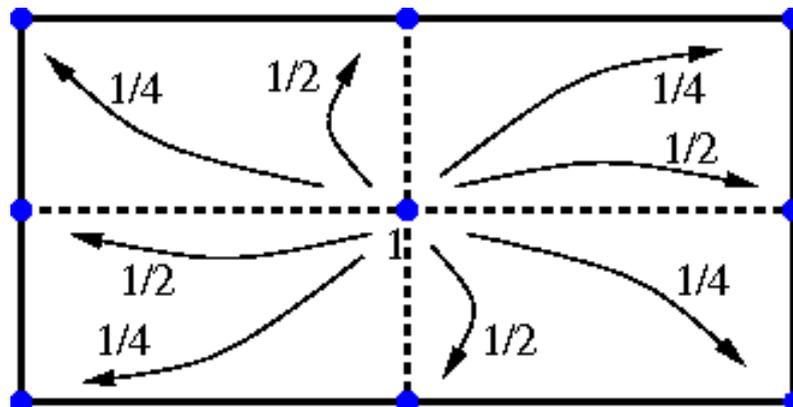


Figure 6.10: full-weighting prolongation for two-dimensional cell-vertex problems

In this figure, the dashed and solid lines define the fine grid cells, whereas the dashed lines define the grid lines of the coarse grid. The numbers next to each line shows the local weight factor between the fine and the coarse mesh functions. In the case of cell-centred finite volume grids the constant weight factors are often replaced by local volume weights. The prolongation operator that has been implemented in the turbulent multigrid

solver used in this research is a constant-weight *bilinear interpolation*. This prolongation operator uses four coarse grid values to compute the fine grid solution (or a correction to the fine grid solution estimate in the case of nonlinear problems). The weighing values of each cell can be seen from the figure 6.11.

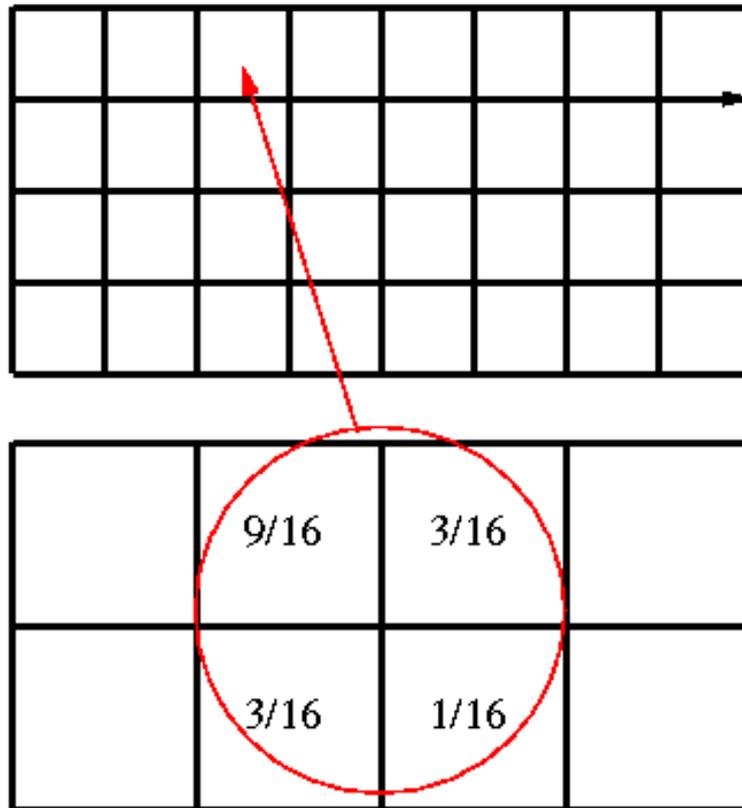


Figure 6.11: bilinear prolongation for two-dimensional cell-centred problems

6.5 Multigrid for nonlinear problems

The typical form of nonlinear problems is $A(u) = f$, where A denotes a system of nonlinear equations, u is the array of unknowns, and f is a constant vector. The numerical error is defined by equation (6.2.1) and the residuals can be found using the equation:

$$r = f - A(v). \quad (6.5.1)$$

However, for nonlinear problems equation (6.2.3) is no longer valid, as the equations are nonlinear, and for this reason the residual equation has a different form defined as:

$$A(u) - A(v) = r \quad (6.5.2)$$

and

$$A(v + e) - A(v) = r. \quad (6.5.3)$$

As one sees from equation (6.5.3) it is not possible to compute the numerical error directly (solve the system of equation (6.2.3)) and use it to correct the approximate solution of the fine grid, as, due to nonlinearity, the set of expressions $A(u)$ cannot be written as the product of a constant-coefficient matrix and the unknown array u . There are two main approaches one can use to circumvent this difficulty. The first approach is the so-called Newton's method [158], which linearizes the equations and makes them suitable for the linear multigrid algorithm, whereas the second approach is the so-called *full approximation scheme* (FAS), which applies the multigrid algorithm directly to the nonlinear equations. In this study the second method has been used and will be explained in detail in the following paragraphs.

The FAS multigrid cycle for nonlinear problems starts by applying one or more sweeps of an iterative method on the fine grid where a new approximation v to the sought solution is computed, and its new residual r is evaluated, similar to the linear problem case. Using a restriction operator, the new residual is then transferred to a coarser grid, like in the linear multigrid algorithm, and represents the right hand side of the equation. However, due to the nonlinearity of the system A it is not possible to compute the numerical error directly on the coarse grid, using only the restricted residuals. For this reason one has to transfer the approximate solution v computed on the fine grid to the coarser grid with a suitable restriction operator. If one considers the restriction operators I_h^{2h} and \hat{I}_h^{2h} which transfer the solution v and the residuals r , respectively, from a fine to a coarse grid then equation (6.5.3) can be written as:

$$I_h^{2h}\{A(v + e) - A(v)\} = \hat{I}_h^{2h}\{r\} \quad (6.5.4)$$

or,

$$I_h^{2h}\{A(w)\} = \hat{I}_h^{2h}\{r\} + I_h^{2h}\{A(v)\} \quad (6.5.5)$$

or,

$$A^{2h}(w^{2h}) = r^{2h} + A^{2h}(v^{2h}) \quad (6.5.6)$$

where symbol h denotes that the discrete equations are considered on the fine mesh, which is that on which the solution is sought. This symbol will be used as a subscript or superscript in order to indicate the mesh on which the operators or the solutions will be defined. The right hand side of the equation (6.5.6) is known and the symbol w^{2h} on the left hand side of the same equation is the new array of unknowns of the system on the coarser grid. After several sweeps of an iterative method on the coarse mesh a new approximation solution w^{2h} is computed and it can be used to compute the numerical error e by calculating the difference between the new computed solution w^{2h} and the restricted solution v^h from the fine mesh as:

$$e^{2h} = w^{2h} - I_h^{2h} v^h. \quad (6.5.7)$$

One can transfer the numerical error computed on the coarse mesh using a prolongation operator and then correct the fine mesh solution using the followed equation:

$$u^h = v^h + I_{2h}^h \{w^{2h} - I_h^{2h} v^h\}. \quad (6.5.8)$$

As explained in chapter 5, the large system of algebraic equations, resulting from the space-discretisation of the flow and turbulence model equations, is expressed by equation (5.2.5). Applying the abovesaid FAS-MG method on the RANS and SST turbulence equations, the equation (5.2.5) is expressed as:

$$\frac{\partial \hat{Q}_h}{\partial \tau} = -\frac{1}{V_h} \hat{R}_h(\hat{Q}_h) \quad (6.5.9)$$

where symbol h denotes the fine grid, the entries of the array \hat{Q}_h represent the unknown flow variables at the cells centres of the discretized physical domain and the array \hat{R}_h stores the cell residuals. Following the above procedure, one has to apply one or more sweeps of the RK integration (5.2.31) where the new solution \hat{Q}_h^{l+1} is computed and its new residual \hat{R}_h^{l+1} is evaluated. The new solution is then transferred to a coarse grid using the restriction operator (6.3.1):

$$\hat{Q}_{2h}^{(0)} = I_h^{2h} \hat{Q}_h^{l+1} \quad (6.5.10)$$

where symbol $2h$ denotes the coarse grid and I_h^{2h} is the restriction operator. The restricted solution $\hat{Q}_{2h}^{(0)}$ is then used at the 1st stage of the RK cycle where the new residuals $\hat{R}_{2h}^{(0)}$ are evaluated. In addition, the residuals of fine grid \hat{R}_h^{l+1} are also transferred to the coarser grid using the restriction operator (6.3.3). The difference between the residuals \hat{R}_h^{l+1} , transferred from fine mesh, and the residuals computed on the coarser mesh $\hat{R}_{2h}^{(0)}$ is the so-called *forcing function*. This source term is required in order to maintain the accuracy of the new system in the coarse mesh and to guarantee that the residuals used in the coarse mesh are those from the fine mesh. The expression of the forcing function is equal to:

$$f_{MG} = I_h^{2h} \hat{R}_h^{l+1} - \hat{R}_{2h}^{(0)}. \quad (6.5.11)$$

This source term is computed after the restriction operator and only for the first RK step of the first iteration of the equation (5.2.31). The equations on the coarse grid are then smoothed using the RK scheme, where after several sweeps, an approximation of the correction $\Delta \hat{Q}_{2h}$ is obtained at a reduced computational cost, since there are fewer mesh points, and its expression is:

$$\Delta \hat{Q}_{2h} = \hat{Q}_{2h}^{l+1} - \hat{Q}_{2h}^{(0)}. \quad (6.5.12)$$

Finally the correction $\Delta \hat{Q}_{2h}$ is transferred back to fine mesh in order to generate a new approximation of the solution:

$$\hat{Q}_h = \hat{Q}_h^{l+1} + I_{2h}^h \Delta \hat{Q}_{2h} \quad (6.5.13)$$

where I_{2h}^h is the prolongation operator described in section 6.4.

6.6 Technicalities on the application of multigrid to RANS and SST equations

While MG methods are well defined in a mathematical sense, their efficient application to the RANS equations with two-equation turbulence models is rather difficult [64]. There are two basic approaches to the incorporation of the turbulence models in MG methods. The first one is named *Mean-flow multigrid* (MF-MG) and the second one *Fully-coupled multigrid* (FC-MG).

In MF-MG, the mean-flow equations are solved on all grid levels, while the turbulence model equations are integrated only on the finest grid where the solution is sought, as in single-grid computations. In this approach, turbulence variables are simply injected into coarser grids, where they are frozen. This allows bypassing the numerical difficulties arising from the destabilizing effects produced by the source terms of the turbulence model. However, insufficient acceleration of the turbulence transport equations, due to a partial use of single-grid computations in MF-MG, may result in an overall reduction of convergence rate.

In FC-MG approaches, both the mean-flow and the turbulence model equations are solved on all grid levels, however FC-MG implementation is far from being straightforward. Usually, stabilization techniques are used to damp numerical difficulties encountered in the integration of turbulence transport equations on coarse grid levels of the MG solution. In early stages of the simulation, the fine grid residual values are large. As a result, high mean-flow gradients may appear on coarse grid levels, with the potential of causing excessive values in the source terms of the turbulence model. Moreover, the accuracy of strongly non-linear source terms cannot be fully preserved on coarse grid levels, possibly leading to divergence or inaccurate coarse grid correction. To improve stability, turbulence source terms are computed only on the fine grid, and then restricted into coarser grid levels where they are frozen. This technique is commonly termed source-term freezing. Some researchers, instead of applying source-term freezing, only employ turbulence viscosity freezing on coarse grid levels in order to increase stability of the coupling mechanism between the mean-flow and turbulence variables.

In some cases, straightforward applications may lead to loss of positivity of fine grid turbulence variables, namely the turbulent kinetic energy K and the dissipation rate ω . For this reason the multigrid algorithm which solves jointly the RANS and turbulence

equations requires a positivity preserving scheme. Positivity-preserving difficulties within MG may arise in the smoothing (relaxation) and prolongation stages. Inappropriate numerical treatment of turbulence equations may result in non-physical values of turbulence variables, leading to inefficient fine grid correction and to loss of multigrid robustness. In this study, at the end of each stage of RK integration, if the values of turbulence kinetic energy or dissipation rate are less than a minimum positive value then they are forced to be equal to that minimum positive value. This positivity scheme is applied on all grid levels. In addition, when using MG, the residuals of the ω equation are limited before being restricted to a coarser grid, as proposed in [59]:

$$R_{\omega}^* = R_{\omega} - \text{MAX} \left(0, \frac{(\rho\omega)_{\min} - (\rho\omega)}{\Delta\tau} \right) \quad (6.6.1)$$

where the R_{ω}^* is the new smoothed residual and the $(\rho\omega)_{\min}$ depends on the production term as:

$$(\rho\omega)_{\min} = \gamma\rho\sqrt{P_d}. \quad (6.6.2)$$

The turbulent viscosity is computed on each grid level, but the production terms P_d and Δ^+ are computed only on the finest grid level and reported to coarser levels by using the restriction operator. Moreover, equation (6.5.13) does not prevent the turbulence variables from becoming negative and it is possible that the coarse grid correction of the turbulence variables on the fine-grid lead to non-physical negative values. This issue arises mainly in the first few near-wall cells. A common technique to avoid this issue in the prolongation stage is to employ artificial fixes by either allowing only positive increments, or by locally neglecting corrections that cause loss of positivity. In this study a limiter has been developed to guarantee the positive values of the turbulence variables. Using such a limiter, the equation (6.5.13) is modified as follows:

$$\hat{Q}_h = \hat{Q}_h^{l+1} + \frac{I_{2h}^h \Delta \hat{Q}_{2h}}{1 - \beta \min \left(\frac{I_{2h}^h \Delta \hat{Q}_{2h}}{\hat{Q}_h^{l+1}}, 0 \right)} \quad (6.6.3)$$

where $\Delta \hat{Q}_{2h}$ is the correction. Note that this limiter reduces the value of the correction

$I_{2h}^h \Delta \hat{Q}_{2h}$ only when it is negative. Parameter β affects the level of reduction that will be applied to the correction. Eliasson and Wallin [61] proved that the value of β has very small influence on the convergence rate and should for most cases be greater than 10 for robustness reasons. In this work we set β equal to 100. When the value of the correction $I_{2h}^h \Delta \hat{Q}_{2h}$ becomes small compared to the solution \hat{Q}_h^{l+1} , and only if it is negative, then the reduction of the correction $I_{2h}^h \Delta \hat{Q}_{2h}$ forced by the limiter (6.6.3) becomes small. The higher order restriction combined with the new correction-limiter has substantially improved the robustness of the turbulent MG code used in this research. Oscillations which were obvious in the turbulent viscosity profiles for a variety of test cases have now disappeared. Results relative to the stability of the turbulent multigrid method can be found in chapter 7 “Validation”.

Chapter 7

VALIDATION

This chapter outlines numerical results computed for three external flow problems. Firstly a steady turbulent flat plate boundary layer is considered: the numerical solution of COSA is compared with available theoretical results. The second test case is the NACA4412 airfoil in a subsonic turbulent flow featuring a flow reversal in the rear portion of the suction side. The COSA solution has been compared to available detailed hot-wire boundary layer measurements. The third test case is a transonic flow field past the RAE2822 airfoil. This transonic flow field is characterized by a significant shock/boundary layer separation interaction on the suction side of the airfoil, and such an interaction causes a boundary layer separation in this area. In all test cases, comparison between Wilcox's and Menter's wall boundary condition for the turbulent quantity ω are presented. It is highlighted that the solution variations due to particular choices of this boundary condition are often larger than those caused by the use of different turbulence models. The chapter also presents comparisons of the numerical solutions obtained by using either a first- or a second-order accurate space-discretization of the two turbulence model equations. This analysis highlights that significant differences between the solutions obtained by using either approach exist in the case of both subsonic and transonic flow problems. This is an important result, since available literature on this aspect points to the fact that the use of a second order rather than first order discretisation of the turbulent equations is more important for transonic flows with shocks. Finally the convergence rate of the residuals for all test cases is examined to highlight the significant convergence and stability improvements obtained by using the high-order restriction operator and the other turbulent multigrid adjustments presented in chapter 6.

7.1 Steady turbulent flat plate

The turbulent flow over a flat plate leading to the formation of a turbulent boundary layer is considered. The computational domain is rectangular and the flat plate lies on the lower horizontal boundary. The *leading edge* (LE) of the flat plate is in the origin of the Cartesian system, and its *trailing edge* (TE) is at $x = 1$, where the (vertical) outlet boundary is positioned. The inlet boundary is at $x = -1/3$, and the upper horizontal side is a far-field boundary positioned at $y = 1$. The mesh can be seen in figure 7.1.

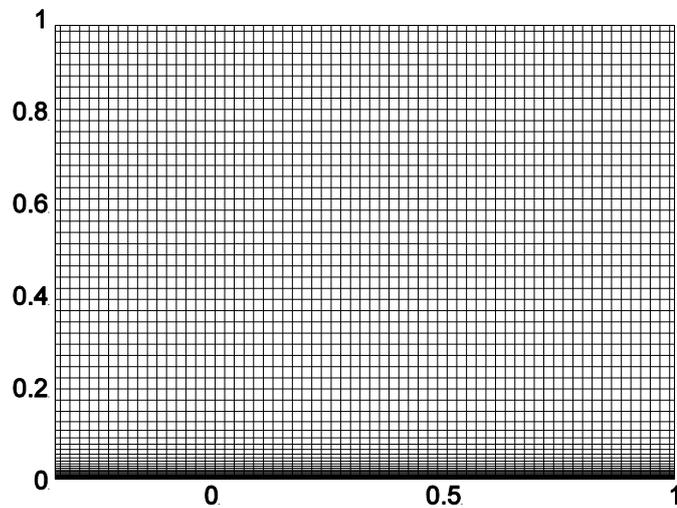


Figure 7.1: *Grid for turbulent flat plate.*

The adopted Cartesian grid is a 2-block grid and it has 384 mesh intervals along y , and the size of these intervals increases from the lower horizontal boundary to the upper horizontal boundary starting from a minimum value of 2.5×10^{-7} with y^+ always less than 1. The grid has 256 equal mesh intervals along x , where 192 are on the flat plate and 64 in the space between the LE and the inlet boundary. The free-stream Reynolds number Re is 6×10^6 , and the free-stream Mach number is $M_\infty = 0.2$. All simulations have been performed using the so-called *improved auxiliary state far-field BCs* for internal flows [69] on the vertical left and right boundaries of the computational domain, and a standard external-flow far-field condition on the top horizontal boundary [81]. Symmetry conditions are imposed on the portion of the lower horizontal boundary between the inlet boundary and the LE of the flat plate, and a no-slip condition is applied on the flat plate.

The turbulent intensity at far-field boundary is equal to 1% and the ratio μ_T/μ is set to 10^{-4} . Mesh refinement analyses conducted for each of the modelling set-ups discussed below have revealed that the solution computed with the grid defined above presents negligible differences from the solution computed by using grids with substantially higher spatial refinement.

This turbulent flow field has been computed using both the $K - \omega$ and the SST turbulence models using a second order upwind scheme for all equations. The multigrid method has been applied to the system of governing equations in order to increase the convergence rate of the residuals. Moreover a high order restriction operator (6.3.3) is used to transfer the residual from a fine grid to a coarser grid and a limiter is added to the prolongation operator, defined by equation (6.6.3), in order to guarantee the positive values of the turbulence variables. Three multigrid levels with V-cycle have been used in all computations where for each multigrid cycle 5 RK iterations have been performed on the first level, 5 RK iterations on the second level and 2 RK iterations on the coarsest level. All computations started from a freestream flow initialisation and used $CFL = 4$. The total number of multigrid cycles was set to 2500. For each turbulence model, three analyses have been performed, one computing ω_w with Wilcox's model (equation (2.6.1)) and $S_R = 100$, one computing ω_w with Menter's model (equation (2.6.7)), and one computing ω_w with equation (2.6.1) and $S_R = 2500$, as proposed in [82]. The four profiles of the non-dimensionalized velocity component parallel to the flat plate on a line orthogonal to the flat plate itself at $x = 0.5$, computed with $K - \omega$ and SST analysis using either equation (2.6.1) with $S_R = 100$ or equation (2.6.7), are reported in the top left subplot of the figure 7.2.

The variable on the horizontal axis is the logarithm in base 10 of y^+ , the nondimensionalized wall distance, and its expression is given by equation (2.6.9). The variable on the vertical axis is u^+ the non-dimensionalized velocity component $u_{||}$ parallel to the wall, which, in this case, is the x -component of the velocity vector. Its expression is $u^+ = u_{||}/u_\tau$. The subplot also reports Spalding's profile, which is a power-series interpolation of experimental data joining the linear sub-layer to the logarithmic region of the turbulent boundary layer occurring on a flat plate in the absence of a stream-wise pressure gradient [74]. It is observed that the velocity profiles computed with either turbulence model using equation (2.6.7) (profiles labelled 'SST, mentw' and ' $K - \omega$, mentw') are very close to each other, and are both fairly close to Spalding's profile. The velocity profiles computed with either turbulence model using equation (2.6.1) with $S_R = 100$ (profiles labelled 'SST, wlc0w' and ' $K - \omega$, wlc0w') are also very close to each

other, but are farther away from the Spalding's estimate with respect to the solutions obtained with (2.6.7).

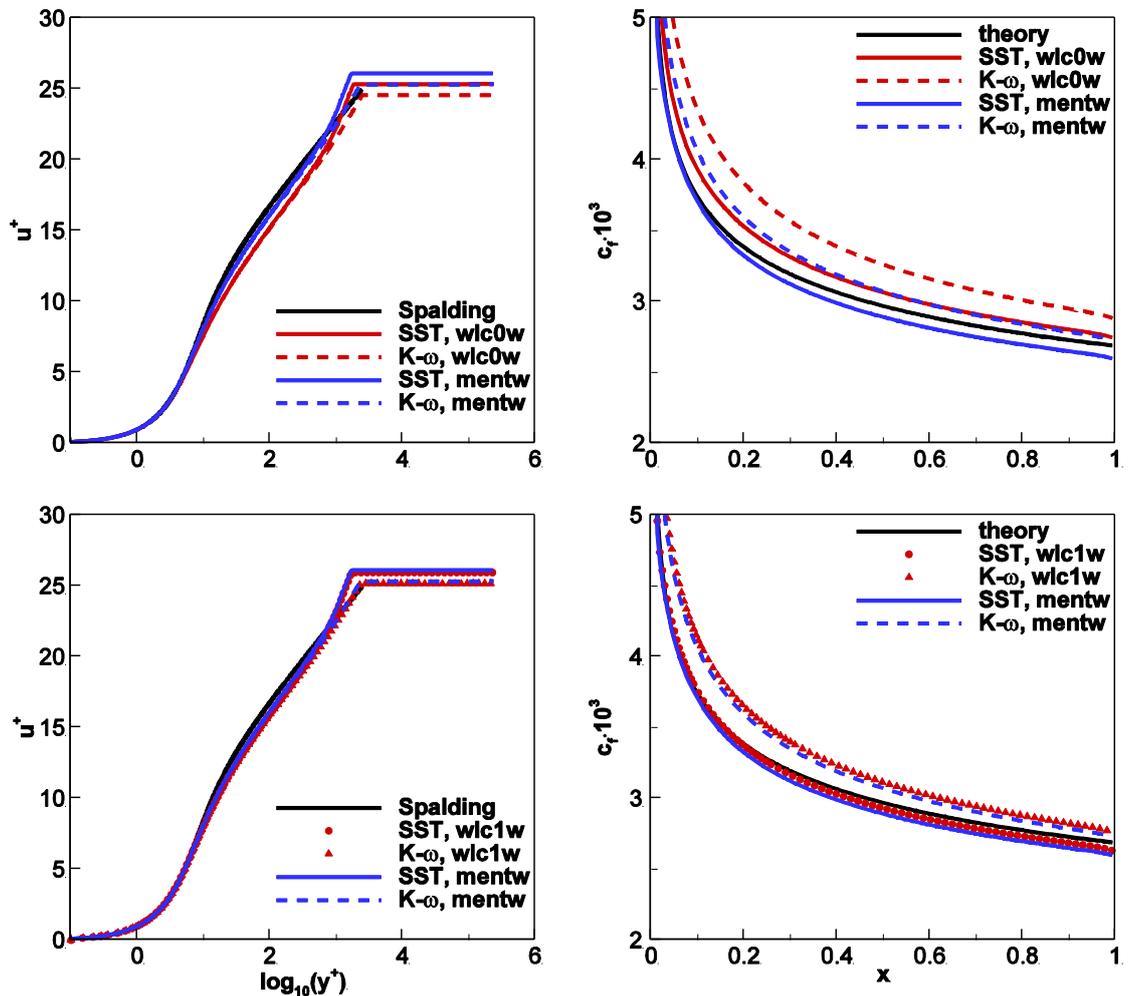


Figure 7.2: Turbulent flat plate analysis. **Top Left:** comparison of Spalding's velocity profile and velocity profiles at $x = 0.5$ computed with $K - \omega$ and SST models using Wilcox's ($S_R = 100$) and Menter's wall BC. **Top Right:** comparison of theoretical skin friction coefficient (c_f) and c_f computed with $K - \omega$ and SST models using Wilcox's ($S_R = 100$) and Menter's wall BC. **Bottom left:** comparison of Spalding's velocity profile and velocity profiles at $x=0.5$ computed with $K - \omega$ and SST models using Wilcox's ($S_R = 2500$) and Menter's wall BC. **Bottom Right:** comparison of theoretical c_f and c_f computed with $K - \omega$ and SST models using Wilcox's ($S_R = 2500$) and Menter's wall BC.

The skin friction coefficient c_f along the flat plate obtained with these four simulations is reported along the vertical axis of the top right subplot of figure 7.2. The theoretical estimate for this problem $\left((c_f)_{theory} = 0.025 * (Re_x)^{-\frac{1}{7}} \right)$ is also reported for reference.

One notices that the differences between the c_f profiles obtained by using the two turbulence models with the same wall BC are of the same order of magnitude as those observed when using one turbulence model with two different wall BCs, and this highlights the impact of the adopted condition for ω_w on the computed solution. The velocity profiles obtained with the $K - \omega$ and SST simulations using either equation (2.6.1) with $S_R = 2500$ or equation (2.6.7) are presented in the bottom left subplot of figure 7.2, where use of the former condition is denoted by the label ‘wlc1w’. The bottom left subplot highlights that the four velocity profiles are substantially closer to each other than in the top subplot and this occurs because the rescaled Wilcox’s wall BC yields wall values of ω closer to those of Menter’s condition. The same conclusion holds for the c_f -profiles of the bottom right subplot. The differences between the two solutions obtained using a given turbulence model with either equation (2.6.1) or equation (2.6.7) with $S_R = 2500$ are significantly smaller than those between the former solution and that obtained with equation (2.6.7) with $S_R = 100$. The remaining significant differences between the computed c_f profiles for a given wall BC and use of either turbulence model are likely to be caused by structural differences between the turbulence models, such as the lower sensitivity of the SST model to the free-stream value of ω . The theoretical value of the drag coefficient C_D for the considered configuration is 3.14×10^{-3} , whereas the values of C_d obtained with the six presented simulations are reported in table 7.1, and these data emphasize again the impact of the wall BC for ω on the computed viscous drag.

BC	SST	$K - \omega$
wlc1w	3.13×10^{-3}	3.35×10^{-3}
wlc0w	3.29×10^{-3}	3.52×10^{-3}
mentw	3.09×10^{-3}	3.31×10^{-3}
theory	3.14×10^{-3}	3.14×10^{-3}

Table 7.1: Flat plate drag coefficient computed with $K - \omega$ and SST turbulence models using different wall boundary conditions for ω .

The convergence histories of the four RANS PDEs and the two turbulence model PDEs of the SST analysis using equation (2.6.7) are reported in the left subplot of figure

7.3, whereas the six convergence histories of the SST analysis using equation (2.6.1) with $S_R = 2500$ are shown in the right subplot of the same figure.

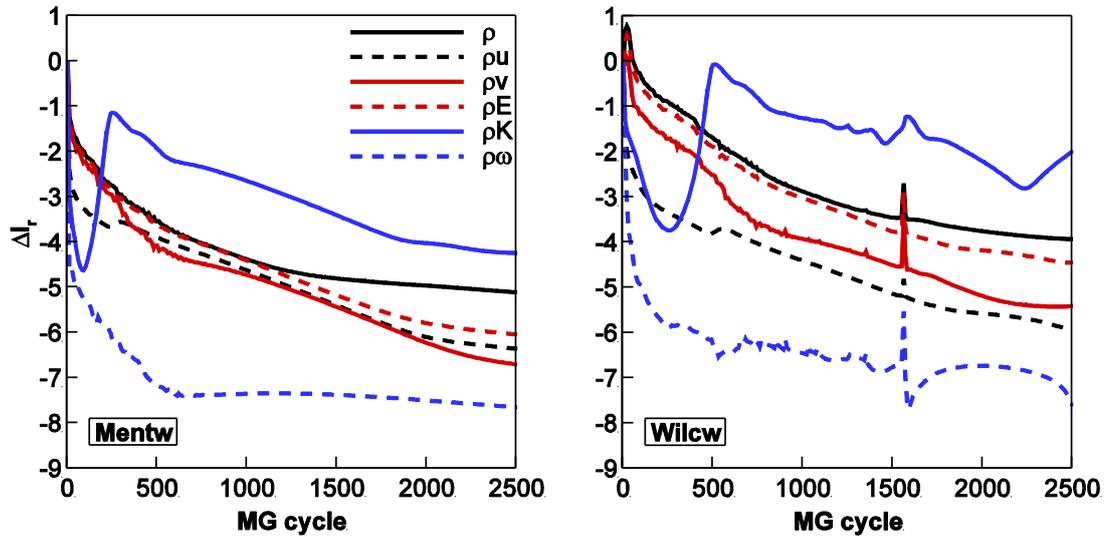


Figure 7.3: Convergence histories of turbulence SST analysis of flat plate using Wilcox's BC ($S_R = 2500$).

The variable on the horizontal axis is the number of MG cycles, and the variable Δl_r on the vertical axis is the logarithm in base 10 of the *root mean square* (RMS) of the cell residuals of the considered conservation equation normalized by the RMS of the cell residuals at the first MG iteration. The convergence histories of the simulation based on equation (2.6.1) are fairly oscillatory, and the overall reduction of the residuals is smaller than that achieved by the simulation based on equation (2.6.7). This is due to the difference in the estimates of the strong stream-wise gradient of ω_w at the LE on the fine and coarser grids, caused by insufficient spatial resolution in the stream-wise direction of the coarser grids. This circumstance results in oscillations of the flow residuals strictly localized to the LE of the flat plate. As expected, this problem does not occur without MG. This phenomenon is also absent when using equation (2.6.7) with or without MG, as this condition does not introduce any significant stream-wise gradient of ω at the wall on any grid level. For flow problems featuring rounded LEs, the oscillatory character of the convergence histories of the turbulent MG solver based on the boundary condition (2.6.1) is less pronounced than in the right subplot of figure 7.3. This is because of the smaller stream-wise gradients of ω_w associated with the use of equation (2.6.1) with respect to the flat plate problem, where the sharp LE essentially leads to a flow singularity. It is also expected that increasing the stream-wise grid refinement in the wall proximity reduces the residual oscillations under analysis, and preliminary numerical tests of this

research appear to confirm this hypothesis. Therefore, in view of its previously discussed advantages, the wall BC (2.6.1) with $S_R = 2500$ has been used for all simulations presented in the chapter 8, unless otherwise stated.

7.2 Steady NACA4412

In this section, the turbulent flow field past the NACA4412 airfoil corresponding to the condition of maximum lift is considered. The free-stream Mach number is 0.2, and the AoA is 13.87° . The Reynolds number based on the airfoil chord and the free-stream velocity is 1.52×10^6 . This operating regime is characterized by a flow reversal in the rear part of the airfoil suction side. Detailed hot-wire boundary layer measurements have been performed at NASA Ames and reported in [182]. Similar numerical studies have been carried out by Menter [40] and Moryossef [179] more recently. Other studies can be found here [64, 183]. The C-grid adopted for the flow simulations reported below is that available on the web site of the NASA CFD code CFL3D [184]. This grid is a 2-block grid and it has 177 points along the airfoil, 41 points in the C-cut, and 81 points in the normal-like direction, giving 20,480 cells (figure 7.4). The far-field boundary is at about 20 chords from the airfoil, and the distance of the first grid points off the airfoil surface from the airfoil surface is 4.0×10^{-3} % of the chord with y^+ always less than 1. The turbulent intensity at the far-field boundary is equal to 1% and the ratio μ_T/μ is set to 10^{-4} . In order to further assess the impact of the wall BC for ω on the accuracy of the computed solution, three analyses have been performed, one computing ω_W with Wilcox's condition (2.6.1) with $S_R = 2500$, one using equation (2.6.1) with $S_R = 100$, and one computing ω_W with Menter's condition (2.6.7). The turbulent flow field under investigation has been computed using only the SST turbulence model making use of both second and first order upwind schemes. The Multigrid method has been applied to the system in order to increase the speed of the computational time making use of the low (6.3.2) and high (6.3.3) order restriction operators. Bilinear interpolation is used for the prolongation operator. Three multigrid levels with V-cycle have been used in all computations where for each multigrid cycle, 5 RK iterations have been done on the first level, 5 RK iterations on the second level and 2 RK iterations on the coarsest level. All

computations started from freestream values using $CFL = 1$ and the total number of multigrid cycles was set to 2500.

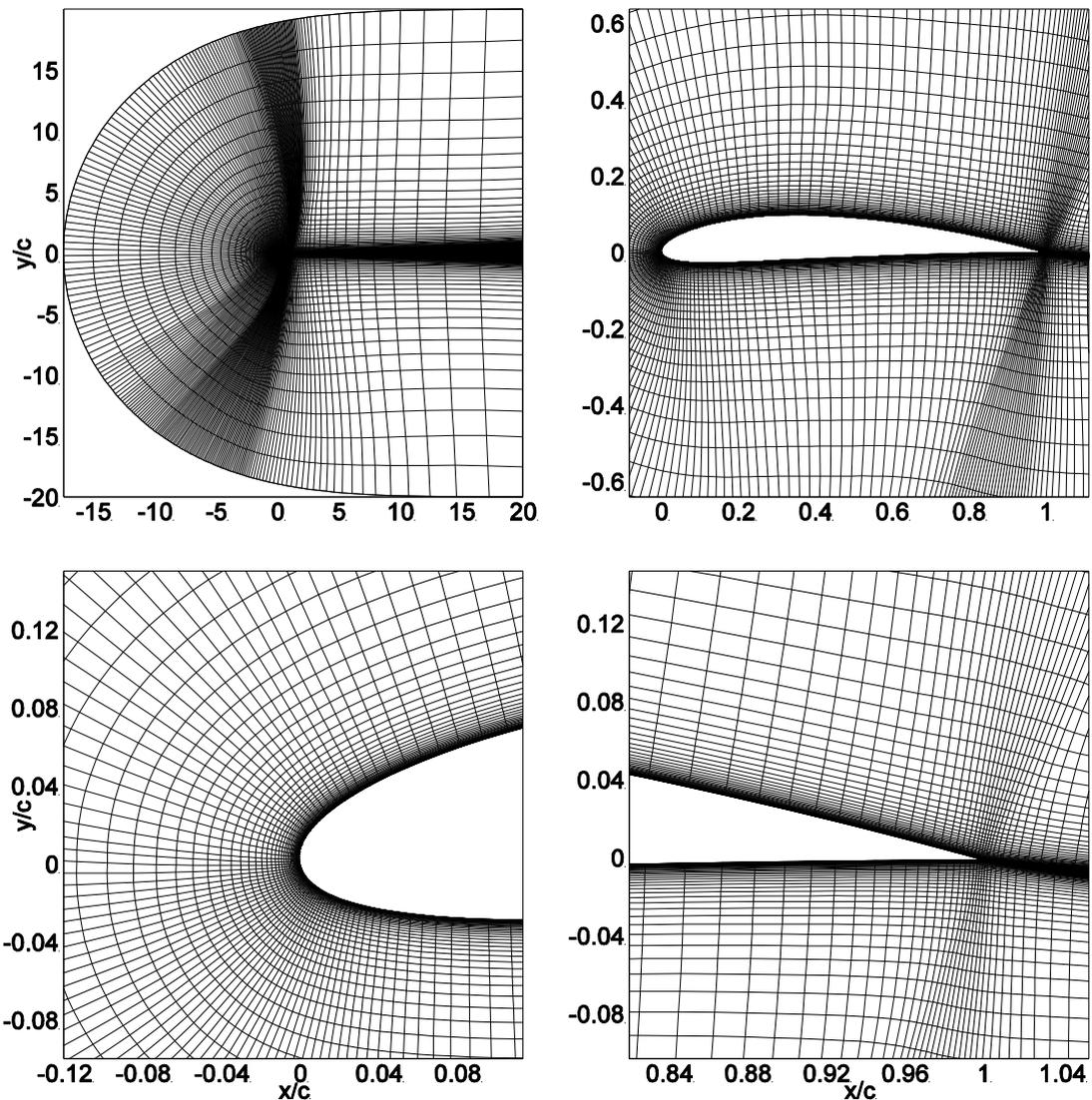


Figure 7.4: Grid for NACA4412. **Top left plot:** computational area. **Top right plot:** airfoil area. **Bottom left plot:** leading edge area. **Bottom right plot:** trailing edge area.

Figure 7.5 presents the comparison between measured and numerical results for the static pressure coefficient around the airfoil surface. For both the experimental and computed results, the variable x/c along the horizontal axis denotes the position along the airfoil chord, and the variable c_p along the vertical axis denotes the static pressure coefficient on the airfoil surface. The computed results labelled ‘wlc1w’ are those obtained using Wilcox’s wall BC with $S_R = 2500$, the computed results labelled ‘wlc0w’

are those obtained using Wilcox's wall BC with $S_R = 100$, and the computed results labelled 'mentw' are those obtained using Menter's wall BC.

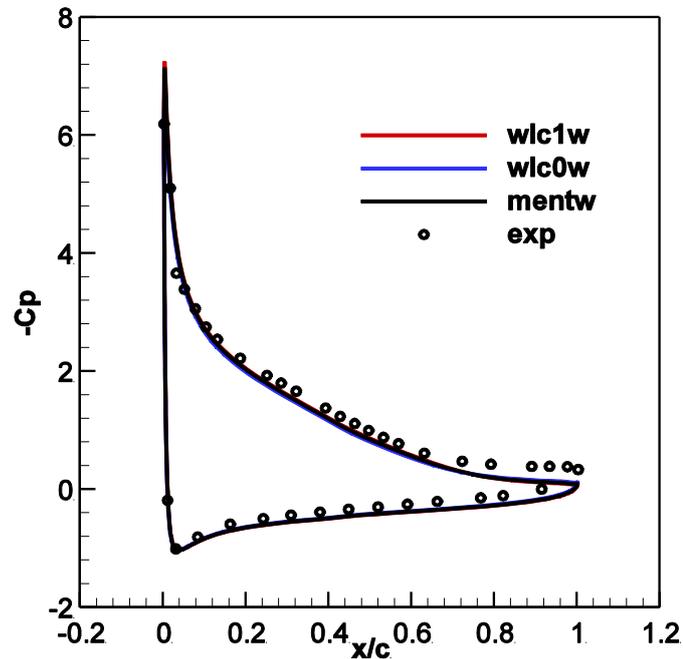


Figure 7.5: Turbulent flow past NACA4412 airfoil at $AoA = 13.87^\circ$, $Re = 1.52 \times 10^6$ and $M = 0.2$. Comparison of measured static pressure coefficient (labelled 'exp') and numerical results computed with SST turbulence model using Wilcox's wall BC with $S_R = 2500$ (profiles labeled 'wlc1w'), Wilcox's wall BC with $S_R = 100$ (profiles labelled 'wlc0w') and Menter's wall BC (profiles labelled 'mentw').

The numerical results for the static pressure coefficient on the surface of the airfoil, using all the different wall boundary conditions, are in very good agreement with the experimental data. The choice of the first or second order discretisation scheme for the turbulent equations had a negligible impact on the computed static pressure coefficient.

The six subplots of figure 7.6 present the comparison between measured and computed velocity profiles at the six chord-wise positions using a second order upwind scheme for the turbulent equations. The position of the profiles reported below is indicated by the value of the x/c variable reported in each subplot. For both experimental and computed results, the variable y/c along the vertical axis denotes the distance from the airfoil, measured along a line orthogonal to the chord and intersecting the chord at the indicated

value of x/c . The variable u/u_∞ along the horizontal axis denotes the velocity component parallel to the chord taken along the above said line orthogonal to the chord itself and non-dimensionalized by the free-stream velocity u_∞ .

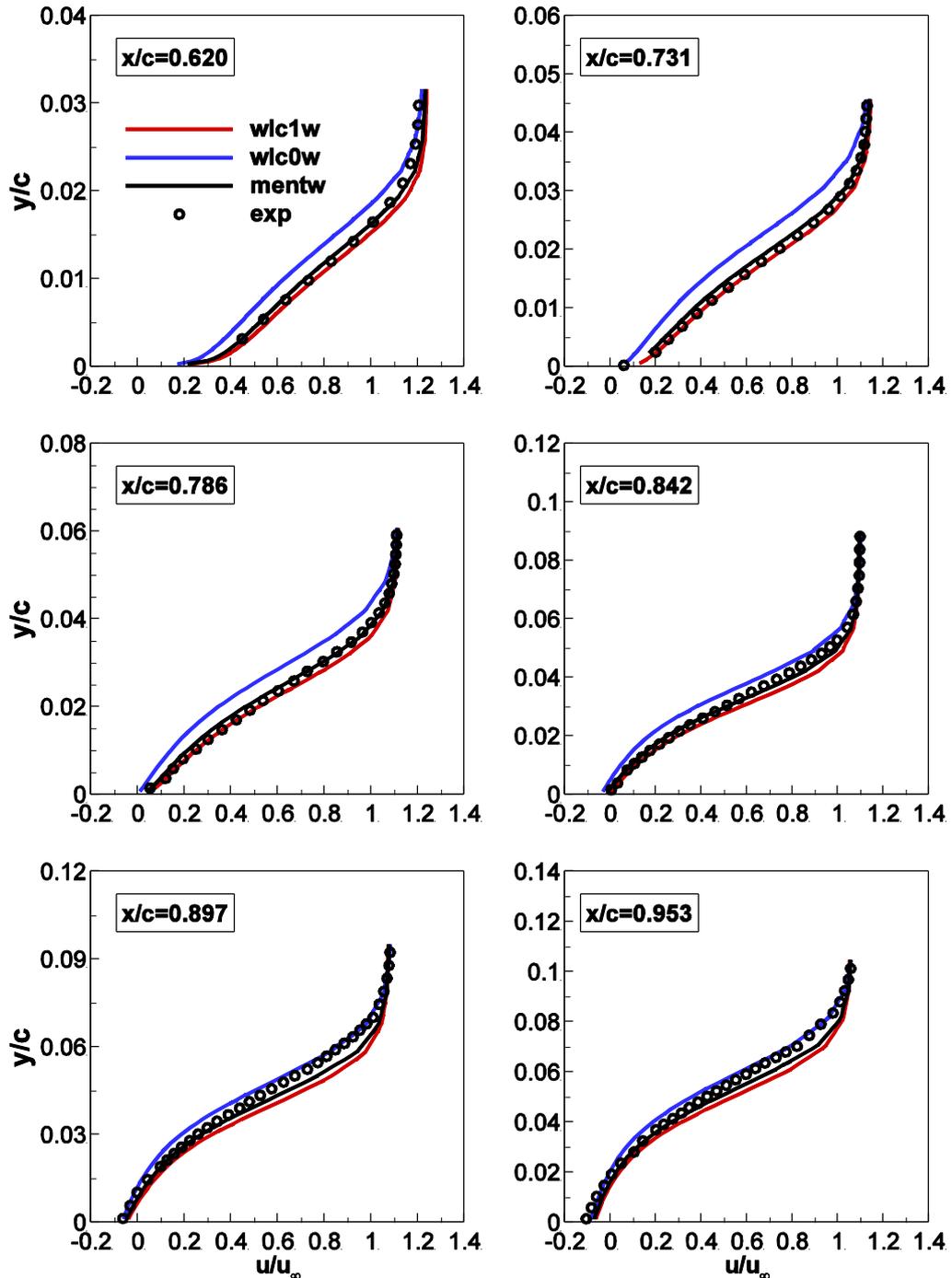


Figure 7.6: Turbulent flow past NACA4412 airfoil at $AoA = 13.87^\circ$, $Re = 1.52 \times 10^6$ and $M = 0.2$. Comparison of measured boundary layer velocity profiles (labelled 'exp') and velocity profiles computed with a second order SST turbulence model using Wilcox's wall BC with $S_R = 2500$ (labelled 'wlc1w'), Wilcox's wall BC with $S_R = 100$ (profiles labelled 'wlc0w'), and Menter's wall BC (labelled 'mentw') at six chord-wise positions.

Examination of the velocity profiles presented in figure 7.6 confirms that the results obtained using different wall BC for ω can differ significantly, and also that the solution obtained using Wilcox's wall BC with $S_R = 2500$ is significantly closer to experimental data compared to that obtained with Wilcox's wall BC $S_R = 100$. This level of agreement is comparable with that of Menter [40]. The agreement between the solutions obtained with Menter's and Wilcox's wall BCs can be further improved by increasing the value of the constant S_R appearing in equation (2.6.1) beyond 2500, but this has not been done because the recalibration of this parameter is beyond the scope of this study.

The six subplots of figure 7.7 present the comparison between measured and computed velocity profiles at the six chord-wise positions, using the first and the second order upwind scheme for the turbulence equations. The position of each profile is indicated by the value of the x/c variable reported in each subplot. The structure of horizontal and vertical axes in figure 7.7 follow the same pattern as in figure 7.6. The computed results labelled 'wlc1wf' and 'mentwf' are obtained by using Wilcox's wall BC with $S_R = 2500$ and Menter's wall BC, respectively, with a first order upwind scheme. Similarly, the computed results labelled 'wlc1ws' and 'mentws' are those obtained using Wilcox's wall BC with $S_R = 2500$ and Menter's wall BC, respectively, with a second order upwind scheme. Examination of the velocity profiles presented in figure 7.7 confirms that the results obtained using first order upwind scheme for the turbulent equations fail to predict correctly the boundary layer profile. This failure can be seen at the positions $x/c = 0.620$, $x/c = 0.731$ and $x/c = 0.788$ of figure 7.7. For the three above said positions, the values of the velocities for both wall boundary conditions (Wilcox and Menter) are farther away from the experimental data than the values computed using the second order scheme for the turbulence equations. In the other positions presented in figure 7.7, namely $x/c = 0.842$, $x/c = 0.897$ and $x/c = 0.953$, one sees that the experimental data appear to be between the results obtained using first order upwind scheme for the turbulence equations and the results obtained using second order upwind scheme. Other studies [64, 179] used a first order upwind scheme for the turbulence equations, most likely to reduce the computational cost and improve the numerical stability. Also Menter [40], commenting on the turbulent flow analysis of this test case using a similar grid to that adopted in the present analysis stated that the solution is virtually independent of the order of the scheme used for the turbulence equations, but this statement appears to contradict the outcome of the analysis discussed above.

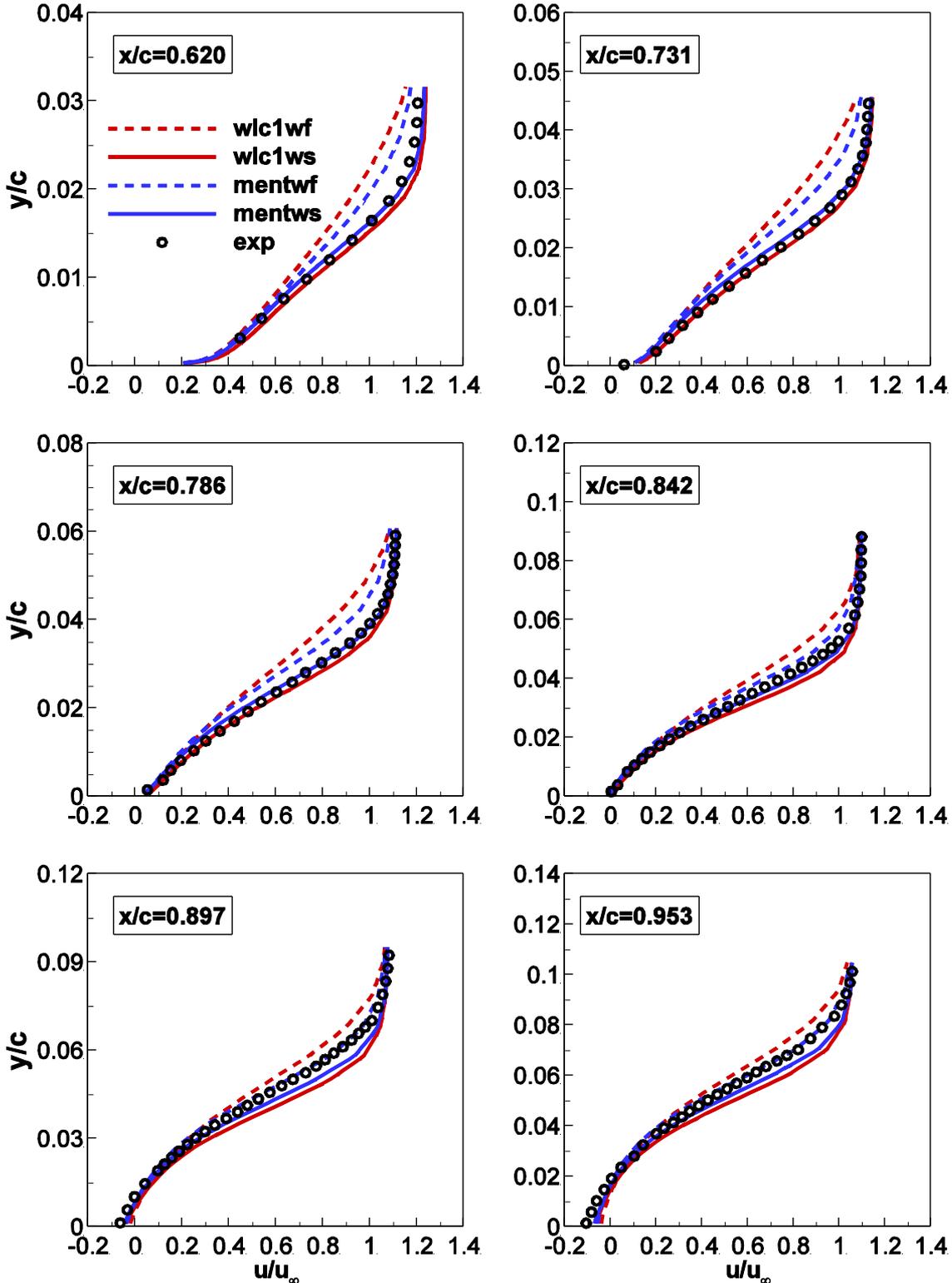


Figure 7.7: Turbulent flow past NACA4412 airfoil at $AoA = 13.87^\circ$, $Re = 1.52 \times 10^6$ and $M = 0.2$. Comparison of measured boundary layer velocity profiles (labelled 'exp') and velocity profiles computed with first (profiles labelled 'mentwf' and 'wlc1wf') and second (labelled 'mentws' and 'wlc1ws') order SST turbulence model using Wilcox's wall BC with $S_R = 2500$ and Menter's wall BC at six chord-wise positions.

In order to further investigate the impact of using a first or second order upwind scheme for the turbulence equations, one additional comparative study has been carried out. The turbulent flow field past the NACA4412 airfoil has been computed by keeping the same values of the Reynolds and Mach number, but decreasing the AoA from 13.87° to 3.87° . In this way the flow will not be reversed in the rear part of the airfoil suction side and no separation will occur. The same type of comparative analysis reported in figure 7.7 has been carried out for this new flow regime and it has been found that the velocity profiles obtained using a first order upwind scheme for the turbulence equations and the results obtained using a second order upwind scheme now present negligible differences. This indicates that the order of the discretization of the turbulence equations is highly important in order to predict correctly the flow field when separation occurs.

One explanation of the differences apparent in the velocity profiles between the first and second order schemes when separation occurs in the flow field, is that the grid refinement is insufficient. More precisely, using the given grid to solve fluid problems with strong gradients, such as those associated with the flow reversal under analysis, the discretisation error of the first order scheme is likely to be significantly higher than that of the second order scheme. This large difference explains the solution differences observed in the high-AoA case. When the flow reversal is suppressed by lowering the AoA, however, flow gradients are substantially reduced. In this circumstance, the discretisation errors of the first and second order schemes using the same given grid are substantially closer, and no significant solution difference is observed. Other numerical experiments performed within this research but not reported herein for brevity have revealed that, even in the case of flow fields featuring significant flow gradients like flow reversals (turbulent flow field past the NACA4412 airfoil under analysis) and shocks (turbulent flow field past the RAE2822 airfoil with Mach number equal to 0.73), the solutions of the first and second order discretisation becomes asymptotically close as the grid refinement is increased. These investigations suggest that a second order accuracy of the turbulence model should always be used particularly in the case of flow separations (a feature that can affect both subsonic and transonic problems), and shocks. In these cases, the use of the second order discretisation enables a high solution accuracy using computational grids with moderate spatial refinement. These considerations highlight the importance of the robust fully-coupled turbulent multigrid solution procedure developed in this research. The high robustness of this implementation allows one to use the second order discretisation of the turbulence model equations even with relatively coarse grids, whereas many other published research works use a first order discretisation in the interest

of numerical stability. Unfortunately, that choice may also have significant detrimental effects on the solution accuracy.

The values of the lift coefficient C_L , the drag coefficient C_D , and the pitching moment coefficient (computed about the point lying on the airfoil chord at 25% of the chord from leading edge) C_M obtained with the three COSA SST simulations are reported in Table 7.2. The significant variations of the force coefficients with the wall BC for ω , highlight once again the impact of this modelling choice on the computed forces. Also the different orders of the upwind scheme for the turbulent equations has a significant impact on the values of the forces. As one sees the value of the drag coefficient has been increased when a first order upwind scheme has been used for the turbulence equations, compared to the value computed with a second order scheme.

BC	C_L	C_D	C_M
wlc1wf	1.624	0.04441	0.05895
wlc1ws	1.696	0.03743	0.06346
wlc0ws	1.643	0.04219	0.06088
mentwf	1.653	0.06157	0.05845
mentws	1.680	0.03884	0.06257
exp	1.669		

Table 7.2: Turbulent flow past NACA4412 airfoil at $AoA = 13.87^\circ$, $Re = 1.52 \times 10^6$ and $M = 0.2$: lift, drag and pitching moment coefficients computed with SST turbulence model using different wall boundary conditions for ω with first and second order upwind scheme for the turbulent equations

Figure 7.8 presents the comparison between the first and the second order numerical results for the skin friction coefficient around the airfoil surface. The variable x/c along the horizontal axis denotes the position along the airfoil chord, and the variable c_f along the vertical axis denotes the skin friction coefficient on the airfoil surface.

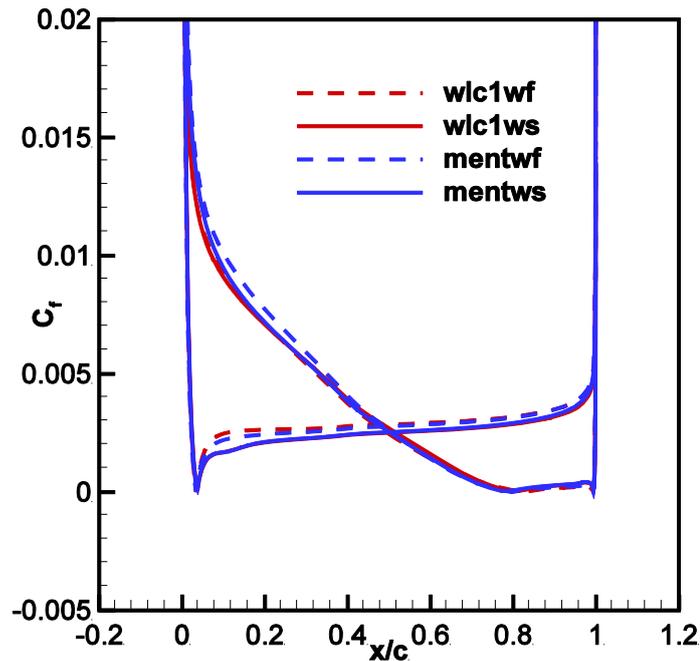


Figure 7.8: comparison of skin friction coefficients at $AoA = 13.87^\circ$, $Re = 1.52 \times 10^6$ and $M = 0.2$ between results computed with first (labelled 'mentwf' and 'wlc1wf') and second (labelled 'mentws' and 'wlc1ws') order discretisation using Wilcox's wall BC with $S_R = 2500$ and Menter's wall BC.

Figure 7.8 highlights that there is a significant difference between the calculated results obtained using a first order upwind scheme and the calculated results obtained using a second order upwind scheme. As a result, these differences affect the values of the drag coefficients presented on the table 7.2. Moreover, one sees that at 80% of the airfoil chord on the suction side, the skin friction coefficient is nearly zero due to the fact that separation occurred.

Also, numerical experiments for the turbulent flow field past the NACA4412 airfoil have been done, using the exact SST equation (5.2.34) for the update process of the RK scheme. This analysis has highlighted that both the computed solution and the convergence histories of the code using either the exact SST update (equation (5.2.34)) or the approximate update (equation (5.2.32)) present negligible differences. For this reason all the analyses presented in chapter 8 for low speed cases are based on the use of equation (5.2.32).

The convergence histories of the four RANS PDEs and the two turbulence model PDEs of the SST analysis using equation (2.6.7) are reported in the top subplots of figure 7.9, whereas the six convergence histories of the SST analysis using equation (2.6.1) with $S_R = 2500$ are shown in the bottom subplots of the same figure.

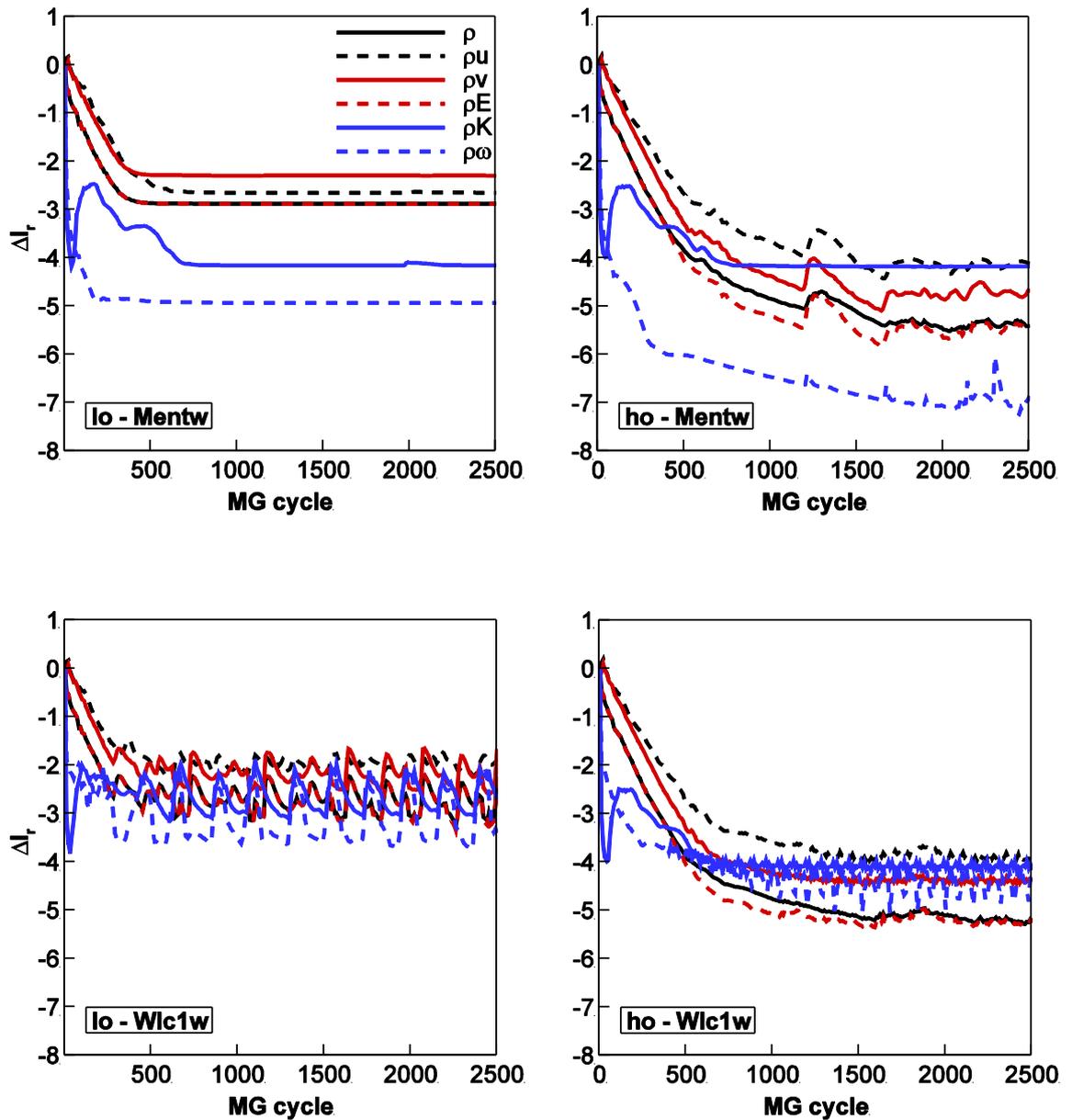


Figure 7.9: Convergence histories of turbulent SST analyses of flow field past NACA4412 airfoil at $AoA = 13.87^\circ$, $Re = 1.52 \times 10^6$, $M = 0.2$. **Top Left plot:** simulation using Menter's BC with low order prolongation operator. **Top Right plot:** simulation using Menter's BC with high order prolongation operator. **Bottom Left plot:** simulation using Wilcox's BC ($S_R = 2500$) with low order prolongation operator. **Bottom Right plot:** simulation using Wilcox's BC ($S_R = 2500$) with high order prolongation operator.

The two subplots on the left hand side of figure 7.9 are the residuals using the low order restriction operator defined by equation (6.3.2), whereas the residuals obtained with the high order restriction operator defined by equation (6.3.3) are presented in the subplots on the right hand side of the same figure. Bilinear interpolation is used for the prolongation operator whereas a limiter is added to the prolongation operator, defined by equation (6.6.3), in order to guarantee the positive values of the turbulence variables. Examination of the convergence histories of the simulation presented in figure 7.9 confirms that the low order restriction operator fails to reduce the residuals more than three orders of magnitude. Comparing the plots on the right hand side of the figure 7.9 and the plots on the left hand side of the same figure one sees the high importance of using a high order restriction operator in order to achieve a converged solution. The convergence histories of the simulation based on equation (2.6.7) are smoother than those of the simulation based on (2.6.1). In both cases, however, all residuals decrease by at least four orders of magnitude when a high order restriction operator is used. It is also expected that increasing the spatial refinement of the computational grid reduces the residual oscillations under analysis, and preliminary numerical tests appear to confirm this hypothesis.

7.3 Steady RAE2822

In this section, the turbulent flow field past the RAE2822 airfoil with sharp trailing edge is considered. This operating regime is characterized by a transonic flow where shock boundary layer separation appears at 55% of the airfoil chord. The free-stream Mach number is 0.73, and the AoA is 2.8° . The Reynolds number based on the airfoil chord and the free-stream velocity is 6.5×10^6 . Detailed experimental measurements have been performed by Cook et al. and reported in [185]. Similar numerical studies have been done by Fassbender [162] and Swanson [140] recently. The C-grid has been generated using the NUMECA package [186]. This grid has two blocks and it has 449 points along the airfoil, 97 points in the C-cut, and 129 points in the normal-like direction, giving 81,920 cells (figure 7.10). The far-field boundary is at about 20 chords from the airfoil, and the distance of the first grid points off the airfoil surface from the airfoil surface is

$1.0 \times 10^{-3}\%$ of the chord with y^+ always less than 1. The turbulent intensity at the far-field boundary is equal to 1% and the ratio μ_T/μ is set to 10^{-4} .

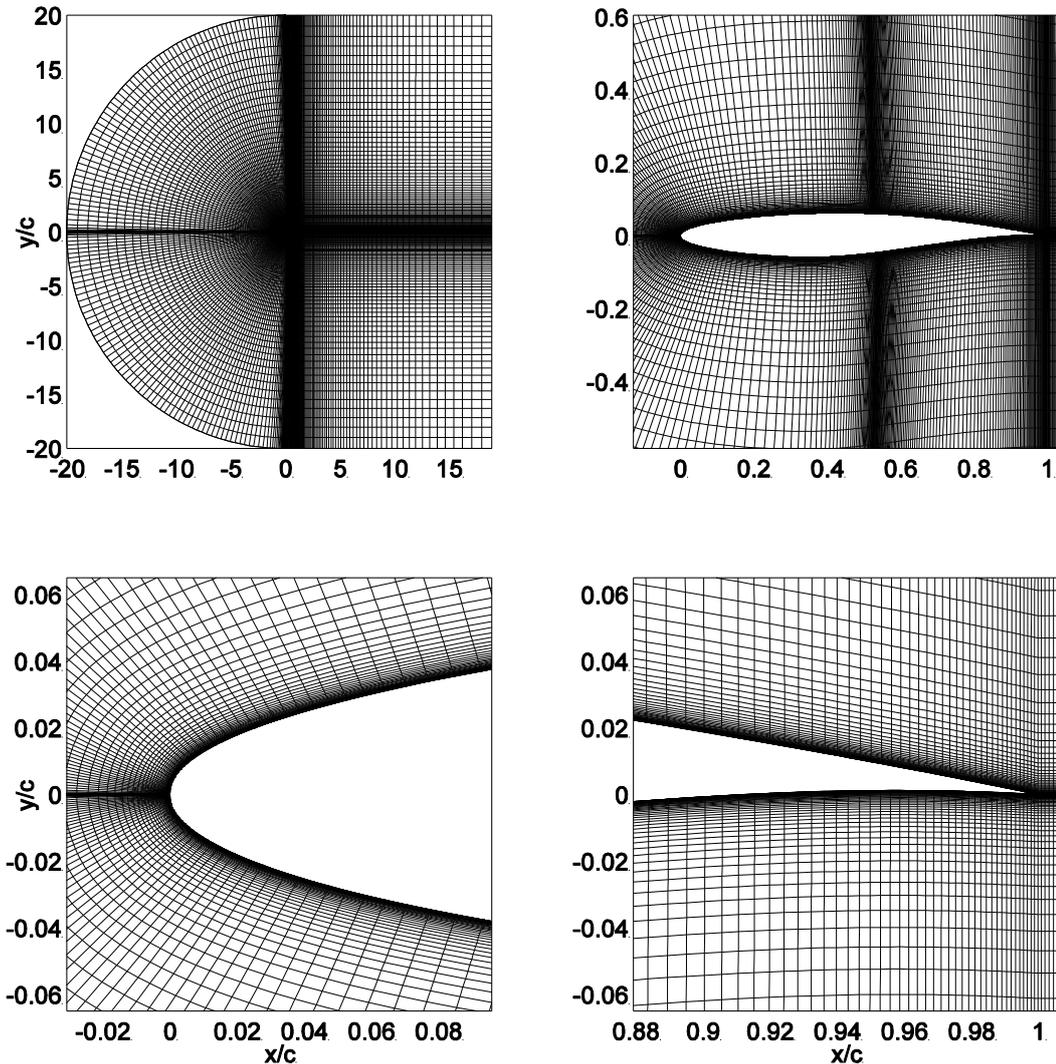


Figure 7.10: *Grid for RAE2822. Top left plot: computational area. Top right plot: airfoil area. Bottom left plot: leading edge area. Bottom right plot: trailing edge area.*

In order to further assess the impact of the wall BC for ω on the accuracy of the computed solution, two analyses have been performed, one computing ω_W with Wilcox's condition (2.6.1) with $S_R = 2500$ and one computing ω_W with Menter's condition (2.6.7). The turbulent flow field under investigation has been computed using only the SST turbulence model making use of both second and first order upwind scheme. The multigrid method has been applied to the system in order to increase the convergence rate of the simulation. Both the calculation using the low-order restriction operator provided by equation (6.3.2), and that using the high-order restriction operator given by equation

(6.3.3) have been carried out. Bilinear interpolation is used for the prolongation operator. Three multigrid levels with V-cycle have been used in all computations. For each multigrid cycle, 5 RK iterations have been done on all grid levels. All computations started from the freestream initialisation, and the total number of multigrid cycles was set to 2,500. CFL ramping was used to increase the maximum value of the CFL number for all equations. The maximum CFL number for the NS and SST equations were set to 4 and 6, respectively.

Figure 7.11 presents the comparison between measured data and numerical results for the static pressure coefficient around the airfoil surface. For both the experimental data and the computed results, the variable x/c along the horizontal axis denotes the position along the airfoil chord, and the variable c_p along the vertical axis denotes the static pressure coefficient on the airfoil surface. The computed results labelled ‘wlc1w’ are those obtained using Wilcox’s wall BC with $S_R = 2500$, and the computed results labelled ‘mentw’ are those obtained using Menter’s wall BC. The symbol ‘f’ and ‘s’ at the end of each label indicate if the computation used first or second order upwind scheme for the turbulence equations, respectively. The symbol ‘ent1’ and ‘ent0’ indicate if the computation used the entropy fix limiter (4.4.1) or no entropy fix, respectively.

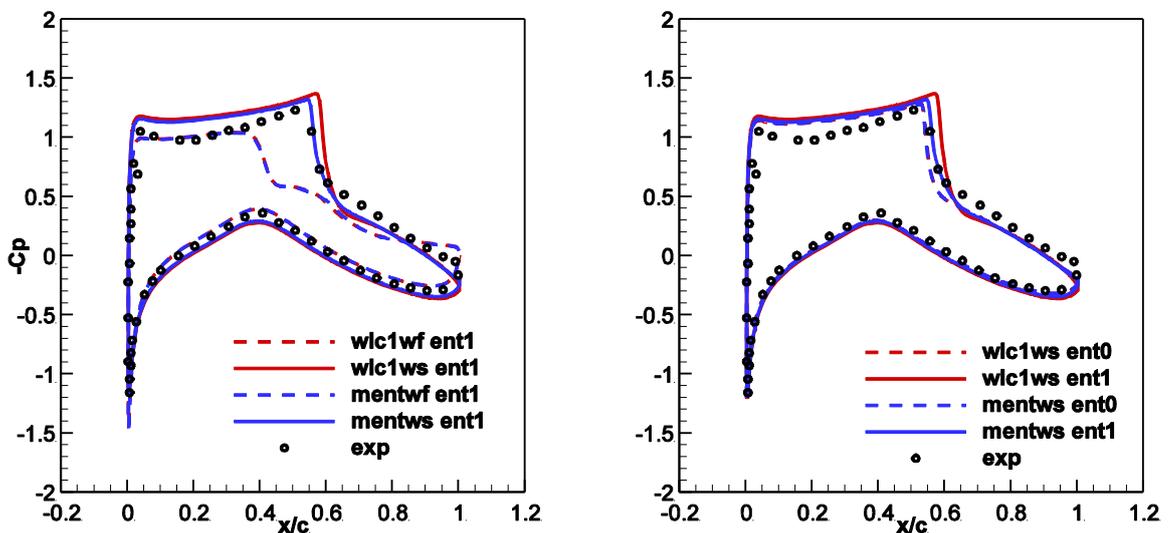


Figure 7.11: Turbulent flow past RAE2822 airfoil at $AoA = 2.8^\circ$, $Re = 6.5 \times 10^6$ and $M = 0.73$. Comparison of measured static pressure coefficient (labelled ‘exp’) and numerical results computed with SST turbulence model using Wilcox’s wall BC with $S_R = 2500$ and Menter’s wall BC. **Left plot:** comparison of the first and second order discretisation scheme for the turbulence equations. **Right plot:** comparison with or without entropy fix.

The numerical results for the static pressure coefficient on the surface of the airfoil, using the second order scheme for the turbulence equations for both wall boundary conditions, are in very good agreement with the experimental data (left plot). Both wall BCs capture correctly the shock that appears at 55% of the chord. It is evident that Menter's wall BC is slightly better than Wilcox BC compared to the experimental data but this difference is negligible. The choice of the first order upwind scheme for the turbulent equations fails to predict correctly the area where the shock-induced boundary layer separation occurs. As expected the order of discretization of the turbulence equations is highly important in order to predict correctly the pressure coefficient when separation occurs. Park et al [83] briefly investigated this topic and found that the use of a first order scheme for the turbulence equations, results in large discrepancies of the velocity profiles compared to the use of a second order scheme. This transonic test case has been used by many researches [59, 61] for validation purposes and they always use second order schemes to compute the convective fluxes of the turbulent equations. The subfigure on the right hand side of figure 7.11 highlights the effect of the entropy-fix. The entropy fix prevents the occurrence of unphysical features like expansion shocks, which violate the second law of thermodynamics. An entropy-fix correction is commonly used when the flow is transonic.

Figure 7.12 provides the comparison between the two different wall BC (wlc1w, mentw) for the skin friction coefficient against x/c using both the first and the second order discretisation schemes for the turbulence equations. Examination of the skin friction coefficient presented in figure 7.12 confirms that the two different wall BCs using a second order discretisation scheme capture correctly the shock boundary layer separation that appears at the 55% of the airfoil chord, despite the significant differences between the mathematical forms of wall BCs. One sees that, close to the trailing edge, the c_f computed by Wilcox BC takes higher values than in the case where Menter BC is used. Moreover the skin friction coefficient presented in figure 7.12 confirms once again that the calculation of the flow field using the first order discretisation scheme (dash lines), with any type of wall boundary condition, fails to predict correctly the shock-induced boundary layer separation that appears at 55% of the airfoil chord, and also differs significantly with respect to the solution obtained with a second order scheme.

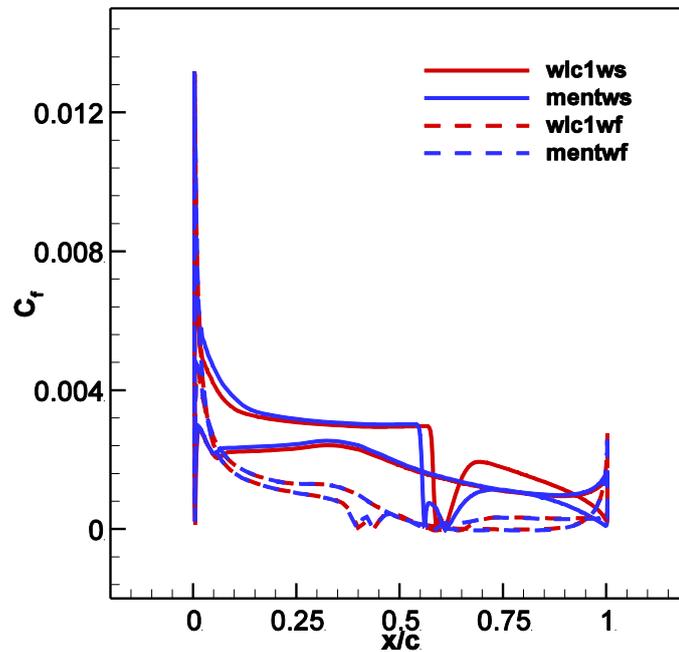


Figure 7.12: Skin Friction coefficient of a turbulent flow past RAE2822 airfoil at $AoA = 2.8^\circ$, $Re = 6.5 \times 10^6$ and $M = 0.73$ using menter's wall BC (labelled 'mentw') and wilcox BC $S_R = 2500$ (labelled 'wlc1w') computed with SST turbulence model.

BC	C_L	C_D	C_M
wlc1ws	0.876	0.019	0.110
mentws	0.832	0.017	0.100
wlc1wf	0.572	0.013	0.059
mentwf	0.569	0.013	0.058
exp	0.801	0.018	

Table 7.3: Turbulent flow past NACA4412 airfoil: lift, drag and pitching moment coefficients computed with SST turbulence model using different wall boundary conditions for ω .

As a result the maximum value of the Mach number computed by the first order scheme is lower than the maximum value of the Mach number computed by the second order scheme. This can be seen also in figure 7.13, where the subfigure on the left hand side is the solution of the flow computed using the first order scheme and the subfigure on the right hand side is the solution of the flow computed using the second order scheme. This figure presents the Mach number contours around the airfoil.

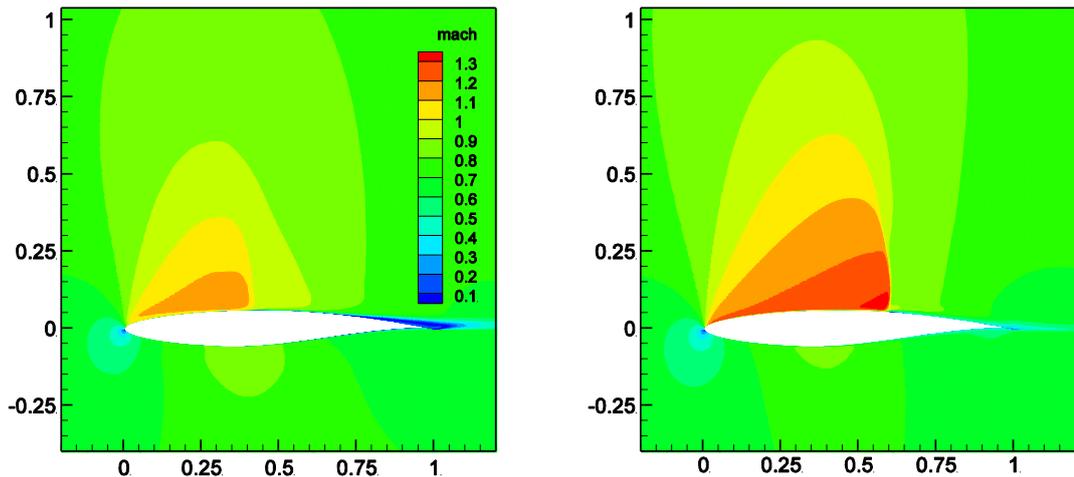


Figure 7.13: Mach contours of a turbulent flow past RAE2822 airfoil at $AoA = 2.8^\circ$, $Re = 6.5 \times 10^6$ and $M = 0.73$ using wilcox BC $S_R = 2500$ computed with SST turbulence model. **Left plot:** first order discretisation scheme for the turbulent equations. **Right plot:** second order discretisation scheme for the turbulence equations.

The values of the lift coefficient C_L , the drag coefficient C_D , and the pitching moment coefficient (computed about the point lying on the airfoil chord at 25% of the chord from leading edge) C_M obtained with the four COSA SST simulations are reported in Table 7.3. The significant variations of the force coefficients using first or second order for the turbulence equations highlight once again the impact of this choice on the computed forces.

The turbulent transonic flow field past the RAE2822 airfoil has also been computed, using equation (5.2.34) for the exact update of the SST equations. Similarly, to the low speed NACA4412 case, the results present negligible differences with respect to the solution computed with the approximate update of the SST equations (choice of equation (5.2.32)). For this reason all the analyses presented in chapter 8 are based on the use of equation (5.2.32).

The convergence histories of the four RANS PDEs and the two turbulence model PDEs of the SST analysis using equation (2.6.7) are reported in the top subplots of figure 7.14, whereas the six convergence histories of the SST analysis using equation (2.6.1) with $S_R = 2500$ are shown in the bottom subplots of the same figure. The two subplots on the left hand side of figure 7.14 are the residuals using low order multigrid operators whereas the residuals obtained with high order operators are presented in the subplots of the right hand side of the same picture.

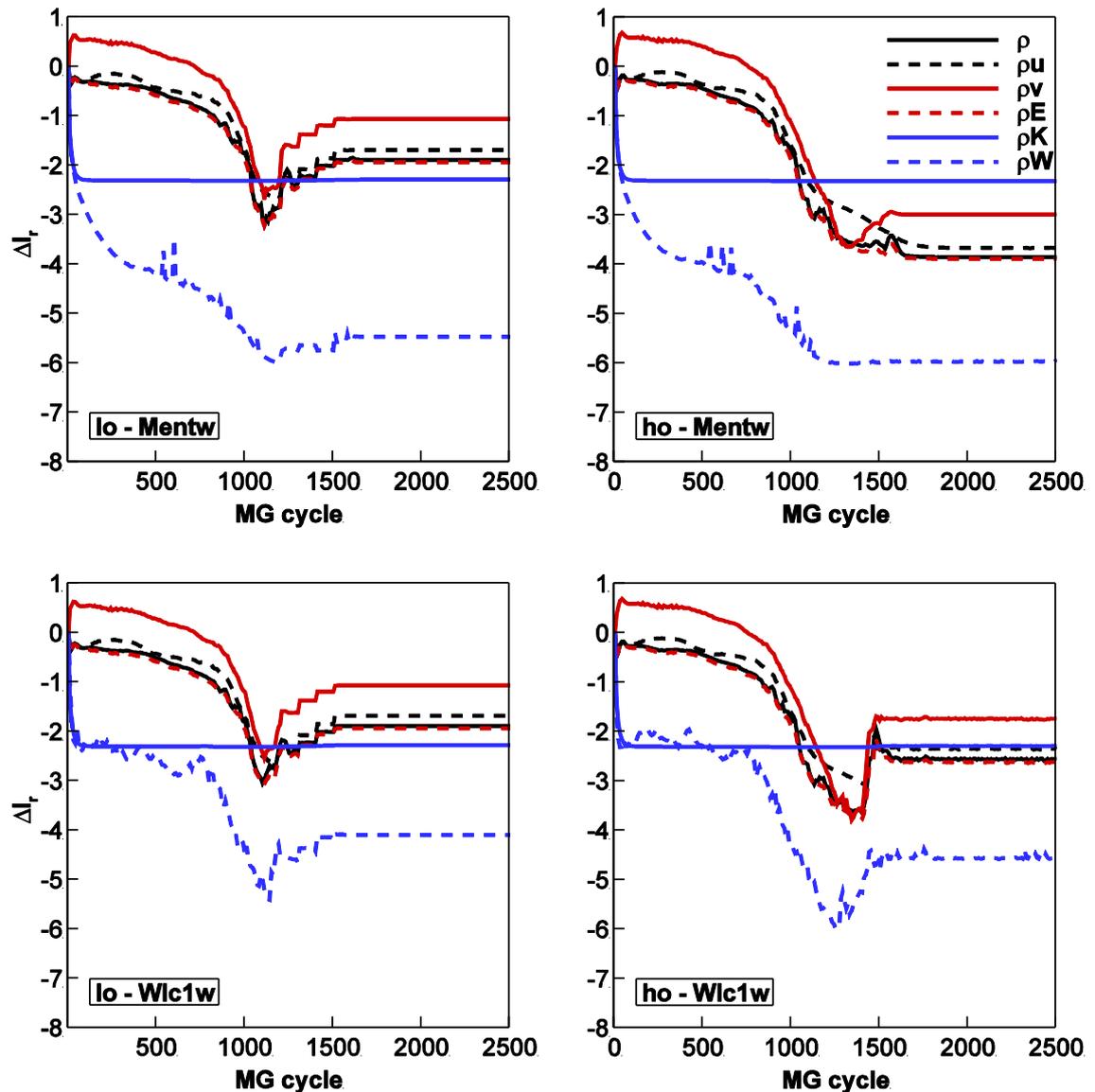


Figure 7.14: Convergence histories of turbulent SST analyses of flow field past RAE2822 airfoil at $AoA = 2.8^\circ$, $Re = 6.5 \times 10^6$ and $M = 0.73$. **Top Left plot:** simulation using Menter's BC with low order prolongation operator. **Top Right plot:** simulation using Menter's BC with high order prolongation operator. **Bottom Left plot:** simulation using Wilcox's BC ($S_R = 2500$) with low order prolongation operator. **Bottom Right plot:** simulation using Wilcox's BC ($S_R = 2500$) with high order prolongation operator.

Examination of the convergence histories of the simulation presented in figure 7.14 confirms again that the results obtained using the low order restriction operator fail to reduce the residuals more than two orders of magnitude. Like for the NACA4412 test case, one notices the importance of using a high order restriction operator in order to achieve a converged solution. Moreover, the plot on the top right hand side of the figure 7.14 confirms that all residuals, with Menter's wall BC, decrease by at least three orders of magnitude when a high order restriction operator is used. For Wilcox wall BC, the residuals decrease by at least 2 orders of magnitude. Although it is not confirmed in this study, the refinement results of the turbulent flat plate and the NACA4412 test case discussed in the previous sections, suggest that a similar improvement of the convergence histories would be expected by increasing the spatial refinement of the computational grid.

Chapter 8

RESULTS

Oscillating airfoils are representative of many diverse engineering problems, ranging from energy engineering to aeronautical engineering. This chapter presents results regarding two substantially different engineering applications. The first case is the flow analysis of the hydrokinetic turbine, an oscillating wing device presently being considered for the exploitation of marine energy. The fluid dynamics of this device is thoroughly analysed using the laminar and the turbulent solvers of COSA code. These CFD simulations are performed at a low subsonic Mach number, and the results of the laminar and turbulent simulations are used to explain the fluid dynamics phenomena accounting for the high efficiency of this device. These results highlight the differences between the unsteady aerodynamic characteristics observed in the two regimes, and hence report the reasons for the different achieved performance. All time-domain simulations of the hydrokinetic turbine are also performed to highlight the prediction capabilities of the COSA time-domain solver: all results are in excellent agreement with the numerical analyses of the same device performed with the commercial CFD code FLUENT and published by researchers of the University of Quebec [35, 38]. The second engineering application is the aerodynamic analysis of a transonic oscillating airfoil, representative of the cross section of a transonic aircraft wing. This type of transonic problem is used to analyse the flutter characteristics of transonic wings. The transonic problem is investigated using both the time-domain and the harmonic balance solvers of COSA code, and comparison with available experimental data are also presented. The main purpose of the transonic time-domain analysis is to highlight the predictive capabilities of the COSA solver, whereas the harmonic balance simulations are performed to highlight the reduction of run-time of the harmonic balance with respect to the run-time of the time-domain analysis. The harmonic balance and time domain comparison analysis is then repeated using a subsonic oscillating airfoil where even larger run-time reduction of the harmonic balance with respect to the run-time of the time domain is highlighted. Overall, the results of this chapter demonstrate how accurately and efficiently a single CFD code, COSA, can solve a wide range of fluid dynamics engineering problems of practical interest.

8.1 Hydrokinetic turbines

This section focuses on the 2D turbulent analysis of an oscillating wing simultaneously heaving and pitching, and is subdivided into three subsections. The first subsection presents the fundamental theory of the hydrokinetic turbines, the second subsection presents the physical and computational set-up of the flow analyses, along with the assessment of the temporal and spatial refinement of the grids used in this study, whereas the third subsection investigates in detail the unsteady flow mechanisms enabling the efficiency of the energy extraction to be controlled and possibly maximized, and it presents an analysis of the qualitative and quantitative differences between a turbulent flow regime with Reynolds number of $1.5 \cdot 10^6$ and the laminar regime analysed in [35, 36].

8.1.1 Fundamentals of hydrokinetic turbines

Here an oscillating wing is defined as an airfoil experiencing simultaneous pitching $\theta(t)$ and heaving $h(t)$ motions. The following mathematical representation of the imposed motion is that adopted in [35]. Taking a pitching axis located on the chord line at position x_p from the *leading edge* (LE), the airfoil motion is expressed as:

$$\theta(t) = \theta_0 \sin(\omega t) \rightarrow \Omega(t) = \theta_0 \omega \cos(\omega t) \quad (8.1.1)$$

$$h(t) = h_0 \sin(\omega t + \varphi) \rightarrow v_y(t) = h_0 \omega \cos(\omega t + \varphi) \quad (8.1.2)$$

where θ_0 and h_0 are respectively the pitching and heaving amplitudes, $\Omega(t)$ is the pitching velocity, $v_y(t)$ is the heaving velocity, ω is the angular frequency and φ is the phase between heaving and pitching. In this study, φ is set to 90° , and the NACA0015 airfoil is selected. The free stream velocity is denoted by u_∞ and the angular frequency ω is linked to the oscillation frequency f by the relationship $\omega = 2\pi f$. The prescribed oscillating motion is depicted in the top sketch of Figure 8.1.

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

$$a(t) = \arctan(-v_y(t)/u_\infty) - \theta(t) \quad (8.1.3)$$

$$v_e(t) = \sqrt{u_\infty^2 + v_y(t)^2} \quad (8.1.4)$$

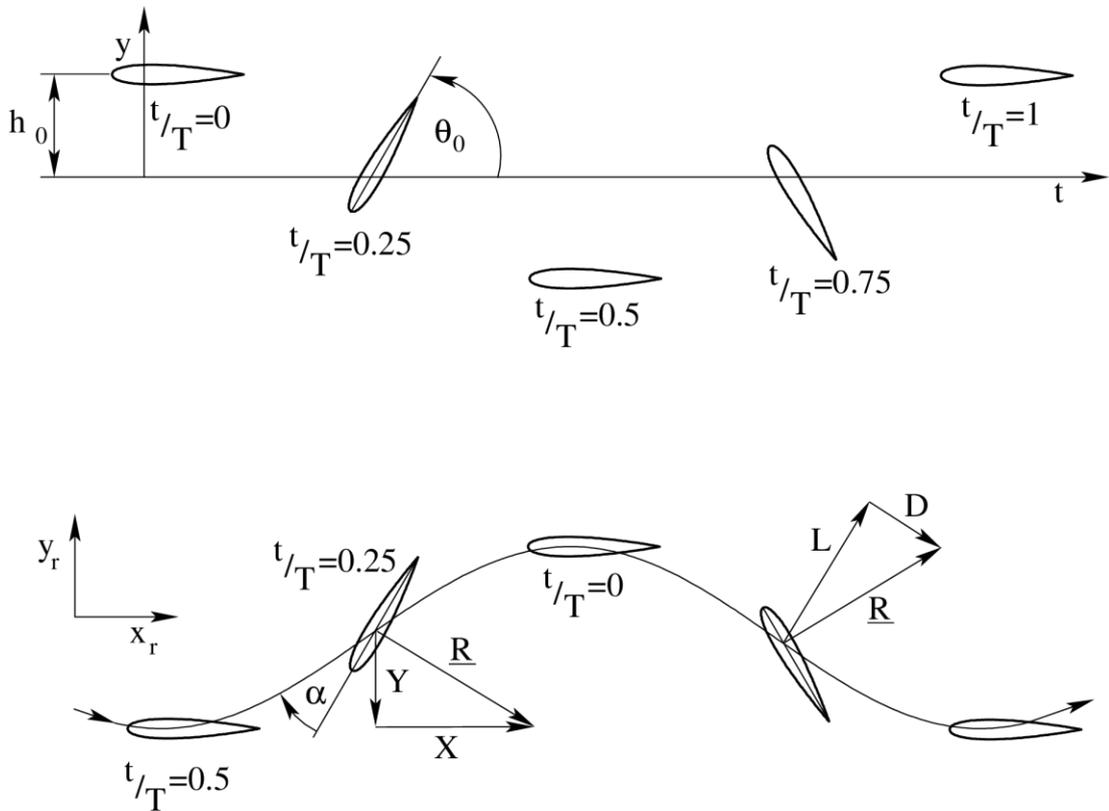


Figure 8.1: Top plot: prescribed motion of oscillating wing. **Bottom plot:** airfoil motion in reference system moving with free stream velocity for power-extracting mode.

The maximum values of a 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 [35], the power-extracting regime (in a mean sense, over one cycle) occurs when $\alpha(T/4) < 0$. This condition is represented in the bottom sketch of figure 8.1, which provides a time-sequence viewed in a reference frame moving

with the far-field flow at u_∞ , so that the effective AoA $\alpha(t)$ is made visible from the apparent trajectory of the airfoil. In this sketch, the resultant force \underline{R} is first 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 airfoil 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 aerodynamic force takes place. This is clearly the case since the airfoil does not move horizontally. The aerodynamic analyses of [35] and also those presented in the paper [39] highlight that the aerodynamic phenomena taking place during the wing oscillation are substantially more complex than the quasi-steady model discussed above. More specifically, the extent and efficiency of the energy extraction are heavily influenced by the occurrence of unsteady *leading edge vortex shedding* (LEVS) associated with dynamic stall and the LEVS timing with respect to the airfoil motion.

Taking a wing span of one unit length, 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 $M(t)$ is the resulting torque about the pitching centre p_x . Denoting by c the airfoil chord, and $C_P \equiv \frac{P}{(\frac{1}{2}\rho_\infty u_\infty^3 c)}$ a power coefficient, the non-dimensional power extracted over one cycle is given by:

$$\bar{C}_P = \bar{C}_{P_y} + \bar{C}_{P_\theta} = \frac{1}{T} \int_0^T \left[C_Y(t) \frac{v_y(t)}{u_\infty} + C_M(t) \frac{\Omega(t)c}{u_\infty} \right] dt \quad (8.1.5)$$

Where $C_Y(t) = Y(t)/(\frac{1}{2} \rho_\infty u_\infty^2 c)$ and $C_M(t) = M(t)/(\frac{1}{2} \rho_\infty u_\infty^2 c^2)$. The efficiency η of the power extraction is defined as the ratio of the extracted mean total power \bar{P} and the total available power P_a in the oncoming flow passing through the swept area (the flow window):

$$\eta = \frac{\bar{P}}{P_a} = \frac{\bar{P}_y + \bar{P}_\theta}{\frac{1}{2} \rho_\infty u_\infty^3 d} = \bar{C}_P \frac{c}{d} \quad (8.1.6)$$

where d is the overall vertical extent of the airfoil motion. This distance depends on the heaving and pitching motion parameters h_0 , θ_0 and φ , and, for the kinematic conditions considered in this report, takes values slightly larger than $2h_0$. The power extraction

efficiency η defined by equation (8.1.6) corresponds to the classical power coefficient obtained by means of Betz's analysis [187]. Such a theory is based on the 1D mass and linear momentum balance applied to a stationary inviscid streamtube surrounding a generic power-extraction device, and it shows that the upper limit of η is $\frac{16}{27} \cdot 100 \approx 59.3\%$. Therefore, equation (8.1.6) provides the relationship between the mean power coefficient \bar{C}_p defined by equation (8.1.5) and Betz's theory power coefficient (η).

8.1.2 Physical and numerical problem set-up

The wing section selected for this study is the NACA0015 airfoil. Here two operating conditions are considered: one characterized by a high efficiency of the energy extraction in the laminar flow regime considered in [36] (caseA), and the other characterized by a lower efficiency in the same laminar regime (caseB). In both case A and B, the heaving amplitude h_0 equals one chord and the pitching center is at $x_p = 1/3$. Case A is characterized by a pitching amplitude θ_0 of 76.33° and a non-dimensionalized frequency $f^* = fc/u_\infty$ of 0.14, where f is the frequency in Hertz. In case B, $\theta_0 = 60^\circ$ and $f^* = 0.18$. In both case A and B, $M_\infty = 0.1$. In the turbulent simulations of both operating conditions, the value of the Reynolds number based on the freestream velocity and the airfoil chord is $Re = 1.5 \cdot 10^6$; for the laminar simulations of both operating conditions reported in [36] and further analysed in the following subsection, $Re = 1100$.

The time-dependent turbulent flow fields past the oscillating wing have been computed using multi-block moving grids. In all simulations, the whole computational grid is animated simultaneously by a heaving and a pitching motion component defined by equation (8.1.1) and (8.1.2) respectively. The grid does not deform, and it undergoes a rigid-body motion corresponding to the prescribed motion of the wing. In order to assess the sensitivity of the turbulent solutions to the level of spatial refinement, the operating condition A has been simulated using three C-grids with different spatial resolution, namely a mesh with 102,400 cells (coarse), a mesh with 368,640 cells (medium) and one with 1,474,560 cells (fine). Four local views of the medium-refinement grid adopted for the analyses reported in the following subsection are provided in figure 8.2. The coarse and medium grids have been divided into 256 blocks, whereas the fine grid has been

divided into 1024 blocks and a parallel computation has been done to decrease the computational time. In all cases, the grid coordinates are non-dimensionalized by the chord of the airfoil, and the farfield boundary is at about 50 chords from the airfoil.

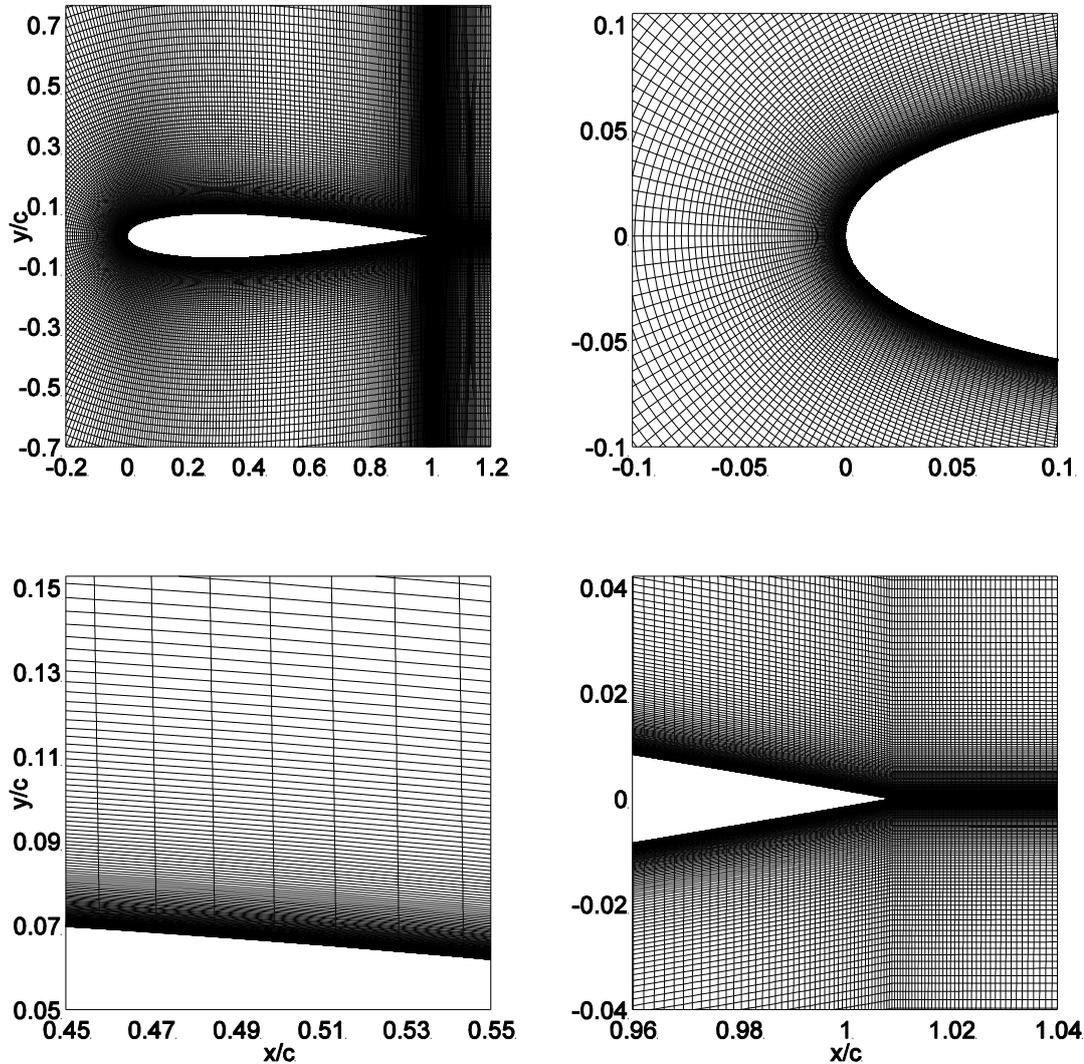


Figure 8.2: Medium-refinement grid for oscillating wing analyses. **Top left plot:** near-airfoil area. **Top right plot:** leading edge area. **Bottom left plot:** airfoil upper side area at midchord. **Bottom right plot:** trailing edge area.

The number of mesh intervals on the airfoil (n_A), the number of intervals on the C-cut (n_C), the number of intervals in the normal-like direction (n_N), and the distance of the first grid points off the airfoil from the airfoil surface (Δ_W) for the three grids are reported in Table 8.1.

Grid	n_A	n_C	n_N	Δ_W	N_{cell}
Coarse	440	100	160	3.2×10^{-6}	102,400
Medium	432	360	320	3.2×10^{-6}	368,640
Fine	864	720	640	1.6×10^{-6}	1,474,560

Table 8.1: Main geometric parameters of three grids adopted for assessing impact of spatial refinement on computed oscillating wing turbulent flow field.

In order to assess the sensitivity of the turbulent solutions to the level of temporal refinement, the operating condition A has also been simulated with the coarse grid using 128, 256, 512 and 1024 time-intervals per period. All turbulent simulations have been run until the maximum difference between C_Y over the last two oscillation cycles became less than 0.7% of the maximum C_Y over the last cycle. The number of oscillation cycles typically required to fulfil this requirement has varied between two and ten depending on the spatial and temporal refinement, and also on whether the simulation has been started from a freestream condition or restarted from the solution of a simulation using the same grid but different temporal refinement. The only exception is the fine grid simulation using 256 time-intervals per period, which has required the simulation of 12 oscillations cycles starting from a freestream condition to achieve the aforementioned periodicity error. It has been chosen to monitor the periodicity error of C_Y because the vertical force component gives the highest contribution to the extracted power. The periodicity error of the other force coefficients at a given period is similar to that of C_Y . The threshold of 0.7 has been chosen because it gives the best trade-off between accuracy and computational cost of the analyses. Reducing the C_Y periodicity error below this value yields insignificant variations of the periodic solutions with respect to those reported in the thesis. For all turbulent analyses of the oscillating wing presented in this thesis, y^+ has been found to be smaller than one at all grid points and all times of the periodic flow field. The periodic profiles of the horizontal force coefficient C_X , the vertical force coefficient C_Y and the pitching moment coefficient C_M computed with the coarse grid using the abovesaid four levels of time-refinement (set-ups 1 to 4) are reported in figure 8.3. The vertical axis of each subplot reports the force coefficient and the horizontal axis reports a time variable non-dimensionalized by the period.

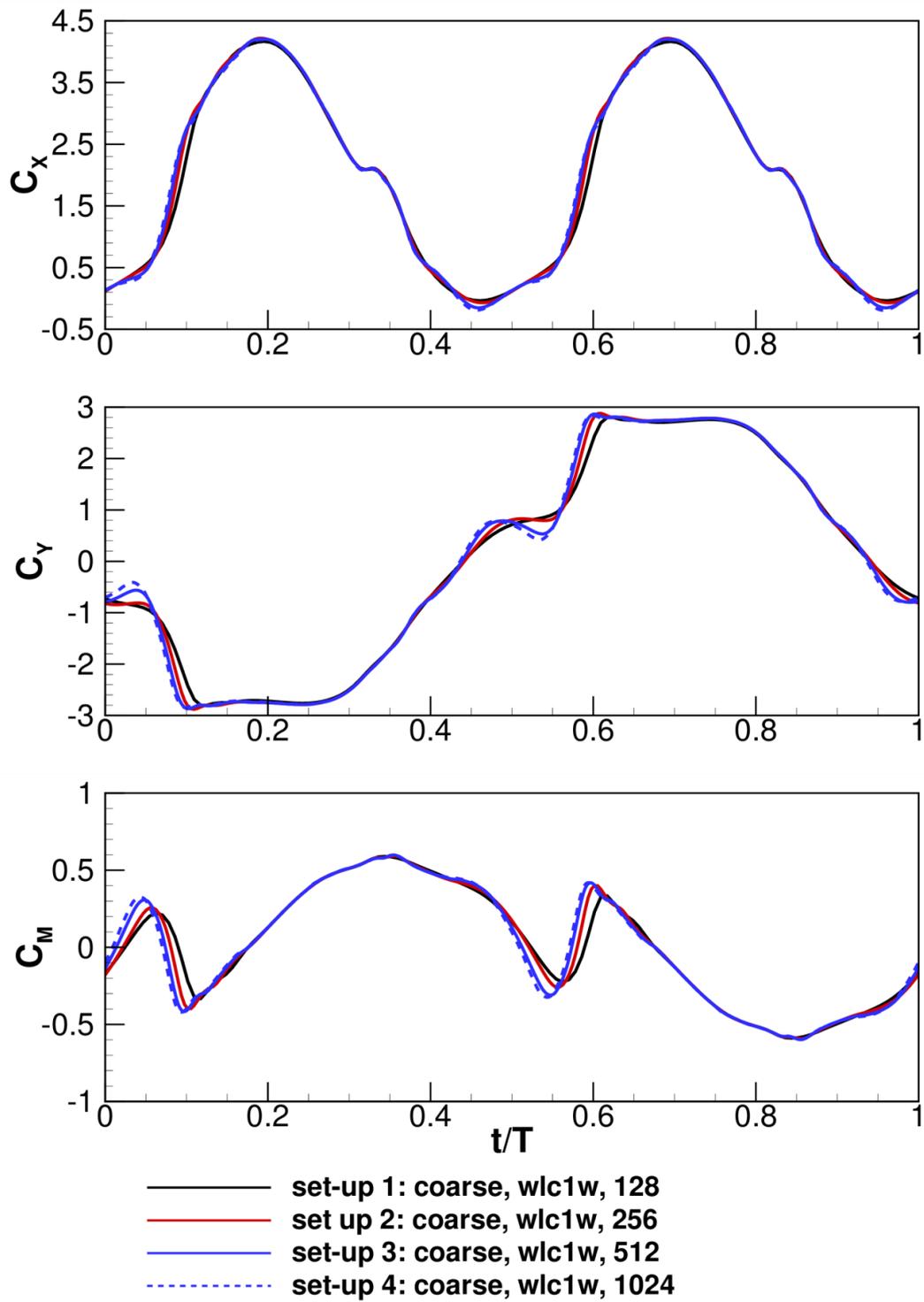


Figure 8.3: Comparison of force coefficients of oscillating wing (operating condition A), varying time-refinement. **Top plot:** horizontal force coefficient. **Middle plot:** vertical force coefficient. **Bottom plot:** pitching moment coefficient.

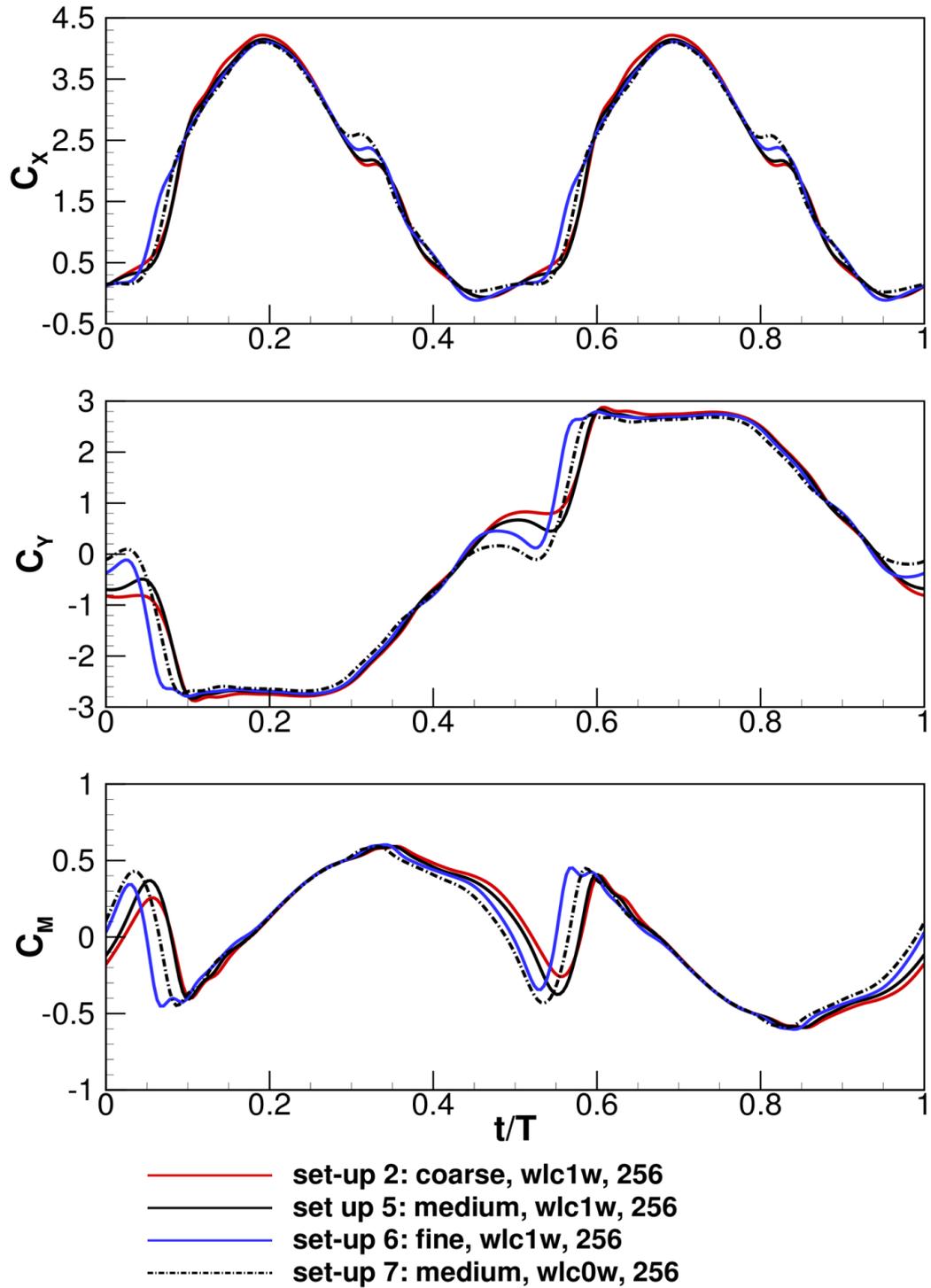


Figure 8.4: Comparison of force coefficients of oscillating wing (operating condition A), varying space-refinement and ω_w BC. **Top plot:** horizontal force coefficient. **Middle plot:** vertical force coefficient. **Bottom plot:** pitching moment coefficient.

	Set-up 1	Set-up 2	Set-up 3	Set-up 4	Set-up 5	Set-up 6	Set-up 7
\bar{C}_P	1.001	1.013	1.017	1.017	1.014	1.020	1.019
$\eta(\%)$	39.06	39.55	39.70	39.70	39.57	39.81	39.76

Table 8.2: Overall power coefficient and energy extraction efficiency of oscillating wing (operating condition A), varying space-refinement, time-refinement and $\omega_W BC$.

These profiles highlight that the computed solution is largely independent of the number of intervals per period provided that at least 512 time-intervals per period are used. Some small variations between the solutions using 256 and 512 intervals per oscillation cycle are observed. The periodic profiles of the three force coefficients computed using 256 time-intervals per period with the coarse, medium and fine grids (set-ups 2, 5 and 6 respectively) are reported in Figure 8.4. It is seen that some differences exist among all three solutions, indicating that the solution computed with the medium grid is not completely grid-independent and the solution computed with the fine grid may also not be fully grid-independent. It should be noted that the spatial resolution of the coarse and medium-refinement grids past the airfoil is very similar. The medium-refinement grid has many more cells than the coarse grid in the area behind the airfoil, and this choice has been made to better resolve the vortex propagation behind the wing. The medium-refinement and the fine grids, conversely, are topologically similar over the entire domain, which means that the medium-refinement grid has been obtained by removing every second grid line of the fine grid. The fact that the differences between the solutions of the coarse and medium-refinement grids are smaller than those observed using the medium-refinement and fine grids may indicate that the resolution of the vorticity field behind the wing does not have a strong effect on the value of the forces acting on the wing. A similar occurrence has been reported in computational aerodynamics studies of stalled horizontal axis wind turbine blades [23]. Figure 8.3 also reports the force coefficients determined with the medium-refinement grid using 256 intervals per period and Wilcox’s wall BC (2.6.1) with $S_R = 100$ (set-up 7). Comparing these profiles with all other results of Figure 8.3 and 8.4 reveals that the variations resulting from using different $\omega_W BC$ s with given temporal and spatial resolutions are larger than those resulting from varying the spatial and temporal resolutions with a given turbulent wall BC.

The values of the overall power coefficient averaged over the considered cycle \bar{C}_P and the power extraction efficiency η obtained with these seven analyses are reported in the first and second row of Table 8.2 respectively. These data indicate that:

- the difference of \bar{C}_p , one of the main integral functional of engineering interest, obtained with the coarse grid using 256 time-intervals per cycle (set-up 2) and 512 time-intervals per cycle (set-up 3) is only about 0.4%, and
- the difference of the same functional obtained using 256 time-intervals per cycle with the medium-refinement grid (set-up 5) and the fine grid (set-up 6) is only about 0.6%.

For these reasons, and also to keep computational costs within affordable limits, set-up 5 has been used for all turbulent analyses presented in the remainder of this study. This has been done also because, as shown below, the differences between the examined turbulent and laminar regimes are substantially larger than the solution variations highlighted in this subsection. Consequently, the use of an imperfect temporal and spatial grid-independent computational set-up for the turbulent analyses is not believed to affect significantly the conclusions of the following study.

The CFL number of all turbulent simulations has been set to 3, 2500 MG iterations per physical time-step have been performed, and CFL ramping has been used. All calculations have been performed using the PIRK MG iteration, since the maximum CFL number that could be used retaining the numerical stability of the FERK integration has been found to be 1.

As reported in [36], the laminar analyses of both operating conditions have been instead performed with 128 time-intervals per period and running the simulations for eight cycles of oscillation, starting from a freestream condition. This has resulted in the maximum difference between C_Y over the last two oscillation cycles being about 0.1% of the maximum C_Y over the last cycle.

8.1.3 Aerodynamic analysis

The evolution of the main kinematic parameters, the force and power coefficients of the oscillating wing for the operating condition *A* are analysed in figure 8.5, the top subplot of which reports the time-dependent values of the vertical position h of the wing, its angular position θ and the effective AoA a computed with equation (8.1.3). One notes that the maximum AoA achieved in case *A* is about 35° .

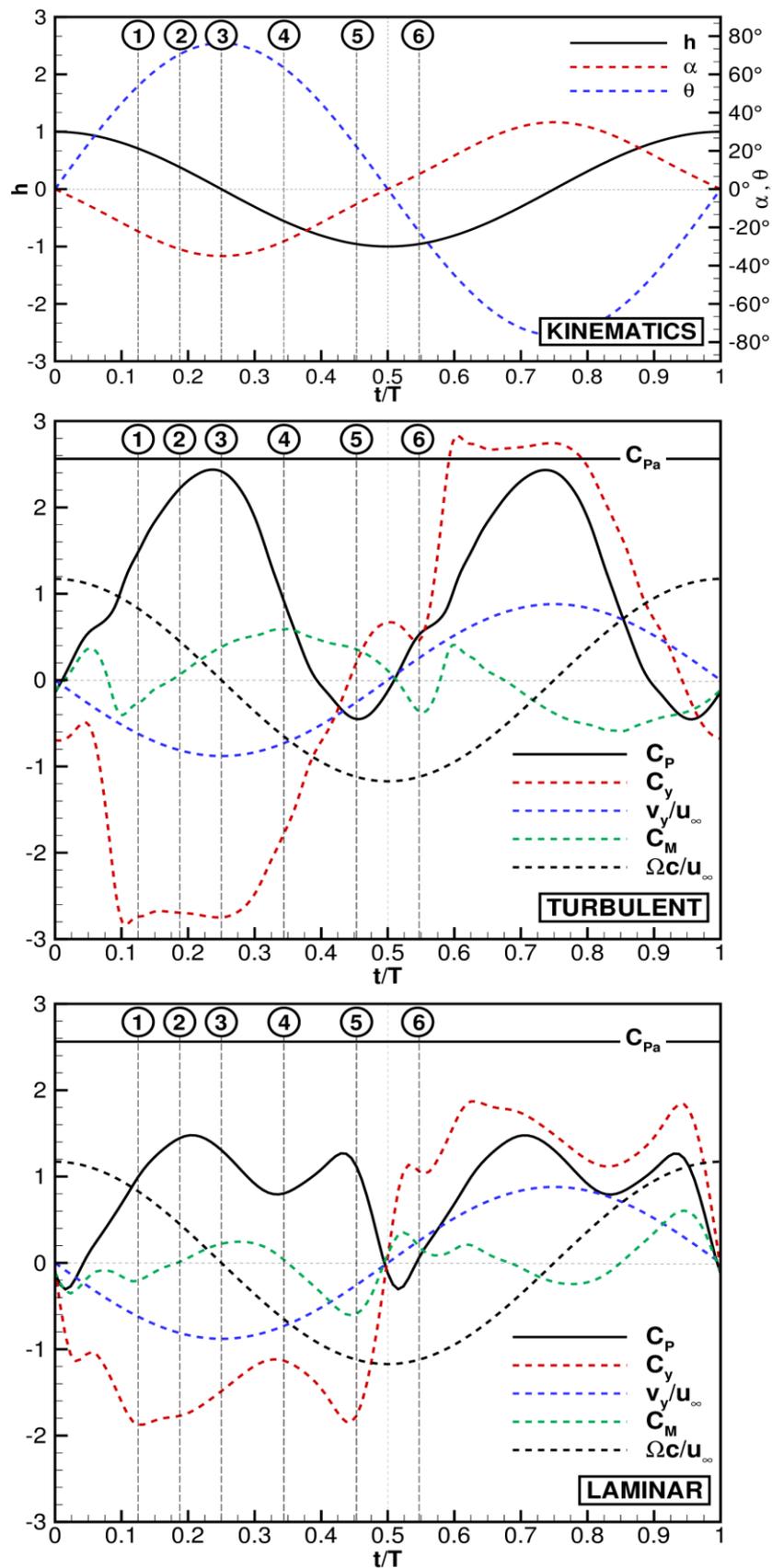


Figure 8.5: Analysis of oscillating wing with $\theta_0 = 76.33^\circ$ and $f^* = 0.14$ (case A). **Top plot:** kinematic parameters. **Middle plot:** turbulent regime dynamics. **Bottom plot:** laminar regime dynamics.

The middle and bottom subplots of figure 8.5 provide the main power and force coefficients over one period of oscillations in the turbulent and laminar regime respectively. There are two important differences between the curve of the overall power coefficient C_p in the turbulent and laminar regimes. One is that the laminar C_p curve over one semi-period has two positive peaks, one at 20.3% and the other at 43.0% of the period, whereas the turbulent C_p curve has only one pronounced peak at 23.4% of the period. The analyses reported in [35, 36] show that the high values of the laminar C_p after the first peak and the following secondary peak (that at 43.0% of the period) are due to the optimal synchronization of the wing motion and the LEVS associated with dynamic stall. This results in the heaving force and the heaving velocity being in phase over most of the cycle. As shown later in this section, however, in the turbulent regime the generation of the LE vortex starts later than in the laminar regime; the vortex travels close to the airfoil towards its TE with the same speed it has in the laminar regime, and therefore it leaves the TE region later than it does in the laminar regime. As a consequence, the optimal synchronization between wing motion and LEVS seen in the laminar regime is lost in the turbulent regime if the same kinematic conditions are used in both cases. The other difference between the turbulent and laminar C_p curves is that the peak value of the former (2.43) is significantly higher than the peak value of the latter (1.48). As shown later in this section, this is due to the fact that, before the LEVS starts, the turbulent boundary layer does not experience significant flow reversals, unlike that seen in the laminar regime, in which separation starts at the TE and travels to the LE until the laminar LEVS commences. As a consequence, in the phase preceding the turbulent LEVS, the turbulent lift is higher than its laminar counterpart, and the turbulent drag is lower than its laminar counterpart for a given wing position. This phenomenon explains why the turbulent heaving force coefficient C_y is significantly higher than its laminar counterpart until about 30% of the period, when the turbulent flow separation at the LE starts. At this point, due to the delayed start of the turbulent LEVS, the heaving force decreases much more rapidly than in the laminar case, leading to a reduction of the extracted heaving power. The middle and bottom subplots of figure 8.5 also report the non-dimensionalized heaving velocity v_y/u_∞ , and the comparison of these two subplots shows that, due to the aforementioned lack of synchronization between wing motion and turbulent LEVS, the laminar heaving power production remains positive over a longer portion of the period with respect to the turbulent case. The sign of the power contribution of the pitching moment to the overall power at each point of the cycle can be determined by comparing the sign of the non-dimensionalized angular velocity $\Omega c/u_\infty$ reported in the last two

subplots of figure 8.5 and that of the pitching moment coefficient C_M . The general shape of the C_M curve in the turbulent and laminar regimes is similar, but the turbulent C_M curve is shifted to the right with respect to its laminar counterpart, and the amplitude of the turbulent pitching moment is slightly larger than that of its laminar counterpart. As a result, the turbulent mean value of the pitching power \bar{C}_P turns out to be negative, whereas this parameter is positive in the laminar regime. The value of the power coefficient C_P associated with the available power in case *A* is also reported in the last two subplots of figure 8.5 for reference.

To examine in greater detail the aforementioned flow phenomena, the flow snapshots at the six positions denoted by the circled labels 1–6 in figure 8.5 are considered. These labels denote the wing configuration at 12.5%, 18.7%, 25.0%, 34.4%, 45.3% and 54.7% of the cycle respectively. The four left subplots of figure 8.6 depict the contours of the flow vorticity Ω_f and the streamlines when the wing is at positions 1–4 in the turbulent regime, whereas the four right subplots present the same analysis for the laminar regime. At position 1, the TE separation on the lower airfoil side starts in the laminar regime, whereas no TE separation is visible in the turbulent regime. Additionally, the vorticity contours close to the lower side of the airfoil highlight that the laminar boundary layer is substantially thicker than the turbulent boundary layer, due to Reynolds number effects. At position 2, the laminar separation has already passed the mid-chord position of the airfoil, whereas only a small TE separation has now appeared in the turbulent regime. At position 3, the laminar separation has reached the LE and the LEVS begins, whereas the turbulent TE separation has slightly increased and the turbulent boundary layer on the lower side has thickened. At position 4, the strong laminar vortex has already achieved the mid-chord position, whereas the turbulent LEVS has just started. These four snapshots confirm that the turbulent boundary layer of the high-Reynolds number regime is substantially thinner and less prone to separation than its laminar counterpart in the low Reynolds number regime, and this explains the higher levels of extracted power before the turbulent LEVS starts. The evolution of the turbulent and laminar LEVS is compared in figure 8.7, where the four left subplots report the Ω_f contours when the wing is at positions 3 to 6 in the turbulent regime, whereas the four right subplots present the same variable for the laminar regime. It is noted that, in the laminar case, the LEVS starts at position 3 and the main vortex leaves the TE region at position 5, whereas, in the turbulent case, the LEVS starts at position 4 and the main vortex leaves the TE region at position 6.

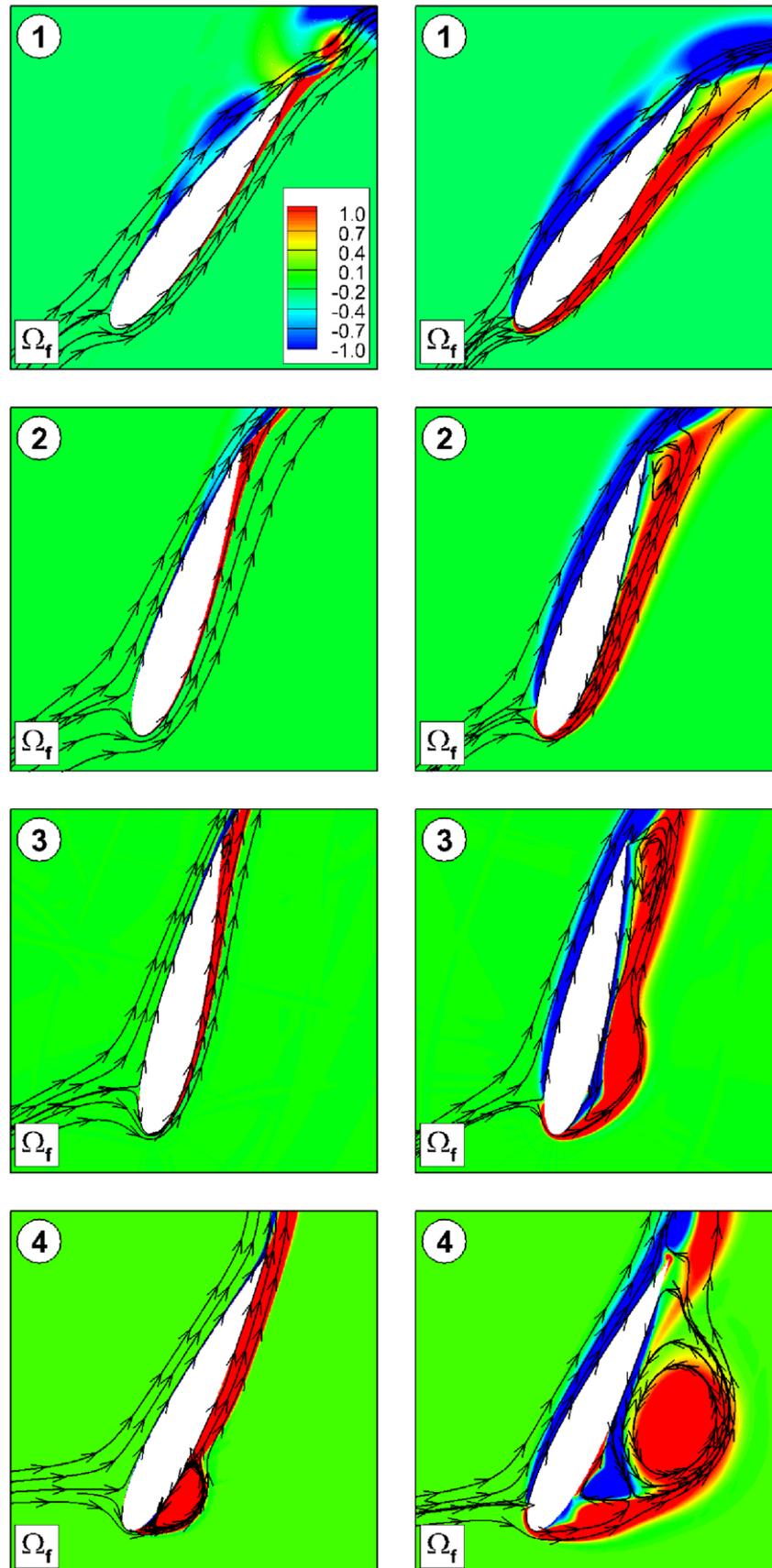


Figure 8.6: Streamlines and vorticity contours for wing positions labelled 1–4 in figure. 8.5. **Left plots:** turbulent regime. **Right plots:** laminar regime.

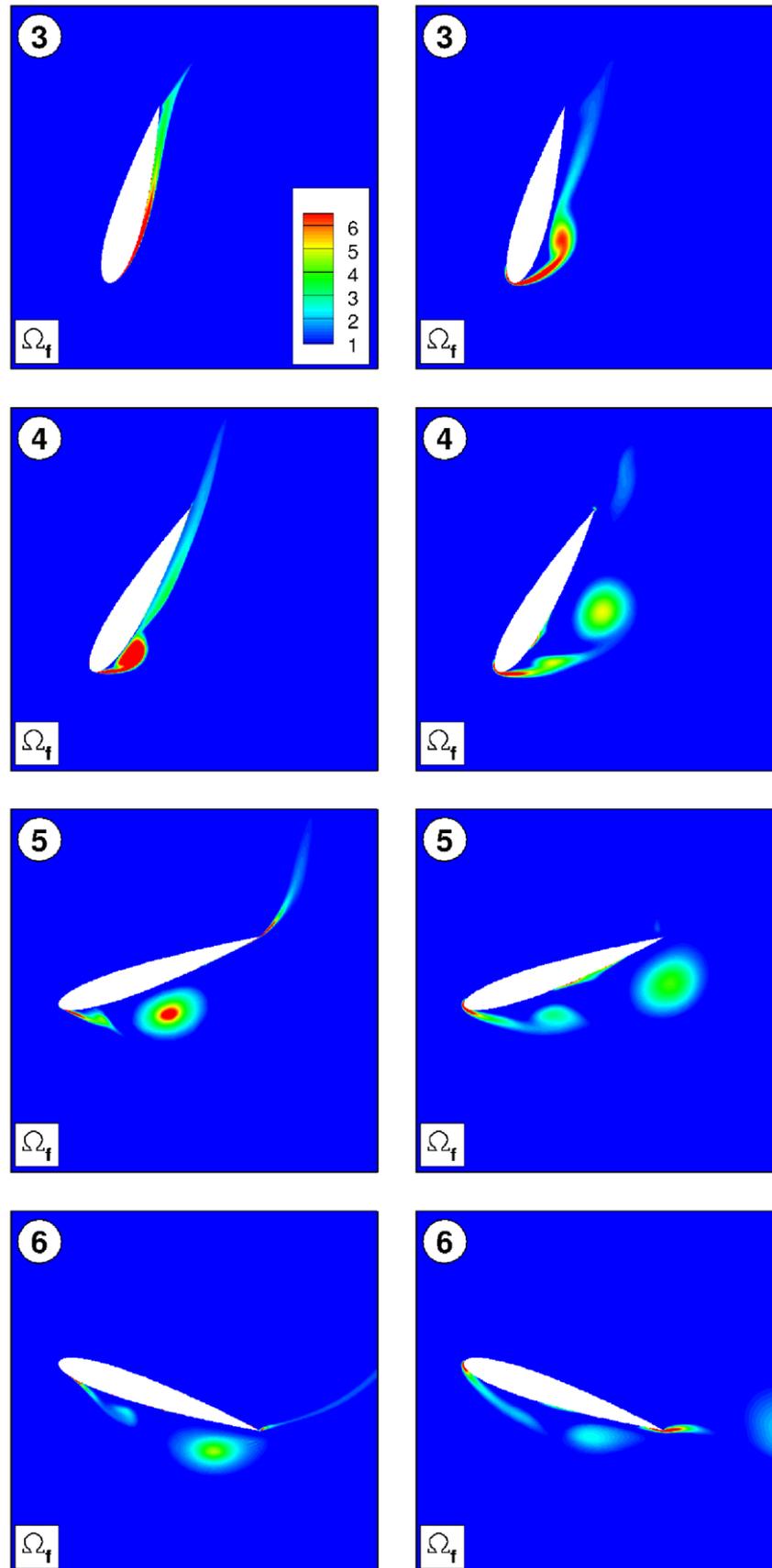


Figure 8.7: Vorticity contours for wing positions labelled 3–6 in figure. 8.5. **Left plots:** turbulent regime. **Right plots:** laminar regime.

Given that the temporal distance between positions 3 and 5, and that between positions 4 and 6 is the same, the streamwise velocity of the main vortex is the same in the laminar and turbulent regimes. These observations highlight the loss of optimal timing between LEVS and wing motion occurring in the turbulent regime caused by the delayed onset of LEVS. On the other hand, comparing the vorticity levels of the main turbulent vortex (position 5) and those of the main laminar vortex (position 4), it is noted that the former vortex is stronger, and this has a beneficial effect on the heaving power production. In addition, by comparing the skin friction coefficient between the laminar and turbulent regime in all six positions, one sees exactly when the flow is separated. These images can be found in *Appendix J*.

The evolution of the main kinematic parameters, the force and power coefficients of the oscillating wing for the operating condition B are analysed in figure 8.8. The time-dependent values of h , θ and α are reported in the top subplot of figure 8.8, which highlights that the maximum AoA achieved in case B is about 11.5° . The analyses reported in [35, 36] show that no LEVS exists in the laminar regime for the operating condition B . The turbulent analyses reported in the present study reveal that also in the turbulent regime there is no LEVS for this operating condition. The middle and bottom subplots of figure 8.8 provide the main power and force coefficients over one period of oscillations in the turbulent and laminar regime respectively.

The shape of the turbulent and laminar C_p curves is very similar, pointing to the fact that in this operating regime the main aerodynamic features are the same in both flow regimes. This is confirmed by the fact that the same observation is also made for the heaving force and pitching moment coefficients. As a consequence, the phase between the heaving velocity and force, and that between the angular velocity and pitching moment are also very close in both flow regimes. The main difference between the turbulent and the laminar regimes of the operating condition B is that the amplitude of all turbulent force and power coefficients is larger than the amplitude of the laminar curves. This difference is due to the reduction of the heaving force and pitching moment caused, in the laminar regime only, by thicker boundary layers and flow separation in the portion of the cycle where the AoA increases.

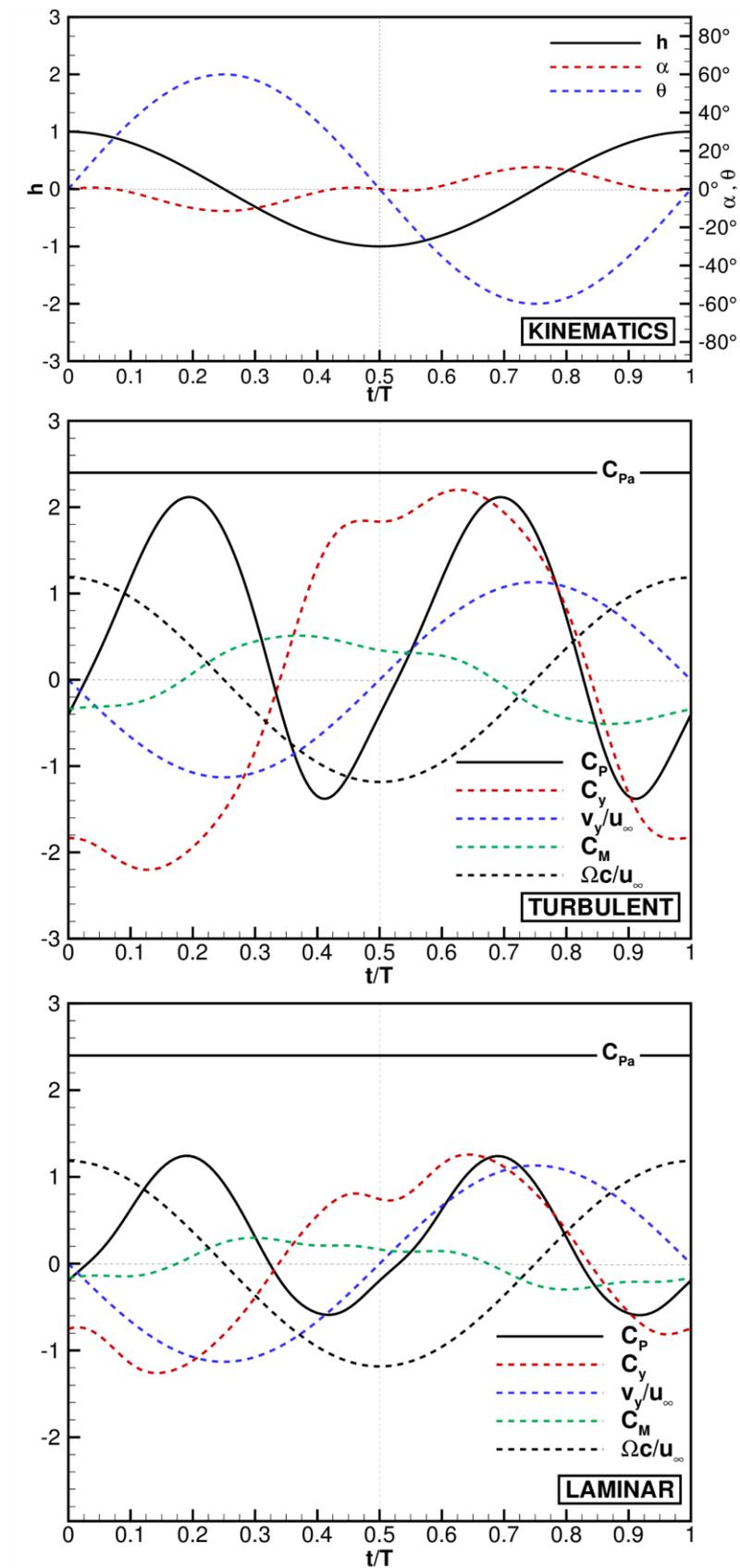


Figure 8.8: Analysis of oscillating wing with $\theta_0 = 60^\circ$ and $f^* = 0.18$ (case B). **Top plot:** kinematic parameters. **Middle plot:** turbulent regime dynamics. **Bottom plot:** laminar regime dynamics.

The mean values of the overall power coefficient \bar{C}_P , the heaving power coefficient \bar{C}_{P_y} , and the pitching power coefficient \bar{C}_{P_θ} , and the efficiency η of the power extraction process computed by means of equation (8.1.6) for cases *A* and *B* in the turbulent and laminar regimes are reported in table 8.3. In case *A* the efficiency in the turbulent condition is nearly 40%, about 6 percentage points more than the laminar analysis predicts. This difference is caused by the significantly higher values of the turbulent C_{P_y} . As discussed above, this occurrence is due to the fact that the increase of the turbulent C_Y due to a thinner and more stable boundary layer outweighs the reduction of heaving power caused by the loss of synchronization between heaving motion and LEVS. Note also that the turbulent \bar{C}_{P_θ} is negative, whereas the laminar \bar{C}_{P_θ} is positive. This is a consequence of the loss of synchronization between pitching motion and LEVS. In the operating regime *B*, the turbulent efficiency is nearly 19%, about 7 percentage points more than the laminar analysis predicts. No LEVS is present in either flow regime for this operating condition, and this efficiency difference is entirely due to the thinner turbulent boundary layer and the absence of flow separation in the turbulent regime.

	Turbulent				Laminar			
	\bar{C}_P	\bar{C}_{P_y}	\bar{C}_{P_θ}	η (%)	\bar{C}_P	\bar{C}_{P_y}	\bar{C}_{P_θ}	η (%)
<i>A</i>	1.014	1.166	-0.152	39.6	0.862	0.824	0.038	33.6
<i>B</i>	0.450	0.707	-0.257	18.8	0.274	0.397	-0.123	11.4

Table 8.3: Power coefficient and power extraction efficiency in turbulent and laminar regimes for operating conditions *A* and *B*.

As for the convergence of the turbulent simulations, the choice of the numerical control parameters reported in the preceding subsection has resulted in all force coefficients achieving full convergence at all physical times of the wing motion, and the RMS of the residuals of the mean flow and the ω equations dropping by at least four orders of magnitude. The residual of the K equation has instead dropped by between 1.5 and 2 orders in all cases. This is highlighted in the convergence histories reported in the six subplots of figure 8.9, which refer to the six wing positions labelled 1–6 in figure 8.5.

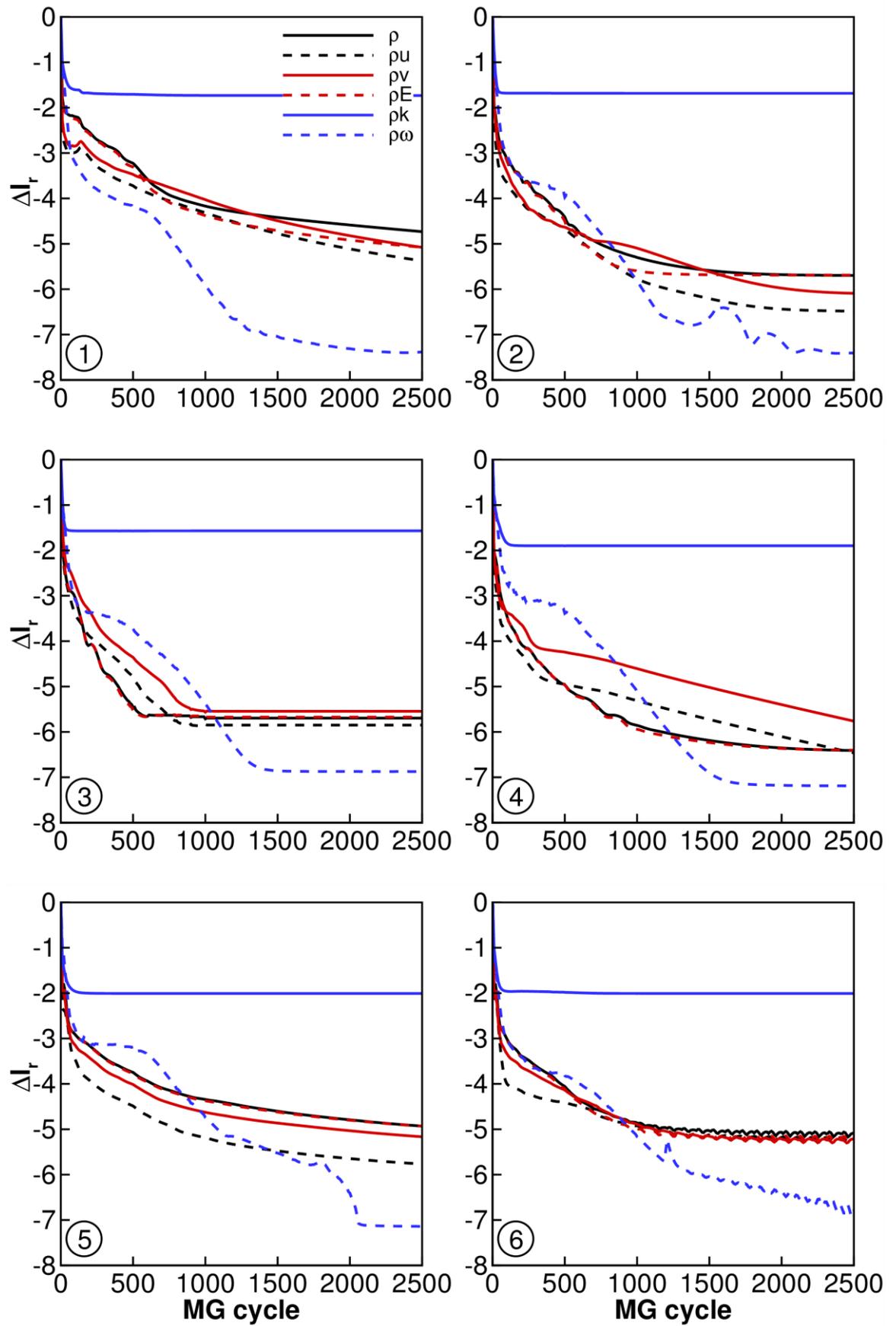


Figure 8.9: Convergence histories of turbulent simulations of oscillating wing for wing positions labelled 1–6 in figure 8.5.

Similar residual drops have also been observed solving test cases of the same type of the oscillating wing using a commercial package with numerical methods and modelling features similar to those of COSA.

Due to the complex turbulent aerodynamics of the oscillating wing, the use of *Large Eddy Simulation* (LES) may yield more accurate predictions of the flow field. However, it is hard to foresee the level of improvement in the prediction of the time dependent force and power components. Furthermore, LES simulations for the considered high Reynolds number would require extremely large computational resources. The turbulent analyses presented above provide a significant contribution to the aerodynamic knowledge base of this device based on the relatively cheaper URANS technology. These URANS data will be a valuable source for cross comparisons with detailed high-Reynolds number measurements, which are likely to be available before LES data.

8.2 Oscillating transonic wings

This section focuses on the 2D turbulent analysis of a pitching airfoil in a transonic regime, and is subdivided into five subsections. The first subsection presents the fundamental theory of the oscillating wings. The second subsection presents the physical and computational set-up of the flow analyses, along with the assessment of the temporal refinement used in the time-domain analyses of this test case. The third subsection investigates in detail the turbulent unsteady flow regime of the transonic pitching motion by comparing the numerical results obtained from the TD solver developed in this study with both the numerical results obtained from a commercial CFD code, and available experimental data. The fourth subsection presents results obtained from the Harmonic Balance solver developed in this study, while the last subsection presents the analysis of the solution run-time speed-up obtained by using the HB solver than the standard TD solver.

8.2.1 Fundamentals of oscillating wings

Here the case of a transonic oscillating airfoil is considered. This type of problem is representative of transonic aircraft wing aerodynamics. The oscillating motion is taken to feature only a pitching motion component. Viewing the problem in the absolute frame of reference, where the freestream direction is constant, the pitching motion of the airfoil results in a harmonic variation of the angle of attack. The mathematical representation of the imposed motion is that adopted in [51]. Taking a pitching axis located on the chord line at position x_p from the LE, the airfoil motion is expressed as:

$$a(t) = a_m + a_0 \sin(\omega t) \quad (8.3.1)$$

where a represents the instantaneous angle of attack, a_m is the mean angle of attack (i.e. the constant direction of the freestream in the absolute frame of reference), a_0 is the amplitude of pitching, and ω is the angular frequency of the motion. The frequency ω is related to the reduced frequency λ by the equation

$$\lambda = \frac{\omega c}{2 u_\infty} \quad (8.3.1)$$

where u_∞ is the freestream velocity and c is the airfoil chord. The prescribed pitching motion is depicted in figure 8.10.

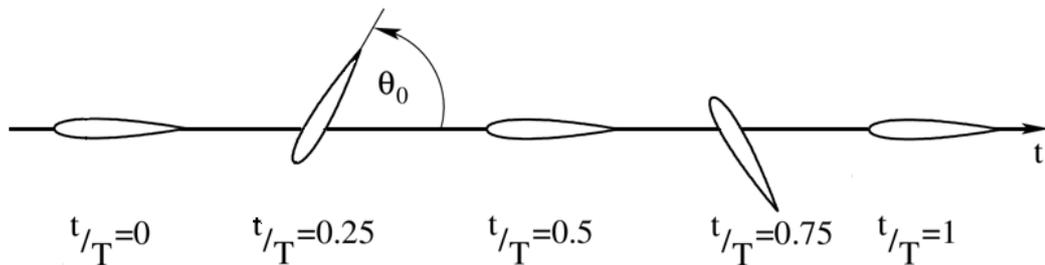


Figure 8.10: prescribed motion of pitching wing

8.2.2 Physical and numerical problem set-up

The wing section selected for this study is the NACA0012 airfoil and the operating condition is characterized by a significantly nonlinear periodic flow due to the harmonic pitching motion. This test case, known as AGARD CT5 is described in reference [51]. In this periodic flow field, a shock forms at the LE of the airfoil and moves towards the TE until it reaches approximately 45% of the chord at a maximum value of angle of attack. Then the shock travels upstream to the LE until it disappears. This cycle is repeated on the opposite side of the airfoil with a phase of about 180 degrees. This is because the AoA is nearly zero and the airfoil is symmetric. It should also be noted that the mean steady flow corresponding to the constant α_m is shock-free. The flow remains attached throughout the cycle of unsteadiness. This pitching motion is characterized by pitching amplitude a_0 of 2.51° with mean angle of attack α_m of 0.016° and a reduced frequency λ of 0.0814. The pitching centre is located at $x_p = 0.25c$, and the free stream Mach number M_∞ is 0.755. The flow conditions are given in table 8.4. The value of the Reynolds number based on the freestream velocity and the airfoil chord is $Re = 5.5 \cdot 10^6$. A similar analysis has been done by Batina [188] and more recently by Da Ronch et al [53] for solving the Euler equations using the same operating conditions. This motion was chosen because there are available experimental data [51]. The experiment was conducted by the Advisory Group for Aerospace Research & Development in France, where the static pressure was measured for at least 30 locations on the surface of the airfoil, for several time intervals.

parameter	value
M	0.755
α_m	0.016°
a_0	2.51°
λ	0.0814
Re	$5.5 \cdot 10^6$

Table 8.4: Flow conditions for the NACA0012 airfoil

The time-dependent turbulent flow field past the pitching airfoil has been computed using multi-block moving grids. In all simulations, the whole computational grid is animated by pitching motion defined by equation (8.3.1). The grid does not deform, and it undergoes a rigid-body motion corresponding to the prescribed motion of the airfoil. The flow field has been simulated using a C-grid with 32.768 cells. The grid coordinates are non-dimensionalized by the chord of the airfoil, and the farfield boundary is at about 50 chords from the airfoil. The grid has been divided into 32 blocks and a parallel computation has been performed to decrease the computational time. Four local views of the grid adopted for the analysis reported in the following subsection are provided in figure 8.11.

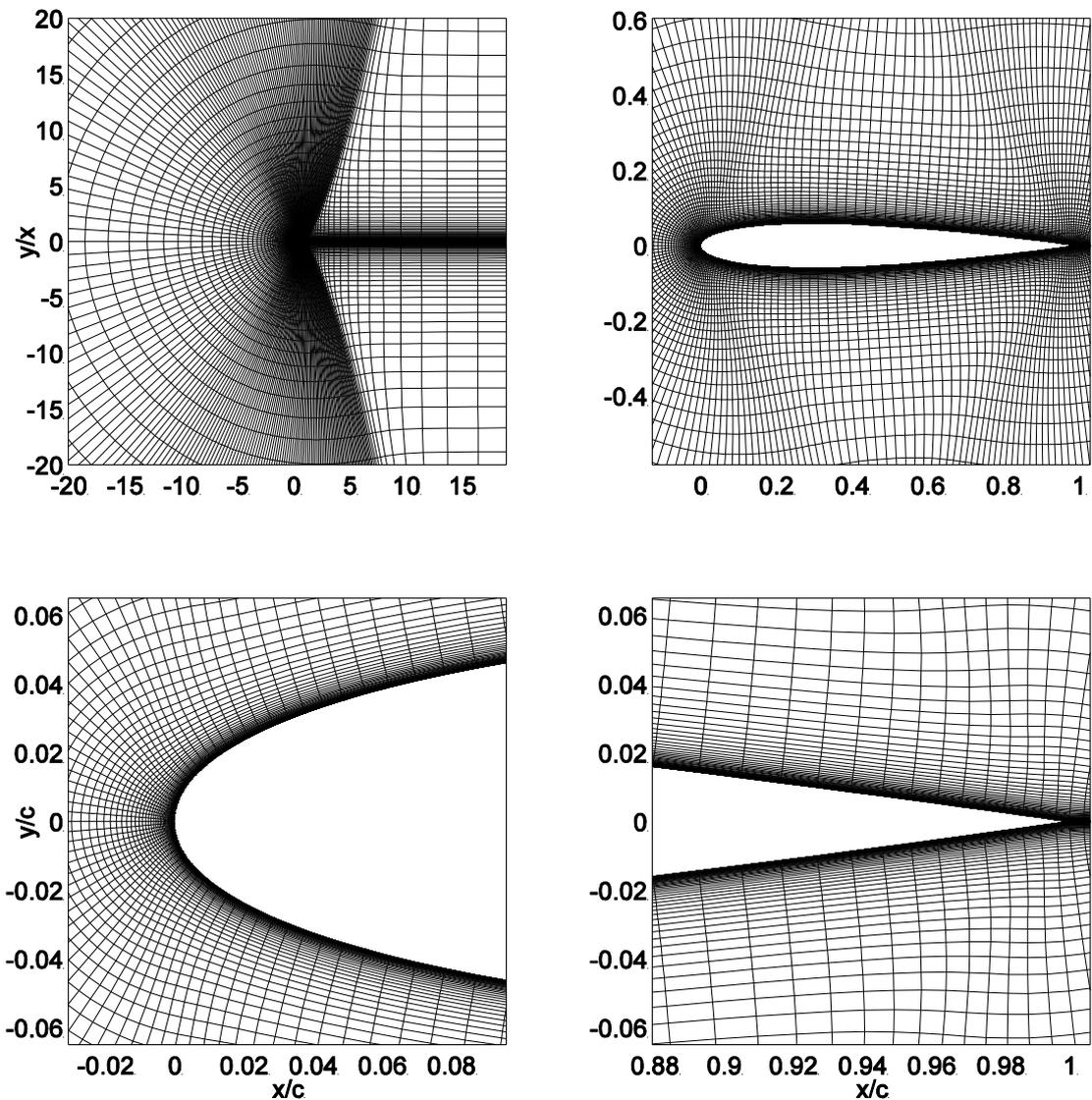


Figure 8.11: grid for the pitching NACA0012 airfoil. **Top left plot:** near-airfoil area. **Top right plot:** leading edge area. **Bottom left plot:** airfoil upper side area at mid-chord. **Bottom right plot:** trailing edge area.

The number of mesh intervals on the airfoil (n_A), the number of intervals on the C-cut (n_C), the number of intervals in the normal-like direction (n_N), and the distance of the first grid points off the airfoil from the airfoil surface (Δ_W) are reported in Table 8.5.

Grid	n_A	n_C	n_N	Δ_W	N_{cell}
Fine	176	40	128	$1.25 \times 10^{-6}c$	32,768

Table 8.5: Main geometric parameters of the grid used to compute the pitching airfoil.

In order to assess the sensitivity of the turbulent solutions to the level of temporal refinement using the TD solver, the flow field has been simulated using 32, 64, and 128 time-intervals per period. All turbulent simulations have been run until the maximum difference between C_L over the last two pitching cycles became less than 0.1% of the maximum C_L over the last cycle. The number of pitching cycles typically required for fulfilling this requirement varied between three and five depending on the temporal refinement, and also on whether the simulation has been started from a freestream condition or restarted from the solution of a simulation using different temporal refinement. The periodicity error of the other force coefficients at a given period is similar to that of C_L . The threshold of 0.1 has been chosen because it gives the best trade-off between accuracy and computational cost of the analyses. Reducing the C_L periodicity error below this value yields insignificant variations of the periodic solutions with respect to those reported in the figure 8.12. For all turbulent analyses of the pitching airfoil presented in this report, y^+ has been found to be smaller than one at all grid points and all times of the periodic flow field. The periodic profiles of the lift coefficient C_L , the drag coefficient C_D and the pitching moment coefficient C_M computed with above said three levels of time-refinement are reported in figure 8.12.

The vertical axis of each subplot reports the force coefficient and the horizontal axis reports a time variable non-dimensionalized by the period. These profiles highlight that the computed solution is almost independent of the number of intervals per period used to calculate the flow field. Some small variations between the solutions using 32 and 64 intervals per pitching cycle are instead observed. In order to keep computational cost low, the case with 64 time intervals per period has been used for all turbulent analyses presented in the remainder of this study.

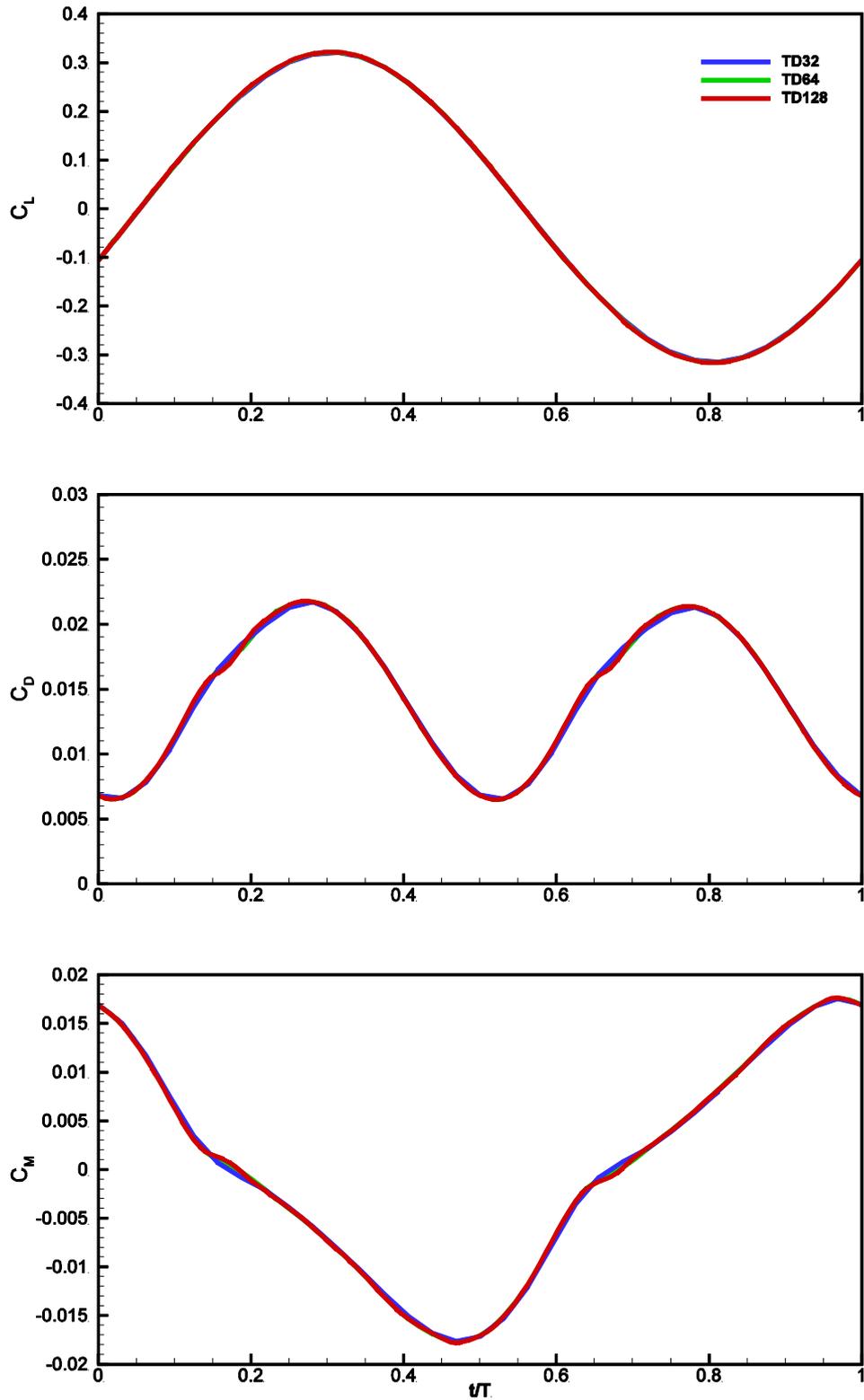


Figure 8.12: Comparison of force coefficients of oscillating wing NACA0012, varying time-refinement. **Top plot:** lift force coefficient. **Middle plot:** drag force coefficient. **Bottom plot:** pitching moment coefficient.

However, in the case of oscillating wing, which is experiencing simultaneous pitching and heaving motions (section 8.2), a spatial refinement analysis took place. In CT5 test case it is no longer necessary to find a grid independent solution as the purpose of this study is to validate the accuracy and the speed up of the harmonic balance solver. In addition, the CFL number of all turbulent simulations has been set to 2 and CFL ramping has been used. 2500 MG iterations per physical time-step have been performed and the scheme of multigrid was chosen to be the V-cycle with 2 smoothed Runge–Kutta cycles on the coarsest grid level and 5 Runge-Kutta cycles on the medium and fine grids. All calculations have been performed using the PIRK MG iteration, since the maximum CFL number that could be used retaining the numerical stability of the FERK integration has been found to be 0.5.

8.2.3 Time-domain analysis

This subsection presents unsteady time accurate numerical results obtained with the TD solver described in section 5.2. Figures 8.13 and 8.14 illustrate the comparison of the hysteresis cycle of the lift and pitching moment coefficient respectively, between numerical results obtained with the TD solver, and experimental data. The horizontal axis of both figures 8.13 and 8.14 represents the instantaneous angle of attack while the vertical axis represents the lift and pitching moment coefficient, respectively. Two different numerical results have been simulated using the TD solver developed in this study. The first one is computed using the turbulent TD solver with Menter’s wall boundary condition described by equation (2.6.7) and labelled in figures as “mentw TD05”, and the second one is computed using the same solver but with Wilcox’s wall boundary condition described by equation (2.6.1) with $S_R = 2500$ and labelled in figures as “wlc1w TD05”. The TD05 symbol indicates that the solver used to compute these results is the TD solver and the period presented in figures 8.13 and 8.14 is equal to 5. Also, results obtained from commercial software NUMECA [186] and experimental data [51] have been included in the figure for validation, labelled as “numeca” and “exp” respectively.

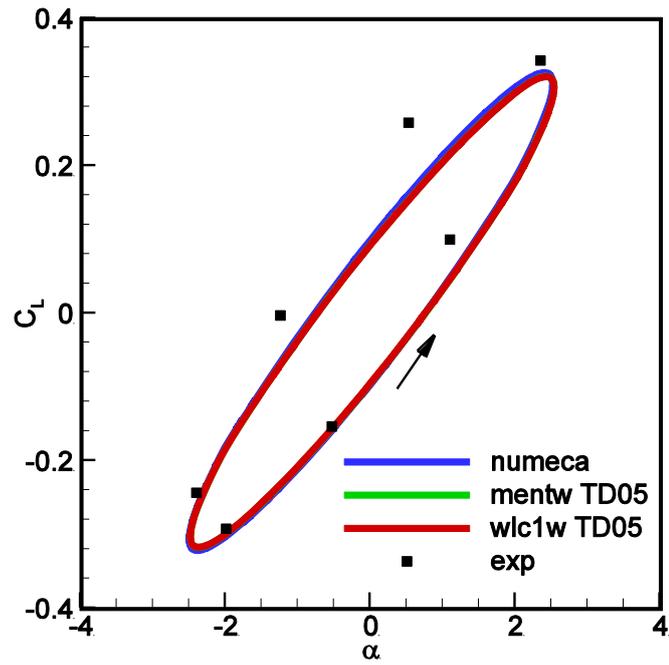


Figure 8.13: hysteresis cycle of lift coefficient for the pitching NACA0012 airfoil

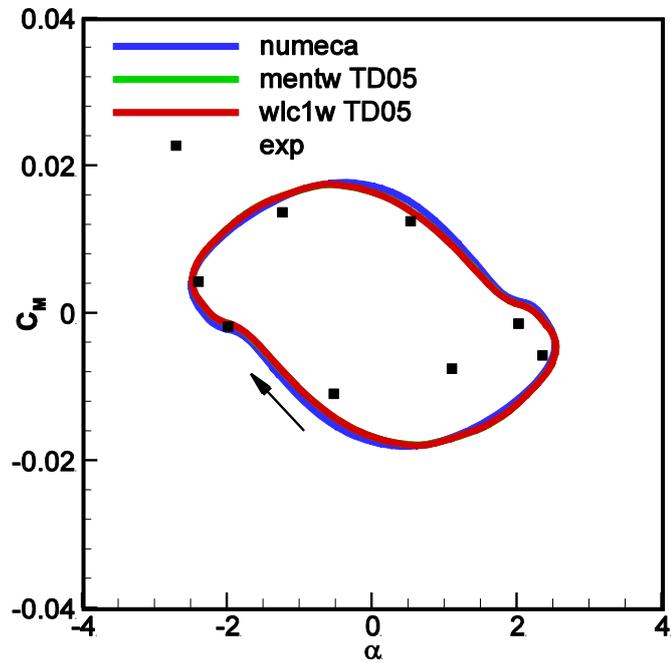


Figure 8.14: hysteresis cycle of pitching moment coefficient for the pitching NACA0012 airfoil

The compared results between simulations obtained with the TD solver developed in this study and experimental data are in very good agreement during the upstroke. The source of the discrepancies between the data during the downstroke is unknown, however this is consistent with other studies [53, 189, 190]. In addition the results computed using the two different wall boundary conditions for the calculation of the dissipation rate ω of the SST turbulence model give zero difference for the prediction of lift and pitching moment coefficient. Numeca results are also in very good agreement with the numerical results obtained with COSA TD solver. Only some small variations appear in the pitching moment coefficient compared to COSA results but it is negligible difference. The arrow indicates the direction of the lift coefficient with respect to the time evolution.

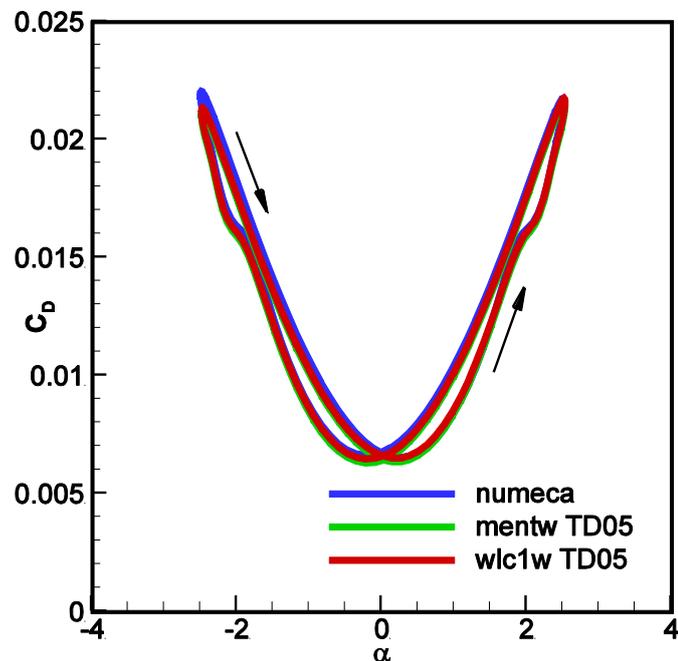


Figure 8.15: hysteresis cycle of drag coefficient for the pitching NACA0012 airfoil

Figure 8.15 illustrates the comparison of hysteresis cycle of the drag coefficient, between numerical results simulated using the COSA TD solver and results obtained with a commercial CFD code [186]. The horizontal axis of figure 8.15 represents the instantaneous angle of attack while the vertical axis represents the drag force coefficient. Similar to figures 8.13 and 8.14, two different numerical results have been simulated by the COSA TD solver where the first one uses Menter's wall boundary condition described by equation (2.6.7) and the second one uses Wilcox's wall boundary condition described by equation (2.6.1) with $S_R = 2500$. The compared results between the

simulations obtained with TD solver developed in this study and the results obtained with commercial software NUMECA are in very good agreement. In addition the results computed using the two different wall boundary conditions for the calculation of the dissipation rate ω of the SST turbulence model give almost zero difference for the prediction of drag coefficient.

To examine in greater detail the shock that appears in the flow field, the pressure coefficient C_p at six positions along one period is considered (figure 8.16). Each subplot of figure 8.16 is labelled at its bottom right hand side with the value of instantaneous angle of attack of the airfoil. The arrow beside each value indicates if the angle of attack is increasing or decreasing. The positions of the airfoil were chosen based on the available experimental data and their instantaneous angle of attack is 1.09° , 2.34° and -0.54° when the AoA is increasing and 0.52° , -1.25° and -2.41° when the AoA decreasing, respectively. The horizontal axis of figure 8.16 represents the x-coordinates of the airfoil nondimensionalized by the chord while the vertical axis represents the pressure coefficient C_p . As discussed above, two different numerical results have been simulated with the COSA TD solver, one is computed using Menter's wall boundary condition, and the second one is computed using Wilcox's wall boundary condition. Also, results obtained with commercial code NUMECA [186] and experimental data [51] have been included to the figure 8.16 for validation purposes. The numerical results obtained from the COSA TD solver are in very good agreement with tunnel measurements with some minor deviations around the shock wave. Furthermore these results are almost in excellent agreement with those obtain from the commercial software NUMECA. Also, one sees that the different set up for the calculation of the dissipation rate ω on the wall has zero influence on the pressure coefficient C_p at all positions along the period.

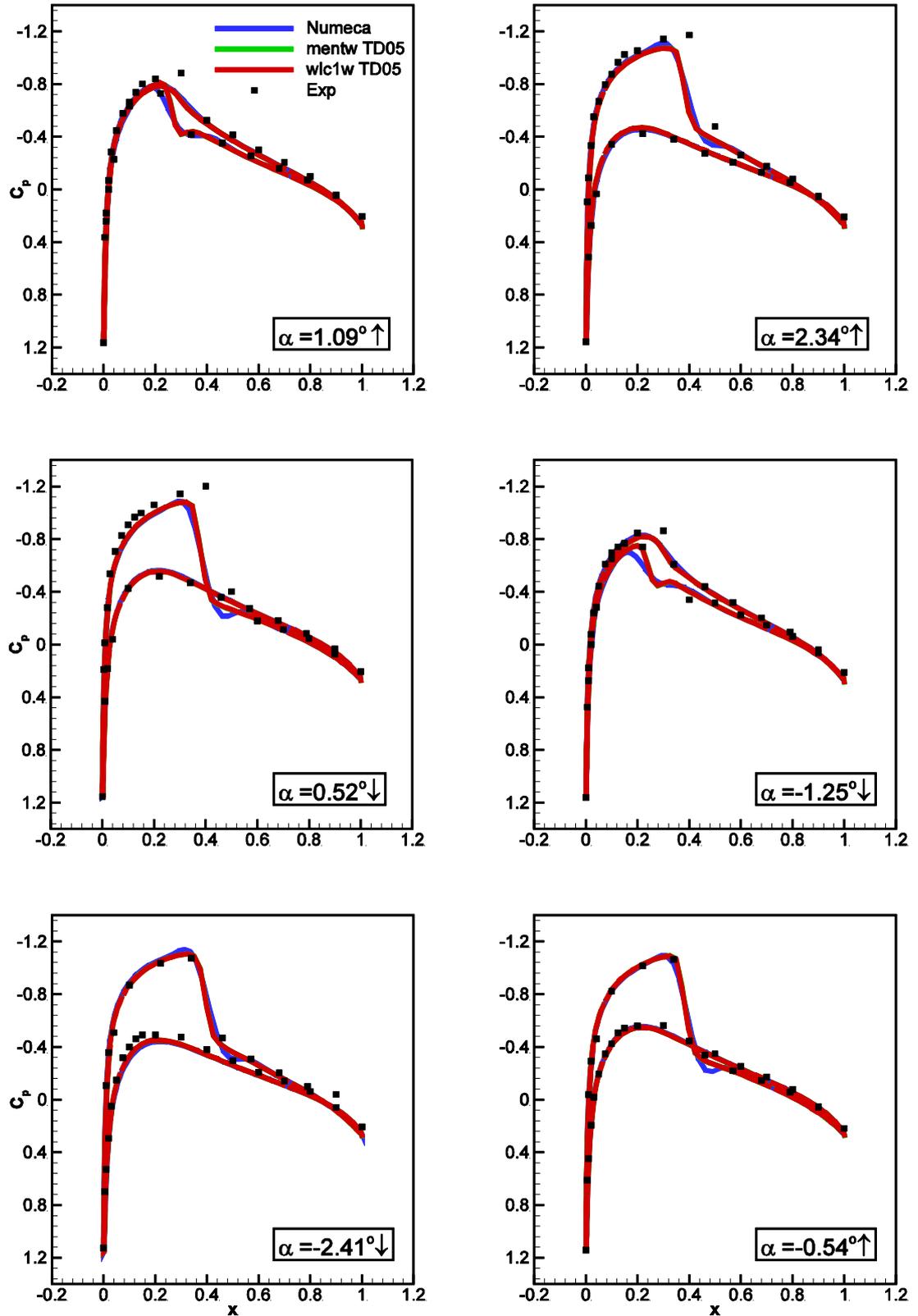


Figure 8.16: numerical results of instantaneous pressure coefficient at several positions along one period compared with experimental data

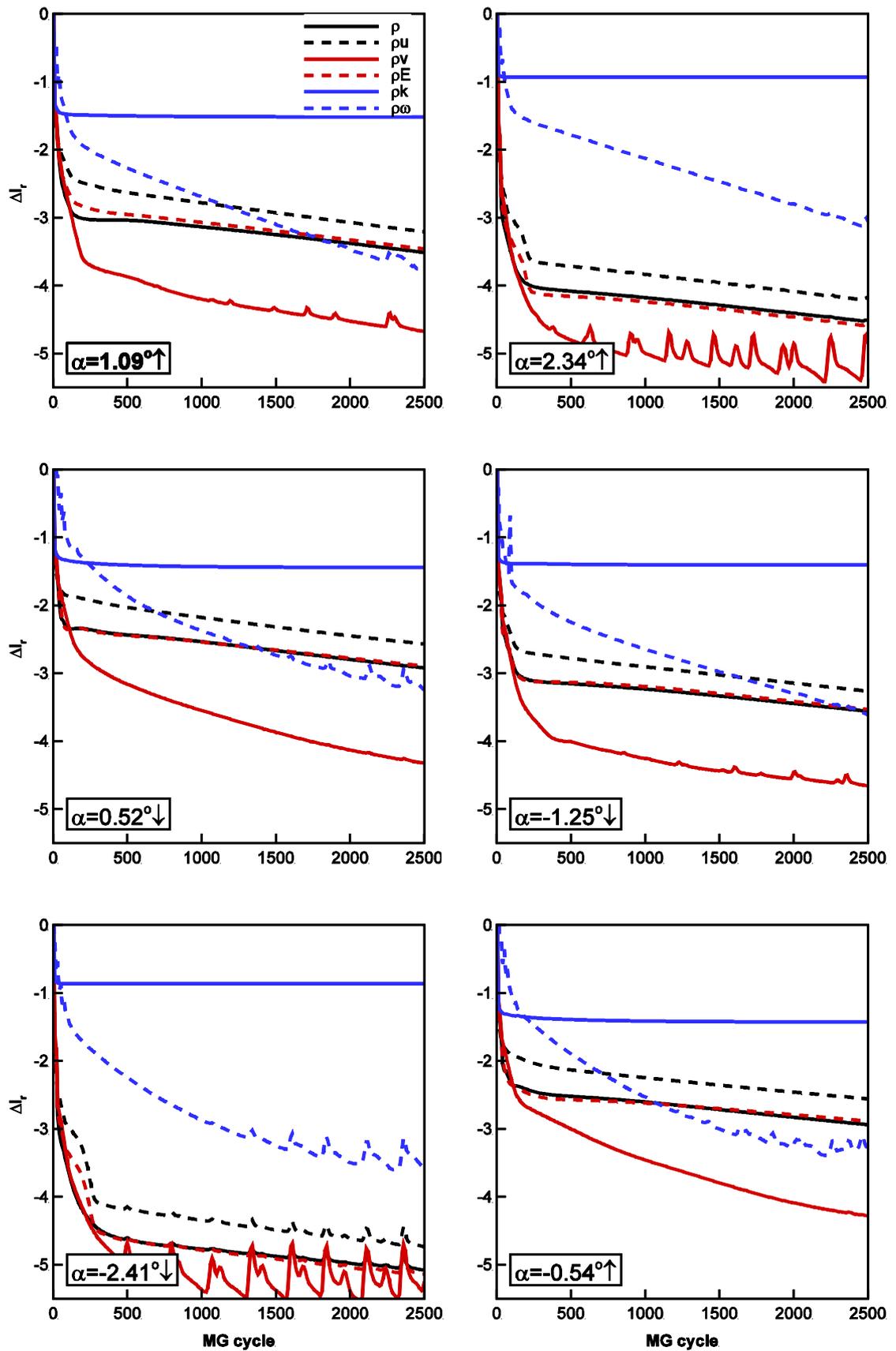


Figure 8.17: Convergence histories of turbulent SST analyses of flow field past the pitching NACA0012 airfoil using Menter's wall boundary condition.

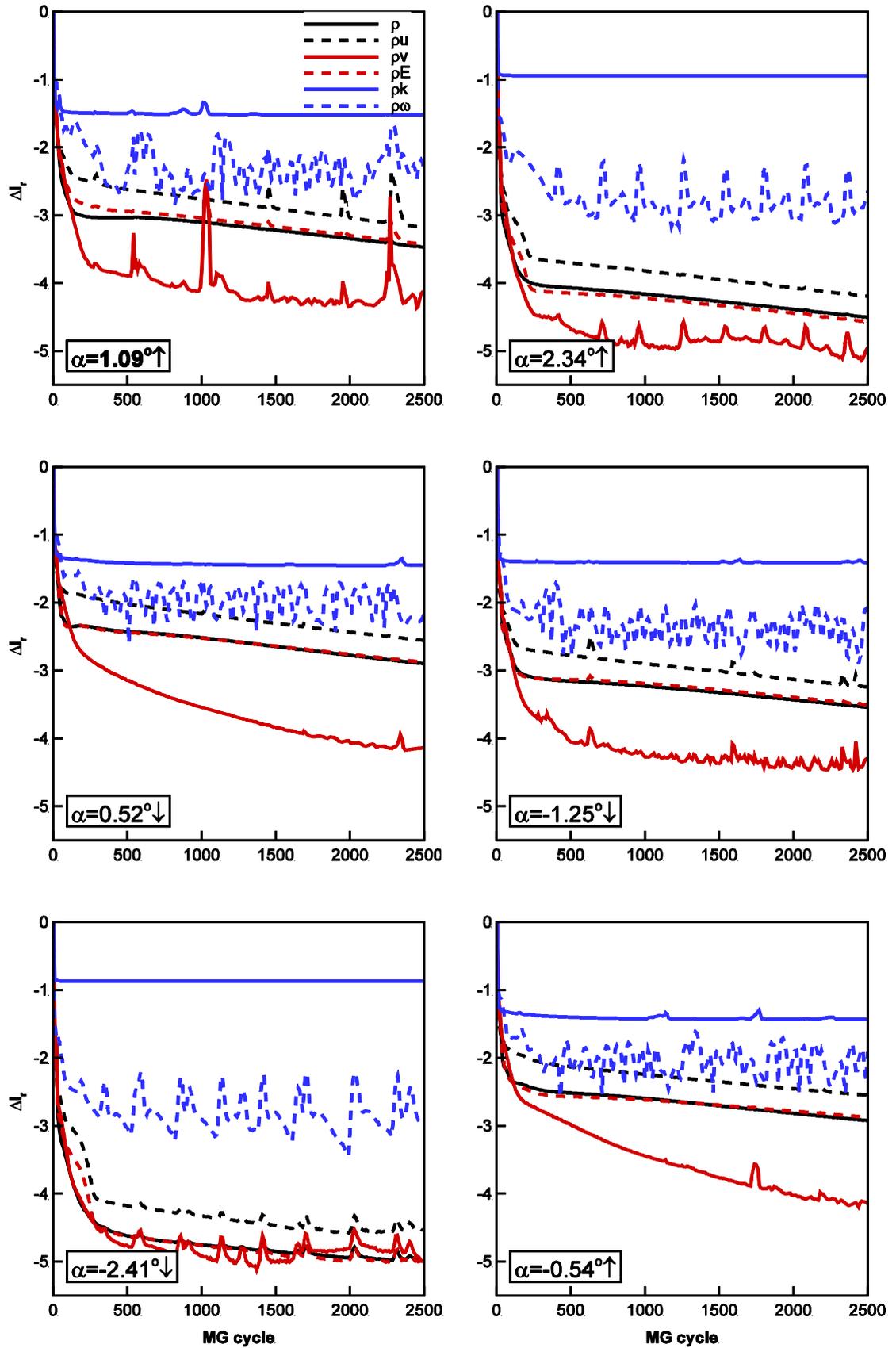


Figure 8.18: Convergence histories of turbulent SST analyses of flow field past the pitching NACA0012 airfoil using Wilcox's boundary condition with $S_R = 2500$

The convergence histories of the four RANS PDEs and the two PDE's associated with the turbulence model using equation (2.6.7) in the six positions mentioned above, are reported in the subplots of figure 8.17, whereas the six convergence histories using equation (2.6.1) with $S_R = 2500$ are shown in the subplots of figure 8.18. All calculations have been performed using the multigrid algorithm where a high order scheme for the restriction operator and a bilinear interpolation for the prolongation operator have been used. In figure 8.17 one sees that the convergence rate of all residuals in all positions has been dropped by at least 2.5 orders of magnitude. However the residual of the K equation has been dropped between 1 and 1.5 orders in all positions as expected. Similar residual drops have also been observed, solving the same test case with commercial package NUMECA, by applying numerical methods and modelling features similar to those of COSA. In the case where the Wilcox wall BC has been used, the convergence rate in all residuals at all positions is fairly worse than those computed with Menter's wall BC. For the same reasons discussed in the chapter 7, the convergence histories of the simulation using Wilcox's wall BC are expected to improve further by increasing the spatial refinement of the computational grid.

8.2.4 Frequency-domain analysis

In the previous subsection a TD analysis applying 64 intervals per period took place using a turbulent flow field past a pitching NACA0012 airfoil. As discussed in chapter 3, an alternative method to solve the RANS equations and the turbulence model equations for a periodic flow problem is the High Dimensional Harmonic Balance Method. In this subsection this HDHB method is used to solve the RANS and SST equations for the same pitching motion. In order to compare with great detail the solution obtained from the turbulent harmonic balance solver and the one obtained from the turbulent TD solver, described in section 8.3.3, harmonic balance simulations with a number of complex harmonics ranging from 1 to 11 have been performed. Figures 8.19 and 8.20 present respectively the comparison of hysteresis cycle of the lift and pitching moment coefficient, computed with the TD simulation and the HB analyses. The numerical results selected to validate the frequency-domain solver are those obtained from the TD solver using Wilcox's wall boundary condition described by equation (2.6.1) with $S_R = 2500$

and labelled in figures 8.19 and 8.20 as “TD05”. As mentioned in subsection 8.2.3, 5 periods have been simulated in order to achieve the aforementioned periodicity error using the TD solver. The horizontal axis of both figures 8.19 and 8.20 represents the instantaneous angle of attack while the vertical axis represents the lift and pitching moment coefficient, respectively. Each figure has 7 different lines represented by different colours. The first 6 lines denote the solution of the force coefficient of lift (figure 8.19) and pitching moment (figure 8.20), respectively, using different numbers of complex harmonics whereas the last line (black) denotes the solution obtained from the TD solver, analysed in the previous subsection.

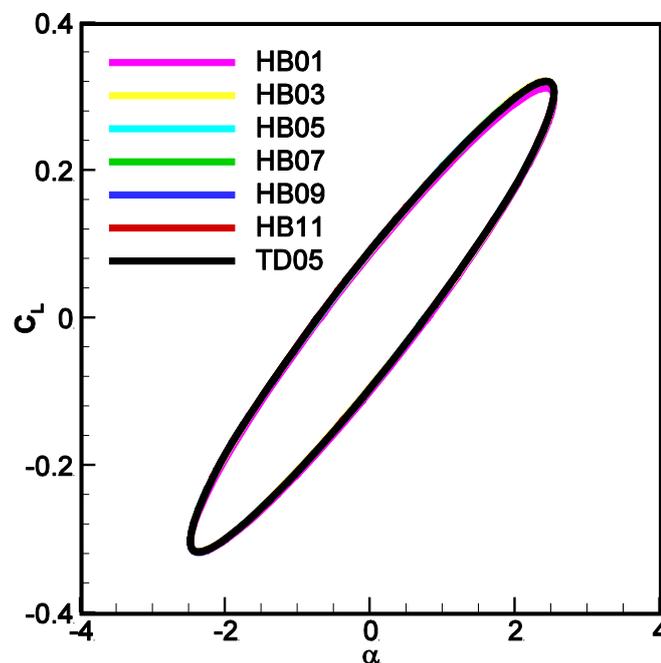


Figure 8.19: hysteresis cycle of lift coefficient for the pitching NACA0012 airfoil using 11 harmonics

The compared results between the frequency domain and the time domain solver are in very good agreement. As one sees from figure 8.19, the number of complex harmonics that one needs to calculate the lift coefficient of a transonic flow field past the NACA0012 airfoil and make it consistent with the TD solution, is greater than one. Using only one complex harmonic, brings the solution very close to TD solution but it is not fully identical, thus more harmonics are needed. In figure 8.20 one sees that the number of complex harmonics one needs to calculate the pitching moment coefficient of a transonic flow field past the NACA0012 airfoil and make it consistent with the TD solution, is

greater or equal to three. As shown in the same figure, the solution using only one complex harmonic fails to predict correctly the moment coefficients. For both lift and pitching moment coefficient, one observes that increasing the number of harmonics has almost zero effect on the force solutions.

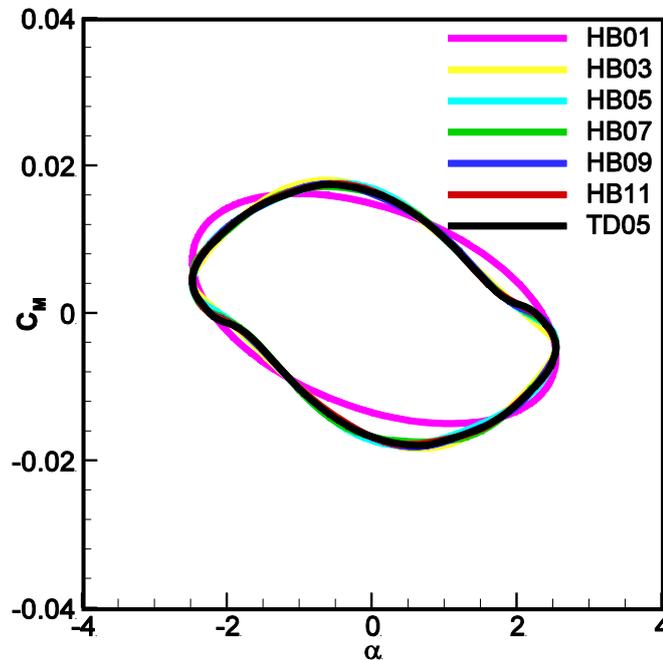


Figure 8.20: hysteresis cycle of pitching moment coefficient for the pitching NACA0012 airfoil using 11 harmonics

Figure 8.21 illustrates the comparison of the hysteresis cycle of the drag coefficient, obtained from the TD analysis and the harmonic balance analyses. The horizontal axis of figure 8.21 reports the instantaneous angle of attack while the vertical axis reports the drag force coefficient. Similarly to the case of figures 8.19 and 8.20, only the harmonic balance solution using Wilcox wall BC described by equation (2.6.1) with $S_R = 2500$ has been performed. The 7 lines that appear in figure 8.21 have the same structure as in figures 8.19 and 8.20. The compared results of the drag coefficient between the frequency domain and the time domain solver are in very good agreement. Similarly to the results of the pitching moment coefficient, the solution using only one complex harmonic fails to predict correctly the drag coefficient. However, the solution obtained with three or five complex harmonics is still not sufficient to predict the drag coefficient correctly and be identical with the results of TD solver. At least 7 harmonics are needed for the drag

coefficient. Increasing the number of harmonics by 7 or more, has almost zero effect on the drag coefficient. Figures 8.19, 8.20 and 8.21 illustrate that the higher the number of harmonics, the more accurate the predictions of the force coefficients are.

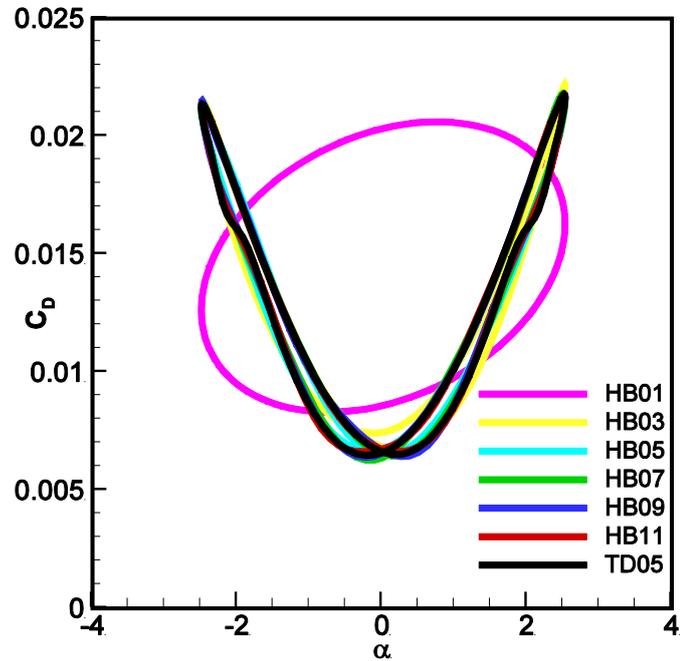


Figure 8.21: *hysteresis cycle of drag coefficient for the pitching NACA0012 airfoil using 11 harmonics*

In order to get further insight on the performance of the frequency domain solver, results of the zeroth flow complex harmonic are presented. The zeroth harmonic represents the average values through a period of unsteadiness. Using these coefficients one can calculate the average value of the surface pressure coefficient on each cell through a period, known as the mean pressure coefficient. The mean surface pressure coefficient is presented in figure 8.22.

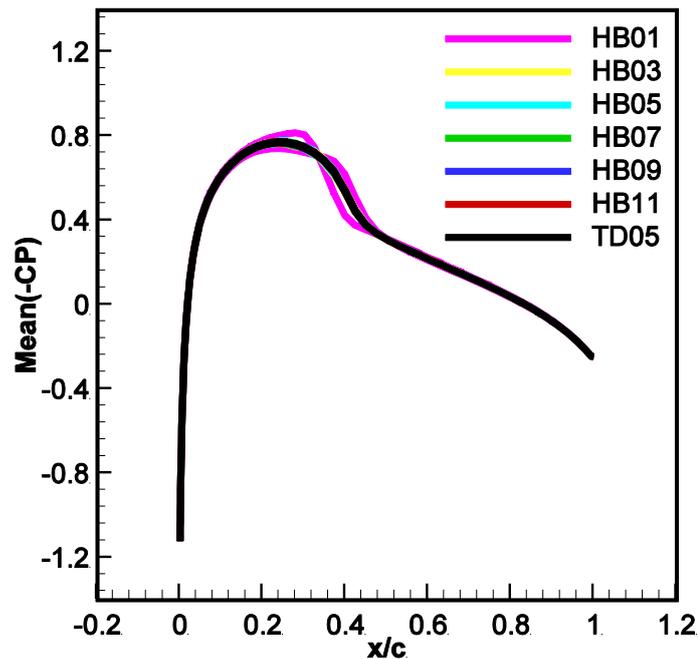


Figure 8.22: zeroth surface pressure coefficients

The horizontal axis of figure 8.22 reports the position along the airfoil chord, and the vertical axis reports the zeroth harmonic unsteady surface pressure coefficients. Similarly to figure 8.21, each line of figure 8.22 denotes the solution of the average surface pressure coefficient using different numbers of complex harmonics. The black line denotes the solution of average surface pressure coefficients obtained from the TD solver, where each pressure coefficient is defined as the sum of all pressure coefficients of a particular cell calculated at all physical times steps divided by the total number of physical time steps of one period. One sees that the results of zeroth surface pressure coefficient, calculated by using only one harmonic do not predict correctly the mean pressure coefficient compared to the TD results. Differences are mostly obvious between 10% and 45% of the airfoil chord because of the shock near the LE of the airfoil. This effect applies to both sides of the airfoil. On the other hand the results of zeroth surface pressure coefficient, calculated using three or more harmonics are in very good agreement with the mean pressure coefficient computed with the TD solver.

In order to get further insight on the performance of the frequency domain solver, results from the real and imaginary parts of the first flow harmonic are presented. Similarly to plot 8.22 the horizontal axis of figures 8.23 and 8.24 represents the non-dimensionalized coordinates of the airfoil and the vertical axis represents the real and the

imaginary part of the first harmonic surface pressure coefficient, respectively. The lines of figures 8.23 and 8.24 have the same structure as in figure 8.22.

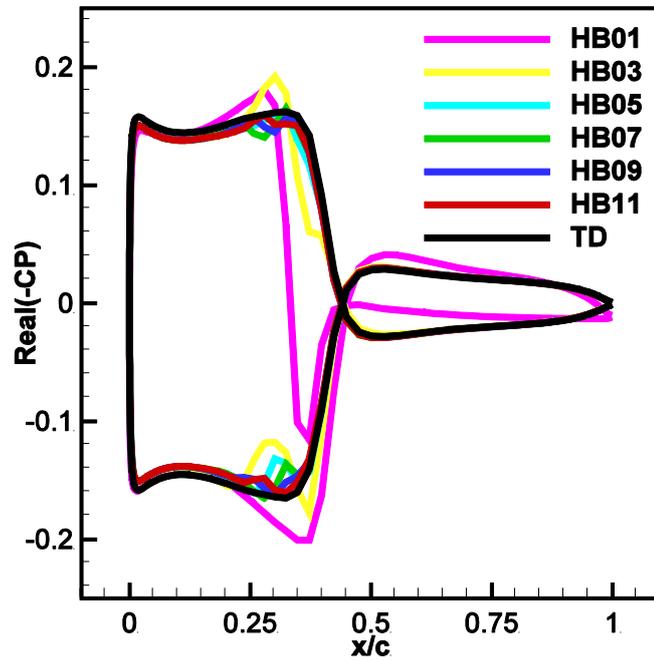


Figure 8.23: first surface pressure coefficients

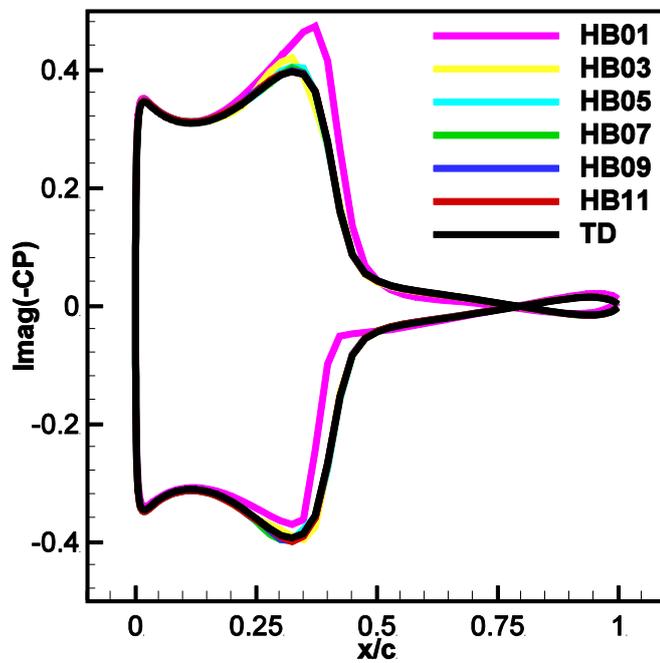


Figure 8.24: second surface pressure coefficients

In figure 8.23, one sees that the results using one or three complex harmonics do not predict correctly the real part of pressure coefficient, compared to the TD results. By increasing the number of harmonics, the solution is moving closer to the one obtained from the TD solver. As one sees in figure 8.23, nine harmonics are sufficient to achieve a good agreement with the results obtained from the TD solver. On the other hand the results of imaginary part of surface pressure coefficient, calculated with three or more harmonics are in very good agreement with the pressure coefficient computed with the TD solver (figure 8.24).

To examine in greater detail if the moving shock dynamics is resolved with sufficient accuracy by the harmonic balance analysis, the pressure coefficient C_p at six positions along one period is considered in figure 8.25. Similarly to figure 8.16, each subplot of figure 8.25 is labelled at its bottom right hand side with the value of instantaneous angle of attack of the airfoil. The arrow beside each value, indicates if the angle of attack is increasing or decreasing. The positions of the airfoil are the same as those of figure 8.16. The horizontal axis represents the x-coordinates of the airfoil non-dimensionalized by the chord, while the vertical axis represents the pressure coefficient C_p . Each line of each subfigure denotes the solution calculated with a different number of harmonics. The black line denotes the solution of pressure coefficient obtained from the TD solver.

One sees that the pressure coefficient of the top subplot on the left hand side ($\alpha = 0.52^\circ \uparrow$) of figure 8.25 and of the middle subplot on right hand side ($\alpha = 1.25^\circ \downarrow$) of the same figure, requires at least 11 harmonics in order to predict the same results with respect to the TD solver. On the other hand, the pressure coefficient on the remaining positions requires only 5 harmonics in order to be consistent with the TD results. As one sees, the accuracy of the shock is well established by the harmonic balance solver and the pressure coefficient C_p is found to be consistent to the TD solver.

The convergence histories of the four RANS PDEs and the two turbulence model PDEs of the SST analysis using equation (2.6.1) with $S_R = 2500$ at the six positions mentioned above for each number of harmonics, are reported in the subplots of figure 8.26. All calculations have been done using multigrid scheme with a high order restriction operator and a bilinear prolongation operator. In figure 8.26 one sees that the convergence rate of all residuals in all positions has been dropped by at least 2.5 orders of magnitude. It should be noted that increasing the number of harmonics retained in truncated Fourier-series representation of the HB analysis improves the convergence rate of the HB solver.

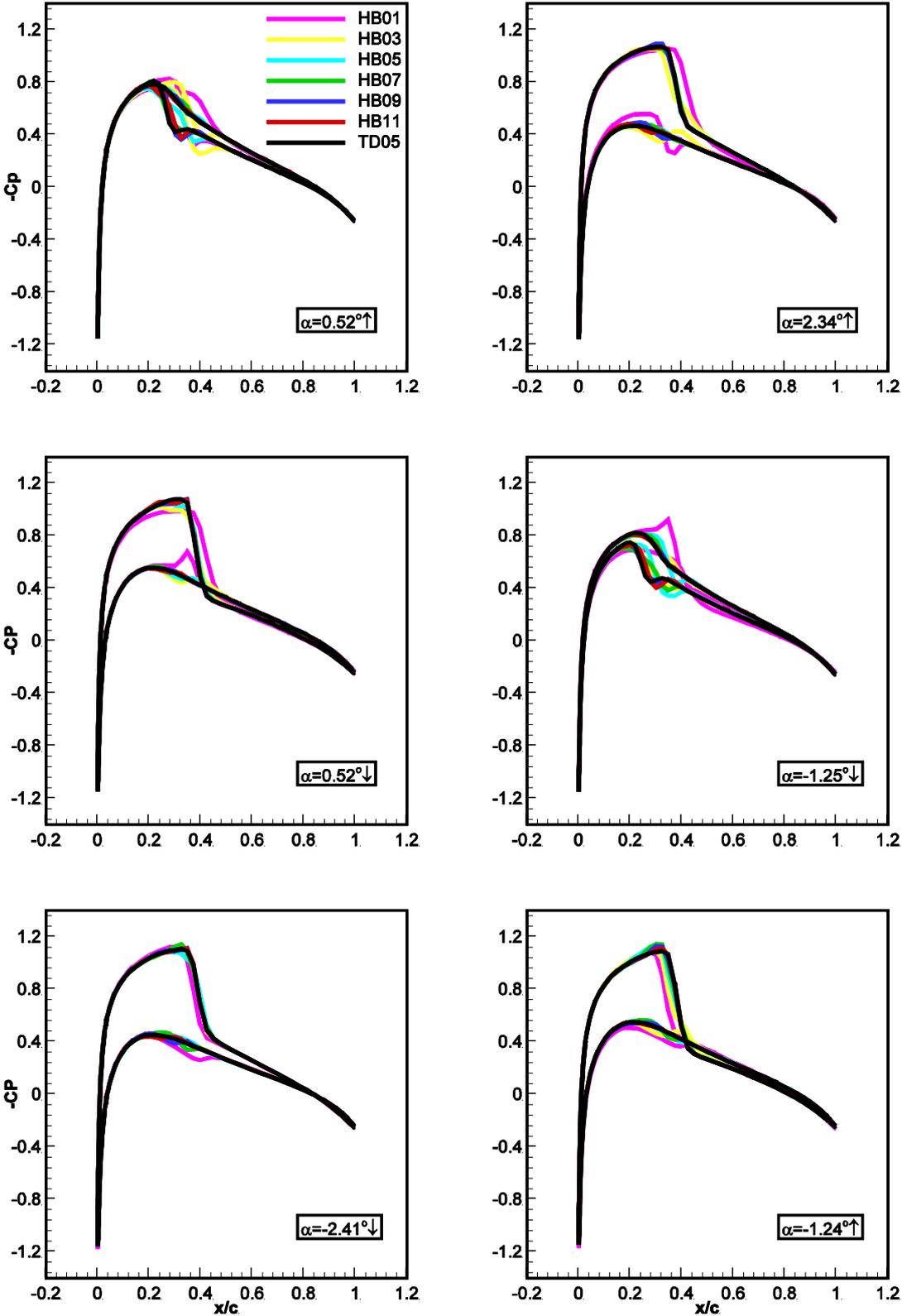


Figure 8.25: numerical comparison instantaneous pressure coefficient at several positions calculated by the harmonic balance and time domain solver

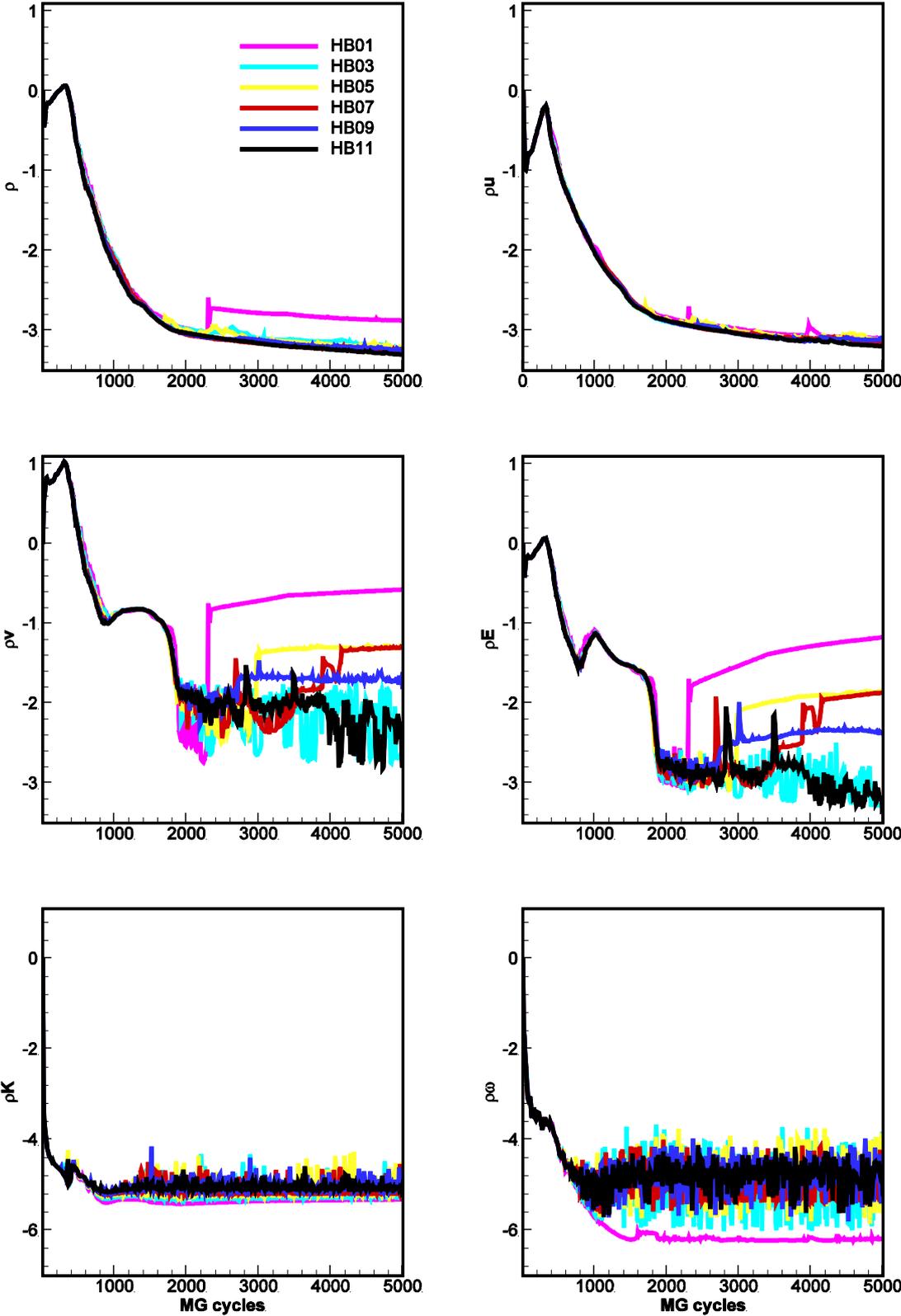


Figure 8.26: Convergence histories of turbulent SST analyses of flow field past the pitching NACA0012 airfoil using the harmonic balance solver

8.2.5 Computational efficiency

In the previous subsection a harmonic balance flow analysis was carried out considering the turbulent flow field past a pitching NACA0012 airfoil. In this section a computational efficiency assessment of the HB solvers and the TD solvers is reported. For the TD solver the flow field has been simulated using 64 time intervals per period until the maximum difference between C_L over the last two pitching cycles became less than 0.1% of the maximum C_L over the last cycle. The number of pitching cycles required for fulfilling this requirement was five periods. In addition the CFL number has been set to 2, a multigrid acceleration is used and 2500 MG iterations per physical time-step were needed to achieve a converged solution. For the harmonic balance solver the flow field has been simulated using 10,000 multigrid iterations with the same CFL number and the same RK scheme like the TD solver. In table 8.6 one sees the computational efficiency of the HB solver with respect to the TD.

<i>Solver</i>	<i>CPU time(h)</i>	<i>speed up</i>
TD	125.86	1.00
HB01	1.02	123.39
HB03	3.37	37.35
HB05	6.90	18.24
HB07	10.66	11.80
HB09	14.30	8.80
HB11	19.09	6.59

Table 8.6: CPU time speed up of the harmonic balance methods with respect to the time domain method.

The HB solver using 11 harmonics is at least 6.5 times more efficient in terms of computational cost with respect to the TD solver. From table 8.6 one sees that with only one harmonic the speed up of HB solver is more than 100 times faster than the TD solver, but the results are not accurate and in most cases are wrong. By increasing the number of harmonics of the HB solver the computational efficiency is decreasing and as a consequence the computational cost is increasing. The computational efficiency of the HB methods depends on the numerical integration method. When using an explicit RK

strategy with multigrid acceleration technique, the computational cost of the HB method grows linearly with the number of harmonics. Moreover for engineering applications usually only the force coefficients are subject of investigation and in this case, as seen above (section 8.3.4) only 7 harmonics were needed, which results in a speed up of at least 10 times.

8.3 Oscillating subsonic wings

This section focuses on the 2D turbulent flow analysis of a pitching airfoil in low speed subsonic operating conditions, and is subdivided into three subsections. The first subsection presents the physical and computational set-up of the flow analysis, along with the assessment of the spatial refinement of the grid used in this test case. The second subsection investigates in detail the turbulent unsteady flow resulting from the harmonic pitching motion of the airfoil and compares the results of the TD and HB analyses. The last subsection reports the analysis of the run-time using the TD and HB simulations for different number of complex harmonics. This subsection also provides the speed-up of the HB flow analysis over the TD analysis for typical values of the number of complex harmonics.

8.3.1 Physical and numerical problem set-up

The wing section selected for this study is the NACA0015 airfoil, which is animated by harmonic pitching motion and is immersed in a low-speed subsonic freestream. The flow remains attached throughout the oscillation cycle. This pitching motion is characterized by a pitching amplitude a_0 of 4.0° with mean angle of attack a_m of 10.0° and a reduce frequency λ of 0.2. The pitching centre is located at $x_p = 0.25c$, and the free stream Mach number M_∞ is 0.2. The flow conditions are given in table 8.7. The value of the Reynolds number based on freestream velocity and airfoil chord is $Re = 1.5 \cdot 10^6$.

parameter	value
M	0.2
a_m	10.0°
a_0	4.0°
K	0.2
Re	$1.5 \cdot 10^6$

Table 8.7: Flow conditions
for the NACA0015 airfoil

The time-dependent turbulent flow field past the pitching airfoil has been computed with multi-block moving grids. In all simulations, the whole computational grid is animated by pitching motion defined by equation (8.3.1). The grid does not deform, and it undergoes a rigid-body motion corresponding to the prescribed motion of the airfoil. The flow field has been simulated using a C-grid with 102,400 cells. The grid coordinates are non-dimensionalized by the chord of the airfoil, and the farfield boundary is at about 50 chords from the airfoil. The grid has been divided into 256 blocks and a parallel computation has been done to decrease the computational time. The number of mesh intervals on the airfoil (n_A), the number of intervals on the C-cut (n_C), the number of intervals in the normal-like direction (n_N), and the distance of the first grid points off the airfoil from the airfoil surface (Δ_W) are reported in table 8.8. Four local views of the grid adopted for the analyses reported in the following subsection are provided in figure 8.27.

Grid	n_A	n_C	n_N	Δ_W	N_{cell}
Fine	400	100	160	$1.6 \times 10^{-6}c$	102,400

Table 8.8: Main geometric parameters of the grid used to compute the pitching airfoil.

In order to assess the sensitivity of the turbulent solutions to the level of temporal refinement using the TD solver, the flow field has been simulated using 32, 64, and 128 time-intervals per period. All turbulent simulations have been run until the maximum difference between C_L over the last two pitching cycles became less than 0.1% of the maximum C_L over the last cycle. The number of pitching cycles typically required for fulfilling this requirement varied between four and six depending on the temporal refinement, and also on whether the simulation has been started from a freestream

condition or restarted from the solution of a simulation using different temporal refinement. The periodicity error of the other force coefficients at a given period is similar to that of C_L . Reducing the C_L periodicity error below this value yields insignificant variations of the periodic solutions with respect to those reported in the figure 8.28. For all turbulent analysis of the pitching airfoil presented in this report, y^+ has been found to be smaller than one at all grid points and all times of the periodic flow field. The periodic profiles of the lift coefficient C_L , the drag coefficient C_D and the pitching moment coefficient C_M computed with above said three levels of time-refinement are reported in figure 8.28.

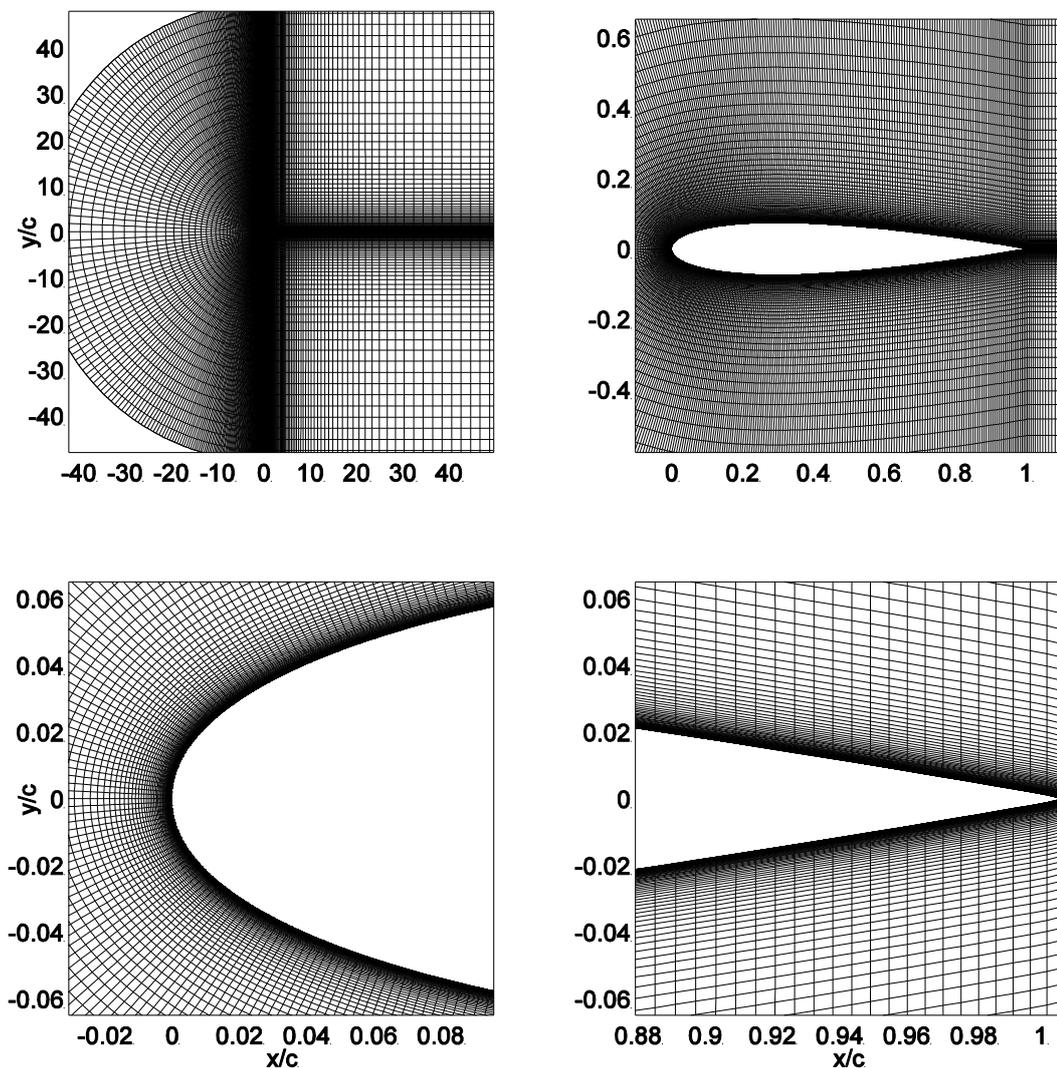


Figure 8.27: grid for the pitching NACA0015 airfoil. **Top left plot:** near-airfoil area. **Top right plot:** leading edge area. **Bottom left plot:** airfoil upper side area at mid-chord. **Bottom right plot:** trailing edge area.

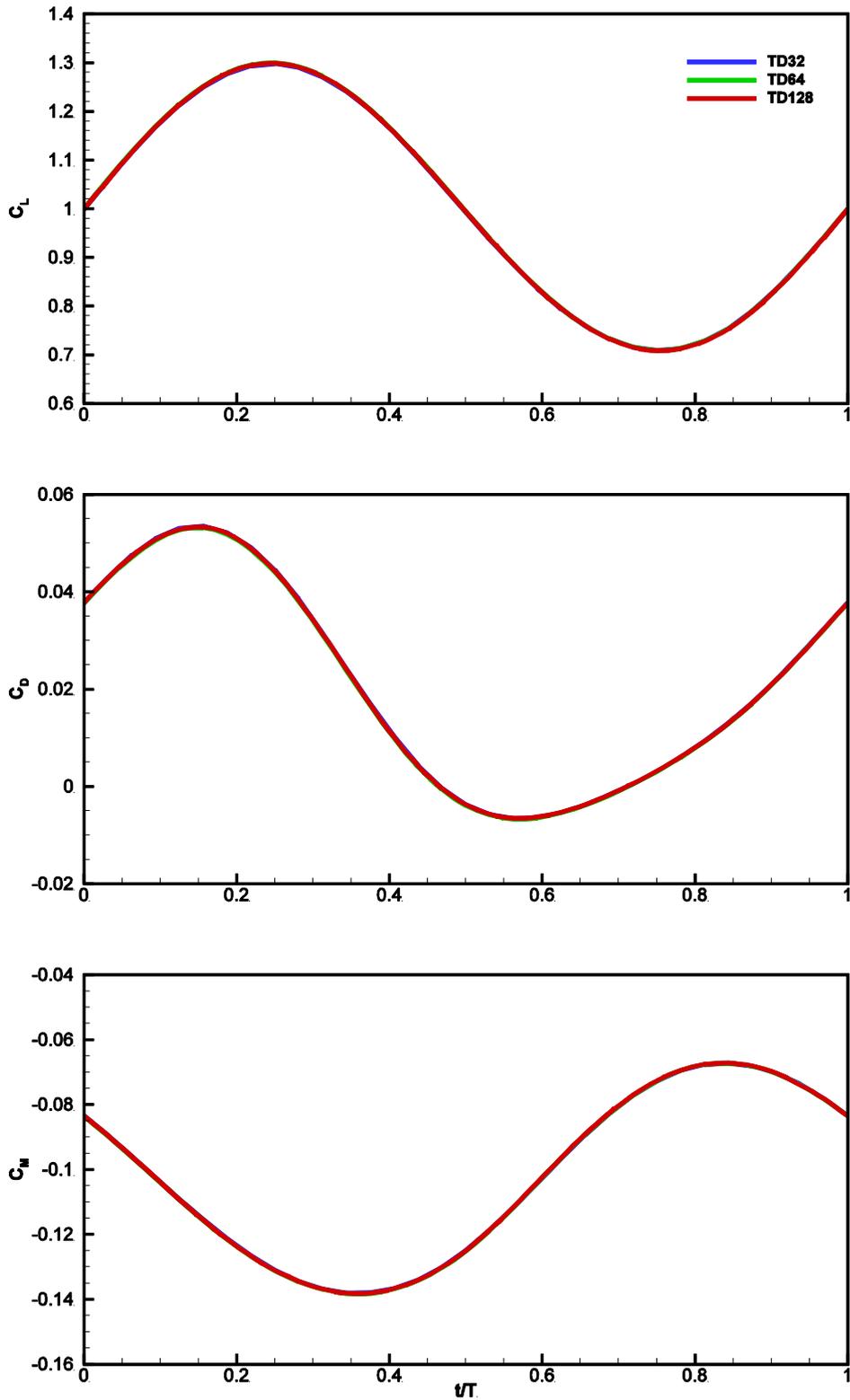


Figure 8.28: Comparison of force coefficients of oscillating wing NACA0015, varying time-refinement. **Top plot:** lift force coefficient. **Middle plot:** drag force coefficient. **Bottom plot:** pitching moment coefficient.

The vertical axis of each subplot reports the force coefficient and the horizontal axis reports a time variable non-dimensionalized by the period. These profiles highlight that the computed solution is almost independent of the number of intervals per period used to calculate the flow field. Some small variations between the solutions using 32 and 64 intervals per pitching cycle are instead observed. In order to keep low computational costs, the case with 64 time intervals per period has been used for all turbulent analyses presented in the remainder of this pitching airfoil. In addition, the CFL number of all turbulent simulations has been set to 3 and CFL ramping has been used. 2000 MG iterations per physical time-step have been performed and the scheme of multigrid was chosen to be the V-cycle with 2 smoothed Runge–Kutta cycles on the coarsest grid level and 5 Runge-Kutta cycles on the medium and fine grids. All calculations have been performed using the PIRK MG iteration.

8.3.2 Time-domain and Frequency-domain analysis

This subsection presents unsteady results obtained from the time domain analysis and the harmonic balance flow analysis described in section 5.2. All numerical results have been simulated using Wilcox’s wall boundary condition described by equation (2.6.1) with $S_R = 2500$. Figures 8.29 and 8.30 illustrate the comparison of the hysteresis cycle of the lift and pitching moment coefficient respectively, and provide numerical results obtained from the time-domain solver, and the harmonic balance solver, respectively. The horizontal axis of both figures 8.29 and 8.30 represents the instantaneous angle of attack while the vertical axis represents the lift and pitching moment coefficient, respectively. In figure 8.29, the TD results are labelled as “TD06”. For the frequency domain solver, several harmonic balance solutions have been simulated up to 4 harmonics. Each figure has 5 different lines. The first 4 lines denote the solution of the force coefficient of lift (figure 8.29) and pitching moment (figure 8.30), respectably, using different number of harmonic whereas the last line (black) denotes the solution obtained from the TD solver.

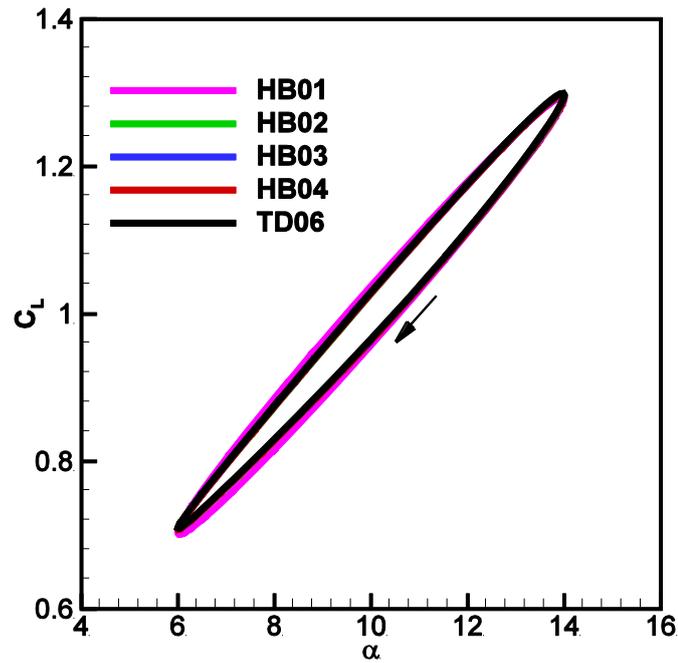


Figure 8.29: *hysteresis cycle of lift coefficient for the pitching NACA0015 airfoil using 4 harmonics*

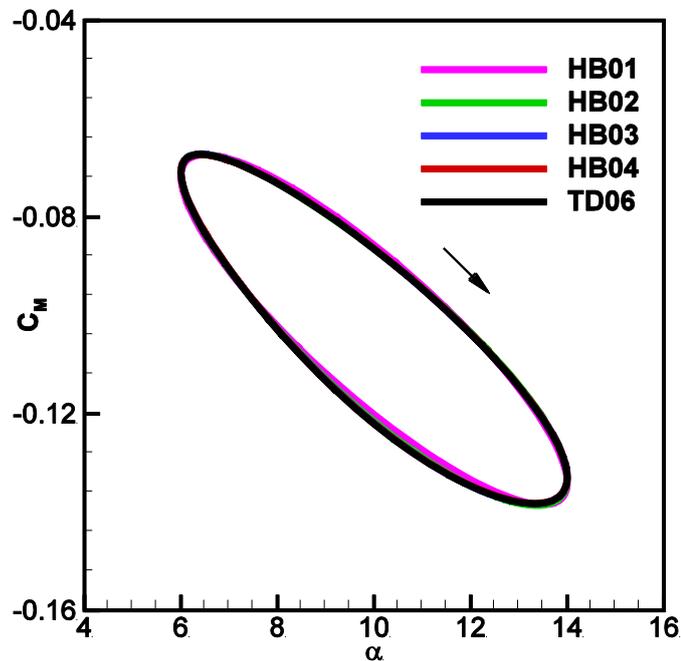


Figure 8.30: *hysteresis cycle of pitching moment coefficient for the pitching NACA0015 airfoil using 4 harmonics*

The compared results between the simulations obtained from the frequency domain solver and the time domain solver, developed in this study are in very good agreement. As one sees in figure 8.29 and 8.30 the number of complex harmonics one needs to calculate the lift and pitching moment coefficient, respectively, of a low speed flow field past the NACA0015 airfoil and obtain an accurate solution comparable to that of the TD solution is greater than one. Using only one complex harmonic the solution is very close to the TD solution but is not fully identical and more harmonics are needed. Further increasing the number of complex harmonics has almost zero effect on the force solutions. The arrows in figures 8.29 and 8.30 indicate the direction of the lift and pitching moment coefficient, respectively, of the time evolution of the TD solver.

Figure 8.31 illustrates the comparison of the hysteresis cycle of the drag coefficient, obtained from the TD analysis and the harmonic balance analyses. The horizontal axis of figure 8.31 represents the instantaneous angle of attack while the vertical axis represents the drag force coefficient. Similar to figures 8.29 and 8.30, only Wilcox wall BC described by equation (2.6.1) with $S_R = 2500$ has been simulated for both solvers.

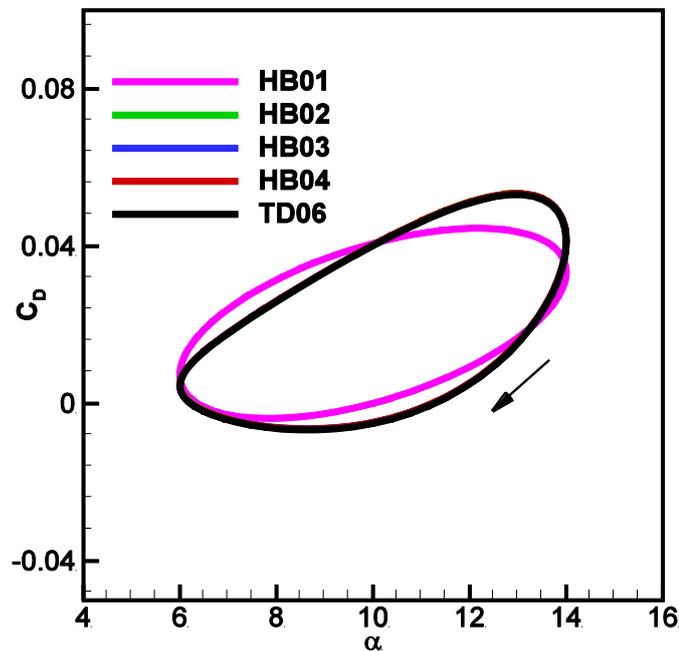


Figure 8.31: *hysteresis cycle of drag coefficient for the pitching NACA0015 airfoil using 4 harmonics*

The five lines that appear in figure 8.31 have the same structure as in figures 8.29 and 8.30. The compared results for the drag coefficient between the simulations obtained from

the TD solver and the frequency domain solver are in very good agreement. Similarly conclusions as discussed for the lift and pitching moment coefficients, the number of complex harmonics that one needs to calculate the drag coefficient of a low speed flow field past the NACA0015 airfoil and be consistent with the TD solution is greater than one. On the other hand, the solution obtained using only one harmonic fails completely to predict correctly the drag coefficient. From the figures 8.19, 8.20 and 8.21 one can observe that the higher the number of harmonics, the more accurate the predictions of the force coefficients are.

Similar analysis for the zeroth flow complex harmonic, like the one presented for the transonic flow field past the NACA0012, is reported in figure 8.32. The horizontal axis of figure 8.32 represents the position along the airfoil chord and the vertical axis represents the zeroth harmonic unsteady surface pressure coefficients.

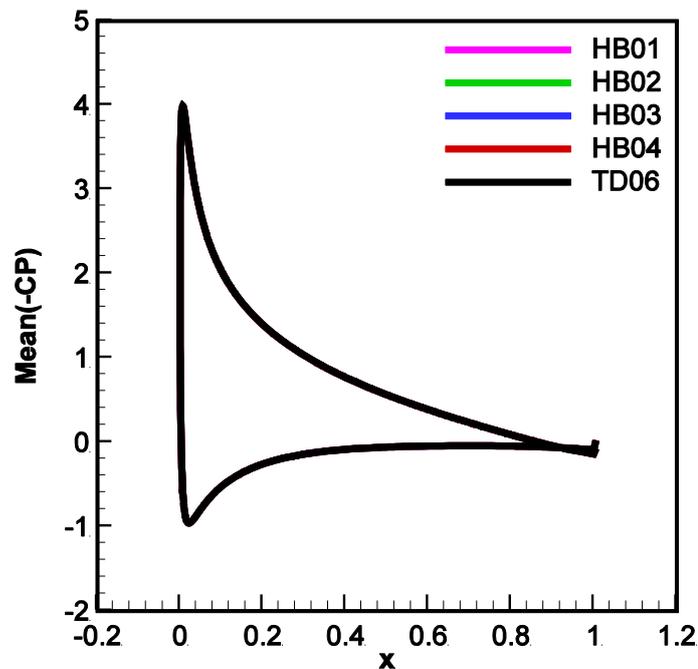


Figure 8.32: zeroth surface pressure coefficients

Similarly to figure 8.31, each line of figure 8.32 denotes the solution of average surface pressure coefficient using different number of complex harmonics. The black line denotes the solution of average surface pressure coefficients obtained from the TD solver. One sees that all results using different number of harmonics predict correctly the zeroth surface pressure coefficient as they are superimposed with the results obtained from the TD solver.

In order to get further insight on the performance of the frequency domain solver, results from the real and imaginary parts of the first flow harmonic are presented. Similarly to plot 8.32 the horizontal axis of figures 8.33 and 8.34 represents the non-dimensionalized coordinates of the airfoil and the vertical axis represents the real and imaginary part of the first harmonic surface pressure coefficient, respectively.

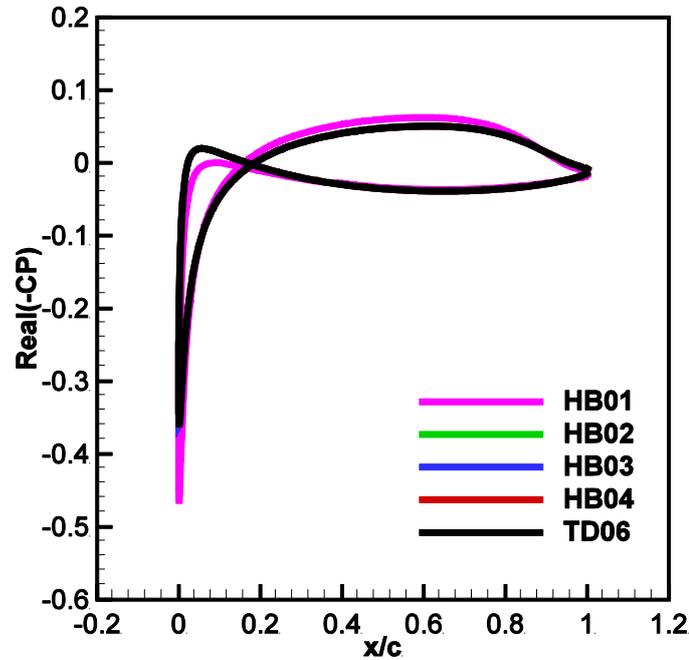


Figure 8.33: first surface pressure coefficients

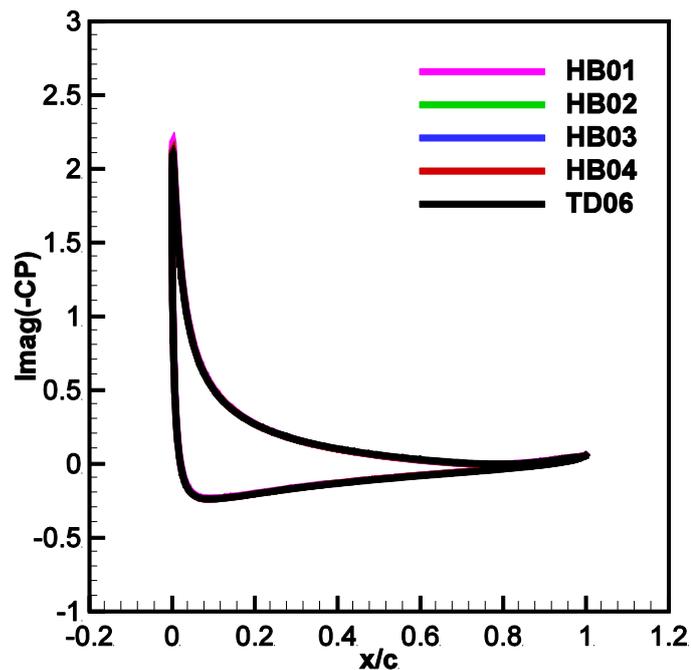


Figure 8.34: second surface pressure coefficients

The lines of figures 8.33 and 8.34 have the same structure as in figure 8.32. One sees that the results in figure 8.33, calculated with only one harmonic, do not predict correctly the real part of pressure coefficient, compared to the ones obtained from the TD analyses. By increasing the number of harmonics, the solution becomes consistent with the one obtained from the TD solver. Moreover, one sees that all results using different numbers of harmonics correctly predict the imaginary part of surface pressure coefficient as they are superimposed with the ones obtained from the TD solver.

To examine in greater detail the low speed flow field past the NACA0015 and to validate the accuracy of the harmonic balance solver, the pressure coefficient C_p at six positions along one period is considered in figure 8.35. Each subplot of figure 8.35 is labelled at its bottom right hand side with the value of instantaneous angle of attack of the airfoil. The arrow besides each value, indicates if the angle of attack is increasing or decreasing. Their instantaneous angle of attack is 10.00° , 14.00° and 8.11° when the AoA is increasing and 12.22° , 7.46° and 8.00° when the AoA is decreasing, respectively. The horizontal axis of figure 8.35 represents the x-coordinates of the airfoil non-dimensionalized by the chord while the vertical axis represents the pressure coefficient C_p . Each line of each subfigure denotes the solution obtained from a different number of harmonics. The black line denotes the solution of pressure coefficient obtained from the TD solver.

One sees that all results for the pressure coefficient C_p using different number of harmonics are almost identical to the results obtained from the TD solver in all different positions along a period. It is important to highlight that the minimum number of harmonics needed to predict the transonic flow field past the NACA0012 and obtain an accuracy level comparable to that of the TD results, was 11. Conversely, in the present low speed case, two complex harmonics are sufficient. This big difference comes from the fact that the pitching NACA0012 case features a strong shock discontinuity and many harmonics are needed to obtain the same very high accuracy of the TD solution.

The convergence histories of the four RANS PDEs and the two SST turbulence model PDEs using the TD solver at the six positions mentioned above are reported in the subplots of figure 8.36. All calculations have been done using high order multigrid operators. One sees in figure 8.36 that the convergence rate of all residuals in all positions has been dropped by at least 3 orders of magnitude. As expected the residual of the K equation has instead been dropped by only one order in all positions.

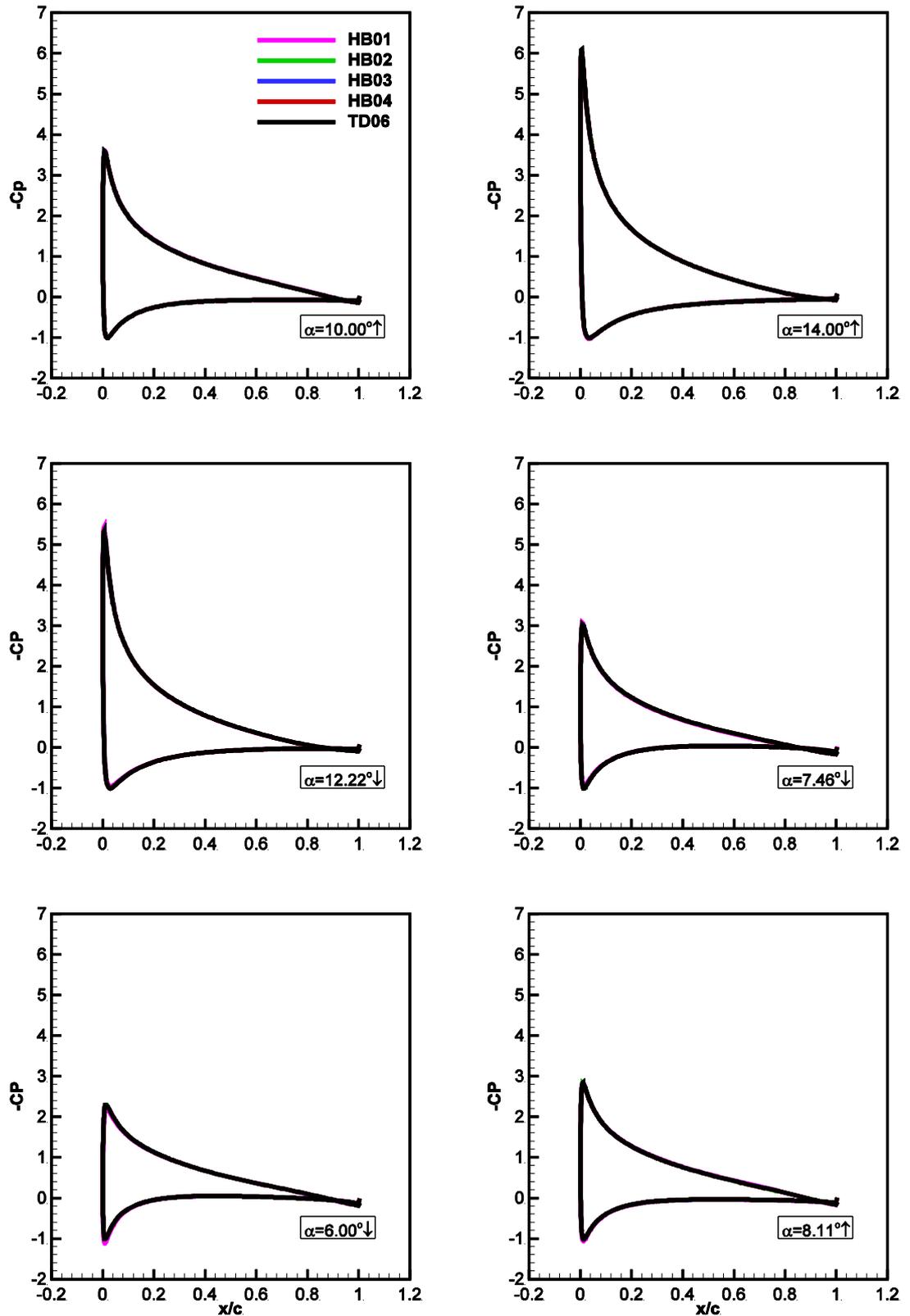


Figure 8.35: numerical comparison instantaneous pressure coefficient at several positions calculated by the harmonic balance and time domain solver

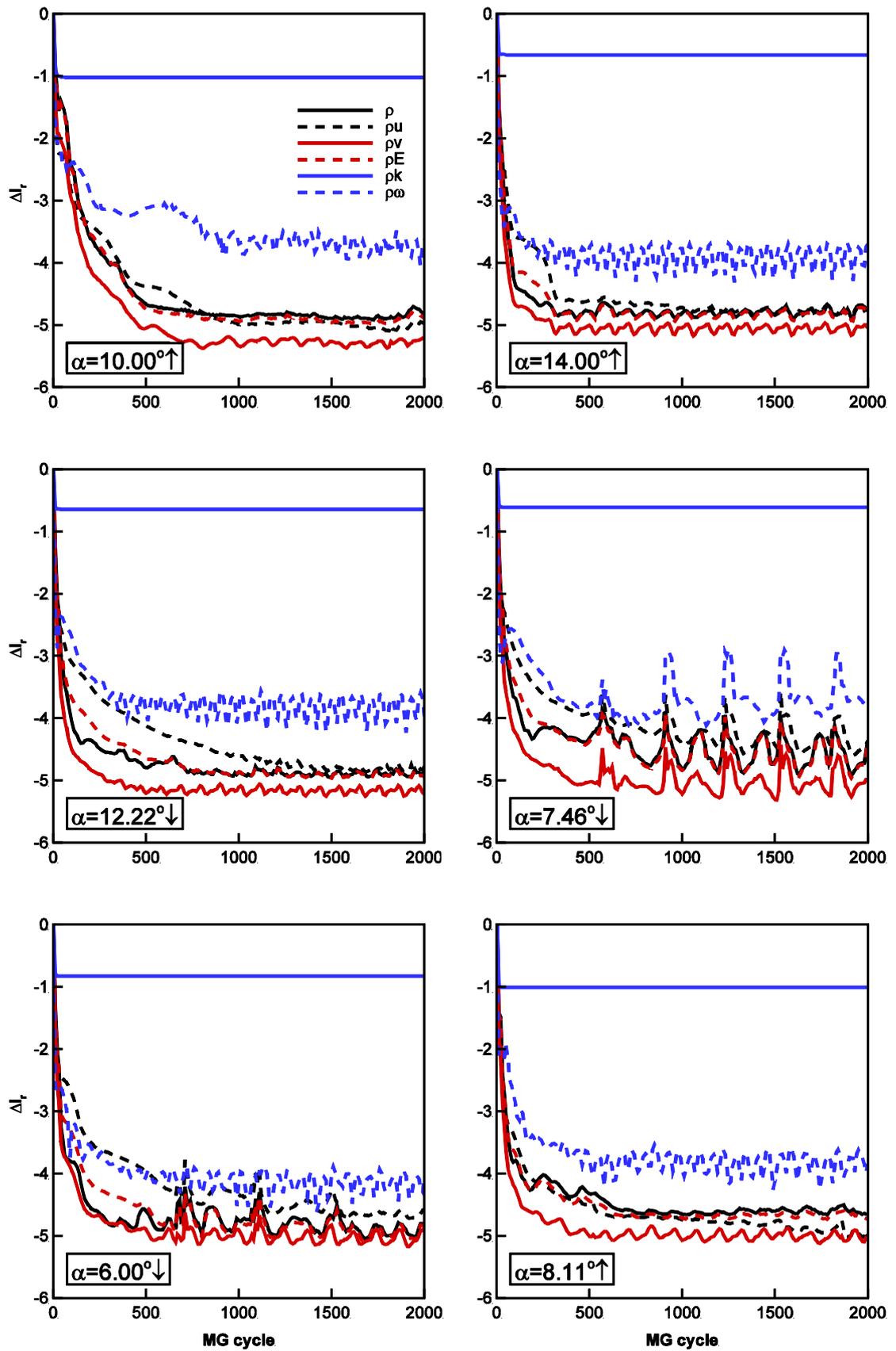


Figure 8.36: Convergence histories of turbulent SST analyses of flow field past the pitching NACA0015 airfoil using Wilcox's boundary condition with $S_R = 2500$

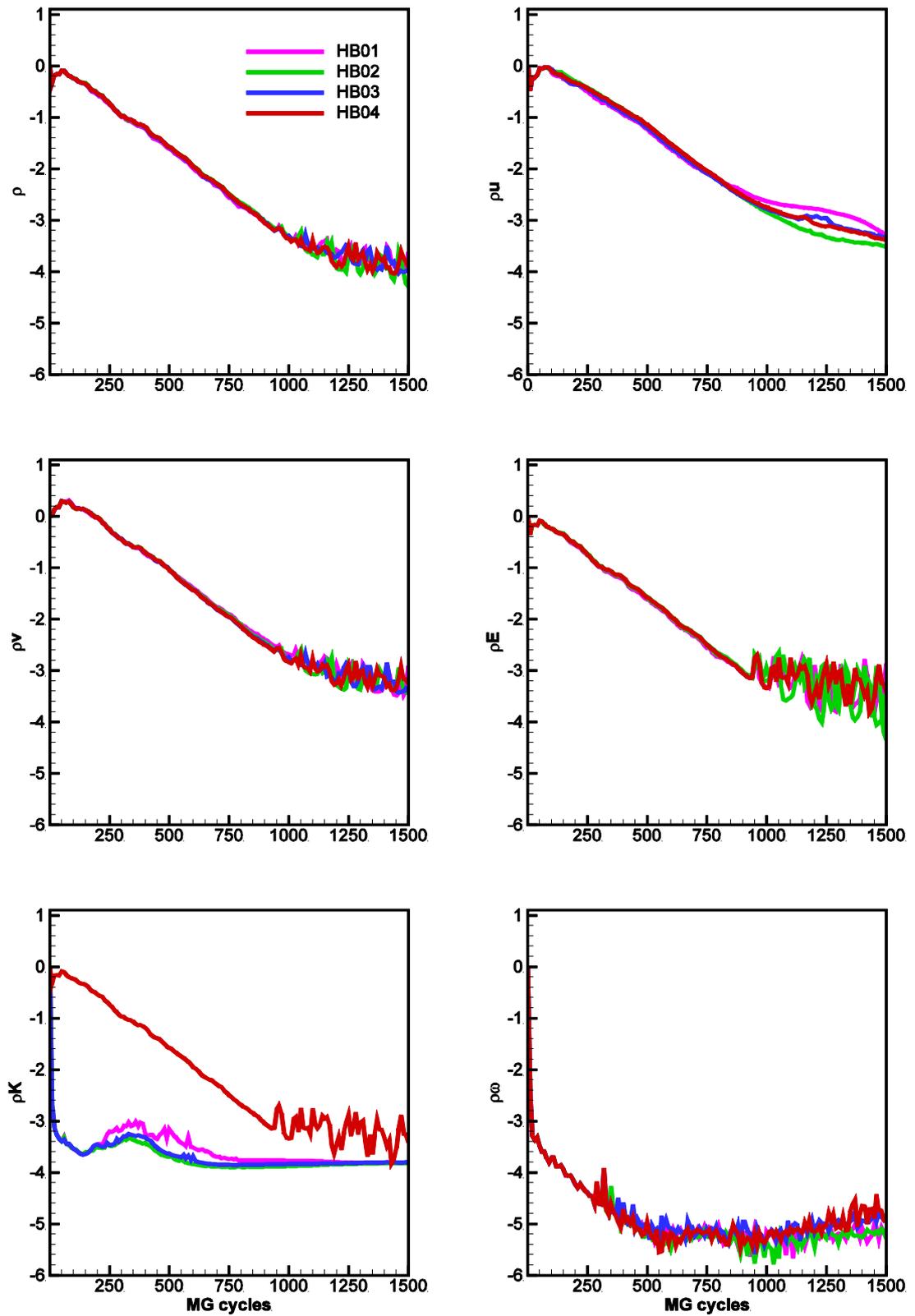


Figure 8.37: Convergence histories of turbulent SST analyses of flow field past the pitching NACA0015 airfoil using the harmonic balance solver.

The convergence histories of the four RANS PDEs and the two turbulence model PDEs of the SST analysis using the harmonic balance solver for each number of

harmonics are reported in the subplots of figure 8.37. All calculations have been done using the multigrid scheme with a high order restriction operator and a bilinear prolongation operator. As one sees in figure 8.37, the convergence rate of all residuals in all positions has been dropped for at least 3 orders of magnitude.

8.3.3 Computational efficiency

In the previous subsection a harmonic balance analysis was carried out using a turbulent flow field past a pitching NCACA0015 airfoil. In this section a computational efficiency assessment of the HB solvers and the TD solver is reported. For the TD solver the flow field has been simulated using 64 time intervals per period until the maximum difference between C_L over the last two pitching cycles became less than 0.1% of the maximum C_L over the last cycle. The number of pitching cycles required for fulfilling this requirement was six periods. In addition the CFL number has been set to 3, a multigrid acceleration is used and 2000 MG iterations per physical time-step were needed to achieve a converged solution. For the harmonic balance solver the flow field has been simulated using 10,000 multigrid iterations using the same CFL number and the same RK scheme like the time domain solver. From table 8.9 one sees the computational efficiency of the HB solver with respect to the TD.

<i>Solver</i>	<i>CPU time(h)</i>	<i>speed up</i>
TD	6.88	1.00
HB01	0.26	26.46
HB02	0.41	16.78
HB03	0.59	11.66
HB04	0.80	8.60

Table 8.9: CPU time speed up of the harmonic balance methods with respect to the time domain method.

The HB solver using 4 harmonics is at least 8.5 times more efficient in terms of computational cost with respect to the TDsolver. From table 8.9 one sees that using only one harmonic the speed up of HB solver is more than 26 times faster than the TD solver,

but the results are not so accurate. By increasing the number of harmonics of the HB solvers the computational efficiency is decreasing and as a consequence the computational cost is increasing. From the results mention above one can consider the solution using 2 harmonics as in all abovesaid analysis the HB02 results were the same as the time domain results. In this case the HB solver is at least 16 times faster than the TD solver.

Chapter 9

CONCLUSIONS

9.1 Summary and concluding remarks

A novel robust explicit multigrid fully-coupled integration approach to the solution of the steady, time-domain and harmonic balance Reynolds-Average and shear stress transport turbulence model equations has been developed, validated and used for the analysis of complex unsteady engineering problems.

9.1.1 Algorithmic conclusions

Thorough theoretical and numerical analyses of all developed algorithmic features have been presented. The key features of the fully-coupled turbulent integration include:

- the use of a restriction operator with higher order with respect to that used for the Euler and Navier-Stokes equations,
- the use of an ad-hoc limiter of the prolonged multigrid corrections,
- an effective point-implicit treatment of the negative source terms of the turbulence model equations, and
- a modelling approximation enabling update of the second turbulent variable, ω , independently of the first turbulent flow variable, the turbulent kinetic energy.

It is found that the developed enhancement of the restriction and prolongation multigrid algorithm plays a crucial role in the achievement of a stable, i.e. robust integration method. In addition the approximation applied on the update of the turbulence equations yields a significant reduction of the run-time of the harmonic balance flow analysis with negligible impact on the accuracy of the computed solution for both low-speed and transonic flow problems.

The presented analyses have also highlighted an important modelling aspect regarding the use of Wilcox's solid wall boundary condition for the turbulent variable ω . This boundary condition depends on a constant, the variations of which have been found to induce significant variations in the computed solution. It is not obvious how to select the most suitable value of this constant. The original value of 100 proposed by Wilcox appears not to match the results of an approximate boundary condition proposed by Menter whereas a value of 2500 appears to greatly reduce the solution differences obtained by either Wilcox's or Menter's wall boundary conditions with a given turbulence model. Interestingly, it is found that

- the solution differences arising by using the original Wilcox's boundary condition and Menter's approximate boundary condition can be similar or even larger than the solution differences observed using the same wall boundary condition with two different turbulence models.

This highlights that the choice of the wall boundary condition for the two-equation turbulence models is an additional source of uncertainty in the RANS analysis of the turbulent flows which is often overlooked.

An additional algorithmic aspect addressed in this thesis is the impact on the solution accuracy of using either a first or a second order space-discretisation of the turbulence model equations. Published literature reveals that both approaches are regularly adopted, but the motivation of such studies for using either approach is often unclear. The additional computational cost associated with the use of a second rather than a first order discretisation of the turbulence model is very small, and, for this reason, it is likely that the main reason why certain RANS codes use a first order discretisation of the turbulence model is a higher numerical stability with respect to the case in which the first order discretisation is used. However, the impact of using a first order discretisation on the computed solution should be assessed. The comprehensive analyses of this issue reported in this thesis highlight that:

- the solution differences due to the use of first or second order discretisation of the turbulence model are significant when significant flow gradients exist, and this is common to both subsonic and transonic flow problems, and
- the solution differences obtained by using either a first or a second order discretisation of the turbulence model become asymptotically small as the grid refinement increases even in the presence of significant flow gradients like those induced shocks.

This is not surprising as the discretisation error of first and second order approximations become closer as the mesh refinement increases, but it highlights the importance of adopting second order discretisation, as, in the presence of significant gradients like flow separations in low- or high-speed flows, this choice enables a higher solution accuracy with coarser grids and thus with lower computational costs.

9.1.2 Fluid dynamics conclusions

The main computational results of this research regard a low-speed renewable energy application and an aeronautical engineering application. The former application is the comparative analysis of the unsteady aerodynamics of an energy-extracting oscillating wing device in two limiting laminar and turbulent regimes. The main motivation for analysing this problem has been both to highlight:

- the predictive capabilities and the numerical robustness of the developed turbulent time-domain flow solver for complex realistic problems, and
- to shed more light on the complex physics of this emerging renewable energy device.

The comparative aerodynamic analysis of the oscillating wing device operating in a laminar flow regime at $Re = 1100$ and a fully turbulent regime at $Re = 1.5 \times 10^6$ has been performed for two different kinematic operating conditions: a high-laminar

efficiency condition (case A), characterized by the occurrence of unsteady LEVS associated with dynamic stall, and a low laminar efficiency condition (case B), characterized by the absence of LEVS. The comparative turbulent/laminar analysis of case A has shown that:

- the optimal synchronization between wing motion and LEVS observed in the laminar regime is lost when operating in the turbulent regime, and
- the power extraction efficiency in the turbulent regime is nearly 40%, whereas that in the laminar regime is about 34%.

The dependence of the phase between wing motion and LEVS on the Reynolds number points to the importance of incorporating turbulent flow effects in the optimization of the kinematic parameters of this device when its operating regime is fully turbulent. By doing so, efficiency even higher than that of 40% reported herein for the considered turbulent Reynolds number is likely to be achieved. The operating condition B is characterized by the absence of LEVS. The comparative turbulent/laminar analysis of case B has shown that:

- the turbulent estimate of the efficiency is 7% higher than its laminar counterpart, but this difference appears to be caused by the different characteristics of the turbulent and laminar boundary layers at the considered Reynolds numbers.

The aeronautical engineering application has been performed using the time-domain and harmonic balance turbulent flow analysis of a transonic wing section animated by pitching motion. The main motivation of these analyses has been both to highlight:

- the computational benefits achievable by using the harmonic balance solution of the RANS and SST equations rather than the conventional time-domain solution, and
- demonstrate the predictive capabilities of the developed CFD system.

Detailed comparisons of the COSA and NUMECA time-domain solutions have confirmed the turbulent flow predictive capabilities of the COSA solver. The comparison

of the run-time associated with the COSA time-domain and harmonic balance analyses highlights that

- the latter approach reduces at least one order of magnitude the run-time required to determine the periodic flow solution with respect to the run-time required by the time-domain analysis.

A similar comparative assessment of the time-domain and harmonic balance solvers has been carried out using a subsonic unsteady periodic flow field resulting from the harmonic pitching of an airfoil. For this problem, it has been found that the use of the turbulent harmonic balance solver results in an even higher reduction of the run-time with respect to the time-domain solver, with insignificant solution accuracy reduction with respect to the latter approach.

9.2 Future work

On the algorithmic side, future extensions of this work include the development of three-dimensional flow capabilities, the implementation of Large Eddy Simulation and Detached Eddy Simulation modelling capabilities. Indeed, the extension of the existing code to three-dimensional flow capabilities is already underway.

On the application side, as the COSA code uses a compressible flow formulation, and it also features an effective low-speed preconditioner for modelling low-speed flow regions, subject to the implementation of application-dependent features (e.g. periodicity boundary conditions for turbomachinery problems), the code can be applied to many diverse engineering problems, including the analysis of the flow field of gas turbine compressor and turbine blades, helicopter rotors and aircraft wings. One of the forthcoming applications of the harmonic balance solver is the solution of horizontal and vertical axis turbine unsteady periodic flows.

Appendices

A) Derivation of Reynolds-Average Navier-Stokes equations

Reynolds proposed that, the instantaneous flow velocity $u_i(x, t)$ is the sum of fluctuating velocity $u_i'(x, t)$ and the mean velocity $\bar{u}_i(x)$,

$$u_i(x, t) = \bar{u}_i(x) + u_i'(x, t) \quad (\text{A.1})$$

where the time average or mean $\bar{u}_i(x)$ is defined as:

$$\bar{u}_i(x) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} u_i(x, t) dt \quad (\text{A.2})$$

where t denotes the time and T has to be large enough compared to the typical time scale of the fluctuations. Calculating the average of the equation (A.1), one sees that the time average of the fluctuating velocity $u_i'(x, t)$ is equal to 0. This can be easily proved using the condition that the average of the mean velocity $\bar{u}_i(x)$ is identical to the mean velocity $\bar{u}_i(x)$. The definition of the Reynolds time average can be written in a general form of any quantity φ .

$$\bar{\varphi}_i(x) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} \varphi_i(x, t) dt. \quad (\text{A.3})$$

Taking the average of two variables $\varphi\psi$ and ignoring all products of mean and a fluctuating quantity as the mean value of them are zero, we obtain the expression below.

$$\overline{\varphi\psi} = \overline{(\bar{\varphi} + \varphi')(\bar{\psi} + \psi')} = \bar{\varphi}\bar{\psi} + \overline{\varphi'\psi'}. \quad (\text{A.4})$$

One sees that the product of the mean values $\bar{\varphi} \bar{\psi}$ differs from the mean value of the product $\overline{\varphi \psi}$ as, in general, the average of the product $\overline{\varphi' \psi'}$ is not zero. One can follow the same procedure for a triple product, obtaining:

$$\overline{\varphi \psi \zeta} = \bar{\varphi} \bar{\psi} \bar{\zeta} + \overline{\varphi' \psi' \zeta} + \overline{\varphi' \zeta' \psi} + \overline{\zeta' \psi' \varphi} + \overline{\varphi' \psi' \zeta'}. \quad (\text{A.5})$$

If we consider the case where the density is constant, in order to avoid the triple correlations, then the Navier-Stokes equations (2.1.1), (2.1.2) and (2.1.7) can be solved for the mean values by averaging them over the turbulence time-scales. This procedure leads to a system of PDEs which formally differs from the original NS equations only because of the appearance of new term which form the components of the so-called *Reynolds stress tensor*:

$$\tau_{ij}^R = -\rho \overline{u_i' u_j'} = -\rho (\overline{u_i u_j} - \bar{u}_i \bar{u}_j). \quad (\text{A.6})$$

This tensor accounts for the transfer of momentum due to turbulent fluctuations, but additional equations are needed in order to compute these extra terms and complete or close the system of equations to be solved. The Reynolds averaging procedure is suitable for incompressible flows. More detail on this procedure and the complete set of Reynolds-averaged Navier-Stokes equations can be found in the book of Wilcox [15] in details.

In the case of a compressible flow, the density is no longer constant. In the mean conservation equations new terms appear due to the additional correlations involving density fluctuations. This has been highlighted with the derivation of equation (A.5). In this case the derivation of the averaged equations and the averaged system itself become very complex. A viable alternative is applying the Reynolds-Favre averaging to the NS equations, a procedure proposed by Favre in 1965. Favre used the density-weighted averaging and he introduced the mass-averaged velocity, defined as:

$$\tilde{u}_i(x) \equiv \frac{\lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} \rho(x, t) u_i(x, t) dt}{\lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} \rho(x, t) dt}. \quad (\text{A.7})$$

Before explaining in detail the procedure of Favre averaging the instantaneous variables that appear in the NS equations must be split in two parts, a mean value and a fluctuating component. Each variable is then given by the sum of two such components. The density ρ , the pressure p , and the components of the flux vector \mathbf{q} are averaged using the Reynolds approach. On the other hand the velocity components u_i , the temperature T , the internal energy e and the static enthalpy h are averaged using the Favre approach

$$\rho = \bar{\rho} + \rho' \quad (\text{A.8})$$

$$p = \bar{p} + p' \quad (\text{A.9})$$

$$q_i = \bar{q}_i + q'_i \quad (\text{A.10})$$

$$u_i = \tilde{u}_i + u_i'' \quad (\text{A.11})$$

$$T = \tilde{T} + T'' \quad (\text{A.12})$$

$$e = \tilde{e} + e'' \quad (\text{A.13})$$

$$h = \tilde{h} + h'' . \quad (\text{A.14})$$

The symbols $\bar{}$ and \prime refer to the Reynolds averaging approach and, on the other hand, the symbols $\tilde{}$ and $''$ refer to the Favre averaging approach. Taking into account the Reynolds averaging definition (A.3), the numerator of equation (A.7) is the mean value $\overline{\rho u_i}$ and the denominator is the mean value $\bar{\rho}$. Replacing these values the definition of Favre leads to the expression:

$$\bar{\rho} \tilde{u}_i \equiv \overline{\rho u_i} . \quad (\text{A.15})$$

Also, looking at the above expression and replacing the right hand side of the same equation using the equation (A.4), one can find that $\overline{\rho u_i''}$ is equal to 0. However, making use of the equations (A.11) and (A.15) we have $u_i'' = -\frac{\overline{\rho' u_i'}}{\bar{\rho}}$, which indicates that the fluctuating velocity u_i'' is not equal to 0. More correlations can be found in *Appendix A*.

Performing a mass averaging on the NS equations (2.1.9) by making use of all the above relations along with the correlations included in *Appendix A* and the equations (A.4), (A.5), a new set of equations (RANS) can be obtained in a differential form as:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_j} (\bar{\rho} \tilde{u}_j) = 0 \quad (\text{A.16})$$

$$\frac{\partial}{\partial t} (\bar{\rho} \tilde{u}_i) + \frac{\partial}{\partial x_j} (\bar{\rho} \tilde{u}_j \tilde{u}_i) = - \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} (\bar{\tau}_{ij} - \overline{\rho u_i'' u_j''}) \quad (\text{A.17})$$

$$\begin{aligned} & \frac{\partial}{\partial t} \left(\bar{\rho} \tilde{e} + \bar{\rho} \frac{\tilde{u}_i \tilde{u}_i}{2} + \frac{1}{2} \overline{\rho u_i'' u_i''} \right) \\ & + \frac{\partial}{\partial x_j} \left(\bar{\rho} \tilde{u}_j \tilde{h} + \bar{\rho} \tilde{u}_j \frac{\tilde{u}_i \tilde{u}_i}{2} + \tilde{u}_j \frac{1}{2} \overline{\rho u_i'' u_i''} \right) \\ & = \frac{\partial}{\partial x_j} \left(\tilde{u}_i [\bar{\tau}_{ij} - \overline{\rho u_i'' u_j''}] - \bar{q}_j - \overline{\rho u_j'' h''} + \overline{\tau_{ij} u_i''} \right. \\ & \left. - \frac{1}{2} \overline{\rho u_j'' u_i'' u_i''} \right). \end{aligned} \quad (\text{A.18})$$

The continuity equation is formally identical to its unaveraged form, but a new term $\overline{\rho u_i'' u_i''}$ appears in the momentum and energy equations. This term is the co-called *Reynolds-Favre stress tensor* τ_{ij}^F and accounts for the transfer of momentum due to turbulent fluctuations. This tensor is symmetric and has six independent components

$$\tau_{ij}^F = - \overline{\rho u_i'' u_j''}. \quad (\text{A.19})$$

The left hand side of the averaged energy equation contains a new term. This term is related to the kinetic energy per unit volume of the turbulent fluctuations and is defined as:

$$\bar{\rho} K = \frac{1}{2} \overline{\rho u_i'' u_i''} \quad (\text{A.20})$$

So, we can rewrite the definition of the total energy and total enthalpy as:

$$E = \tilde{e} + \frac{1}{2} \tilde{u}_i \tilde{u}_i + K, \quad H = \tilde{h} + \frac{1}{2} \tilde{u}_i \tilde{u}_i + K. \quad (\text{A.21})$$

In the case of a perfect gas, this definition of the total energy E implies that the static pressure \bar{p} is defined as:

$$\bar{p} = (\gamma - 1) \left[\bar{\rho} E - \bar{\rho} \frac{1}{2} \tilde{u}_i \tilde{u}_i - \bar{\rho} K \right]. \quad (\text{A.22})$$

Another new term (A.23) which appears in the equation (A.18) is due to the turbulent transport of heat:

$$q_{Tj} = \overline{\rho u_j'' h''}. \quad (\text{A.23})$$

The last two terms of the right hand side of the energy equation (A.18) represent the molecular diffusion and turbulent transport of turbulent kinetic energy, respectively.

B) Closure approximations

An approximation, the classical Reynolds analogy [72] between momentum and heat transfer, is very commonly used by the researchers to model the turbulent heat flux vector \hat{q}_T . This flux vector results from the sum of a laminar and turbulent contribution. Note that the molecular viscosity, the constant pressure specific heat and the thermal diffusivity are linked by the Prandtl number P_r and the turbulent counter part of those terms are linked by the turbulent Prandtl number P_{rT} [15], respectively:

$$P_r = \frac{\mu c_p}{k_L}, \quad P_{rT} = \frac{\mu_T c_p}{k_T}. \quad (\text{B.1})$$

Thus, one can write that the total thermal conductivity coefficient k as:

$$k = k_L + k_T = c_p \left(\frac{\mu}{P_r} + \frac{\mu_T}{P_{rT}} \right) \quad (\text{B.2})$$

and the component can be written as

$$\hat{q}_j = \bar{q}_j + q_{Tj} = - \left[\frac{\mu}{P_r} + \frac{\mu_T}{P_{rT}} \right] \frac{\partial \tilde{h}}{\partial x_j}. \quad (\text{B.3})$$

The last two terms of the right hand side of the energy equation (A.18) are the molecular Diffusion and Turbulent Transport, respectively. One of the most commonly used approximation for these two terms is:

$$\overline{\tau_{ij} u_i''} - \frac{1}{2} \overline{\rho u_j'' u_i'' u_i''} = (\mu + \sigma_K \mu_T) \frac{\partial K}{\partial x_j}. \quad (\text{B.4})$$

So, the new system of equations is formally identical to the system of Equations (2.1.9) except for three features: the molecular stress tensor τ_{ij} which appears in diffusive flux vector $\hat{\Phi}_d$ has been replaced by the total stress tensor \hat{t}_{ij} . The molecular heat flux vector q_j has been replaced by the total heat flux vector \hat{q}_j , and there is an additional diffusive term in the energy equation depending on the eddy viscosity μ_T and the components of the gradient of the turbulent kinetic energy K . The diffusive terms (2.1.15) and (2.1.16) of the RANS equations are defined as:

$$\begin{aligned} \hat{E}_d \\ = \left[0 \quad \hat{t}_{xx} \quad \hat{t}_{xy} \quad \tilde{u}_x \hat{t}_{yy} + \tilde{u}_y \hat{t}_{xy} + (\mu + \sigma_K \mu_T) \frac{\partial K}{\partial x} - \hat{q}_x \right]' \end{aligned} \quad (\text{B.5})$$

$$\begin{aligned} \hat{F}_d \\ = \left[0 \quad \hat{t}_{xy} \quad \hat{t}_{yy} \quad \tilde{u}_x \hat{t}_{xy} + \tilde{u}_y \hat{t}_{yy} + (\mu + \sigma_K \mu_T) \frac{\partial K}{\partial y} - \hat{q}_y \right]' \end{aligned} \quad (\text{B.6})$$

where the symbol σ_K in the energy equation denotes one of the constants of the turbulence model and will be defined in the next section. Thus, making use of Boussinesq approximation, the main part of the Reynolds stress tensor is taken to be proportional to the strain rate tensor through an eddy viscosity parameter μ_T . In the turbulence models, this parameter, the value of which depends on the position in the computational domain, is determined by solving two additional transport equations, one for the turbulent kinetic energy K and the other for the so-called *specific dissipation rate* ω . Since μ_T is a function of K and ω , the value of μ_T at any point of the computational domain can be determined by using K and ω values at the same location. From now on, we will omit the symbols $\bar{}$ and $\hat{}$, as all the variables will be meant to be averaged values.

C) Correlations

The correlations needed to average the NS equations are written below:

$$\bar{\rho} \tilde{u}_i \equiv \overline{\rho u_i} = \bar{\rho} \bar{u}_i + \overline{\rho u_i'} \quad (\text{C.1})$$

$$\tilde{u}_i = \bar{u}_i + \frac{\overline{\rho' u_i'}}{\bar{\rho}} \quad (\text{C.2})$$

$$\overline{\rho u_i''} = 0 \quad (\text{C.3})$$

$$\overline{u_i''} = -\frac{\overline{\rho' u_i'}}{\bar{\rho}} \quad (\text{C.4})$$

$$\overline{\rho u_i u_j} = \bar{\rho} \tilde{u}_i \tilde{u}_j + \overline{\rho u_i'' u_j''} \quad (\text{C.5})$$

$$\begin{aligned} \overline{\rho u_j u_i u_i} &= \bar{\rho} \tilde{u}_j \tilde{u}_i \tilde{u}_i + \tilde{u}_j \overline{\rho u_i'' u_i''} + 2 \tilde{u}_i \overline{\rho u_i'' u_j''} \\ &+ \overline{\rho u_j'' u_i'' u_i''} \end{aligned} \quad (\text{C.6})$$

D) Non-dimensionalisation

Before the implementation in the CFD solver, the governing equations have to be non-dimensionalized. By doing so, the order of magnitude of the variables appearing in the RANS and turbulence model equations become of order 1, and this occurrence contributes to the reduction of round-off errors. Unfortunately, even using non-dimensionalized equations, it is not possible to achieve comparable order of magnitudes of all flow variables for flows with very low Mach number. Additionally, for all flow regimes, the variable ω is several orders of magnitude larger than the other variables. The adopted basis of reference variables is:

$$[\alpha_\infty, \rho_\infty, T_\infty, \mu_\infty, l] \quad (\text{D.1})$$

where α_∞ is the free-stream sound speed, ρ_∞ is the free-stream density, T_∞ is the free-stream temperature, l is a reference length, and μ_∞ is the free-stream molecular viscosity. All dimensional variables appearing in the (dimensional) governing equations are then expressed as the product of a reference dimensional variable and the unknown non-dimensional variables. The reference dimensional variables resulting from the choice of basis (D.1) are:

$$\rho_{\text{ref}} = \rho_\infty, \quad u_{\text{ref}} = \alpha_\infty, \quad l_{\text{ref}} = l, \quad Re_{\text{ref}} = \frac{\rho_\infty \alpha_\infty l}{\mu_\infty}, \quad (\text{D.2})$$

$$p_{\text{ref}} = \rho_\infty \alpha_\infty^2 = \gamma p_\infty, \quad t_{\text{ref}} = \frac{l}{\alpha_\infty}, \quad \tau_{\text{ref}} = \mu_\infty \frac{\alpha_\infty}{l}, \quad (\text{D.3})$$

$$x_{\text{ref}} = l, \quad k_{\text{ref}} = \alpha_\infty^2, \quad \omega_{\text{ref}} = \frac{\alpha_\infty}{l}, \quad \mu_{\text{ref}} = \mu_\infty. \quad (\text{D.4})$$

Using the Mach number at infinity, M_∞ , defined by:

$$M_\infty = \frac{u_\infty}{\alpha_\infty} \quad (\text{D.5})$$

it is possible to rewrite Re_{ref} as:

$$Re_{\text{ref}} = \frac{Re}{M_\infty} \quad (\text{D.6})$$

where Re is the Reynolds number based on the freestream velocity. Using the reference variables (D.1) one can easily transform the dimensional RANS equation into their non-dimensional counterpart. The non-dimensional Favre stress tensor can be written as

$$\tau_{ij}^F = 2 \mu_T \left(\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \frac{Re}{M_\infty} \rho K \delta_{ij} \quad (\text{D.7})$$

and the non-dimensional stress tensor $\hat{\tau}_{ij}$, which is the sum of the molecular and the Favre stress tensor, is:

$$\begin{aligned}
 \hat{\tau}_{ij} &= \tau_{ij} + \tau_{ij}^F \\
 &= (\mu + \mu_T) \left(\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) \\
 &\quad - \frac{2}{3} \frac{Re}{M_\infty} \rho K \delta_{ij}
 \end{aligned} \tag{D.8}$$

where the non-dimensional eddy viscosity is

$$\mu_T = \frac{Re}{M_\infty} \frac{\alpha_1 \rho K}{\max(\alpha_1 \omega, \Omega F_2)} \tag{D.9}$$

$$F_2 = \tanh(\arg_2^2) \tag{D.10}$$

$$\arg_2 = \max \left(\frac{2 \sqrt{K}}{\beta^* \omega d}, \frac{500 \mu}{\rho \omega d^2} \frac{M_\infty}{Re} \right). \tag{D.11}$$

The non-dimensional molecular viscosity μ is computed with the non-dimensional Sutherland's law:

$$\mu = T^{3/2} \frac{1 + S/T_{ref}}{T + S/T_{ref}}. \tag{D.12}$$

The non-dimensional component of the heat flux vector is:

$$\hat{q}_j = - \frac{1}{\gamma - 1} \left[\frac{\mu}{Pr} + \frac{\mu_T}{Pr_T} \right] \frac{\partial T}{\partial x_j}. \tag{D.13}$$

Given that the non-dimensional equation of state is $p \gamma = \rho T$, the non-dimensionalized total energy and total enthalpy per unit mass, respectively, are:

$$E = \frac{T}{\gamma(\gamma - 1)} + \frac{u_i u_i}{2} + K = \frac{p}{\rho(\gamma - 1)} + \frac{u_i u_i}{2} + K \tag{D.14}$$

$$H = \frac{T}{(\gamma - 1)} + \frac{u_i u_i}{2} + K = \frac{\gamma p}{\rho(\gamma - 1)} + \frac{u_i u_i}{2} + K. \tag{D.15}$$

Having assumed a perfect gas, the definition above leads to:

$$p = (\gamma - 1) \left[\rho E - \frac{1}{2} \rho u_i u_i - \rho K \right].$$

(D.16)

E) Compact divergence form of the URANS and SST equations

The non-dimensional URANS and the $K - \omega$ SST equations can be written in a compact vector form as follows:

$$\frac{\partial \hat{U}}{\partial t} + \frac{\partial(\hat{E}_c - \hat{E}_d)}{\partial x} + \frac{\partial(\hat{F}_c - \hat{F}_d)}{\partial y} = \hat{S} \quad (\text{E.1})$$

where the array \hat{U} collects the conservative flow variables of all transport equations, the arrays \hat{E}_c and \hat{F}_c collect the x- and y- components of the convective fluxes of all equations respectively, the array \hat{E}_d and \hat{F}_d collect the x- and y- components of the diffusive flux of all equations respectively, and the array \hat{S} contains the source terms of the K and ω equations. The definitions of the arrays \hat{U} , \hat{E}_c and \hat{F}_c are:

$$\hat{U} = \begin{bmatrix} \rho \\ \rho u_x \\ \rho u_y \\ \rho E \\ \rho K \\ \rho \omega \end{bmatrix}, \quad \hat{E}_c = \begin{bmatrix} \rho u_x \\ \rho u_x^2 + p \\ \rho u_x u_y \\ \rho u_x H \\ \rho u_x K \\ \rho u_x \omega \end{bmatrix}, \quad \hat{F}_c = \begin{bmatrix} \rho u_y \\ \rho u_y u_x \\ \rho u_y^2 + p \\ \rho u_y H \\ \rho u_y K \\ \rho u_y \omega \end{bmatrix} \quad (\text{E.2})$$

where the total energy E per unit mass and the total enthalpy H per unit mass are defined by the equations (E.14) and (E.15) respectively. The definitions of the diffusive flux vectors \hat{E}_d and \hat{F}_d are:

$$\hat{E}_d = \frac{M_\infty}{Re} \begin{bmatrix} 0 \\ \hat{t}_{xx} \\ \hat{t}_{xy} \\ u_x \hat{t}_{xx} + u_y \hat{t}_{xy} + (\mu + \sigma_K \mu_T) \frac{\partial K}{\partial x} - \hat{q}_x \\ (\mu + \sigma_K \mu_T) \frac{\partial K}{\partial x} \\ (\mu + \sigma_\omega \mu_T) \frac{\partial \omega}{\partial x} \end{bmatrix}, \quad (E.3)$$

$$\hat{F}_d = \frac{M_\infty}{Re} \begin{bmatrix} 0 \\ \hat{t}_{xy} \\ \hat{t}_{yy} \\ u_x \hat{t}_{xy} + u_y \hat{t}_{yy} + (\mu + \sigma_K \mu_T) \frac{\partial K}{\partial y} - \hat{q}_y \\ (\mu + \sigma_K \mu_T) \frac{\partial K}{\partial y} \\ (\mu + \sigma_\omega \mu_T) \frac{\partial \omega}{\partial y} \end{bmatrix}. \quad (E.4)$$

The definition of the array \hat{S} is:

$$\hat{S} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ P_K - D_K \\ P_\omega - D_\omega + CD_{term} \end{bmatrix} \quad (E.5)$$

where

$$P_K = \frac{M_\infty}{Re} \tau_{ij}^F \frac{\partial u_i}{\partial x_j}, \quad (E.6)$$

$$D_K = \beta^* \rho \omega K, \quad (E.7)$$

$$P_\omega = \frac{\gamma \omega}{K} \tau_{ij}^F \frac{\partial u_i}{\partial x_j}, \quad (E.8)$$

$$D_\omega = \beta \rho \omega^2, \quad (E.9)$$

$$CD_{term} = 2 \rho (1 - F_1) \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial K}{\partial x_j} \frac{\partial \omega}{\partial x_j}. \quad (E.10)$$

F) Quasi-linear form of the URANS and SST equations

The quasi-linear form of the 2D time-dependent RANS equations coupled to the two transport equations of the $K - \omega$ turbulence model in non-dimensional form is:

$$\frac{\partial \hat{U}}{\partial t} + \frac{\partial(\hat{\Phi}_c - \hat{\Phi}_d)}{\partial \hat{U}} \nabla \hat{U} = \hat{S}, \quad (\text{F.1})$$

or explicitly

$$\frac{\partial \hat{U}}{\partial t} + \tilde{A} \frac{\partial \hat{U}}{\partial x} + \tilde{B} \frac{\partial \hat{U}}{\partial y} = \hat{S}, \quad (\text{F.2})$$

where \hat{U} , $\hat{\Phi}_c$, $\hat{\Phi}_d$ and \hat{S} can be found from equations (2.1.10), (2.1.11), (2.1.14) and (E.5). The symbols \tilde{A} and \tilde{B} are the Jacobian matrices of the flux vector $\hat{\Phi}$ and they are defined as:

$$\tilde{A} = \frac{\partial(\hat{E}_c - \hat{E}_d)}{\partial \hat{U}} \quad \tilde{B} = \frac{\partial(\hat{F}_c - \hat{F}_d)}{\partial \hat{U}} \quad (\text{F.3})$$

where \hat{E}_c , \hat{E}_d , \hat{F}_c and \hat{F}_d can be found from equations (2.1.12), (2.1.13), (B.5) and (B.6), respectively.

G) Harmonic balance solution of an ODE example

The comparison between the HB solution obtained by solving system (3.2.17) and the numerical solution obtained by solving the ODE (3.2.8) with a MATLAB ODE solver, is presented. In order to solve the system of equations (3.2.17) one has to express equations (3.2.10), (3.2.11), (3.2.12) and (3.2.13) using $N_H = 1$:

$$x(t) = \hat{x}_0 + \hat{x}_1 \cos(\omega t) + \hat{x}_2 \sin(\omega t) \quad (\text{G.1})$$

$$\dot{x}(t) = -\omega \hat{x}_1 \sin(\omega t) + \omega \hat{x}_2 \cos(\omega t) \quad (\text{G.2})$$

$$\ddot{x}(t) = -\omega^2 \hat{x}_1 \cos(\omega t) - \omega^2 \hat{x}_2 \sin(\omega t) \quad (\text{G.3})$$

$$x^3(t) = \hat{r}_0 + \hat{r}_1 \cos(\omega t) + \hat{r}_2 \sin(\omega t). \quad (\text{G.4})$$

Solving the harmonic integrals apparent in equations (3.2.14), (3.2.15) and (3.2.16) with the same number of harmonics, one finds the following expressions of the \hat{r}_n coefficients:

$$\hat{r}_0 = \hat{x}_0^3 + \frac{3}{2} \hat{x}_0 \hat{x}_1^2 + \frac{3}{2} \hat{x}_0 \hat{x}_2^2 \quad (\text{G.5})$$

$$\hat{r}_1 = 3 \hat{x}_0^2 \hat{x}_1 + \frac{3}{4} (\hat{x}_1^3 + \hat{x}_1 \hat{x}_2^2) \quad (\text{G.6})$$

$$\hat{r}_2 = 3 \hat{x}_0^2 \hat{x}_2 + \frac{3}{4} (\hat{x}_2^3 + \hat{x}_2 \hat{x}_1^2) \quad (\text{G.7})$$

Inserting expressions (G.1) – (G.7) into equation (3.2.8) and balancing the harmonics of the same order, yields a system of three equations for the three unknown harmonic coefficients. Collecting the 0th harmonic terms, the equation can be written as:

$$\hat{x}_0 \left[1 + a \left(A_0^2 + \frac{3}{2} A_1^2 \right) \right] = 0 \quad (\text{G.8})$$

where

$$A_0 = \sqrt{\hat{x}_0^2}, \quad A_1 = \sqrt{\hat{x}_1^2 + \hat{x}_2^2}. \quad (\text{G.9})$$

For the term $\cos(\omega t)$ of the 1st harmonic the equation is:

$$(1 - \omega^2) \hat{x}_1 + 2\zeta \omega \hat{x}_2 + a \left(3A_0^2 + \frac{3}{4} A_1^2 \right) \hat{x}_1 = 0 \quad (\text{G.10})$$

and finally collecting the terms $\sin(\omega t)$ of the 1st harmonic the equation can be written as:

$$-2\zeta\omega\hat{x}_1 + (1 - \omega^2)\hat{x}_2 + a\left(3A_0^2 + \frac{3}{4}A_1^2\right)\hat{x}_2 = F. \quad (\text{G.11})$$

One now has to solve this system of three equations with three unknowns $(\hat{x}_0, \hat{x}_1, \hat{x}_2)$ to find the solution of the Duffing's oscillator problem. For the linear case where $a = 0$ the solution is equal to:

$$\hat{x}_0 = 0 \quad (\text{G.12})$$

$$\hat{x}_1 = -\frac{F\omega 2\zeta}{(1 - \omega^2)^2 + (\omega 2\zeta)^2} \quad (\text{G.13})$$

$$\hat{x}_2 = -\frac{F(1 - \omega^2)}{(1 - \omega^2)^2 + (\omega 2\zeta)^2} \quad (\text{G.14})$$

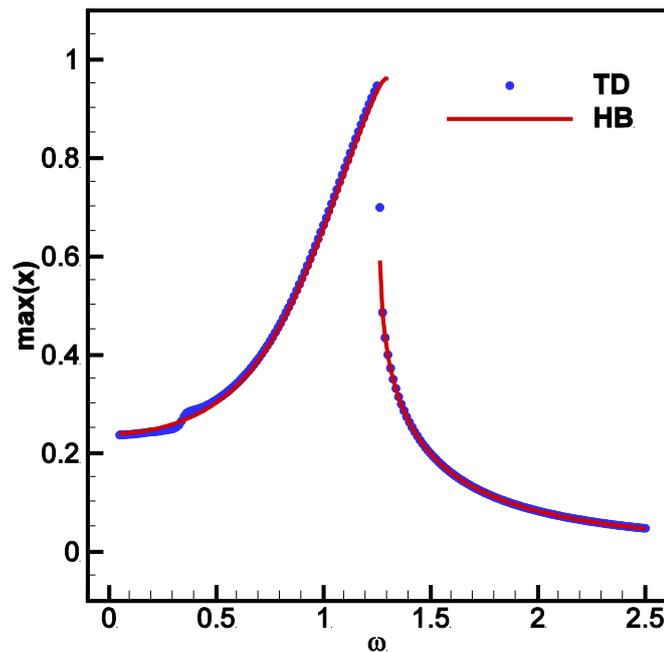


Figure G.1: Duffing's oscillator solution using a matlab solver and the harmonic balance approach

Figure G.1 presents a comparison analysis of the Duffing's oscillator between a solution obtained from a TD matlab solver and a solution obtained from the harmonic balance method. The red line indicates the harmonic balance solution whereas the blue symbols indicates the time-domain solution. The horizontal axis of figure G.1 represents different

values of frequency ω and the vertical axis represents the maximum value of the solution within one period. As one sees, the two different approaches are in excellent agreement. More details about the results of the Duffing's oscillator can be found in [101].

H) Numerical dissipation

In this section, the calculation of the numerical dissipation $|\tilde{K}_U|\delta\hat{U}$ will be explained. As highlighted by Equation (4.3.2), the numerical dissipation is proportional to $|\tilde{K}_U|\delta\hat{U}$, and this term can be written as:

$$\delta\hat{\Phi} = |\tilde{K}_U|\delta\hat{U} = \tilde{\mathbf{P}}|\tilde{\Lambda}|\tilde{\mathbf{P}}^{-1}\delta\hat{U} = \tilde{\mathbf{P}}|\tilde{\Lambda}|\delta\hat{W} \quad (\text{H.1})$$

where $\tilde{\mathbf{P}}$ is the matrix of the right eigenvectors of \tilde{K}_U (more specifically the columns of $\tilde{\mathbf{P}}$ are the right eigenvectors of \tilde{K}_U), $\tilde{\Lambda}$ is the matrix of the eigenvalues of \tilde{K}_U (more specifically $\tilde{\Lambda}$ is the diagonal and its nonzero entries are the eigenvalues of \tilde{K}_U), and $\delta\hat{W}$ are the characteristic variables, defined by $\delta\hat{W} = \tilde{\mathbf{P}}^{-1}\delta\hat{U}$. The symbol $\tilde{\mathbf{P}}^{-1}$ denotes the matrix of the left eigenvectors of \tilde{K}_U (more specifically the rows of $\tilde{\mathbf{P}}^{-1}$ are the left eigenvectors of \tilde{K}_U). Equation (H.1) highlights that the construction of the numerical dissipation only requires the calculation of the eigenvalues and the eigenvectors of \tilde{K}_U , the expression of which is:

$$\tilde{K}_U = \begin{array}{c} \begin{array}{cccccc} 0 & \mathbf{i} & \mathbf{j} & 0 & 0 & 0 \\ \psi q^2 \mathbf{i}/2 - u_i \mathbf{U}_n & \mathbf{U}_n - \varphi u_i \mathbf{i} & u_i \mathbf{j} - \psi u_j \mathbf{i} & \psi \mathbf{i} & -\psi \mathbf{i} & 0 \\ \psi q^2 \mathbf{j}/2 - u_j \mathbf{U}_n & u_j \mathbf{i} - \psi u_i \mathbf{j} & \mathbf{U}_n - \varphi u_j \mathbf{j} & \psi \mathbf{j} & -\psi \mathbf{j} & 0 \\ \psi q^2 \mathbf{U}_n/2 - H \mathbf{U}_n & H \mathbf{i} - \psi u_i \mathbf{U}_n & H \mathbf{j} - \psi u_j \mathbf{U}_n & \gamma \mathbf{U}_n & -\psi \mathbf{U}_n & 0 \\ -K \mathbf{U}_n & K \mathbf{i} & K \mathbf{j} & 0 & \mathbf{U}_n & 0 \\ -\omega \mathbf{U}_n & \omega \mathbf{i} & \omega \mathbf{j} & 0 & 0 & \mathbf{U}_n \end{array} \end{array}$$

where $q^2 = u_i^2 + u_j^2$, $\psi = \gamma - 1$, $\varphi = \gamma - 2$ and \mathbf{U}_n denotes the component of the flow velocity along the outwards face normal vector \mathbf{n} defined by:

$$\mathbf{U}_n = u_i \mathbf{i} + u_j \mathbf{j}. \quad (\text{H.2})$$

The direct calculation of the eigenmodes of $\tilde{\mathbf{K}}_U$ is a lengthy process, due to the dense structure of $\tilde{\mathbf{K}}_U$. The construction of the numerical dissipation can be simplified by considering the convective terms of the RANS and K-omega equations obtained by writing the equations in a differential form with respect to a new set of independent variables, namely the primitive variables \hat{V} defined as:

$$\frac{\partial \hat{V}}{\partial t} + \hat{A} \frac{\partial \hat{V}}{\partial x_i} + \hat{B} \frac{\partial \hat{V}}{\partial x_j} = \hat{Q} \quad (\text{H.3})$$

where

$$\hat{V} = \begin{bmatrix} \rho \\ u_i \\ u_j \\ p \\ K \\ \omega \end{bmatrix} \quad (\text{H.4})$$

with

$$\hat{A} = \begin{vmatrix} u_i & \rho & 0 & 0 & 0 & 0 \\ 0 & u_i & 0 & 1 & 0 & 0 \\ 0 & 0 & u_i & \rho & 0 & 0 \\ 0 & \rho c^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_i & u_i & 0 \\ 0 & 0 & 0 & 0 & 0 & u_i \end{vmatrix} \quad (\text{H.5})$$

$$\hat{B} = \begin{vmatrix} u_j & 0 & \rho & 0 & 0 & 0 \\ 0 & u_j & 0 & 0 & 0 & 0 \\ 0 & 0 & u_j & 1/\rho & 0 & 0 \\ 0 & 0 & \rho c^2 & u_j & 0 & 0 \\ 0 & 0 & 0 & 0 & u_j & 0 \\ 0 & 0 & 0 & 0 & 0 & u_j \end{vmatrix}$$

where c^2 denotes the square of the sound speed, which relates to the static temperature through the equation $c^2 = T$. The process of constructing the required numerical dissipation can be simplified by considering the Jacobian $\tilde{\mathbf{K}}_V$, given by:

$$\tilde{K}_V = \begin{vmatrix} \mathbf{U}_n & \rho \mathbf{i} & \rho \mathbf{j} & 0 & 0 & 0 \\ 0 & \mathbf{U}_n & 0 & \mathbf{i}/\rho & 0 & 0 \\ 0 & 0 & \mathbf{U}_n & \mathbf{j}/\rho & 0 & 0 \\ 0 & \rho c^2 \mathbf{i} & \rho c^2 \mathbf{j} & \mathbf{U}_n & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{U}_n & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{U}_n \end{vmatrix}. \quad (\text{H.6})$$

It can be easily shown that

$$\tilde{K}_U = \tilde{M} \tilde{K}_V \tilde{M}^{-1} \quad (\text{H.7})$$

where \tilde{M} is the Jacobian matrix of the transformation from primitive to conservative variables, and \tilde{M}^{-1} is its inverse. Their expressions are respectively:

$$\tilde{M} = \frac{\partial \hat{U}}{\partial \hat{V}} = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ u_i & \rho & 0 & 0 & 0 & 0 \\ u_j & 0 & \rho & 0 & 0 & 0 \\ q^2/2 + K & \rho u_i & \rho u_j & 1/(\gamma - 1) & \rho & 0 \\ K & 0 & 0 & 0 & \rho & 0 \\ \omega & 0 & 0 & 0 & 0 & \rho \end{vmatrix} \quad (\text{H.8})$$

$$\tilde{M}^{-1} = \frac{\partial \hat{V}}{\partial \hat{U}} = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -u_i/\rho & 1/\rho & 0 & 0 & 0 & 0 \\ -u_j/\rho & 0 & 1/\rho & 0 & 0 & 0 \\ (\gamma - 1)q^2/2 & -(\gamma - 1)u_i & -(\gamma - 1)u_j & \gamma - 1 & 1 & 0 \\ -K/\rho & 0 & 0 & 0 & 1/\rho & 0 \\ -\omega/\rho & 0 & 0 & 0 & 0 & 1/\rho \end{vmatrix} \quad (\text{H.9})$$

Equation (H.7) defines a similarity transformation from \tilde{K}_U to \tilde{K}_V and vice versa. The matrices \tilde{K}_U and \tilde{K}_V are similar, and this implies that they have the same eigenvalues and also that their eigenvectors are related through the transformation \tilde{M} . By inserting equation (H.7) into equation (H.1) one finds:

$$\begin{aligned} \delta \hat{\Phi} &= |\tilde{K}_U| \delta \hat{U} = \tilde{M} |\tilde{K}_V| \tilde{M}^{-1} \delta \hat{U} = \tilde{M} \tilde{L} |\tilde{\Lambda}| \tilde{L}^{-1} \tilde{M}^{-1} \delta \hat{U} \\ &= \tilde{M} \tilde{L} |\tilde{\Lambda}| \tilde{L}^{-1} \delta \hat{V} \end{aligned} \quad (\text{H.10})$$

where the columns of \tilde{L} are the right eigenvectors of \tilde{K}_V and the rows of \tilde{L}^{-1} are the left

eigenvectors of \tilde{K}_V . By comparing equations (H.1) and (H.10), one finds that:

$$\tilde{P} = \tilde{M} \tilde{L} \quad (\text{H.11})$$

and

$$\delta \hat{W} = \tilde{L}^{-1} \delta \hat{V}. \quad (\text{H.12})$$

Given that the calculation of the eigenmodes of \tilde{K}_V is simpler than the calculation of the eigenmodes of \tilde{K}_U , the matrix of the right eigenvalues \tilde{P} and the characteristic variables \hat{W} which appear in the equation (H.1), are determined by means of equations (H.11) and (H.12) respectively, namely by using the matrix of the left eigenvalues \tilde{L} and the right eigenvectors \tilde{L}^{-1} . The eigenvalues of \tilde{K}_V and \tilde{K}_U are

$$\lambda_1 = \lambda_2 = \lambda_5 = \lambda_6 = \mathbf{U}_n \mathbf{n} \quad (\text{H.13})$$

$$\lambda_3 = \mathbf{U}_n \mathbf{n} + c |\mathbf{n}| \quad (\text{H.14})$$

$$\lambda_4 = \mathbf{U}_n \mathbf{n} - c |\mathbf{n}|. \quad (\text{H.15})$$

The eigenvalue $\mathbf{U}_n \mathbf{n}$ yields the solution:

$$[k_1 \quad k_2 \quad -k_2 \mathbf{i}/\mathbf{j} \quad -k_1/c^2 \quad k_3 \quad k_4] \quad (\text{H.16})$$

where the constants k_i can be chosen arbitrarily. Making a typical choice for these constants, yields the following 4 left eigenvectors:

$$[1 \quad 0 \quad 0 \quad -1/c^2 \quad 0 \quad 0] \quad (\text{H.17})$$

$$[0 \quad \rho \mathbf{j} \quad -\rho \mathbf{i} \quad 0 \quad 0 \quad 0] \quad (\text{H.18})$$

$$[0 \quad 0 \quad 0 \quad 0 \quad \rho \quad 0] \quad (\text{H.19})$$

$$[0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \rho]. \quad (\text{H.20})$$

The eigenvalues $\mathbf{U}_n \mathbf{n} + c|\mathbf{n}|$ and $\mathbf{U}_n \mathbf{n} - c|\mathbf{n}|$ yield respectively the eigenvectors:

$$[0 \quad \rho \mathbf{i}/2c \quad \rho \mathbf{j}/2c \quad 1/2c^2 \quad 0 \quad 0] \quad (\text{H.21})$$

$$[0 \quad -\rho \mathbf{i}/2c \quad -\rho \mathbf{j}/2c \quad -1/2c^2 \quad 0 \quad 0]. \quad (\text{H.22})$$

The sought matrix of the left eigenvectors is thus:

$$\tilde{\mathbf{L}}^{-1} = \begin{vmatrix} 1 & 0 & 0 & -1/c^2 & 0 & 0 \\ 0 & \rho \mathbf{j} & -\rho \mathbf{i} & 0 & 0 & 0 \\ 0 & \rho \mathbf{i}/2c & \rho \mathbf{j}/2c & 1/2c^2 & 0 & 0 \\ 0 & -\rho \mathbf{i}/2c & -\rho \mathbf{j}/2c & 1/2c^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \rho & 0 \\ 0 & 0 & 0 & 0 & 0 & \rho \end{vmatrix} \quad (\text{H.23})$$

and its inverse is:

$$\tilde{\mathbf{L}} = \begin{vmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & \mathbf{j}/\rho & \mathbf{ci}/\rho & -\mathbf{ci}/\rho & 0 & 0 \\ 0 & -\mathbf{i}/\rho & \mathbf{cj}/\rho & -\mathbf{cj}/\rho & 0 & 0 \\ 0 & 0 & c^2 & c^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/\rho & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/\rho \end{vmatrix}. \quad (\text{H.24})$$

The matrix P with the right eigenvectors of $\tilde{\mathbf{K}}_U$ is computed by means of equation (H.11)

$$\tilde{P} = \tilde{M} \tilde{L} = \begin{vmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ u_i & \mathbf{j} & u_i + \mathbf{ci} & u_i - \mathbf{ci} & 0 & 0 \\ u_j & -\mathbf{i} & u_j + \mathbf{cj} & u_j - \mathbf{cj} & 0 & 0 \\ q^2/2 + K & \mathbf{U}_t & H + \mathbf{U}_n c & H - \mathbf{U}_n c & 1 & 0 \\ K & 0 & K & K & 1 & 0 \\ \omega & 0 & \omega & \omega & 0 & 1 \end{vmatrix} \quad (\text{H.25})$$

where

$$\mathbf{U}_t = u_i \mathbf{j} - u_j \mathbf{i} \quad (\text{H.26})$$

$$H = \frac{u_i u_i}{2} + K + \frac{c^2}{\gamma - 1}. \quad (\text{H.27})$$

The characteristic variables $\delta\widehat{W}$ can be computed by means of equations (H.12). Their expressions are:

$$\delta W_1 = \delta\rho - \frac{1}{c^2} \delta p \quad (\text{H.28})$$

$$\delta W_2 = \rho j \delta u_i - \rho i \delta u_j = \rho \delta \mathbf{U}_t \quad (\text{H.29})$$

$$\delta W_3 = \frac{\rho i}{2c} \delta u_i + \frac{\rho j}{2c} \delta u_j + \frac{1}{2c^2} \delta p = \frac{\delta p}{2c^2} + \frac{\rho \delta \mathbf{U}_n}{2c} \quad (\text{H.30})$$

$$\delta W_4 = -\frac{\rho i}{2c} \delta u_i - \frac{\rho j}{2c} \delta u_j + \frac{1}{2c^2} \delta p = \frac{\delta p}{2c^2} - \frac{\rho \delta \mathbf{U}_n}{2c} \quad (\text{H.31})$$

$$\delta W_5 = \rho \delta K \quad (\text{H.32})$$

$$\delta W_6 = \rho \delta \omega. \quad (\text{H.33})$$

Equation (H.1) can also be written as:

$$\delta\widehat{\Phi} = P |\Lambda| \delta\widehat{W} = \sum_{k=1}^6 |\lambda_k| \delta W_k r_k \quad (\text{H.34})$$

where δW_k is the k^{th} component of $\delta\widehat{W}$ and r_k is the k^{th} right eigenvector of \widetilde{K}_U , which is also the k^{th} column of P . The expanded counterpart of equation (H.35) provides the sought flux differences:

$$\begin{aligned}
 \delta\hat{\Phi} &= \sum_{k=1}^6 |\lambda_k| \delta W_k r_k \\
 &= |\lambda_1| \delta W_1 \begin{vmatrix} 1 \\ u_i \\ u_j \\ \frac{q^2}{2} + K \\ K \\ \omega \end{vmatrix} + |\lambda_2| \delta W_2 \begin{vmatrix} 0 \\ \mathbf{j} \\ -\mathbf{i} \\ \mathbf{U}_t \\ 0 \\ 0 \end{vmatrix} \\
 &+ |\lambda_3| \delta W_3 \begin{vmatrix} 1 \\ u_i + c\mathbf{i} \\ u_j + c\mathbf{j} \\ H + \mathbf{U}_n c \\ K \\ \omega \end{vmatrix} + |\lambda_4| \delta W_4 \begin{vmatrix} 1 \\ u_i - c\mathbf{i} \\ u_j - c\mathbf{j} \\ H - \mathbf{U}_n c \\ K \\ \omega \end{vmatrix} \\
 &+ |\lambda_5| \delta W_5 \begin{vmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{vmatrix} + |\lambda_6| \delta W_6 \begin{vmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{vmatrix}.
 \end{aligned} \tag{H.35}$$

In order to maximize the computational efficiency of the numerical implementation, the flux differences provided by equation (H.35) are computed as described below. Firstly, one computes a set of intermediate variables defined as:

$$\alpha_1 = |\lambda_1| \left(\delta\rho - \frac{\delta p}{c^2} \right) \tag{H.36}$$

$$\alpha_2 = |\lambda_2| \rho \tag{H.37}$$

$$\alpha_3 = |\lambda_3| \left(\frac{\delta p}{c^2} + \frac{\rho \delta \mathbf{U}_n}{c} \right) / 2 \tag{H.38}$$

$$\alpha_4 = |\lambda_4| \left(\frac{\delta p}{c^2} - \frac{\rho \delta \mathbf{U}_n}{c} \right) / 2. \tag{H.39}$$

The components of $\delta\hat{\Phi}$ are then computed by means of the expressions:

$$\delta\hat{\Phi}_1 = a_1 + a_3 + a_4 \tag{H.40}$$

$$\delta\hat{\Phi}_2 = a_1 u_i + a_2 \delta \mathbf{U}_t \mathbf{j} + a_3 (u_i + c \mathbf{i}) + a_4 (u_i - c \mathbf{i}) \tag{H.41}$$

$$\delta\hat{\Phi}_3 = a_1 u_j - a_2 \delta \mathbf{U}_t \mathbf{i} + a_3 (u_j + c \mathbf{j}) + a_4 (u_j - c \mathbf{j}) \tag{H.42}$$

$$\begin{aligned} \delta\Phi_4 = a_1 \left(\frac{q^2}{2} + K \right) + a_2 \mathbf{U}_t \delta\mathbf{U}_t + a_3 (H + c \mathbf{U}_n) \\ + a_4 (H - c \mathbf{U}_n) + a_2 \delta K \end{aligned} \quad (\text{H.43})$$

$$\delta\Phi_5 = a_1 K + a_3 K + a_4 K + \alpha_2 \delta K \quad (\text{H.44})$$

$$\delta\Phi_6 = a_1 \omega + a_3 \omega + a_4 \omega + \alpha_2 \delta\omega. \quad (\text{H.45})$$

When dealing with moving-grid problems, the convective fluxes include the contribution associated with the velocity of the cell boundaries \mathbf{v}_b . In this circumstance, the array of convective fluxes $\hat{\Phi}_c$ is given by equation (2.1.19). The convective fluxes at the boundary of each cell become:

$$\hat{\Phi}_c = (\hat{\Phi}_{c,f} \cdot \mathbf{n}) dS = (\hat{\mathbf{E}}_c \mathbf{i} + \hat{\mathbf{F}}_c \mathbf{j} - \hat{U} v_{bn}) dS \quad (\text{H.46})$$

with

$$v_{bn} = \mathbf{v}_b \cdot \mathbf{n}. \quad (\text{H.47})$$

The expression of the numerical flux at each cell boundary is formally identical to equation (4.3.2). The only practical differences are that a) the analytical fluxes $\hat{\Phi}_{c,f}(\hat{U}_L)$ and $\hat{\Phi}_{c,f}(\hat{U}_R)$ include the flux contribution associated with the boundary velocity and b) the expression of the generalized flux Jacobian evaluated at the face under analysis becomes:

$$\widetilde{K}'_U = \frac{\partial \hat{\Phi}_c}{\partial \hat{U}} = \left(\frac{\partial \hat{\mathbf{E}}_c}{\partial \hat{U}} \mathbf{i} + \frac{\partial \hat{\mathbf{F}}_c}{\partial \hat{U}} \mathbf{j} \right) - I v_{bn} = \widetilde{K}_U - I v_{bn}. \quad (\text{H.48})$$

Since the operator \widetilde{K}'_U and \widetilde{K}_U differs only by the diagonal term, they share the same eigenvalues. Their eigenvectors differ only because of the constant offset v_{bn} . Specifically, the eigenvalues of \widetilde{K}'_U are:

$$\lambda_1 = \lambda_2 = \lambda_5 = \lambda_6 = (\mathbf{U}_n - \mathbf{v}_b) \cdot \mathbf{n} \quad (\text{H.49})$$

$$\lambda_3 = (\mathbf{U}_n - \mathbf{v}_b) \mathbf{n} + c |\mathbf{n}| \quad (\text{H.50})$$

$$\lambda_4 = (\mathbf{U}_n - \mathbf{v}_b) \mathbf{n} - c |\mathbf{n}|. \quad (\text{H.51})$$

Consequently, the expression of the flux differences $\delta\widehat{\Phi}$ for problems with moving grids is formally identical to expressions (H.40) - (H.45) and the only difference with respect to problems with motionless grids is the appearance of the boundary velocity term in the eigenvalues of \widetilde{K}'_U .

I) Diffusive fluxes

As explained in chapter 4.5, we introduce a mapping function (4.5.8), (4.5.9) between the Cartesian and the curvilinear coordinates system (figure 4.5) where the spacing $\Delta\xi$ and $\Delta\eta$ are taken to be 1. Using the chain rule, the velocity derivatives can be expressed as:

$$\frac{\partial u_x}{\partial x} = \frac{\partial u_x}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial u_x}{\partial \eta} \frac{\partial \eta}{\partial x} \quad (\text{I.1})$$

$$\frac{\partial u_x}{\partial y} = \frac{\partial u_x}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial u_x}{\partial \eta} \frac{\partial \eta}{\partial y} \quad (\text{I.2})$$

$$\frac{\partial u_y}{\partial x} = \frac{\partial u_y}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial u_y}{\partial \eta} \frac{\partial \eta}{\partial x} \quad (\text{I.3})$$

$$\frac{\partial u_y}{\partial y} = \frac{\partial u_y}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial u_y}{\partial \eta} \frac{\partial \eta}{\partial y}. \quad (\text{I.4})$$

The ξ - and η - derivatives of u_i are computed with second order centred finite-differences. Therefore, on face $i + 1/2, j$ (AB) we have:

$$\frac{\partial u_x}{\partial \xi} = \frac{u_x(i + 1, j) - u_x(i, j)}{2 \Delta\xi/2} = u_x(i + 1, j) - u_x(i, j). \quad (\text{I.5})$$

The derivative $\frac{\partial u_x}{\partial \eta}$ on the same face is instead:

$$\frac{\partial u_x}{\partial \eta} = \frac{u_{x_B} - u_{x_A}}{2 \Delta \eta / 2} = u_{x_B} - u_{x_A}. \quad (1.6)$$

The required values u_{x_A} and u_{x_B} , however, are not readily available, as they are not values at cell centres. Such values have to be expressed as functions of available cell centre data. For this reason, these two variables are evaluated using the expressions:

$$\begin{aligned} u_{x_A} &= \frac{u_x(i, j-1) + u_x(i+1, j-1) + u_x(i+1, j) + u_x(i, j)}{4} \end{aligned} \quad (1.7)$$

$$\begin{aligned} u_{x_B} &= \frac{u_x(i, j) + u_x(i+1, j) + u_x(i+1, j+1) + u_x(i, j+1)}{4} \end{aligned} \quad (1.8)$$

which instead use only cell centre data. The same approach is used for calculating $\frac{\partial u_y}{\partial \xi}$ and $\frac{\partial u_y}{\partial \eta}$ on the face $i + 1/2, j$.

On the faces having normal along the j family of grid lines, the velocity derivatives in the η direction are easily computed using cell centre data. For example, on the face $i, j + 1/2$, one has:

$$\frac{\partial u_x}{\partial \eta} = \frac{u_x(i, j+1) - u_x(i, j)}{2 \Delta \eta / 2} = u_x(i, j+1) - u_x(i, j) \quad (1.9)$$

On these faces, however, the ξ derivatives cannot be determined directly using cell centre data. For example, the ξ derivative of u_x is:

$$\frac{\partial u_x}{\partial \xi} = \frac{u_{x_B} - u_{x_C}}{2 \Delta \xi / 2} = u_{x_B} - u_{x_C} \quad (1.10)$$

But the values of u_{x_C} and u_{x_B} are not readily available, as they do not refer to cell centres. As seen above, the solution is to express these values as function of values at surrounding cell centres, that is:

would need to be computed from the coordinates of the given grid. In alternative to this, the derivatives of the x and y coordinates at the face $i + 1/2, j$ in the ξ direction are computed by averaging ξ derivatives at faces with normal in the η direction, which are instead computed using only coordinates of grid vertices. It thus follows that:

$$\frac{\partial y}{\partial \xi} = \frac{1}{4} \left(\frac{\partial y}{\partial \xi}_{i,j+\frac{1}{2}} + \frac{\partial y}{\partial \xi}_{i+1,j+\frac{1}{2}} + \frac{\partial y}{\partial \xi}_{i,j-\frac{1}{2}} + \frac{\partial y}{\partial \xi}_{i+1,j-\frac{1}{2}} \right) \quad (\text{I.17})$$

Where

$$\frac{\partial y}{\partial \xi}_{i,j+\frac{1}{2}} = y_{i+\frac{1}{2},j+\frac{1}{2}} - y_{i-\frac{1}{2},j+\frac{1}{2}} \quad (\text{I.18})$$

$$\frac{\partial y}{\partial \xi}_{i+1,j+\frac{1}{2}} = y_{i+\frac{3}{2},j+\frac{1}{2}} - y_{i+\frac{1}{2},j+\frac{1}{2}} \quad (\text{I.19})$$

$$\frac{\partial y}{\partial \xi}_{i,j-\frac{1}{2}} = y_{i+\frac{1}{2},j-\frac{1}{2}} - y_{i-\frac{1}{2},j-\frac{1}{2}} \quad (\text{I.20})$$

$$\frac{\partial y}{\partial \xi}_{i+1,j-\frac{1}{2}} = y_{i+\frac{3}{2},j-\frac{1}{2}} - y_{i+\frac{1}{2},j-\frac{1}{2}} \quad (\text{I.21})$$

$$\frac{\partial x}{\partial \xi} = \frac{1}{4} \left(\frac{\partial x}{\partial \xi}_{i,j+\frac{1}{2}} + \frac{\partial x}{\partial \xi}_{i+1,j+\frac{1}{2}} + \frac{\partial x}{\partial \xi}_{i,j-\frac{1}{2}} + \frac{\partial x}{\partial \xi}_{i+1,j-\frac{1}{2}} \right) \quad (\text{I.22})$$

where

$$\frac{\partial x}{\partial \xi}_{i,j+\frac{1}{2}} = x_{i+\frac{1}{2},j+\frac{1}{2}} - x_{i-\frac{1}{2},j+\frac{1}{2}} \quad (\text{I.23})$$

$$\frac{\partial x}{\partial \xi}_{i+1,j+\frac{1}{2}} = x_{i+\frac{3}{2},j+\frac{1}{2}} - x_{i+\frac{1}{2},j+\frac{1}{2}} \quad (\text{I.24})$$

$$\frac{\partial x}{\partial \xi}_{i,j-\frac{1}{2}} = x_{i+\frac{1}{2},j-\frac{1}{2}} - x_{i-\frac{1}{2},j-\frac{1}{2}} \quad (\text{I.25})$$

$$\frac{\partial x}{\partial \xi}_{i+1,j-\frac{1}{2}} = x_{i+\frac{3}{2},j-\frac{1}{2}} - x_{i+\frac{1}{2},j-\frac{1}{2}} \quad (\text{I.26})$$

The same approach explained above is adopted to calculate the metrics on the face $i, j +$

1/2.

As for the term (4.5.10) the expression is:

$$\begin{aligned}
 & [(u_x \hat{t}_{xx} n_x + u_y \hat{t}_{xy} n_x + u_x \hat{t}_{xy} n_y + u_y \hat{t}_{yy} n_y) \Delta S]_{i+1/2,j} \\
 & + [(u_x \hat{t}_{xx} n_x + u_y \hat{t}_{xy} n_x + u_x \hat{t}_{xy} n_y + u_y \hat{t}_{yy} n_y) \Delta S]_i \\
 & + [(u_x \hat{t}_{xx} n_x + u_y \hat{t}_{xy} n_x + u_x \hat{t}_{xy} n_y + u_y \hat{t}_{yy} n_y) \Delta S]_i \\
 & + [(u_x \hat{t}_{xx} n_x + u_y \hat{t}_{xy} n_x + u_x \hat{t}_{xy} n_y + u_y \hat{t}_{yy} n_y) \Delta S]_i
 \end{aligned} \tag{I.27}$$

The procedure followed to discretize each term of equation (I.27) is the same explained for the case of the two components of the momentum equation. The only addition in the case of the diffusive flux of the energy equation is the need to calculate the velocity components on the cell faces. These values are computed by using face averages based on the cell centre values of the two cells adjacent to the considered face.

The general expression of the discrete representation of the heat term (4.5.11) on the cell face is:

$$(\hat{q}_x n_x + \hat{q}_y n_y) \Delta S \tag{I.28}$$

From equation (D.13) we have:

$$\hat{q}_x = - \frac{1}{\gamma - 1} \left[\frac{\mu}{P_r} + \frac{\mu_T}{P_{rT}} \right] \frac{\partial T}{\partial x} \tag{I.29}$$

$$\hat{q}_y = - \frac{1}{\gamma - 1} \left[\frac{\mu}{P_r} + \frac{\mu_T}{P_{rT}} \right] \frac{\partial T}{\partial y} \tag{I.30}$$

The derivatives of the temperature at the face centres can also be written using the chain rule to combine the temperature derivatives with respect to the generalised curvilinear coordinates and the grid metrics:

$$\frac{\partial T}{\partial x} = \frac{\partial T}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial T}{\partial \eta} \frac{\partial \eta}{\partial x} \tag{I.31}$$

$$\frac{\partial T}{\partial y} = \frac{\partial T}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial T}{\partial \eta} \frac{\partial \eta}{\partial y} \tag{I.32}$$

Rather than calculating the derivatives of the temperature with respect to the generalised curvilinear coordinates, however, an alternative approach has been used: The temperature difference across the faces has been expressed in terms of the density, pressure and temperature at the face centre, and the density and pressure variations across the face. The relationship among these quantities,

$$dT = T \left(\frac{dp}{p} - \frac{d\rho}{\rho} \right) \quad (\text{I.33})$$

has been obtained by differentiating the perfect gas equation of state. As a result, the temperature derivatives are expressed as:

$$\frac{dT}{dx} = T \left(\frac{1}{p} \frac{dp}{dx} - \frac{1}{\rho} \frac{d\rho}{dx} \right) \quad (\text{I.34})$$

$$\frac{dT}{dy} = T \left(\frac{1}{p} \frac{dp}{dy} - \frac{1}{\rho} \frac{d\rho}{dy} \right) \quad (\text{I.35})$$

The calculation of the two components of the temperature gradient in the CFD code used in this study is based on equations (I.34) and (I.35). The use of these equations requires the calculation of the Cartesian derivatives of both ρ and p , rather than the calculation of the Cartesian derivatives of T . The procedure used to calculate numerically the derivatives of all three variables is the same as that used for calculating the Cartesian derivatives of the velocity components. It has been preferred to calculate the temperature gradient using equation (I.34) and (I.35) rather than equations (I.31) and (I.32) because the overall computational cost of the former approach in the CFD code used in this study is lower than the computational cost of the latter approach.

J) Hydrokinetic turbine: turbulent and laminar C_f comparison

In this section a skin friction coefficient comparison between the laminar and the turbulent regime for the wing section NACA0015 characterized by a high efficiency of the energy extraction (case A), is presented. Figures J.1 and J.2 present the skin friction coefficients in the laminar and turbulent regime. Each figure includes 6 subplots. Subplots on the left hand side represent the skin friction coefficients obtained from a laminar regime whereas subplots on the right hand side of the same figures represent the skin friction coefficients obtained from a turbulent regime. Each subplot of figures J.1 and J.2 is labelled at its top left hand side with a number which denotes the wing configuration at 12.5%, 18.7%, 25.0% for figure J.1 and the wing configuration at 34.4%, 45.3% and 54.7% for figure J.2. The horizontal axis of each subplot for both figures J.1 and J.2 represents the x-coordinates of the airfoil non-dimensionalized by the chord while the vertical axis represents the skin friction coefficient C_f . All skin friction coefficient results in the turbulent regime are computed using Wilcox's wall boundary condition described by equation (2.6.1) with $S_R = 2500$.

One sees that the order of magnitude of the skin friction coefficients between the turbulent and the laminar regime is not the same. This happens because in the turbulent regime the Reynolds number is substantial larger than the one used for the calculation of the laminar regime. As a consequence, in the case of turbulence regime, the boundary layer is stronger and thinner, and for this reason one sees that the values of skin friction coefficients in turbulent regime are smaller by one order of magnitude compared to the laminar counterpart. The subplot on the left hand side of figure J.1, labelled as 1, reports a small separation at the rear of the lower airfoil side as the values of the skin friction coefficient are almost zero at these positions. This can also be confirmed in figure 8.6.

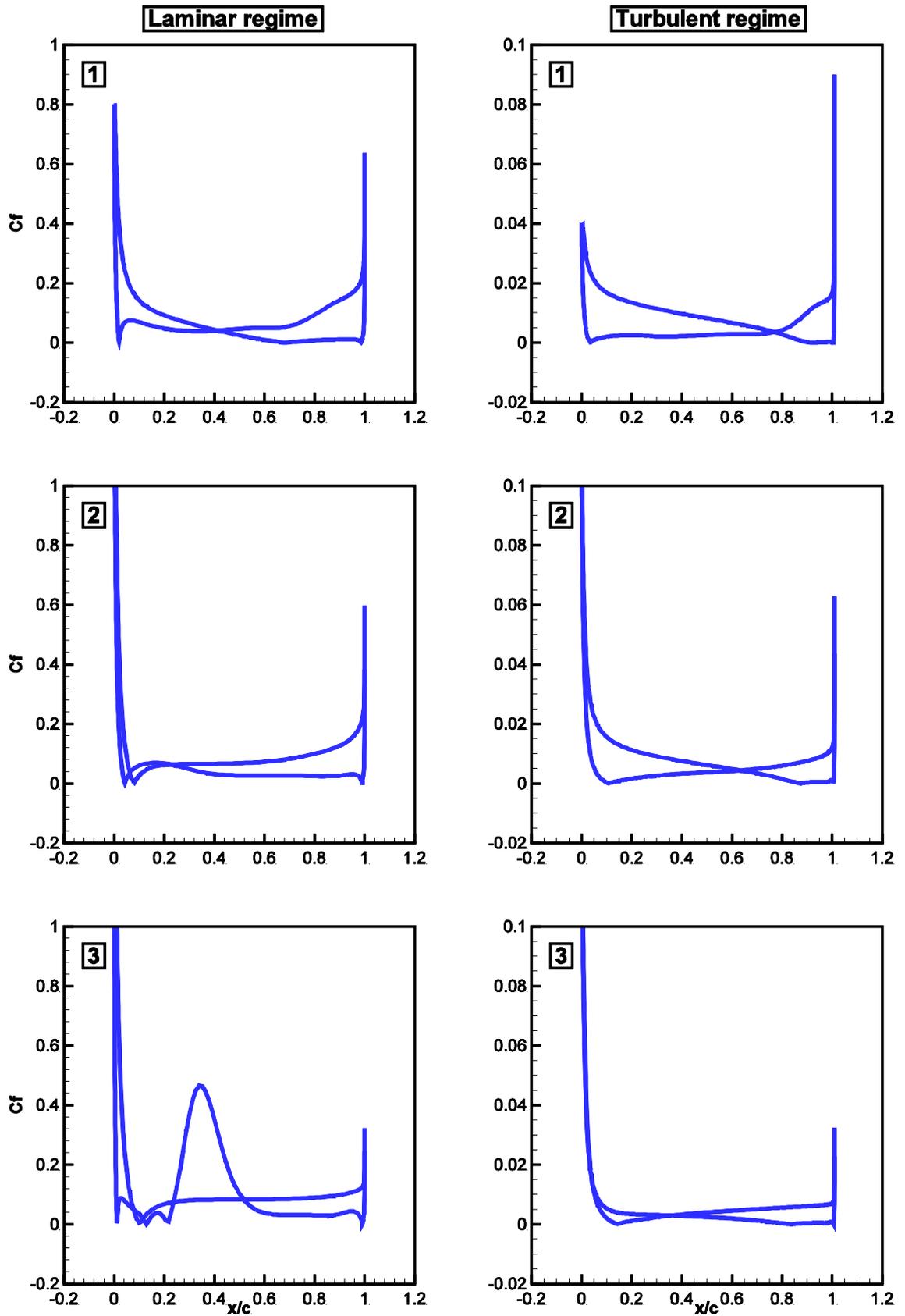


Figure J.1: skin friction coefficients for wing positions labelled 1-3

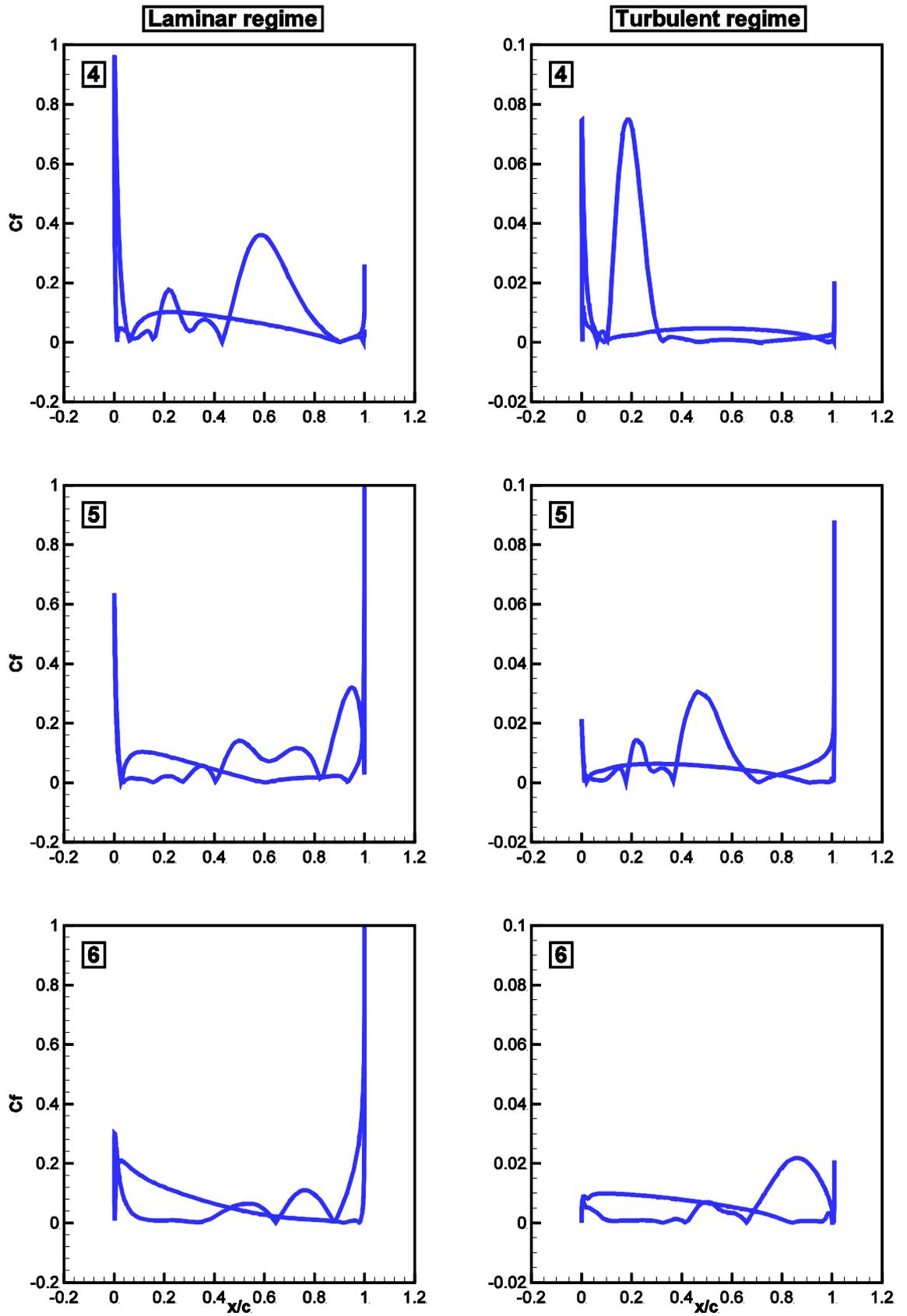


Figure J.2: skin friction coefficients for wing positions labelled 4-6

On the other hand, no separation occurs at the same wing configuration for the turbulent regime. One sees in subplot labelled as 2 of figure J.1 that the separation has rapidly grown for laminar regime and it has passed the mid-chord position. Moreover the position of the flow separation that occurs at the LE, has slightly changed compared to position 1, which is due to the airfoil motion. In addition, for the turbulent regime, one sees a minor separation, which occurs at the TE of the airfoil. Moving on to the third wing configuration, one sees that the values of the skin friction coefficient for the laminar regime increase rapidly at 20% of the chord and decrease again to normal values at 40% of the chord. This is due to the fact that LEVS begin. At position 4 of figure J.2, the laminar LEVS has already reached the mid chord position, as the values of the skin friction coefficient rapidly increase at 40% and decrease at 60% of the chord, respectively. Also the skin friction coefficient reports the same behaviour as described above, for position close to the LE of the airfoil which is due to the fact that a second vortex is generated. For the turbulent counterpart, a turbulent LEVS commences and can be easily located in subplot 4. On the rest of the subplots the values of the skin friction coefficient are decreasing until the vortex leaves the airfoil.

Bibliography

- [1] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone and J. C. Phillips, "GPU Computing," *Proceeding of the IEEE*, vol. 96, no. 5, pp. 879-899, 2008.
- [2] J. Blazek, *Computational Fluid Dynamics: Principles and Applications*, Elsevier, 2001.
- [3] P. K. Kundu and I. M. Cohen, *Fluid Mechanics*, second edition, Elsevier, 2002.
- [4] C. Hirsch, *Numerical Computation of Internal and External Flows*, John Wiley & Sons, 1994.
- [5] F. Zahle, C. Bak, S. Guntur, N. N. Sorensen and N. Troldborg, "Comprehensive Aerodynamic Analysis of a 10 MW Wind Turbine Rotor Using 3D CFD," in *AIAA SciTech / 32nd ASME Wind Energy Symposium*, National Harbor, Maryland, 2014.
- [6] Y. X. Xiao, C. J. Zeng, J. Zhang, Z. G. Yan and Z. W. Wang, "Numerical Analysis of the Bucket Surface Roughness Effects in Pelton Turbine," in *6th International Conference on Pumps and Fans with Compressors and Wind Turbines*, 2013.
- [7] B. Walther and S. Nadarajah, "Constrained Adjoint-Based Aerodynamic Shape Optimazation of a Single-Stage Transonic Compressor," in *ASME Turbo Expo*, Copenhagen, Denmark, 2012.
- [8] E. J. Nielsen and W. K. Anderson, "Aerodynamic Design Optimization on Unstructured Meshes Using the Navier-Stokes Equations," *AIAA Journal*, vol. 37, no. 11, pp. 1411-1419, 1999.
- [9] C. S. Johnson and G. N. Barakos, "A Framework for the Optimisation of a BERP-like Blade," in *AIAA Conference*, Grapeville, Texas, 2013.
- [10] J. Reuther, A. Jameson, J. J. Alonso, M. J. Rimlinger and D. Saunders, "Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers," in *AIAA Conference*, Reno, 1997.
- [11] H. Ahmed and S. Chacko, "Computational Optimization of Vehicle Aerodynamics," in *23th International DAAAM Symposium*, Vienna, Austria, 2012.
- [12] B. K. Kestner, J. S. Schutte, J. C. Gladin and D. N. Mavris, "Ultra High Bypass Ratio Engine Sizing and Cycle Selection Study for a Subsonic Commercial Aircraft in the N+2 Timeframe," in *ASME Turbo Expo*, Vancouver, Canada, 2011.
- [13] J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*, Springer, 2002.
- [14] A. W. Cook and J. J. Riley, "Direct Numerical Simulation of a Turbulent Reactive Plume on a Parallel Computer," *J. of Computational Physics*, vol. 129, pp. 263-283, 1996.
- [15] D. C. Wilcox, *Turbulence Modeling for CFD*, DCW Industries, 1994.

- [16] B. S. Baldwin and H. Lomax, "Thin-Layer Approximation and Algebraic Model for Separated Turbulent Flows," in *AIAA aerospace sciences meeting*, Huntsville, Alabama, 1978.
- [17] P. R. Spalart and S. R. Allmaras, "A One-Equation turbulence Model for Aerodynamic Flows," *La Rech Aerospatiabe*, vol. 1, pp. 5-21, 1994.
- [18] W. P. Jones and B. E. Launder, "The Calculation of Low-Reynolds-Number Phenomena with a Two-Equation Model of Turbulence," *J.Heat Mass Transfer*, vol. 16, pp. 1119-1130, 1973.
- [19] D. C. Wilcox, "Reassessment of the scale-determining equation for advanced turbulence," *AIAA Journal*, vol. 26, no. 11, pp. 1299-1310, 1988.
- [20] B. E. Launder and G. J. Reece, "Progress in the Development of a Reynolds-Stress Turbulence Closure," *J. Fluid Mechanics*, vol. 68, pp. 537-566, 1975.
- [21] T. D. Dreeben and S. B. Pope, "Probability Density Function and Reynolds-Stress Modeling of Near-Wall Turbulent Flows," *Journal of Physics Fluids*, vol. 9, no. 1, pp. 154-163, 1996.
- [22] U. Piomelli, "Large-Eddy Simulation: Present State and Future Perspectives," in *AIAA paper*, 98-0534, 1998.
- [23] J. Johansen, N. N. Sorensen, J. A. Michelsen and S. Schreck, "Detached-Eddy Simulation of Flow Around the NREL Phase-VI Blade," *Wind Energy*, vol. 5, no. 2-3, pp. 185-197, 2002.
- [24] P. R. Spalart, W.-H. Jou, M. Strelets and S. R. Allmaras, "Comments on the Feasibility of LES for Wings, and on a Hybrid RANS/LES Approach," *Proceeding of the First AFOSR International Conference on DNS/LES*, 1997.
- [25] N. V. Nikitin, F. Nicoud, B. Wasistho, K. D. Squires and P. R. Spalart, "An Approach to Wall Modelling in Large-Eddy Simulations," *Journal of Physics and Fluids*, vol. 12, no. 7, pp. 1629-1632, 2000.
- [26] A. Bechmann and N. N. Sorensen, "Hybrid RANS/LES Method for Wind Flow over Complex Terrain," *Wind Energy*, vol. 13, pp. 36-50, 2010.
- [27] H. S. Im and G. C. Zha, "Delayed Detached Eddy Simulation of a Stall Flow Over NACA0012 Airfoil Using High Order Schemes," in *AIAA Meeting*, Oriando, Florida, 2011.
- [28] P. R. Spalart, S. Deck, K. D. Shur, K. D. Squires, M. K. Strelets and A. Travin, "A New Version of Detached-Eddy Simulation, Resistant to Ambiguous Grid Densities," *Theor. Comput. Fluid Dyn.*, vol. 20, pp. 181-195, 2006.
- [29] U. Piomelli, E. Balaras, H. Pasinato, K. Squires and P. R. Spalart, "The Inner-Outer Layer Interface in Large-Eddy Simulations with Wall-Layer Models," *Journal of Heat and Fluid Flows*, vol. 24, pp. 538-550, 2003.
- [30] N. N. Sorensen, A. Bechmann and F. Zahle, "3D CFD Computations of Transitional Flows using DES and a Correlation Based Transition Model," *Wind Energy*, vol. 14, pp. 77-90, 2011.

- [31] K. W. Ng, W. H. Lam and K. C. Ng, "2002-2012: 10 Years of Research Progress in Horizontal-Axis Marine Current Turines," *Journal of Energy*, vol. 6, pp. 1497-1526, 2013.
- [32] S. H. Salter and J. R. Taylor, "Vertical-Axis Tidal-Current Generators and the Pentland Firth," *J. Power and Energy*, vol. 221, pp. 181-199, 2007.
- [33] W. McKinney and J. DeLaurier, "The Wingmill: an Oscillating-Wing Windmill," *Journal of Energy*, vol. 5, no. 2, pp. 109-115, 1981.
- [34] K. D. Jones, K. Lindsey and M. F. Platzer, "An Investigation of the Fluid-Structure Interaction in an Oscillating-Wing Micro-Hydropower Generator," *Fluid Structure Interaction 2*, pp. 73-82, 2004.
- [35] T. Kinsey and G. Dumas, "Parametric Study of an Oscillating Airfoil in a Power-Extraction Regime," *AIAA Journal*, vol. 46, no. 6, pp. 1318-1330, 2008.
- [36] 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 & Fluids*, vol. 67, pp. 26-40, 2012.
- [37] T. Kinsey, G. Dumas, G. Lalande, J. Ruel, A. Mehut, P. Viarouge, J. Lemay and Y. Jean, "Prototype Testing of a Hydrokinetic Turbine Based on Oscillating Hydrofoils," *Renewable Energy*, vol. 36, pp. 1710-1718, 2011.
- [38] T. Kinsey and G. Dumas, "Computational Fluid Dynamics Analysis of a Hydrokinetic Turbine Based on Oscillating Hydrofoils," *Journal of Fluids Eng.*, vol. 134, no. 2, 2012.
- [39] 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 Turbulent Model," *Computers & Fluids*, vol. 88, pp. 136-155, 2013.
- [40] F. R. Menter, "Improved two-equation eddy-viscosity turbulence models for engineering," *AIAA Journal*, vol. 32, no. 8, pp. 1598-1605, 1994.
- [41] T. Kinsey and G. Dumas, "Optimal Operating Parameters for an Oscillating Foil Turbine at Reynolds Number 500,000," *AIAA Journal*, 2014.
- [42] M. F. Platzer, J. Young and J. S. Lai, "Flapping-Wing Technology: The Potential for Air Vehicle Propulsion and Airborne Power Generation," in *26th International Congress of the Aeronautical Sciences*, Canberra, Australia, 2008.
- [43] R. A. Bradley and M. F. Platzer, "Oscillating-Wing Power Generator with Flow-Induced Pitch-Plunge phasing.," *US patent*, p. A1, 2009.
- [44] M. A. Ashraf, J. Young, J. S. Lai and M. F. Platzer, "Numerical Analysis of an Oscillating-Wing and Hydropower Generator," *AIAA Journal*, vol. 49, no. 7, pp. 1374-1386, 2011.
- [45] M. F. Platzer, M. A. Ashraf, J. Young and J. S. Lai, "Development of a new Oscillating-Wing Wind and Hydropower Generator.," in *47th AIAA Science meeting*, Orlando, Florida, 2009.

- [46] J. Wang, C. Liang and M. Miesch, "High-Order Accurate CFD Simulation of an Oscillating-Wing Wind Power Generator," in *51st AIAA Sciences Meeting*, Dallas, Texas, 2013.
- [47] J. Young, J. C. Lai and M. F. Platzer, "A Review of Progress and Challenges in Flapping Foil Power Generation," *Progress in Aerospace Sciences*, p. in Press, 2014.
- [48] G. H. Bryan, "Stability in Aviation," *Macmillan, London*, pp. 19-37, 1911.
- [49] D. R. McDaniel, R. M. Cummings, K. Bergeron, S. A. Morton and J. P. Dean, "Comparisons of Computational Fluid Dynamics Solutions of Static and Manoeuvring Fighter Aircraft with Flight Test Data," *J. Aerospace Engineering*, vol. 223, pp. 323-340, 2009.
- [50] R. D. Kimberlin, *Flight Testing of Fixed-Wing Aircraft*, AIAA Education Series, 2003.
- [51] R. H. Landon, "NACA 0012. Oscillating and Transient Pitching," AGARD, France, 1982.
- [52] O. J. Boelens, K. J. Badcock, A. Elmilgui, K. S. Abdol-Hamid and S. J. Massey, "Comparison of Measured and Block Structured Simulations for the F-16XL Aircraft," in *AIAA papar*, 2009.
- [53] A. Da Ronch, A. McCracken, K. J. Badcock, M. Widhalm and M. S. Campobasso, "Linear Frequency Domain and Harmonic Balance Predictions of Dynamic Derivatives," *Journal of Aircraft*, vol. 50, no. 3, pp. 694-707, 2013.
- [54] A. Zanon, P. Giannattasio and C. J. S. Ferreira, "A Vortex Panel Model for the Simulation of the Wake Flow Past a Vertical Axis Wind Turbine in Dynamic Stall," *J. Wind Energy*, 2012.
- [55] N. A. Pierce and J. J. Alonso, "Efficient Computation of Unsteady Viscous Flows by a Implicit Preconditioned Multigrid Method," *AIAA Journal*, vol. 36, no. 3, pp. 401-408, 1998.
- [56] V. Ramesh and S. M. Deshpande, "Unsteady computations for Flow Past Two Oscillating Airfoils," in *9th AeSI CFD symposium*, Bangalore, India, 2006.
- [57] A. Da Ronch, D. Vallespin, M. Ghoreyshi and K. J. Badcock, "Evaluation of Dynamic Derivatives Using Computational Fluid Dynamics," *AIAA Journal*, vol. 50, no. 2, pp. 470-484, 2012.
- [58] J. Thompson, N. T. Frink and P. C. Murphy, "Guidelines for Computing Longitudinal Dynamic Stability Characteristics of a Subsonic Transport," *AIAA paper*, 2010.
- [59] F. Liu and X. Zheng, "A Strongly Coupled Time-Marching Method for Solving the Navier-Stokes and $k-\omega$ Turbulence Model Equations with Multigrid," *J. of Computational Physics*, vol. 128, no. 0211, pp. 289-300, 1996.
- [60] F. B. Lin and F. Sotiropoulos, "Strongly-Coupled Multigrid Method for 3-D incompressible Flows Using Near-Wall Turbulence Closures," *ASME*, vol. 119, pp. 314-324, 1997.

- [61] P. Eliasson and S. Wallin, "A Positive Multigrid Scheme for Computations with Two-Equation Turbulence Models," in *ECCOMAS*, Barcelona, 2000.
- [62] F. Liu and X. Zheng, "Staggered Finite Volume Scheme for Solving Cascade Flow with a k - ω Turbulence Model," *AIAA Journal*, vol. 32, no. 8, pp. 1589-1597, 1994.
- [63] S. Lee and D. W. Choi, "On Coupling the Reynolds-Averaged Navier–Stokes Equations with Two-Equation Turbulence Model Equations," *J. for Numerical Methods in Fluids*, vol. 50, no. 2, pp. 165-197, 2006.
- [64] M. Wasserman, Y. Mor-Yossef, I. Yavneh and J. B. Greenberg, "A robust implicit multigrid method for RANS equations with two-equation turbulent models," *J. of Computational Physics*, vol. 229, pp. 5820-5842, 2010.
- [65] N. D. Melson, M. D. Sanetrik and H. L. Atkins, "Time-Accurate Navier-Stokes Calculations with Multigrid Acceleration," in *AIAA Computational Fluid Dynamics*, Orlando, France, 1993.
- [66] A. Jameson, "Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings," in *AIAA Computational Fluid Dynamics*, Honolulu, Hawaii, 1991.
- [67] K. C. Hall, J. P. Thomas and W. S. Clark, "Computation of Unsteady Nonlinear Flows in Cascades Using a Harmonic Balance Technique," *AIAA Journal*, vol. 40, no. 5, pp. 879-886, 2002.
- [68] M. S. Campobasso, "EPSRC," 2008. [Online]. Available: <http://gow.epsrc.ac.uk/NGBOVViewGrant.aspx?GrantRef=EP/F038542/1>.
- [69] M. S. Campobasso and M. H. Baba-Ahmadi, "Ad-hoc Boundary Bonditions for CFD Analyses of Turbomachinery Problems with Strong Flow Gradients at Farfield Boundaries," *ASME Journal of Turbomachinery*, vol. 133, no. 4, 2011.
- [70] 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," *J. Turbomach*, vol. 134, no. 6, 2012.
- [71] J. Yao, A. Jameson, J. J. Alonso and F. Liu, "Development and Validation of a Massively Parallel Flow Solver for Turbomachinery Flows," in *AIAA 38th Aerospace Sciences Meeting and Exhibit*, Reno,NV, 2000.
- [72] O. Reynolds, "On the extent and action of the heating surface for steam boilers," *Proc. Manchester Lit. Phil. SOC.*, vol. 14, pp. 7-12, 1874.
- [73] A. A. Townsend, *The Structure of Turbulent Shear Flow*, Cambridge University press, 1980.
- [74] D. Spalding, "A single formula for the law of the wall," *Journal of Applied Mechanics*, no. 83, p. 455, 1961.
- [75] W. Rodi and G. Scheurer, "Scrutinizing the k - ϵ model under adverse pressure gradient conditions," *J. Fluid Eng.*, no. 108, pp. 174-179, 1986.
- [76] D. G. Koubogiannis, A. N. Athanasiadis and K. C. Giannakoglou, "One- and Two-Equation Turbulence Models for the Prediction of Complex Cascade Flows using Unstructured Grids," *Computers and Fluids*, vol. 32, no. 3, pp. 403-430, 2003.

- [77] J. A. Ekaterinaris and F. R. Menter, "Computation of Oscillating Airfoil Flows with One- and Two-Equation Turbulence Models," *AIAA Journal*, vol. 32, no. 12, pp. 2359-2365, 1994.
- [78] D. C. Wilcox, "Formulation of the $K - \omega$ Turbulence Model Revisited," in *45th AIAA Aerospace Sciences Meeting*, Reno, Nevada, 2007.
- [79] B. S. Baldwin and T. J. Barth, "A One-Equation Turbulence Transport Model for High Reynolds Number Wall-Bounded Flows," in *NASA Technical Memorandum*, Ames Research Center, California, 1990.
- [80] P. R. Spalart, "Strategies for Turbulence Modeling and Simulations," *International Journal of Heat and Fluid Flow*, vol. 21, pp. 252-263, 2000.
- [81] A. Jameson and T. J. Baker, "Solution of the Euler Equations for Complex Configurations.," in *6th AIAA computational fluid*, Danvers, Massachusetts, 1983.
- [82] E. Ferrer and X. Munduate, "Cfd predictions of transition and distributed roughness over a wind turbine airfoil," in *National Renewable Energy Centre*, Navarra, Spain, 2009.
- [83] S. H. Park and J. H. Kwon, "Implementation of $k-\omega$ Turbulence Models in an Implicit Multigrid Method," *AIAA JOURNAL*, vol. 42, no. 7, pp. 1348-1357, 2004.
- [84] J. M. Verdon, M. D. Montgomery and H. A. Chuang, "Development of a Linearized Unsteady Euler Analysis with Application to Wake/Blade-Row Interactions," NASA Contractor Report NASA/CR 1999-208879, 1999.
- [85] W. S. Clark and K. C. Hall, "A Time-Linearized Navier-Stokes Analysis of Stall Flutter," *Journal of Turbomachinery*, vol. 122, pp. 467-476, 2000.
- [86] W. Ning and L. He, "Computation of Unsteady Flows Around Oscillating Blades Using Linear and Nonlinear Harmonic Euler Methods," *Journal of Turbomachinery*, vol. 120, pp. 508-514, 1998.
- [87] R. C. Maple, P. I. King, P. D. Orkwis and J. M. Wolff, "Adaptive Harmonic Balance Method for Nonlinear Time-Periodic Flows," *Journal of Computational Physics*, vol. 193, pp. 620-641, 2004.
- [88] L. He, "Harmonic Balance of Unsteady Flow Around Blades with Separation," *AIAA Journal*, vol. 46, no. 6, pp. 1299-1307, 2008.
- [89] R. Corral and J. Crespo, "Development of An Edge-Based Harmonic Balance Method for Turbomachinery Flows," in *ASME Turbo EXPO*, Vancouver, Canada, 2011.
- [90] D. Tang and E. H. Dowell, "Limit-Cycle Hysteresis Response for a High-Aspect-Ratio Wing Model," *Journal of Aircraft*, vol. 39, no. 5, pp. 885-888, 2002.
- [91] J. P. Thomas, E. H. Dowell and K. C. Hall, "Modeling Viscous Transonic Limit-Cycle Oscillation Behavior Using a Harmonic Balance Approach," *Journal of Aircraft*, vol. 41, no. 6, pp. 1266-1274, 2004.
- [92] A. K. Gopinath and A. Jameson, "Time Spectral Method for Periodic Unsteady Computations over Two- and Three- Dimensional Bodies," in *AIAA Aerospace Meeting*, Reno, Nevada, 2005.

- [93] J. C. Vassberg, A. K. Gopinath and A. Jameson, "Revising the Vertical-Axis Wind-Turbine Design using Advance Computational Fluid Dynamics," in *AIAA Aerospace meeting*, Reno, Nevada, 2005.
- [94] P. J. Attar, E. H. Dowell, J. R. White and J. P. Thomas, "Reduced Order Nonlinear System Identification Methodology," *AIAA Journal*, vol. 44, no. 8, pp. 1895-1904, 2006.
- [95] 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," in *ASME Turbo Expo 2014*, Dusseldorf, Germany, 2014.
- [96] D. Hassan and F. Sicot, "A Time-Domain Harmonic Balance Method for Dynamic Derivatives Predictions," in *AIAA Aerospace meeting*, Orlando, Florida, 2011.
- [97] S. M. Murman, "A Reduced-Frequency Approach for Calculating Dynamic Derivatives," in *AIAA Aerospace meeting*, Remo, Nevada, 2005.
- [98] E. Weide, A. K. Gopinath and A. Jameson, "Turbomachinery Applications with the Time Spectral Method," in *AIAA Fluid Dynamics Conference*, Toronto, Orntario, 2005.
- [99] M. A. Woodgate and K. J. Badcock, "Implicit Harmonic Balance Solver for Transonic Flow with Forced Motions," *AIAA Journal*, vol. 47, no. 4, pp. 893-901, 2009.
- [100] J. M. Weiss and K. C. Hall, "Simulation of Unsteady Turbomachinery Flows Using an Implicitly Coupled Nonlinear Harmonic Balance Method," in *ASME Turbo EXPO*, Vancouver, Canada, 2011.
- [101] L. Liu, J. P. Thomas, E. H. Dowell, P. Attar and K. C. Hall, "A Comparison of Classical and High Dimentional Harmonic Balance Approaches for a Duffing Oscillator," *Computational Physics*, pp. 298-320, 2006.
- [102] L. N. Virgin, "On the harmonic response of an oscillator with unsymmetric restoring force," *Journal of Sound and Vibration*, vol. 126, no. 1, pp. 157-165, 1988.
- [103] P. Donecsu, L. N. Virgin and J. J. Wu, "Periodic Solutions of an Unsymmetric Oscillator Including a Comprehensive Study of their Stability Characteristics," *Journal of Sound and Vibration*, vol. 192, no. 5, pp. 959-976, 1996.
- [104] K. Ekici and K. C. Hall, "Harmonic Balance Analysis of Limit Cycle Oscillations in Turbomachinery," *AIAA Journal*, vol. 49, no. 7, pp. 1478-1487, 2011.
- [105] K. Ekici and K. C. Hall, "Nonlinear Analysis of Unsteady Flows in Multistage Turbomachines Using Harmonic Balance," *AIAA Journal*, vol. 45, no. 5, pp. 1047-1057, 2007.
- [106] P. Knupp and S. Steinberg, *Fundamentals of Grid Generation*, 1993.
- [107] J. F. Thompson, B. K. Soni and N. P. Weatherill, *Handbook of Grid Generation*, CRC Press LLC, 1999.
- [108] J. F. Thompson, Z. A. Warsi and C. W. Mastin, *Numerical Grid Generation: Foundations and Applications*, 1997.

- [109] 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," in *ASME*, Vancouver, British Columbia, Canada, 2011.
- [110] A. Jackson and M. S. Campobasso, "Shared-memory, Distributed-memory and Mixed-mode Parallelization of a CFD Simulation Code," *Computer Science Research and Development*, vol. 26, no. 3-4, pp. 187-195, 2011.
- [111] A. Jackson and M. S. Campobasso, "Optimised Hybrid Parallelisation of a CFD code on Many Core architectures," in *15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, Timisoara, Romania, 2013.
- [112] C. J. Fletcher, *Computational Techniques for Fluid Dynamics 1*, Springer-Varlag, 1990.
- [113] F. B. Hildebrand, *Introduction to Numerical Analysis*, Second Edition, New York: Dover Publications, Inc, 1974.
- [114] G. Moretti, "Efficient Euler Solver with Many Applications," *AIAA Journal*, vol. 26, no. 6, pp. 655-660, 1988.
- [115] R. Marsilio and G. Moretti, "Shock-Fitting Method for Two-Dimensional Inviscid Steady Supersonic Flows in Ducts," *Meccanica*, vol. 24, pp. 216-222, 1989.
- [116] M. J. Turner, R. W. Clough, H. C. Martin and L. P. Topp, "stiffness and deflection analysis of complex structures," *J. Aeronautical Society*, vol. 23, p. 805, 1956.
- [117] R. W. Clough, "The Finite Element Method in Plane Stress Analysis," in *Proc 2nd ASCE Conference on Electronic Computation*, 1960.
- [118] O. C. Zienkiewicz and R. L. Taylor, *The finite Element Method.*, Mcgraw-hillboo company, 1991.
- [119] T. J. Chung, *Finite Element Analysis in Fluid Dynamics*, New York: McGraw-Hill Inc, 1978.
- [120] F. Thomasset, *Implementation of Finite Element Methods for Navier-Stokes Equations*, Springer Serieis in Computational Physics, 1981.
- [121] A. J. Baker, *Finite Element Computational Fluid Mechanics*, New York: McGraw-Hill, 1983.
- [122] P. W. McDonald, "The computation of Transonic Flow through Two-Dimensional Gas Turbine Cascades," in *ASME 71-GT-89*, 1971.
- [123] B. V. Leer, "Towards the Ultimate Conservative Difference Scheme. Anew Approach to Numerican Convection," *Journal of Computational Physics*, vol. 23, pp. 276-299, 1977.
- [124] B. V. Leer, "Flux-vector splitting for euler equations," in *8th International Conference on Numerical Methods in Fluid Dynamics*, Germany, 1982.
- [125] J. Morrison, "Flux Difference Split Scheme for Turbulent Transport Equations," in *AIAA Second Intrernational Aerospace Planes Conference*, Orlando, 1990.
- [126] P. L. Roe, "Approximate Riemann Solvers, Parameter Vectors and Difference Schemes," *Journal of Computational Physics*, vol. 43, pp. 357-372, 1981.

- [127] W. K. Anderson, J. L. Thomas and V. B. Leer, "A Comparison of Finite Volume Flux Vector Splitting for the Euler Equations," *AIAA Journal*, vol. 24, no. 9, pp. 1453-1460, 1986.
- [128] G. D. van Albada, B. van Leer and R. W. Roberts, "A Comparative Study of Computational Methods in Cosmic Gas Dynamics," *Astronomy and Astrophysics*, vol. 108, pp. 76-84, 1982.
- [129] D. Drikakis and S. Tsangaris, "On the Solution of the Compressible Navier-Stokes equations using Improved Flux Vector Splitting Methods," *Applied Mathematical Modelling*, vol. 17, no. 6, pp. 282-297, 1993.
- [130] A. Harten, "High Resolution Schemes for Hyperbolic Conservation Laws," *J. of Computational Physics*, vol. 49, pp. 357-393, 1983.
- [131] S. Gottlieb and C. W. Shu, "Total Variation Diminishing Runge Kutta Schemes," *Mathematics of Computation*, vol. 67, no. 221, pp. 73-85, 1998.
- [132] P. Gerlinger, J. Algermissen and D. Bruggermann, "Matrix Dissipation for Central Difference Schemes Combustion," *AIAA Journal*, vol. 33, no. 10, pp. 1865-1870, 1995.
- [133] H. C. Lin, "Dissipation Additions to Flux-Difference Splitting," *J. of Computational Physics*, vol. 117, pp. 20-27, 1995.
- [134] H. C. Yee, "A Class of High-Resolution Explicit and Implicit Shock-Capturing Methods," in *Computational Fluid Dynamics*, Rhode-St-Genese, Belgium, 1989.
- [135] P. Woodward and P. Colella, "The numerical Simulation of Two-Dimensional Fluid Flow with Strong Shocks," *J. of Computational Physics*, vol. 54, pp. 115-173, 1984.
- [136] A. Harten, J. M. Hyman and P. D. Lax, "On Finite-Difference Approximations and Entropy Conditions for Shocks," *Communications on Pure and Applied Mathematics*, vol. XXIX, pp. 297-322, 1976.
- [137] A. Madrane and E. Tadmor, "Entropy stability of Roe-type upwind finite volume methods," *Applied Mathematics*, vol. 67, no. 2, pp. 775-784, 2009.
- [138] E. Tadmor, "Entropy Stability Theory for Difference Approximations of nonlinear Conservation Laws and Related Time-Dependent Problems," *Cambridge University Press*, pp. 451-512, 2003.
- [139] J. Blazek, C. C. Rossow, N. Kroll and R. C. Swanson, "A Comparison of Several Implicit Residual Smoothing Methods In Combination with Multigrid," *Thirteenth International Conference on Numerical Methods in Fluid Dynamics*, vol. 414, pp. 386-390, 1993.
- [140] R. C. Swanson, "An Efficient Flow Solver with a Transport Equation for Modeling Turbulence," in *19th AIAA Computational Fluid Dynamics*, San Antonio, Texas, 2009.
- [141] J. Blazek, N. Kroll, R. Radespiel and C. C. Rossow, "Upwind Implicit Residual Smoothing Method for Multi-Stage Schemes," *AIAA*, pp. 70-80, 1991.
- [142] C. C. Douglas, "Multigrid Methods in Science and Engineering," *IEEE COMPUTATIONAL SCIENCE & ENGINEERING*, pp. 55-68, 1996.

- [143] P. Wesseling, *An Introduction to Multigrid Methods*, New York: John Wiley & Sons, 1992.
- [144] A. Jameson, W. Schmidt and E. Turkel, "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," in *AIAA 14th Fluid and Plasma Dynamics*, Palo Alto, 1981.
- [145] A. Belov, L. Martinelli and A. Jameson, "A New Implicit Algorithm with Multigrid for Unsteady Incompressible Flow Calculations," in *AIAA 33rd Aerospace Sciences Meeting and Exhibit*, Reno, NV, 1995.
- [146] J. P. Singh, "An Improved Navier-Stokes flow computation of AGARD case-10 Flow over RAE2822 airfoil using Baldwin-Lomax Model," *ACTA Mechanica*, vol. 151, pp. 255-263, 2001.
- [147] D. J. Mavriplis and L. Matrinelli, "Multigrid Solution of Compressible Turbulent Flow on Unstructured Meshes Using a Two-Equation Model," *International Journal for Numerical Methods In Fluids*, vol. 18, pp. 887-914, 1994.
- [148] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Minneapolis: Industrial and Applied Mathematics, 2011.
- [149] K. J. Badcock, B. E. Richards and M. A. Woodgate, "Elements of Computational Fluid Dynamics on Block Structured Grids using Implicit Solvers," *Aerospace Sciences*, vol. 36, pp. 351-392, 2000.
- [150] K. J. Badcock and B. E. Richards, "Implicit Time Stepping Methods for the Navier-Stokes equations," *AIAA Journal*, vol. 34, pp. 555-559, 1995.
- [151] M. Nemeć and D. W. Zingg, "Newton-Krylov Algorithm for Aerodynamic Design Using the Navier-Stokes Equations," *AIAA Journal*, vol. 40, no. 6, pp. 1146-1154, 2002.
- [152] M. Blanco and D. W. Zingg, "Newton-Krylov Algorithm with a Loosely Coupled Turbulence Model for Aerodynamic Flows," *AIAA Journal*, vol. 45, no. 5, pp. 980-987, 2007.
- [153] T. T. Chisholm and D. W. Zingg, "A Jacobian-free Newton-Krylov Algorithm for Compressible Turbulent Fluid Flows," *J. of Computational Physics*, vol. 228, pp. 3490-3507, 2009.
- [154] R. M. Beam and R. F. Warming, "An Implicit Factored Scheme for the compressible Navier-Stokes Equations," *AIAA Journal*, vol. 16, pp. 393-402, 1978.
- [155] T. L. Tysinger and D. A. Caughey, "Alternating Direction Implicit Methods for the Navier-Stokes Equations," *AIAA Journal*, vol. 30, no. 8, pp. 2158-2161, 1992.
- [156] W. R. Briley and H. McDonald, "An Overview and Generalization of Implicit Navier-Stokes algorithms and Approximate Factorization," *Computers & Fluids*, vol. 30, pp. 807-828, 2001.
- [157] S. R. Chakravarthy, "Relaxation Methods for unfactored Implicit Upwind Schemes," *AIAA 22nd Aerospace Sciences Meeting*, 1984.

- [158] D. A. Knoll and D. E. Keyes, "Jacobian-free Newton-Krylov Methods: a Survey of approaches and Applications," *J. of Computational Physics*, vol. 193, pp. 357-397, 2004.
- [159] A. Rizzi and L. E. Eriksson, "Transfinite Mesh Generation and Damped Euler Equation Algorithm for Transonic Flow Around Wing-Body Configurations," in *Fifth AIAA Computational Fluid Dynamics Conference*, Palo Alto, 1981.
- [160] C. C. Rossow, "Efficient Computation of Compressible and Incompressible Flows," *J. of Computational Physics*, vol. 220, pp. 879-899, 2007.
- [161] L. A. Catalano and V. E. Daloiso, "Upwinding and Implicit Residual Smoothing on Cell-Vertex Unstructured Grids," *International J. for Numerical Methods in Fluids*, vol. 47, pp. 895-902, 2005.
- [162] J. H. Fassbender, "Robust and efficient computation of turbulent flows around civil transport aircraft at flight Reynolds numbers," *Elsevier, Aerospace Science and Technology*, vol. 9, pp. 672-680, 2005.
- [163] D. Strauss and J. F. Azevedo, "On the Development of an Agglomeration Multigrid Solver for Turbulent Flows," *J. of the Braz. Soc. of Mech. Sci. & Eng.*, vol. XXV, no. 4, pp. 315-324, 2003.
- [164] B. V. Leer, K. G. Tai and K. G. Powell, "Design of Optimally Smoothing Multi-Stage Schemes for the Euler Equations," *AIAA journal*, pp. 40-59, 1989.
- [165] C. H. Tai, J. H. Sheu and B. V. Leer, "Optimal Multistage Schemes for Euler Equations with Residual Smoothing," *AIAA Journal*, vol. 33, no. 6, pp. 1008-1016, 1995.
- [166] M. S. Campobasso, A. Piskopakis and M. Yan, "Analysis of an Oscillating Wing in a Power-Extraction Regime Based on the Compressible Reynolds-Averaged Navier-Stokes Equations and the K- ω SST Turbulence Model," in *ASME Turbo Expo*, San Antonio, Texas, USA, 2013.
- [167] A. Arnone, M. S. Liou and L. A. Povinelli, "Multigrid Time-Accurate Integration of Navier-Stokes Equations," in *Technical Memorandum NASA*, Lewis Research Center, 1993.
- [168] R. C. Swanson, E. Turkel and J. A. White, "An Effective Multigrid Method for High-Speed Flows," in *5th Copper Mountain Conference on Multigrid Methods*, 1991.
- [169] E. M. Constantinescu and A. Sandu, "Multi Timestepping Methods for Hyperbolic Conservation Laws," *J. Sci Computation*, vol. 33, pp. 239-278, 2007.
- [170] M. J. Grote and T. Mitkova, "Explicit Local Time-Stepping Methods for Time-Dependent Wave Propagation," in *Cornell University Library*, 2012.
- [171] E. Turkel, R. C. Swanson and V. N. Vatsa, "Multigrid for Hypersonic Viscous Two- and Three-Dimensional Flows," *AIAA Journal*, pp. 501-517, 1991.
- [172] R. C. Swanson and E. Turkel, "Multistage Schemes with Multigrid for Euler and Navier-Stokes Equations," in *NASA Technical Paper*, Hampton, Virginia, 1997.
- [173] A. Brandt, "Multi-Level Adaptive Solutions to Boundary-Value Problems," *Mathematics of Computation*, vol. 31, no. 138, pp. 333-390, 1977.

- [174] R. C. Swanson and C. C. Rossow, "An Efficient Flow Solver with a Transport Equation for Modeling Turbulence," in *AIAA Computational Fluid Dynamics*, San Antonio, Texas, 2009.
- [175] A. Jameson and T. J. Baker, "Multigrid Solution of the Euler Equations for Aircraft Configurations," in *AIAA 22nd Aerospace Sciences Meeting*, Reno, Nevada, 1984.
- [176] J. Yan, F. Thiele and L. Xue, "A modified Full Multigrid Algorithm for Navier-Stokes Equations," *Computer & Fluids*, vol. 36, pp. 445-454, 2007.
- [177] R. C. Swanson, E. Turkel and C. C. Rossow, "Convergence Acceleration of Runge-Kutta Schemes for Solving the Navier-Stokes Equations," *Journal of Computational Physics*, vol. 224, pp. 365-388, 2007.
- [178] M. Vazquez, M. Ravachol, F. Chalot and M. Mallet, "The Robustness Issue on Multigrid Schemes Applied to the Navier-Stokes Equations for Laminar and Turbulent, Incompressible and Compressible Flows," *Int. J. Numerical Methods In Fluids*, vol. 45, pp. 555-579, 2004.
- [179] Y. Moryossef and Y. Levy, "Unconditionally Positive Implicit Procedure for Two-equation Turbulence Model: Application to $k-\omega$ Turbulence models," *J. of Computational Physics*, vol. 220, pp. 88-108, 2006.
- [180] N. K. Lambropoulos, D. G. Koubogiannis and K. C. Giannakoglou, "Acceleration of a Navier-Stokes Equation Solver for Unstructured Grids using Agglomeration Multigrid and Parallel Processing," *Computer Methods in Applied Mechanics and Engineering*, vol. 193, pp. 781-803, 2004.
- [181] W. L. Briggs, E. Henson and S. F. McCormick, *A Multigrid Tutorial*, Second Edition: SIAM, 2000.
- [182] A. J. Wadcock, "Structure of the turbulent Separated Flow Around a Stalled Airfoil," Contract report NASA CR-152263, NASA Ames Research Center, Moffett Field, CA, USA, 1979.
- [183] S. E. Rogers, N. L. Wiltberger and D. Kwak, "Efficient Simulation of Incompressible Viscous Flow over Single and Multielement Airfoils," *J. of Aircraft*, vol. 30, no. 5, pp. 736-743, 1993.
- [184] R. CL., "CFL3D," <http://cfl3d.larc.nasa.gov/>, Version 6.6.
- [185] P. H. Cook, M. A. McDonald and M. C. Firmin, "Aerofoil RAE2822-Pressure Distributions, Boundary Layer and Wake Measurements," AGARD, 1979.
- [186] "NUMECA International," [Online]. Available: <http://www.numeca.com/en>.
- [187] A. Betz, "Das Maximum der Theoretisch Möglichen Ausnützung des Windes Durch Windmotoren," *Z Gesamte Turbinenwesen*, vol. 20, pp. 307-309, 1920.
- [188] J. T. Batina, "Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes," *AIAA Journal*, vol. 28, no. 8, pp. 1381-1388, 1990.
- [189] A. Da Ronch, "On the Calculation of Dynamic Derivatives Using Computational Fluid Dynamics," Thesis, University of Liverpool, 2012.
- [190] M. K. Koopaee, "Effect of flow regime change from subsonic to transonic on the air loads of an oscillating airfoil," *Journal of Fluids and Structures*, p. press, 2014.

