UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA

AERONÁUTICA Y DEL ESPACIO

Development of a High-Order Method for the
Prediction of Low-Pressure Turbine Losses
using Quasi-Direct Numerical Simulations

TESIS DOCTORAL

Marc Bolinches i Gisbert

Ingeniero Aeronáutico

Madrid, 2020

Departamento de Motopropulsión y
Termofluidodinámica

Escuela Técnica Superior de Ingeniería Aeronáutica y
del Espacio

Development of a High-Order Method for the
Prediction of Low-Pressure Turbine Losses
using Quasi-Direct Numerical Simulations

Autor

Marc Bolinches i Gisbert
Ingeniero Aeronáutico

Director

Roque Corral García
Doctor Ingeniero Aeronáutico

Madrid, 2020

Escuela Técnica Superior de Ingeniería Aeronáutica y del Espacio

UNIVERSIDAD POLITÉCNICA DE MADRID

Tribunal nombrado por el Magnífico y Excelentísimo Señor Rector de la Universidad Politécnica de Madrid. El día de de 2020.

Presidente: Benigno Lázaro

Secretario: Carlos Martel

Vocal: Mehdi Vahdati

Vocal: Alfredo Pinelli

Vocal: Oscar Flores

Suplente: Manuel Burgos

Suplente: Almudena Vega

Realizado el acto de defensa y lectura de la tesis el día de de 2020 en la E.T.S.I. Aeronáutica y del Espacio.

El Presidente

Los Vocales

El Secretario

Aquesta tesi està dedicada als meus pares, Emili Josep Bolinches i Maria Lluïsa Gisbert.

Acknowledgments

First of all I would like to thank Roque Corral for giving me the opportunity to carry out this work. His guidance has been essential for the completion of this work. His knowledge of the field and his determination to improve the tools and methods used today in turbomachinery design have been an inspiration.

Special mention goes to Fernando Gisbert. He has been essential in the integration with the Mu^2s^2T code and the GPU implementation. He has solved so many doubts along the long and tedious way of implementing a CFD tool.

David Cadrecha is also acknowledge here. We had numerous discussions about the experimental setup and the aerodynamics of the test case. I learned a lot from these.

There are many other people who have contributed directly or indirectly towards this work. Most people in the Methods Department of ITP Aero are to be thanked.

A special thank goes to my family. Without their encouragement this work would have not been possible. This thesis is dedicated to them.

Abstract

In this thesis the implementation of a compact high-order Computational Fluid Dynamics (CFD) method and its validations is tackled. The chosen compact scheme is that of the Flux-Reconstruction (FR) method. This method is indeed a family of methods that can recover other schemes such as Discontinuous Galerkin method. These methods are particularly well suited for massively parallel hardware architectures since many operation per degree of freedom are performed which are independent of neighboring cells. The implementation of the CFD method has been made on utility Graphical Processor Units (GPU) in order for the software to take advantage of their parallel architecture.

The GPUs used in this work are single precision hardware. This imposed further action in order to keep the precision of the solution. Particularly, action was needed in the implementation of our time-marching algorithm. This was mainly due to the fact that very small time-steps were imposed by stability conditions. The Kahan summation algorithm was used in order to prevent single-precision errors from appearing as the solution marched in time.

The validation of the method was carried out against a very rich array of high-quality experimental data. This validation case consisted of a low velocity Low Pressure Turbine linear cascade designed by ITP Aero. The experiments were carried out at the Polytechnic University of Madrid. Pressure distribution were measured over the profile; velocity and root-mean-square (RMS) of velocity fluctuations were measured in the boundary layer; and total pressure, velocity and RMS of velocity and angle distributions were measured at the exit measurement station. All these data were compared against the numerical results with outstanding agreement for all the cases. Global quantities such as the cascade turning angle and cascade losses also agreed well

with experimental data. More importantly, the Reynolds lapse rate, *i.e.*, the variation of the cascade loss with Reynolds number, was captured by our simulations. This thorough comparison is expected to increase confidence in compact high-order schemes.

Keywords: computational fluid dynamics, large eddy simulation, linear cascade aerodynamics, high-order methods, flux reconstruction

Resumen

Esta tesis aborda el desarrollo y la implementación de un método compacto de alto orden para mecánica de fluidos computacional. El esquema propuesto es el del método de reconstrucción de flujo (Flux-Reconstruction). Este método es, en realidad, una familia de métodos con la que se pueden recuperar otros esquemas populares tales como el Discontinuous Galerkin. Dado que estos métodos requieren de una gran cantidad de operaciones per grado de libertad, su implementación en ordenadores preparados para la computación en paralelo es especialmente eficiente. De hecho, la implementación hecha en esta tesis es en tarjetas gráficas (GPU).

Las tarjetas gráficas utilizadas en este trabajo son de uso común, *i.e.*, tarjetas no diseñadas para computación. Estas tarjetas tienen una arquitectura hecha para precisión simple y, por tanto, operaciones adicionales son necesarias para preservar la calidad de la solución. En particular, el esquema de integración en el tiempo se tuvo que modificar para evitar error de precisión. Esto es debido al extremadamente pequeño paso de tiempo que impone la estabilidad del esquema. El algoritmo de Kahan se implementó en nuestro método de integración para resolver este problema adicional.

El método desarrollado se validó utilizando datos experimentales de alta calidad. Este caso de validación consiste en una cascada de turbina de baja presión y baja velocidad diseñado por ITP Aero. Los ensayos experimentales se realizaron en la Universidad Politécnica de Madrid. Durante los ensayos se midieron la presión sobre el perfil; la velocidad y sus variaciones sobre la capa límite; y las distribuciones de presión total, velocidad y sus variaciones, y ángulo en la estación de salida. El nivel de acuerdo entre todos los datos medidos experimentalmente y los numéricos es muy alto. En particular, el “Reynolds lapse rate” (variaciones de las pérdidas de la

cascada con el número de Reynolds) es capturado con precisión. Este resultado es especialmente relevante y creemos que ayudará a aumentar la confianza en este tipo de métodos.

Palabras clave: dinámica de fluidos computacional, large eddy simulation, aerodinámica de cascadas lineales, métodos de alto orden, flux reconstruction

Contents

List of Figures	xiii
List of Tables	xv
Nomenclature	xx
1. Introduction	1
1.1. Reynolds number effects on LPT performance	2
1.1.1. Transition in LPTs	3
1.1.2. Simulations of LPTs with scale resolving simulations	5
1.1.3. Lapse rate investigations	9
1.2. Numerical tools	10
1.2.1. Turbulence Resolving Simulations in Low Pressure Turbines	13
1.2.2. High-Order Methods	14
1.2.3. Challenges of High-Order Methods	15
1.3. Contribution of this Thesis	16
1.4. Outline of this Thesis	17
2. Numerical Method	19
2.1. Introduction	19
2.1.1. Second-order CFD methods	20
2.1.2. Element based methods	22
2.2. The Flux Reconstruction Method	28
2.2.1. Summary of the FR method	32

2.2.2.	Treatment of dissipation terms	33
2.3.	Temporal integration	35
2.4.	Numerical properties	39
2.4.1.	Discretization of the 1D linear advection equation	40
2.4.2.	The 1D linear advection-diffusion equation	47
2.5.	Discretization of the 1D Euler equations	51
2.5.1.	The one-dimensional Euler equations	51
2.5.2.	The Riemann problem	52
2.5.3.	Numerical discretization 1D	53

3. Method Implementation 57

3.1.	Navier-Stokes equations	57
3.1.1.	Navier-Stokes equations in the FR Context	59
3.2.	Hardware and Software architecture	61
3.3.	Implementation of the Operators	64
3.3.1.	Data arrangement for maximum achievable bandwidth	66
3.3.2.	Evaluation of the discontinuous derivatives	67
3.3.3.	Evaluation of derivative corrections	70
3.3.4.	Kernels time breakdown	71
3.4.	Parallel use of multiple GPUs	72
3.4.1.	Increase of data transfer bandwidth	74
3.4.2.	Overlap of communication and computation	74
3.4.3.	Application to a representative case	76
3.5.	Explicit treatment of round-off errors	77

4. Boundary Conditions 83

4.1.	Introduction	83
4.2.	Boundary conditions in FR	85
4.3.	Wall boundary condition	85

4.4. Inlet/Outlet boundary conditions	87
4.4.1. NSCBC theory	88
4.4.2. Subsonic outlet BC	92
4.4.3. Subsonic inlet BC	95
5. Test Cases	103
5.1. Introduction	103
5.2. Inviscid 2D vortex in uniform flow	103
5.3. Blasius boundary layer	108
5.4. Taylor-Green vortex	111
5.5. Turbulent channel simulation	114
6. The Validation Case	117
6.1. The experimental setup	117
6.1.1. Considered cases	119
6.1.2. Boundary layer measurement stations	120
6.2. The numerical setup	120
6.2.1. Computational grids	121
6.2.2. Steady state hypothesis	124
6.2.3. Unsteady cases considerations	125
6.2.4. Stream-tube definition	127
6.2.5. Mach number effects	128
6.2.6. Convergence	128
6.2.7. Computational cost	132
7. Results	135
7.1. Steady results	135
7.1.1. Time-averaged results	136
7.1.2. Boundary layer analysis	138
7.1.3. Wake data	142

7.1.4. Cascade Exit Angle and Losses	145
7.2. Unsteady results	147
7.2.1. Time-averaged pressure coefficient	149
7.2.2. Phase-averaged skin friction coefficient	152
7.2.3. Velocity profiles	154
7.2.4. Wake traverse data	157
7.2.5. Cascade exit angle and losses	160
8. Conclusions and Future Work	163
8.1. Summary	163
8.2. Future work	166
A. Extension of the FR method to multiple dimensions	169
A.1. Extension of the method by Huynh to 2D (QUADs)	169
A.2. The CPR method for Triangles	170
A.3. Discretization of the 2D linear advection equation	174
Bibliography	181

List of Figures

1.1.	Cross-cut section of a modern aero-engine with its main components laid out. .	2
1.2.	Velocity field over the suction surface of a LPT aerofoil. The laminar separation bubble is highlighted. The top figure shows the time-averaged flow whereas the bottom figure shows a time instant.	4
1.3.	Sketch of total pressure loss variation with Reynolds number. The solid line is representative of a steady case whereas the dashed line is representative of unsteady cases.	9
2.1.	Schematic of node distribution for the Spectral Difference method. ●: solution points. ■: flux points. Note that the elements have been spaced out for clarity.	25
2.2.	Left- and right Radau polynomials for polynomial order $p = 3$. As it can be seen, the Radau polynomials are of order $p + 1$ ($p + 1$ zeros).	31
2.3.	Graphical representation of the Flux Reconstruction method. Domain split in three elements. ○ denote fluxes at solution points. Black solid lines denotes discontinuous flux function. □ denotes interface fluxes. Red solid line denotes reconstructed continuous flux function.	33
2.4.	Contour plots of amplification factor for the implemented four-step Runge-Kutta method.	39
2.5.	Dispersion relation of the 1D linear wave equation discretised with the FR scheme with $p = 3$ and $p = 5$. The exact relation and the second-order central differences scheme have been added for reference.	44

2.6.	Schematic representation of waves of different wave-number in relation to the element size Δx . Left: single fourth-order element ($p = 3$) with representation of the bad-resolved wave-number $k\Delta x/(p + 1) = \pi$ (dashed line) and well-resolved wave-number $k\Delta x/(p + 1) = \pi/2$ (solid line). Right: four equally-spaced second-order mesh elements for central differences with bad-resolved wave-number $k\Delta x = \pi$ (dashed line) and $k\Delta x = \pi/2$ (solid line).	44
2.7.	Dissipation of the 1D linear wave equation discretized with the FR scheme with $p = 3$ and $p = 5$	45
2.8.	Footprint of the dimensionless spatial eigenvalues, $\lambda\Delta t$, of the 1D linear wave equation for $p = 3$ and $CFL = 0.12$ (red line) and the stability limit, $ G = 1$, of a four-step RK scheme (black line).	46
2.9.	Projection of a single harmonic ($k\Delta x \approx \frac{\pi}{6}$) onto the eigenvector space for a third-order discretization ($p = 2$).	47
2.10.	Effect of the $Re_{\Delta x}$ on the fourth-order discretisation eigenvalues for $CFL_{inv} = 0.12$. The dashed black line encloses the stability region of the RK scheme.	49
2.11.	Scaled fourth-order discretisation eigenvalues for different values of the $Re_{\Delta x}$. The CFL used for each $Re_{\Delta x}$ follows the expression $CFL = CFL_{inv} \left(1 + \frac{2\sigma_{max}}{Re_{\Delta x}^2}\right)^{-1}$. The dashed black line encloses the stability region of the RK scheme.	50
2.12.	Eigenvalues of the discretized 1D Euler equations for $M = 0.3$ for two different Riemann solvers: Roe-type solver (red dots) and Lax-Friedrich (black dots). Black dashed line encloses the stable region of the RK4 scheme.	54
3.1.	Schematic of the hardware architecture where the simulations have been run. Top: configuration of the configuration. Bottom: internal architecture of the computing nodes.	63
3.2.	Sketch of data arrangement and discontinuous derivative operations on the GPU architecture. Global data is copied in local memory. Each thread performs its corresponding matrix-vector operation for each of the variables (a generic variable $\phi_i = \phi(\xi_i)$ is used in the figure for compactness).	68

3.3. Percentage of time spent executing each of the 8 kernels enumerated in section 3.3 for the Taylor-Green vortex case with a 28^3 sixth-order hexahedra elements.	72
3.4. Example of pipelined communication. The communication of a single message (top) is split into smaller messages (bottom) so communication between elements can happen concurrently.	73
3.5. Overlap of computation and communication. The $GPU \rightarrow CPU$ communication, the $CPU \rightarrow CPU$ MPI communication, and the $CPU \rightarrow GPU$ data transfer proceed simultaneously with the kernel executions. As long as the total communication time t_{comm} remains lower than $t_{interiorCells}$, the extra-cost of running parallel simulations will be $t_{gatherScatter} + (1 - GPU_{eff}) t_{frontierCells}$, where GPU_{eff} is the parallel efficiency of the GPU, which is below unity when the number of kernel threads is small.	75
3.6. Parallel speed-up for different number of mesh partitions. The mesh corresponds to the linear cascade case described in Chap. 6 for Reynolds number $Re = 10^5$ and reduced frequency $f_r = 1$. The mesh consists of 183 million points. In the cases presented above, the GPU memory usage varies between 92% when partitioned in 10 sub-domains to 10% at 80 sub-domains.	76
3.7. Mesh of the cylinder case used to validate the Kahan summation algorithm with a close-up around the cylinder.	79
3.8. Entropy snapshot and contour lines of a cylinder simulation after 160k time-steps with a time-step of $\Delta\tilde{t} = 2.2 \times 10^{-4}$. Left: single-precision without Kahan summation. Middle: double-precision simulation without Kahan summation. Right: single-precision simulation using Kahan summation.	81
3.9. Entropy snapshot and contour lines of a cylinder simulation after 160k time-steps with a time-step of $\Delta\tilde{t} = 2.2 \times 10^{-5}$. Left: single-precision without Kahan summation. Middle: double-precision simulation without Kahan summation. Right: single-precision simulation using Kahan summation.	81

4.1.	Schematic of the element boundary flux configuration for element i . Left: element i is a boundary element. Right: element i is an internal element.	84
4.2.	Pressure contours over a NACA0012 profile. Top: extended domain. Bottom: short domain. Pressure is allowed to vary in the y -direction so that the BC does not affect the pressure over the profile.	88
4.3.	Time sequence of a circular pressure wave leaving the computational domain after an impulsive start. Effect of the relaxation parameters on the inlet BC. Top: without relaxation coefficients ($\sigma_i = 0$). Bottom: with relaxation coefficients ($\sigma_i \neq 0$).	101
5.1.	Contour plots of the vertical velocity component of the Euler vortex in uniform flow (left) and linear mesh used for the simulation (right).	105
5.2.	Vertical velocity across the mid section of the vortex for the exact and numerical solution after 40 through-flows for different polynomial orders: $p = 1$, $p = 3$ and $p = 5$	106
5.3.	Vertical velocity across the mid section of the vortex after 40 flow-through (300 vortex diameters) for a discretization $p = 3$ and $p = 5$ with very similar number of nodes across the domain (43 for the $p = 3$ mesh and 41 for the $p = 5$ mesh).	107
5.4.	Convergence of error with mesh refinement for the Euler vortex case for two different polynomial orders, $p = 3$ and $p = 5$	108
5.5.	Schematic of the flat plate test case (left) and one of the meshes used in the study (right).	109
5.6.	The Blasius boundary layer solution (solid line) is compared against the results of the simulations for different meshes (solid dots) with polynomial degree $p = 3$. For clarity in the plot only solutions at the elements boundaries have been plotted.	111

5.7.	Iso-surfaces of vorticity at $t = 0$ (left) and $t = 10t_c$ (right) for the Taylor-Green vortex at $Re = 1600$ for polynomial order $p = 3$ and a mesh with 128 elements across each direction.	112
5.8.	Time evolution of the kinetic energy dissipation.	113
5.9.	Turbulent channel case configuration.	114
5.10.	Left: mean velocity profile in wall units for $Re_\tau = 180$ and $Re_\tau = 325$ along with the log law $u^+ = 1/0.41 \log(y^+) + 5.0$ (dashed line). For clarity the DNS solution at $Re_\tau = 550$ (green line), the iLES solution at $Re_\tau = 325$ (circles) and an additional log-law curve have been shifted upwards with $\Delta u^+ = 5$. Right: comparison of the root-mean-square values of the velocity fluctuations for $Re_\tau = 180$. $\circ$: Mu^2s^2T values; Solid lines: DNS from Lee and Moser[1].	115
6.1.	Schematic of the experimental set-up.	119
6.2.	Location of the boundary layer traverse measurement stations along with the absolute value of the friction coefficient of the steady $Re = 10^5$ case.	120
6.3.	Linear mesh topology (internal nodes to element not shown) used for steady and unsteady $Re = 10^5$ cases with a close-up of the mesh around the profile and trailing edge detail. Black lines in the detail define the second-order mesh and the red lines connect the high-order internal nodes.	122
6.4.	Experimental turbulence intensity as a function of time measured at different axial locations.	125
6.5.	Non-dimensional energy spectra for the $Re = 10^5$ case within the boundary layer at different stream-wise stations. The dashed line stands for the free turbulence energy decay rate ($E \sim f^{-5/3}$).	126
6.6.	Pressure coefficient distribution for different flow conditions (steady $Re = 10^5$).	128
6.7.	Convergence history of RMS velocity peak in the boundary layer of station e (circles) and the wake measurement position (squares) for the steady $Re = 10^5$.	129

6.8.	Convergence history of RMS velocity of the station e boundary layer (a) and the wake measurement position (b) as a function of the number of flow-through times. The insets correspond to the zones containing the largest velocity fluctuations. ($Re = 10^5$)	129
6.9.	Convergence of the RMS of velocity fluctuations at different locations across the computational domain for the $Re = 10^5$ cases. Top: $f_r = 0.5$ cases; bottom: $f_r = 1$ cases. y denotes local vertical distance to the wall.	131
7.1.	Iso-surfaces of Q-criterion coloured with the entropy level at the aft of the aerofoil. Pressure fluctuation field is plotted in the rear plane (steady $Re = 5 \times 10^4$),	136
7.2.	Experimental and numerical steady pressure coefficient distributions for different Reynolds numbers. Locations of boundary layer traverse measurements are indicated with letters a to e	137
7.3.	Steady friction coefficient along the aerofoil for different Reynolds numbers. The suction side is denoted with SS and the pressure side with PS.	137
7.4.	Boundary layer profiles of velocity (top) and root-mean-square of fluctuations of the absolute value of the velocity (bottom) at several stream-wise locations for the steady $Re = 5 \times 10^4$ case. $\circ$ experimental data; — numerical results. . .	139
7.5.	Boundary layer profiles of velocity (top) and root-mean-square of fluctuations of the absolute value of the velocity (bottom) at several stream-wise locations for the steady $Re = 10^5$ case. $\circ$ experimental data; — numerical results. . . .	139
7.6.	Boundary layer profiles of velocity (top) and root-mean-square of fluctuations of the absolute value of the velocity (bottom) at several stream-wise locations for steady $Re = 2 \times 10^5$ case. $\circ$ experimental data; — numerical results. . . .	140

7.7. Summary of steady wake traverses data at measuring station ($57\%c_x$ downstream of the TE) for all three Reynolds numbers ($Re = 5 \times 10^4$, 10^5 and 2×10^5 , from left to right). From top to bottom: Total pressure loss, outlet angle, velocity and RMS of velocity fluctuations in axial- and pitch-directions. Solid line: numerical results; circles: experimental data for the central aerofoil of the cascade; squares: experimental results for the aerofoil above the central one. . .	144
7.8. Steady exit angle as a function of the Reynolds number. Experimental results for the instrumented airfoil (black line) and the contiguous one (red line). . . : dotted line notates the bracket of the measurement errors.	145
7.9. Total-pressure loss coefficient scaled with the coefficient at $Re = 10^5$ ($\omega^* = \omega/\omega_{Re=10^5}$) as a function of the Reynolds number.	146
7.10. Top: iso-surfaces of Q-criterion coloured by Mach number with entropy field in the background. Bottom: velocity fluctuation (the darker the colour the more negative the fluctuation) with entropy field in the background. Case: $Re = 5 \times 10^4$, $f_r = 1$. Images are deformed for confidentiality reasons.	148
7.11. Time-averaged profile pressure coefficient, c_p , for all the tested cases. $\circ$: Steady experimental data; $---$: steady numerical data; $\square$: unsteady experimental data; $-$: unsteady numerical predictions. The arrows mark the locations where the boundary layer profiles have been measured.	149
7.12. Phase-averaged skin friction coefficient. Top row: steady case. Middle row: $f_r = 0.5$. Bottom row: $f_r = 1$. White continuous lines enclose areas of negative skin friction. The dashed lines indicate the location of the boundary layer traverse measurement stations.	151
7.13. Iso-contours of phase-averaged axial velocity perturbation ($v_x^{\text{phase}} - v_x^{\text{mean}}$) for phase $t/T_w = 0.2$ over the aft of the suction side. The centre of the wake is marked with a dashed line. Dotted lines denote rotation of phase-coherent vortices. A patch of wake high turbulence is marked with a greyed area.	153

7.14. Time-average of two-dimensional velocity modulus (lower horizontal axis) and RMS (upper horizontal axis) of fluctuations of the two-dimensional velocity modulus in the boundary layer for the $Re = 5 \times 10^4$ cases. Top: $f_r = 0.5$; Bottom: $f_r = 1$. $\circ$ experimental velocity; $\triangle$ experimental RMS of velocity fluctuations; $-$ numerical velocity; $--$ numerical RMS of velocity fluctuations.	154
7.15. Time-average of two-dimensional velocity modulus (lower horizontal axis) and RMS (upper horizontal axis) of fluctuations of the two-dimensional velocity modulus in the boundary layer for the $Re = 10^5$ cases. Top: $f_r = 0.5$; Bottom: $f_r = 1$. $\circ$ experimental velocity; $\triangle$ experimental RMS of velocity fluctuations; $-$ numerical velocity; $--$ numerical RMS of velocity fluctuations.	155
7.16. Time-average of two-dimensional velocity modulus (lower horizontal axis) and RMS (upper horizontal axis) of fluctuations of the two-dimensional velocity modulus in the boundary layer for the $Re = 2 \times 10^5$ cases. Top: $f_r = 0.5$; Bottom: $f_r = 1$. $\circ$ experimental velocity; $\triangle$ experimental RMS of velocity fluctuations; $-$ numerical velocity; $--$ numerical RMS of velocity fluctuations.	155
7.17. Distribution of time-averaged total pressure defect distribution at the exit measurement station. Top: $f_r = 0.5$; Bottom: $f_r = 1$. $\square, \triangle$: experimental measurements for central and contiguous pitches; Solid line: numerical data.	157
7.18. Distribution of time-averaged velocity distribution at the exit measurement station. Top: $f_r = 0.5$; Bottom: $f_r = 1$. $\square, \triangle$: experimental measurements for central and contiguous pitches; Solid line: numerical data.	158
7.19. Distribution of time-averaged angle distribution at the exit measurement station. Top: $f_r = 0.5$; Bottom: $f_r = 1$. $\square, \triangle$: experimental measurements for the central and contiguous pitches; Solid line: numerical data.	158
7.20. Distribution of time-averaged $u'v'_{rms}$ distribution at the exit measurement station. Top: $f_r = 0.5$; Bottom: $f_r = 1$. $\square, \triangle$: experimental measurements for central and contiguous pitches; Solid line: numerical data.	159
7.21. Time and pitch-wise averaged exit angle variation with Reynolds number and reduced frequency.	161

7.22. Time and pitch-wise averaged total pressure loss variation with the Reynolds number and reduced frequency. The experimental steady loss values have been added for reference. Note the logarithmic scale of the ordinate.	162
A.1. Eigenvalues for a QUAD element with an AR=10 (green dots) along with the eigenvalues of the 1D case (black dots) and the stability region of a four-steps fourth-order RK time integration scheme (black line).	176
A.2. Eigenvalues for a QUAD element with an AR=10 (green dots) along with the eigenvalues of the 1D case (black dots) and the stability region of a four-steps fourth-order RK time integration scheme (black line).	177
A.3. Eigenvalues for a TRIA element with an AR=1 (green dots) along with the stability region of a four-steps fourth-order RK time integration scheme (black line).	177
A.4. Eigenvalues for a TRIA element with an AR=10 (green dots) along with the eigenvalues of the 1D case (black dots) and the stability region of a four-steps fourth-order RK time integration scheme (black line).	178

List of Tables

- 2.1. Coordinates of the Gauss-Lobatto points for different number of points. 29
- 5.1. Summary of the errors and convergence order of the Blasius solution. 111
- 6.1. Summary of the meshes used in the steady simulations for each Reynolds number.123
- 6.2. Summary of the meshes used in the unsteady simulation for each Reynolds
number. 123
- 6.3. Summary of the computational cost and resources in steady cases. 132
- 6.4. Summary of the computational cost and resources in unsteady cases. 133

Nomenclature

Abbreviations

1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
AoS	Array of Structure
AR	Aspect Ratio
BC	Boundary Condition
BL	Boundary Layer
CFD	Computational Fluid Dynamics
CFL	Courant Friedrich Levy
CPU	Central Processing Unit
DG	Discontinuous Galerkin
DNS	Direct Navier-Stokes
FR	Flux Reconstruction
FV	Finite Volume

GPU	Graphics Processing Unit
HO	High-Order
ILES	Implicit Large Eddy Simulation
KH	Kelvin-Helmholtz
LCP	Lifting Collocation Penalty
LDG	Local Discontinuous Galerkin
LDV	Laser Doppler Velocimetry
LES	Large Eddy Simulation
LF	Lax-Friedrich
LPT	Low Pressure Turbine
LSP	Laminar Separation Bubble
MPI	Message Passage Interface
NS	Navier-Stokes
NSCBC	Navier-Stokes Characteristic Boundary Conditions
QDNS	Quasi-Direct Numerical Simulation
QUAD	Quadrilateral
RANS	Reynolds Averaged Navier-Stokes
RHS	Right-Hand-Side
RK	Runge-Kutta
RMS	Root Mean Square

SD	Spectral Difference
SGS	Sub-Grid Scale
SoA	Structure of Arrays
SP	Solution Point
SV	Spectral Volume
TE	Trailing Edge
TRIA	Triangular
uDNS	Underresolved Direct Navier-Stokes

Greek Symbols

α_2	Outlet angle
$\alpha_{i f l}$	Lifting coefficient of element i , face f and face location l
$\gamma(\xi)$	Flux correction function
ω	Pressure loss coefficient
ρ_2	Mean density at outlet
τ_w	Viscous friction at wall
ξ, η	Coordinates in standard element

Roman Symbols

U	Vector of conservative variables
---	----------------------------------

Subscripts

$\square^+$	Quantity $\square$ in wall units
-------------	----------------------------------

$\square^D$	Discontinuous (not corrected) quantity $\square$
$\tilde{V}$	Volume of standard element
c_f	$= \tau_w / (\frac{1}{2} \rho_2 \bar{V}_2^2)$. Friction coefficient
c_p	Pressure coefficient
c_x	Axial chord
F, G	Flux function in x, y direction
g_L, g_R	Left and right Radau polynomial
J	Determinant of the jacobian of coordinate transformation
L	Lagrange polynomial
p	Polynomial order or static pressure
p_2	Static pressure at outlet
P_{t1}	Mean total pressure upstream of cascade
P_{tmo}	Mixed-out total pressure at outlet
Re	Reynolds number
S	Suction side length
t	Time
T_{ft}	Flow-through time
U	Conservative quantities vector
V	Velocity
x, y, z	Physical coordinates

1. Introduction

Contents

1.1. Reynolds number effects on LPT performance	2
1.2. Numerical tools	10
1.3. Contribution of this Thesis	16
1.4. Outline of this Thesis	17

Turbomachinery components are complex systems with involved aerodynamic features. This makes their design a complex process which requires of dedicated wind-tunnel tests at both component and system levels. The efficiency of low-pressure turbines is strongly influenced by the separation, transition and reattachment of the flow on the suction side of the turbine aerofoils. The separation/transition phenomena are very Reynolds dependent and difficult to predict numerically with conventional Reynolds-Average Navier-Stokes (RANS) approaches due to the high number of hypotheses used to derive such models which are not fulfilled in this class of flows. This means that, aerodynamic testing is performed before the actual start of the design to assess new concepts. Scale resolving simulations are intended to replace the less complex tests, namely low- or high-speed linear cascades, or single-stage high-speed annular cascades. However, the cost of these simulations and affordability issues have traditionally prevented a widespread use of numerical tools to replace low Technology Readiness Level (TRL) tests. The development of efficient, accurate, but still flexible, numerical tools to address this problem is of paramount relevance.

Most of the accumulated knowledge today about the separation-transition-reattachment process

Figure 1.1.: Cross-cut section of a modern aero-engine with its main components laid out.

in Low-Pressure Turbine (LPT) has been acquired by means of wind-tunnel testing. RANS-based numerical simulations are used to conduct the design using strict design rules based on the knowledge obtained in previous experimental campaigns and calibrated tools. The description of how this process is conducted is frequently proprietary and goes well beyond the scope of this introduction. In this chapter the problem of Reynolds number effect predictions on LPT performance is discussed along with a literature review of the subject. Additionally, a review of the numerical tools for the analysis of LPTs is also made with emphasis on high-order tools.

1.1. Reynolds number effects on LPT performance

LPTs of modern aero-engines operate under a variety of Reynolds numbers throughout the flight cycle, *e.g.*, Reynolds numbers in the order of a few hundred thousand at take-off conditions and Reynolds numbers in the low hundred thousand or below at high-altitude cruise conditions. Within this range of Reynolds numbers LPT performance can vary substantially. In particular, losses can increase considerably at lower Reynolds numbers. This effect, known as Reynolds lapse, was identified from very early stages and its prediction is business critical for aero-engine makers. A review by Mayle can be found in [2].

Low-pressure turbine profile losses may account for up to an 80% of the total losses in modern

LPTs, and these are mainly controlled by the separation, transition, and reattachment of the flow on the suction side of the aerofoils [3]. Unlike other components, the impact of LPT efficiency on Specific Fuel Consumption is close to 1:1 and air-framers keep a tight control on them as well. As a consequence, the prediction of the different physical phenomena that control LPT losses is, then, of paramount importance for predicting the performance of an LPT and a whole aircraft.

1.1.1. Transition in LPTs

In the absence of perturbations, the laminar Boundary Layer (BL) of the suction side of LPTs may separate due to the adverse pressure gradient present in the rear part of high loaded aerofoils. The shear layer associated with the separated region is naturally unstable and rolls up due to the Kelvin-Helmholtz (KH) instabilities, forming the so-called KH vortices with their axis perpendicular to the flow direction and parallel to the wall. These vortices are further subject to secondary instabilities which distort them initiating a break-down process and transitioning to turbulence further downstream. Depending on the Reynolds number, this transition process can occur over the aerofoil, in which case the flow may reattach forming a closed Laminar Separation Bubble (LSB), or downstream of the airfoil so that the flow presents an open bubble. Figure 1.2 shows an example in which the flow reattaches before the trailing edge of the aerofoil. The top figure shows the time-averaged velocity field whereas at the bottom an instantaneous snapshot is shown. In this snapshot the break-down process into turbulence can be appreciated. The complex phenomena mentioned above are very sensitive to unsteady perturbations [3]. A large body of work has been produced in order to understand the effects of incoming wakes, background turbulence, etc., on the separation-transition-reattachment phenomena in LPTs both experimentally [3, 4, 5, 6, 7, 8] and numerically [9, 10, 11, 12, 5, 13, 14, 15, 16].

Other than background turbulence, there might be two different causes of unsteadiness: potential and wake interaction. Potential interaction refers to the effect of the pressure field generated by the downstream or upstream rows on the concerned aerofoil. Lou *et al.* [5] experimentally studied the laminar separation bubble and transition under a periodically varying free flow

Figure 1.2.: Velocity field over the suction surface of a LPT aerofoil. The laminar separation bubble is highlighted. The top figure shows the time-averaged flow whereas the bottom figure shows a time instant.

with very low inlet turbulence ($Tu \simeq 0.5\%$). They found that the reattachment point responded strongly to flow oscillations resulting in shorter time-averaged bubble lengths. Opoka *et al.* [6] also studied the effect of the unsteady potential field over boundary layer separation and transition with a different experimental setup. In their experiments, two different levels of free-stream turbulence were considered: low level of turbulence ($Tu = 0.5\%$) and high level of turbulence ($Tu = 4\%$). At the lower turbulence levels, they report results very similar to the ones reported by Lou *et al.* In the case of higher turbulence levels, however, bubble transition is bypassed, so that attached flow transition is present. An increase in the time-averaged momentum thickness of the boundary layer for both turbulence levels was reported.

Wake-blade interaction has also been extensively [3, 4, 7, 8, 17] researched in high-lift LPT flows, due to the fact, in part, that the design of modern high-lift profiles relies on the presence of upstream wakes to achieve high-efficiency LPTs [3]. Wakes influence aerodynamic losses in two different manners. Firstly, wakes trigger transition so that they can shorten the bubble length or suppress it completely, which considerably changes the boundary layer development and its associated losses. Secondly, wakes are heavily stretched within the blade passage incurring in additional losses.

Coull and Hodson [4] analysed the transition phenomena in high-lift LPTs with wakes and large levels of background turbulence by means of flat plate experiments under a pressure distribution that resembles that of an LPT aerofoil. Two reduced frequencies were tested, the lower of the two allowed for the boundary layer to return to a quasi-steady state whereas in the higher

frequency the boundary layer was always under the influence of passing wakes. In their very detailed analysis, they concluded that transition is driven by a complex interaction of the longitudinal streaks, boundary layer instabilities (*e.g.*, KH instabilities) and incoming wakes (*e.g.*, wake turbulence, negative jet, etc.). In particular, the intensity and number of the longitudinal streaks and their dependence on background turbulence and incoming wakes was analysed. It was noted that stronger longitudinal streaks have a greater effect on the KH vortices breakdown which results in earlier reattachment. The intensity of the longitudinal streaks was found to be dependent on the turbulence intensity as it passes near the leading edge (where there is a higher receptivity of the boundary layer). For very intense streaks it was found that they can destabilise and form turbulent spots which further accelerates transition. Passing wakes not only influence transition through the generation of longitudinal streaks. The velocity defect associated with it was also identified to intensify the intensity of already present streaks. Additionally, the velocity defect and turbulence associated with the incoming wake can interact directly with the formation and breakdown of the KH vortices.

As it can be seen, although inlet turbulence affects transition and breakdown of KH vortices, for sufficiently high reduced frequencies of incoming wakes their effect can be overcome by that of the incoming wakes. For sufficiently high reduced frequencies it is understood here frequencies at which the dynamics of the bubble is entirely influenced by the passing wakes. This is, the restitution time of the bubble after a wake passing is larger than that between passing wakes. In these cases, the most relevant parameters in transition are the Reynolds number (which controls the height and dynamics of the bubble), the pressure distribution (which controls the separation location) and reduced frequency. For low reduced frequencies, inlet turbulence also becomes an important parameter.

1.1.2. Simulations of LPTs with scale resolving simulations

Numerical simulations of LPT flows prove to be very complex. Reynolds Averaged Navier-Stokes (RANS) simulations make use of models that account for the effect of all scales of turbulence. These models are designed so that accurate results can be obtained for certain

canonical configurations but usually fail when others are analysed. To remedy this deficiency, detailed, but computationally expensive, analyses using Direct Numerical Simulations (DNS) or Large Eddy Simulations (LES) can be carried out. This type of simulations entail no turbulence model (DNS) or, at most, models for the smallest scales of the turbulence. These type of tools have been used for the detailed simulation of LPT flows for some time with mixed success. A non-negligible number of publications have claimed in the past to produce accurate results for the prediction of LPT flow. However, the level of accuracy shown by many authors fall short of replacing actual testing or taking business critical decisions based on that numerical data, *e.g.*, [15, 14, 9].

Some of the numerical simulations have been employed to better understand the unsteady flow characteristics of the separation-transition-reattachment process of LPT bubbles. McAuliffe and Yaras [13] performed a DNS of a flat plate with a pressure gradient distribution with a computational domain that was based on the experimental setup used by their research group. They used ANSYS-CFX for their simulations (second-order central differences) with a backward Euler time integration. Two simulations at different levels of free-stream turbulence were performed: low turbulence, 0.1% using a mesh with 1.1M cells; and high turbulence, 1.45% using a mesh with 4.2M cells. Turbulence was introduced through an array of uniform-velocity square jets that simulated the flow through the turbulence grid used in their experimental setup. Transition to turbulence was studied for the two free-stream turbulence cases. It was concluded that, in the low turbulence case, the transition to turbulence was due to the KH instability. However, in the elevated turbulence case, stream-wise streaks formed near the leading edge which intensified as they travelled downstream. The shear associated with these streaks appeared to be the cause of secondary instabilities that promoted the formation of local turbulent spots which eventually triggered early transition, in line with the findings of Jacobs and Durbin [18]. Good agreement with experimental data was found for the low free-stream turbulence case [19].

Michelassi *et al.* [15] investigated the ability of LES to capture the effect of incoming wakes on transition over the T106 aerofoil using a mesh with 10.5M nodes and an incompressible second-order cell-centred finite volume solver (a three-stage RK was used as the temporal integrator).

Even though the main flow features were captured in their simulation, the transition process was not fully captured being the transition point $10\%c_x$ downstream with respect to that of the DNS by Wu and Durbin [20]. They attributed these differences to a rather coarse mesh and to the Sub-Grid Scale (SGS) model used in their simulations which was the dynamic model by Germano *et al.* [21] with the modifications of Lilly [22]. In this model, the eddy viscosity vanished for well-resolved regions.

Other numerical studies have been devoted to the study of the impact of incoming wakes on the row aerodynamics and their contribution to losses. Michelassi *et al.* [23] studied the effect of inflow turbulence along with incoming wakes for different Reynolds numbers and reduced frequencies using DNS with their solver HiPSTAR which uses a structured fourth-order discretization in the pitch- and chord-wise direction and a spectral discretization in the span direction; it is compressible and uses a five-step fourth-order RK time integrator. The resulting mesh was 18.1M grid points with 32 Fourier modes in the span direction for their lower Reynolds number case and 75.6M points with 64 Fourier modes in the span direction for the higher Reynolds number case. They showed with their simulations that the sole effect of the inlet turbulence can decrease profile losses even in the absence of incoming wakes by shortening the laminar separation bubble while it increases them when combined with incoming wakes possibly due to the interaction of the wakes with the background turbulence. These results are in line with experimental results by Zhang and Hodson [24] but were, however, not verified against experimental data. They also found that the contribution of the wakes to total pressure loss is more important when the wake remains discrete within the blade passage, *i.e.*, when wakes do not merge or interact with previous wakes. This last finding was further verified using LES by Michelassi *et al.* in [25] where they found that a more diffused and weakened wake (due to wake-wake interaction) caused small wake dilation losses which are driven by a reduced production of turbulent kinetic energy in the blade passage area. In this latter experiments they used HiPSTAR on a 12M node mesh with 32 modes in the span-direction.

Moreover, Pichler *et al.* [26] introduced the concept of wake maturity, defining it as the difference between the mixed-out losses and the mass-averaged losses of the wakes before the leading

edge of the cascade. In their simulations they could correlate the losses generated by the wake distortion with the wake maturity. Their simulations comprised the modeling of two airfoils and two different gap sizes between them in the axial direction resulting in meshes of 105 and 118M nodes using HiPSTAR. They found that a more mixed-out incoming wake generates less passage losses what is in line with the work performed by Michelassi et al. [25].

The effect of free inlet turbulence on the profile losses was analysed by Medic and Sharma [14] for different Reynolds numbers through LES using a compressible solver with second-order in time and space discretization and a mesh with 35M cells. In their work, three aerofoils with different loading distributions were analysed and their numerical results compared against experimental data. For every Reynolds number and inlet turbulence intensity, their LES simulations under-predicted the total pressure losses and failed at capturing the change in the trend of the total pressure losses. This was mainly attributed to the difficulty in capturing the closing of the separation bubble. Nevertheless, the simulations predicted the general flow trends and the authors concluded that LES could be used to assess new aerofoil designs and study the flow physics.

Cui *et al.* [10] analysed the different features of the flow around a T106A LPT linear cascade with different levels of free-stream turbulence and incoming wakes using a second-order finite volume LES code (HYDRA) on a 21 million cell mesh. In their simulations, they captured the Klebanoff streaks over the suction surface and the correct behaviour of the separation bubble. Additionally, they discussed the appearance of Görtler vortices over the pressure side and found that, in reality, a combination of Görtler and stretched eddies from the incoming wakes were present. Overall good agreement with experimental data was found at the tested Reynolds number of $Re = 1.6 \times 10^5$ where only a very small bubble was present, but that was totally suppressed when tested at higher levels of inlet turbulence.

More recently, Ameri [9] performed implicit LES of an LPT aerofoil with no inlet turbulence using a second-order finite volume code in a 45M points and compared his results against experimental data. His simulation was fully 3D, *i.e.*, with end walls, and for two incidences (cruise incidence, $\alpha_1 = 40^\circ$ and take-off incidence, $\alpha_1 = -2.5^\circ$) at essentially the same Reynolds num-

Figure 1.3.: Sketch of total pressure loss variation with Reynolds number. The solid line is representative of a steady case whereas the dashed line is representative of unsteady cases.

ber, $Re = 2.21 \times 10^5$ and $Re = 2.15 \times 10^5$, at which a short laminar separation bubble existed. The pressure distribution around the aerofoil agreed very well for both incidences. Losses also agreed well in nominal case but differed slightly for the negative incidence case.

1.1.3. Lapse rate investigations

The dependence of the profile losses with the Reynolds number is especially relevant in aeronautical LPTs since it affects not only the conceptual design of the turbine through the chord and spacing selection but because it heavily influences the engine efficiency variation between sea level and altitude conditions. The increase of profile losses with decreasing Reynolds numbers (Reynolds lapse) has been investigated primarily by means of experiments and several attempts to predict it numerically using RANS models have been performed [2, 27, 28, 29]. Figure 1.3 sketches the behaviour of total pressure losses with Reynolds numbers. For a clean case, *i.e.* very low level of incoming turbulence and no neighbouring row perturbations, it exists a critical Reynolds, Re_c , number below which losses increase steeply. This rapid increase in loss can be explained with the separation-transition-reattachment mechanism explained earlier. This is, as the Reynolds number decreases, the bubble increases in length (the reattachment point moves downstream) and height which makes losses increase. When some level of unsteadiness is introduced, the critical Reynolds number decreases. This is because the transition process is accelerated with an earlier reattachment so that lower losses are encountered at low Reynolds

number. The situation, however, worsens at high Reynolds number where, in addition to friction losses, other loss mechanisms contribute to the overall loss. In the more recent work by Lyall *et al.* [28] the Reynolds lapse of two cascades, one front-loaded and the other aft-loaded, are tested in the wind-tunnel with inlet free stream turbulence at a variety of Reynolds numbers. For the range of tested Reynolds numbers, the front-loaded cascade did not present a change of trend of losses with Reynolds numbers whereas the aft-loaded one did. These cascades were simulated with RANS codes using different transition models. Whereas losses could be predicted accurately with one of the transition models employed for the front-loaded cascade the same was not possible for the aft-loaded case which stresses the difficulty in predicting losses at low Reynolds numbers. The fact that the authors could predict losses accurately in the front-loaded case is, likely, due to the suppression of the laminar separation bubble by the inlet turbulence. This was not the case for the aft-loaded aerofoil which resulted in inaccurate prediction of losses.

It is clear that numerical prediction of the Reynolds effects on losses is both important and difficult. The validation case presented in this thesis aims at challenging the numerical method developed within this work and, at the same time, demonstrating its capabilities to predict Reynolds lapse rate.

1.2. Numerical tools

CFD methods have been in use in the aerospace industry for over 50 years. The different methods of CFD evolved hand-in-hand with computational power. These methods, hence, have gone from being two-dimensional (2D) linearised potential solvers in the early days, to today's three-dimensional (3D) URANS solvers. Due to its low computational requirements (for today computers) RANS simulations have become the main tool in the design process of many industrial applications, particularly in the turbomachinery industry. RANS simulations are not, however, free of limitations which is the reason why offline wind-tunnel testing needs to be done before the design cycles. Wind-tunnel testing serves to assess new design concepts and eventually to calibrate the RANS models.

The limitations of RANS arise from the modelling of turbulence. In particular, in RANS simulations all scales of turbulence are modelled, and because of that, turbulent models must account for all physical effects related to turbulence, including transition. Tuning of the models allows RANS simulations to accurately predict the flow characteristics for particular configurations.

The pitfalls of turbulence modelling are alleviated with the so-called scale resolving simulations such as LES and DNS, in which all or most of the turbulent scales are resolved. In DNS all scales of turbulence are simulated whereas in LES a portion of the turbulent scales (the smallest scales) are modelled. These small scales, typically, have already lost “memory” of the larger structures dominating the flow so that the models used, usually called Sub-Grid Scale (SGS) models, are independent of the outer flow. The fact that these scales are somewhat independent of the outer scales make the models more universal. It is usually assumed that these under-resolved scales have a dissipative effect over the simulated/resolved scales (energy is drawn from the larger scales) which, for most models, impede back-scatter events. The main difficulty with SGS modelling is determining the amount of dissipation over the resolved scales which has received some attention over the last two decades ([30, 31, 32, 33]). Additionally, the modelled SGS dissipation adds to the numerical dissipation which might result in an artificially dissipative scheme.

Most SGS models work well on unbounded flows. These models, however, start failing for wall-bounded flows. Sayadi and Moin [34] performed LES of transition to turbulence. In their numerical experiments they excited the boundary layer with the appropriate Tollmien-Schlichting frequencies by blowing into the domain and examined the capabilities of different SGS models to predict transition to turbulence. The constant coefficient models of Smagorinsky [35] and Vreman [31] both fail to differentiate between laminar and turbulent flow hence damping the linear perturbations leading to transition. The dynamic versions of these models, the dynamic Smagorinsky [36] and dynamic Vreman [37], are both able to predict transition to turbulence. However, they both under-predict the shear stress in the turbulent region.

In dissipative numerical schemes, *i.e.*, up-winding schemes, numerical dissipation can also be used as a SGS model. This is based on the fact that this dissipation behaves in a desirable

manner, *i.e.*, it dissipates the small under-resolved scales whereas no numerical dissipation affects the larger scales. Since there are not explicit SGS terms in the equations this type of simulations are called in the literature in multiple ways: Implicit LES (ILES), under-resolved DNS (uDNS) or Quasi-DNS (QDNS). ILES usually face criticism due to the unphysical nature of the dissipation. In Grinstein *et al.* [38] a mathematical justification and the goodness of ILES schemes is made. In their work a large amount of examples are given where ILES provides excellent agreement with DNS data.

Simulations using ILES/uDNS have proven, nevertheless, to be of sufficient accuracy [39]. Ma *et al.* [40], in a brief of the *Center for Turbulence Research*, compared the performance of Discontinuous Galerkin scheme against a state-of-the-art non-dissipative Finite Volume method to perform under-resolved DNS. Their finite volume method was able to achieve fourth-order accuracy in regular meshes and second-order accuracy in skewed meshes. Their test problem was that of decaying homogeneous isotropic turbulence. They conclude that for smooth flows (without shock waves) the discontinuous Galerkin method performed better than their finite volume method, especially so for skewed meshes. For the case with supersonic turbulence their finite volume method outperformed the discontinuous Galerkin method. This was due to the fact that the finite volume code had a special treatment of shock waves. The discontinuous Galerkin methods was deemed good-suited for under-resolved DNS, specially in complex geometries.

The cost of the simulation depends on the amount of modelling. Hence, RANS are the low-end of the simulations whereas DNS are the most expensive since no model is used. The cost of LES can vary considerably depending on the type of flow being solved and the amount of turbulent energy modelled. It is considered that at least 80% of the turbulent energy content should be simulated [41] though this is difficult to verify *a priori* since it might not be possible to know what the turbulent energy content should be before running the simulation. There are, nevertheless, guidelines that can help constructing the meshes, for instance, in wall resolved simulations [42].

In 1997 Chapman [41] gave a series of estimates of the cost of running scale resolving simulations. He analysed the cost of running LES for a given percentage of the resolved turbulent

energy spectra. It was made clear in his work that the cost of running Wall-Resolved LES (WRLES) was close to that of the DNS due to the fact that the viscous sub-layer needs to be resolved which might be as small as 1% of the boundary layer thickness for large enough Reynolds numbers. This fact has led to numerous approaches to model part of the entirety of the boundary layer for LES. Piomelli [43] provides a summary of such efforts. Thus, WRLES is, in reality, very close to a DNS, *i.e.*, QDNS, which has been the approach followed in this work.

1.2.1. Turbulence Resolving Simulations in Low Pressure Turbines

Turbulence resolving simulations are interesting for a number of reasons such as verifying particular numerical RANS results, or the study of a particular physical flow phenomenon which would be of great difficulty by other means. With the increasing computing power of modern computers and the advent of new software paradigms, however, some turbulence-resolving simulations are now achievable in times that are comparable to the whole experiment design-setup-test-postprocess cycle, which makes it possible to complement or substitute some of experiments.

Some flow features, such as transition, are particularly difficult to deal with for a RANS solver. The effect of mispredicting transition can be dramatic, as is the case in LPT where performance is heavily dependent on Reynolds number.

Scale-resolving simulations have been applied to turbomachinery components with some success. Due to its lower Reynolds numbers, simulation of LPT profiles using high-fidelity models has become popular among researchers in the turbomachinery field, *e.g.*, [9, 44, 45, 46, 47, 14, 15, 23, 25, 26, 48, 49]. This effort has had two main outcomes: *i*) help to better understand the physics of the different mechanisms involved in these flows along experimental and RANS data and *ii*) assess the capabilities of scale-resolving solvers in predicting the performance of the studied component.

Even though some of the simulations performed to predict the performance of the component being studied have been successful, a validation effort is still required. The reason is twofold:

first, performance has been successfully assessed for single point of operation only and; the lack of extensive experimental data has made it impossible to compare different flow quantities at different locations at the same point of operation.

1.2.2. High-Order Methods

Given the necessity of properly simulating turbulence it is important to select a numerical scheme suitable to this purpose. Since in a turbulent process the larger scales of motion are broken down into smaller and smaller scales which need to be accounted for, the amount of points (or degrees of freedom) needed for a turbulent-resolving simulation can be, depending on the Reynolds number, several orders of magnitude more expensive than those of a RANS simulation. It is, therefore, convenient to devise schemes that reduce the amount of points needed to model the smaller vortical structures associated to turbulence. The grid size is usually determined by the amount of dispersion and dissipation a vortical structure undergoes when travelling through the considered mesh.

Most industrial codes today are density-based central second-order numerical schemes. Though these schemes are very well understood and have evolved to a very sophisticated level over the last thirty years they have strong constraints when used in turbulence simulations. This is mainly due to the fact that they offer poor performance on the smaller scales, *i.e.*, dispersion is high for small scales. This means that, in order to avoid dispersion, large amounts of points per wave-length are required. High-Order (HO) methods, *i.e.*, those with order greater than two [50], offer the possibility of relaxing the amount of points per wave-length due to their better dispersion/diffusion characteristics, see for instance [51].

One way in which one could achieve a HO derivative of a variable in a particular node of a mesh is to consider information from nodes beyond the ones in direct contact, *i.e.*, a stencil larger than the three-node stencil used in one-dimensional (1D) second-order schemes. This is readily done on structured meshes which enable very efficient codes. This approach is used by solvers such as HiPSTAR [52] which uses a fourth-order Padé discretization in space optimised for minimum dispersion [53]. For complex geometry, however, a spatial discretization based on

unstructured meshes is usually required. In these meshes a large stencil can result in very inefficient codes due to the large amount of memory indirections required. For these geometries it is generally best to use compact schemes. High-Order compact schemes have been given significant attention in the last two decades as demonstrated by the increasing number of publications in this area [54, 55, 44, 56, 57, 58, 59, 60, 61, 62, 50, 63].

Possibly the most known of the compact HO methods is the Discontinuous Galerkin (DG) method. It was first introduced by Reed and Hill in 1973 [64] to solve time-independent linear hyperbolic problems. The DG was further developed by Cockburn *et al.* [65] to solve the Euler equations, this is, non-linear time-dependent equations.

Another compact HO scheme is the Spectral-Differences (SD) method. This method was first developed by Kopriva and Kolas [66] and they used Chebyshev points for the nodal representation of the solution, which avoids the need for quadratures.

In 2007 Huynh [58] introduced the Flux Reconstruction (FR) scheme which is in itself a family of methods. A generalization of the method was later proposed by Vincent *et al.* [60] and Vincent *et al.* [61] where the corrected function is extended to a one and two parameter family, respectively. They prove that these family of correction functions recover those proposed by Huynh and also demonstrate that they are energy stable.

1.2.3. Challenges of High-Order Methods

Unlike central second-order differences methods, HO methods are just recently being understood. There are a number of issues preventing the adoption of HO methods for LES applications [62]:

1. Complexity of the HO solvers.
2. Generation of HO meshes.
3. Lack of robustness

High-order methods involve, generally, more involved mathematics than their second-order counterparts. This makes these codes more complex, with an increasing number of opera-

tions per degree of freedom as the order is increased. However, these type of solvers are very well-suited for many-core processors such as Graphics Processing Unit (GPU) as these methods increase the ratio of operations-to-memory accesses, which results in better code performance on these architectures.

The second of the issues mentioned above relates to mesh generation. The difficulty here is to generate a meaningful representation of the boundaries. This difficulty arises because in HO methods, the elements require of internal nodes. As demonstrated by Bassi and Rebay [67] the internal node distribution needs to, at least, capture the curvature of the surface the element is on in order to achieve meaningful results. For generic meshes with large curvatures generating this element-conformal meshes can be a difficult task [68, 69] and it is an active research topic.

The lack of robustness is a major concern. Most robustness issues in HO methods are related to aliasing and non-linear effects. These problems can be partially avoided by the addition of explicit numerical dissipation terms [70, 71, 72] or by changing the convective terms formulation to a numerically stable form [73, 74, 75].

The solutions to the previous problems are, as mentioned, active fields of research and, as such, they remain an open question. In this thesis, some of the solutions present already in the literature have been taken which, even though they might not be the best solutions, were available at the time and have enabled to put this work together.

1.3. Contribution of this Thesis

The contributions of this thesis can be summarised in the following two points:

1. Development of an **efficient, single-precision High-Order compressible aerodynamic solver** for generic flows in turbo-machinery on a **utility GPU architecture** that integrates with the tools of an industrial setting.
2. Thorough **validation** of the capabilities of High-Order solvers in the context of implicit LES for low-pressure turbine aerofoils at low Reynolds numbers.

An effort has been made to develop a HO solver that is efficient, takes advantage of the GPU architecture and that is generic for turbomachinery aerodynamic problems. The novelty lays on the fact that the code has been optimized to run on GPUs that are not designed for computation which is a big leap in terms of simulation affordability. A special treatment of round-off errors due to single-precision arithmetic has had to be implemented.

Besides the development, the code has been exhaustively validated against a rich array of experimental data. This step was necessary in order to gain confidence in LES/QDNS simulations using HO codes. This thesis contributes towards this aim as demonstrated by the fact that two of our publications have made its way to the high-impact peer-reviewed *Journal of Turbomachinery*.

1.4. Outline of this Thesis

This thesis is organized as follows. Chapter 2 is dedicated to the numerical method. An overview of the method used in this work is given including: numerical characteristics such as dissipation and dispersion are discussed and stability considerations given a time-marching scheme

Chapter 3 deals with implementation details. Despite the method being well suited for GPU architectures some strategies are needed to mitigate the deficiencies of parallel implementations, specifically communication bottlenecks are dealt with in multiple ways. Additionally, due to the small time-steps required by stability conditions and that single-precision arithmetic are used, a section is devoted to the specific treatment of round-off errors encountered in the simulation.

Chapter 4 describes the boundary condition implementation. Special attention is put on the inlet/outlet boundary conditions since these are critical for internal aerodynamic problems such as those encountered in turbomachinery.

In Chapter 5 several test cases are analysed in order to test the capabilities of the implemented code. Its ability to simulate turbulent flows is assessed here as well as the order of convergence and influence of the polynomial order used.

Chapter 6 and 7 are devoted to the validation case. The experimental and numerical setup are described in Chapter 6 along with the hypothesis made for the simulations and their justification. In Chapter 7 the results of the numerical simulations are presented and discussed.

2. Numerical Method

Contents

2.1. Introduction	19
2.2. The Flux Reconstruction Method	28
2.3. Temporal integration	35
2.4. Numerical properties	39
2.5. Discretization of the 1D Euler equations	51

2.1. Introduction

Advancements in computational power are placing high fidelity simulations such as LES and QDNS within what is affordable for industrial applications. Vortex dominated flows are simulated more efficiently with higher-order discretizations than the state-of-the-art second-order ones because of their low numerical dissipation. Indeed, the last 10 years have seen major steps in the development of compact high-order schemes [54, 50, 62]. HO methods are less dissipative in a per degree-of-freedom basis than conventional second-order methods which is a fundamental requirement for the simulation of vortex dominated flows. In this chapter, a summary of the different discretization methods is given and the basic principles behind the high-order method used in this work outlined.

2.1.1. Second-order CFD methods

In achieving high-order accuracy, one may devise algorithms that would make state-of-the-art CFD methods achieve higher orders of accuracy. However, this can be a difficult task and lead to complex and inefficient algorithms. To understand the limitations of these methods, they are described briefly next. The scalar conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0 \quad (2.1)$$

is used to illustrate the different methods.

The Finite Difference method

The finite difference (FD) method consists of approximating the spatial derivatives of the conservation law 2.1 by finite differences. These finite differences can be constructed with truncated Taylor series approximations around a particular point x_i , since $f = f(u(x))$ then

$$f(x_i \pm \Delta x) = f(x_i) \pm \frac{df(x_i)}{dx} \Delta x + \frac{1}{2!} \frac{d^2 f(x_i)}{dx^2} \Delta x^2 + \mathcal{O}(\Delta x^3) \quad (2.2)$$

Using 2.2 one can, for instance, construct an up-wind first-order derivative by particularising Eq. 2.2 for the next point upstream

$$\frac{df(x_i)}{dx} = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} + \mathcal{O}(\Delta x). \quad (2.3)$$

A second-order finite difference approximation of the first derivative can be constructed by particularizing Eq. 2.2 at both the upstream and downstream points, x_{i-1} and x_{i+1} respectively and eliminating $f(x_i)$ to yield

$$\frac{df(x_i)}{dx} = \frac{f(x_{i+1}) - f(x_{i-1}))}{x_{i+1} - x_{i-1}} + \mathcal{O}(\Delta x^2). \quad (2.4)$$

As it can be seen, the first order (up-wind) and second order (central) approximations only use neighbouring points of the one at which the derivative is being computed. These approximations are said to have a compact stencil. To achieve higher order approximations it is required to use non-compact stencils.

Once the spatial derivative is discretized it can be injected into Eq. 2.1

$$\frac{\partial u_i}{\partial t} + \frac{f(u_{i+1}) - f(u_{i-1}))}{x_{i+1} - x_{i-1}} = 0 \quad (2.5)$$

which yields the semi-discrete form of the conservation law which can now be integrated in time with a time-integration scheme.

Although finite difference methods are very efficient and their high-order extensions are straight forward they rely heavily on structured grids which limits their applicability to most industrial applications.

The Finite Volume method

Unlike finite differences, Finite Volume (FV) methods are based on the integral form of the conservation law. Integrating Eq. 2.1 in an element $\Omega_i = (x_{i-1/2}, x_{i+1/2})$ and applying the Gauss theorem to the spatial derivative term, the conservative form of the differential equation is obtained

$$\int_{\Omega_i} \left(\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0 \right) d\Omega = \int_{\Omega_i} \frac{\partial u}{\partial t} d\Omega + \int_{\partial\Omega_i} f \mathbf{i} \cdot \mathbf{n} dS \quad (2.6)$$

where $\partial\Omega$ is the boundary of the element, $\mathbf{n}$ is the outward pointing normal to the boundary and $\mathbf{i}$ is a normal vector aligned with the x axis pointing downstream. By assuming that the solution is constant within the element Eq. 2.6 resolves to

$$(x_{i+1/2} - x_{i-1/2}) \frac{\partial u_i}{\partial t} + f_{i+1/2} - f_{i-1/2} = 0 \quad (2.7)$$

where $f_{i+1/2}$ and $f_{i-1/2}$ are the fluxes at the element boundaries. These fluxes at the interfaces are replaced by numerical fluxes at such interfaces. Interface fluxes can be calculated by some

solution of the Riemann problem. For instance, for the location at $x_{i+1/2}$ the numerical flux can be computed as $f_{i+1/2}^* = \frac{1}{2} (f(x_{i+1}) + f(x_i)) - \frac{1}{2} \chi \left| \frac{\partial f}{\partial u} \right| (u_{i+1} - u_i)$ where $0 < \chi < 1$. For $\chi = 0$ a central scheme is recovered whereas for $\chi = 1$ the method reduces to a pure upwind scheme.

Once the numerical flux is calculated the discretised equation results in

$$\frac{\partial u_i}{\partial t} + \frac{f_{i+1/2}^* - f_{i-1/2}^*}{x_{i+1/2} - x_{i-1/2}} = 0 \quad (2.8)$$

which, for sufficiently smooth grids, recovers a second-order derivative of the flux. It can be seen that this formulation extends naturally to unstructured grids. However, the extension to higher-order schemes is not straightforward since the integration volume would have to include several grid points.

2.1.2. Element based methods

Contrary to the FD and FV approaches, element-based methods allow high-order discretisations in arbitrary meshes. There exist several element based methods, the most known of them being the Spectral Difference (SD), Discontinuous Galerkin (DG) and the Flux Reconstruction (FR) methods. These methods belong to the family of the so-called discontinuous schemes which have discontinuous solutions across the element interfaces. There also exists a continuous counterpart. As it will be seen next, these methods require the inversion of a mass matrix which makes them less efficient for CFD applications.

The different element-based methods are explained next. Continuous methods are described first followed by brief description of the most popular discontinuous methods.

Continuous Finite Element methods

Finite element methods divide the domain into elements in which additional points are defined (usually called Solution Points (SP)) so that the solution u^e , within the element e , can be con-

structured using some interpolating function as

$$u^e(x) = \sum_{i=0}^p u_{e,p+i} \ell_i(x) \quad (2.9)$$

where the ℓ_i are interpolating functions, u_i is the solution at the different solution points with coordinate x_i , and p the order of the element. Usually the interpolating functions are taken to be Lagrange polynomials. The fluxes can be calculated at the same solution points so that

$$f^e(x) = \sum_{i=0}^p f_{e,p+i} \ell_i(x) \quad (2.10)$$

where $f_i^e = f(u_i^e)$. The solution and flux functions can be extended to all the domain elements by considering piece-wise polynomials $N^j(x)$ such that

$$u^h(x) = \sum_{j=0}^{N-1} u_j N^j(x), \text{ and } f^h(x) = \sum_{j=0}^{N-1} f_j N^j(x) \quad (2.11)$$

where $N = pN_e$ is the number of solution or collocation points. Substituting these polynomial series into Eq. 2.1 one gets

$$\sum_{j=0}^{N-1} \frac{du_j}{dt} N^j(x) + \sum_{j=0}^{N-1} f_j^e \frac{dN^j}{dx}(x) = R^h \quad (2.12)$$

where the residual R^h has been included to note that this is an approximate solution. The residual is required to be orthogonal to a defined space of test functions

$$\int \left[\sum_{i=0}^{N-1} \frac{du_i}{dt} N^i(x) + \sum_{i=0}^{N-1} f_i \frac{dN^i}{dx}(x) \right] N^j(x) dx = 0 \quad (2.13)$$

or

$$\sum_{i=0}^{N-1} \int N^i(x) N^j(x) dx \frac{du_i}{dt} + \sum_{i=0}^{N-1} \int \frac{dN^i}{dx}(x) N^j(x) dx f_i = 0 \quad (2.14)$$

where the space of test functions have been chosen to be the same as the global piece-wise polynomials, $N^i(x)$. The previous equation can be cast in matrix form considering every point

in the domain as

$$\mathbf{M} \frac{d\mathbf{u}}{dt} + \mathbf{S}\mathbf{f} = 0 \quad (2.15)$$

where the vectors $\mathbf{u} = [u_0, \dots, u_{N-1}]^T$ and $\mathbf{f} = [f_0, \dots, f_{N-1}]^T$ contain the unknowns and fluxes at all the solution points of the domain, and the mass matrix, $\mathbf{M}$, and the stiffness matrix, $\mathbf{S}$, are defined as

$$M_{ij} = \int N^i(x) N^j(x) dx, \text{ and } S_{ij} = \int \frac{dN^i}{dx}(x) N^j(x) dx \quad (2.16)$$

Both the mass and stiffness matrices are non-diagonal matrices due to the fact that the interface nodes are shared between contiguous elements. This makes continuous finite element methods costly since a time-accurate integration requires inverting the mass matrix.

Continuous finite element methods are well suited for unstructured meshes and extend to high-order discretisations naturally. However, the problem of the mass matrix inversion puts this method at a disadvantage against discontinuous methods. Some of the discontinuous methods are briefly discussed in what follows.

Spectral Difference method

Similar to what is done in continuous finite element methods, in discontinuous finite element methods, the solution within the element is also constructed by polynomial interpolation¹ of the solution at the element solution points

$$u^e(x) = \sum_{i=0}^p u_i^e \ell_i(x) \quad (2.17)$$

where u_i^e is the solution at the internal point i of element e and where for a polynomial order p , $p + 1$ solution points are needed.

In spectral difference methods an additional set of points are defined at which the fluxes are computed. Since the fluxes are spatially derived, the interpolation polynomials are required to be of order $p + 1$ so that, when derived, the same order polynomials as that of the solution is

¹As with continuous finite element methods, Lagrange interpolation polynomials are considered here.

Figure 2.1.: Schematic of node distribution for the Spectral Difference method. ●: solution points. ■: flux points. Note that the elements have been spaced out for clarity.

achieved. It is, therefore, required that there be $p + 2$ flux points. Since solution points and flux points are located at different locations spectral difference methods are also referred to as staggered grid methods. A representation of the points layout can be seen in Fig. 2.1.

The flux function can be constructed within the element as

$$f^D(x) = \sum_{i=0}^{p+1} f_i^e \hat{\ell}_i(x) \quad (2.18)$$

where the superscript D is used to denote that this flux representation is discontinuous across the element interfaces; $\hat{\ell}_i$ are Lagrange polynomials of order $p + 1$ and different from those used to interpolate the solution (see Eq. 2.17); and f_i^e are the flux values evaluated at the flux points using the expression 2.17, *i.e.*, $f_i^e = f(u_e(x_i))$.

A necessary condition for a conservative scheme is that the flux be conserved across element interfaces. To do that the fluxes at the interface are replaced by a numerical flux such that the element flux function becomes continuous

$$f^C(x) = f_0^* \hat{\ell}_0(x) + \sum_{i=1}^p f_i^e \hat{\ell}_i(x) + f_{p+1}^* \hat{\ell}_{p+1}(x) \quad (2.19)$$

where f_0^* and f_{p+1}^* are the numerical fluxes at the interfaces computed using some Riemann solver that replace the original fluxes at the interfaces, $f^D(0)$ and $f^D(1)$ respectively. This is possible because the fluxes are computed in a different location than the solution points, more specifically at the interfaces. These numerical fluxes connect the element with its neighbours.

The main differences of the SD method with respect the Galerking are three. First, the method uses the strong formulation instead of the weak formulation of the conservation law. Secondly,

the mass matrix is the identity matrix, and no matrix inversion is required. Finally, as in all the discontinuous methods, the elements are only coupled with its neighbours.

Discontinuous Galerkin method

The discontinuous Galerkin method follows a different approach. In this method the flux function is instead interpolated within the element with the same Lagrange polynomials as those used to interpolate the solution, *i.e.*,

$$f_e^D(x) = \sum_{i=0}^p f_i^e \ell_i(x) \quad (2.20)$$

where, again, the superscript D is used to indicate that the interpolated flux is discontinuous across the element interfaces. In this method the fluxes are evaluated at the solution points. Analogously to the continuous finite element methods, a residual is defined as

$$R^e = \frac{\partial u^e}{\partial t} + \frac{\partial f_e^D}{\partial x} \quad (2.21)$$

which is required to be orthogonal to the set of basis functions

$$\int_{\Omega_e} \left(\frac{\partial u^e}{\partial t} + \frac{\partial f_e^D}{\partial x} \right) \ell_j(x) dx = 0 \quad (2.22)$$

where integrating by parts the second term

$$\int_{\Omega_e} \frac{\partial u^e}{\partial t} \ell_j(x) dx - \int_{\Omega_e} f_e^D \frac{d\ell_j}{dx} dx = - [f_e^D \ell_j(x)]_{x_0^e}^{x_p^e}. \quad (2.23)$$

The right-hand-side contains flux values at the element interface. As with the spectral difference method, these fluxes are substituted by the numerical fluxes at the interface, f^* . This leads to the nodal discontinuous Galerkin formulation in weak form

$$\int_{\Omega_e} \frac{\partial u^e}{\partial t} \ell_j(x) dx - \int_{\Omega_e} f_e^D \frac{d\ell_j}{dx} dx = - [f^* \ell_j(x)]_{x_0^e}^{x_p^e} \quad (2.24)$$

Integrating by parts again the second term on the left hand side the discontinuous Galerkin formulation in strong form is obtained:

$$\int_{\Omega_e} \frac{\partial u^e}{\partial t} \ell_j(x) dx + \int_{\Omega_e} \frac{df_e^D}{dx} \ell_j(x) dx = - [(f^* - f^D) \ell_j(x)]_{x_0^e}^{x_p^e} \quad (2.25)$$

where the term on the right-hand-side can be seen as a correction term that takes into account the flux jump at the interface. As it can be seen, the method requires the integration of the terms on the left. These integrals can be computed using high-order Gaussian quadrature rules that can be precomputed so that the integral reduces to matrix-vector operations.

Flux Reconstruction method

In the flux reconstruction approach, the discontinuous flux 2.20 is reconstructed directly so that the interface fluxes are recovered at the interface. This is, the flux function within element e is expressed as

$$f^e(x) = f_e^D(x) + \gamma_e(x) \quad (2.26)$$

where γ_e is the correction flux function and where f_e^D has the same form as in Eq. 2.20. The correction function is expressed as

$$\gamma_e(x) = (f_L^* - f_0^D) g_L(x) + (f_R^* - f_p^D) g_R(x) \quad (2.27)$$

where f_L^* and f_R^* are the numerical interface fluxes at the left and right element boundaries and g_L and g_R are the left and right correction functions. These correction functions are required to satisfy a number of conditions which are described later in this chapter. In the flux reconstruction approach the reconstructed flux is injected into Eq. 2.6 to yield, for each solution point

$$\frac{\partial u_i^e}{\partial t} + \sum_{i=0}^p f_i^e \frac{d\ell_i}{dx}(x_i) + (f_L^* - f_0^D) \frac{dg_L}{dx}(x) + (f_R^* - f_p^D) \frac{dg_R}{dx}(x) = 0. \quad (2.28)$$

As it can be seen the present method is rather simple and depending on the choice of correction function both the spectral difference and nodal discontinuous Galerkin method can be

retrieved [58] (for the linear case). In this regard, the flux reconstruction method can be regarded as a framework, because of its versatility.

As opposed to the spectral difference method, the flux reconstruction method does not require interpolating and extrapolating the solution to the flux points since these are the same as the solution points. Additionally, the size of the solution and flux arrays is different in the spectral difference method which makes its implementation more complex than that of the flux reconstruction method. Both the discontinuous Galerkin and the flux reconstruction method are very similar in terms of number of operations. The latter, however, is formulated in the strong form of the conservation form and is, therefore, clearer. For these reasons, in the present work, the flux reconstruction approach has been chosen as the basis spatial discretization. More detail of this method is given in the following sections.

2.2. The Flux Reconstruction Method

Huynh [58] introduced the FR scheme in 2007. Based on his original work, the basic theory of the FR method is discussed in this section. Some of the previous discussion about the FR method is repeated here in more detail.

The conservation law $\partial u / \partial t + \partial f / \partial x = 0$ is solved for $t > 0$ and $x \in [0, L]$, where L is the length of the computational domain, by discretising the space into N elements $\Omega_e = \Delta x_e = [x_e, x_{e+1}]$, with $e = 0 \dots N - 1$. For each of these elements, the degrees of freedom are the values of the state variable, u , at $p + 1$ Solution Points (SP) defined within the element, *i.e.*, $u_j^e = u^e(x_j, t)$ with $x_j \in \Omega_e$. The internal nodes follow a Gauss-Lobatto distribution, *i.e.*, a Gauss distribution including end-nodes². Gauss-Lobatto points can be calculated as the roots of $P'_{n-1}(x)$ where $P_{n-1}(x)$ is the Legendre polynomial of degree n . The Gauss-Lobatto point distributions for several number of points can be seen in Tab. 2.1.

²Interpolating with high-order polynomials in equispaced intervals can lead to significant interpolating errors. The Gauss node distribution avoids this issue. This distribution is usually employed in Discontinuous Galerkin methods since it provides accurate integration of the mass and stiffness matrices (see for instance [76]). As a side effect, Gauss node distributions help reduce non-linear aliasing errors [77].

Table 2.1.: Coordinates of the Gauss-Lobatto points for different number of points.

# of points	Gauss-Lobatto points, x_i
2	-1, 1
3	-1, 0, 1
4	$-1, -\frac{1}{\sqrt{5}}, \frac{1}{\sqrt{5}}, 1$
5	$-1, -\sqrt{\frac{3}{7}}, 0, \sqrt{\frac{3}{7}}, 1$

For each element we define the local coordinate, $\xi = [-1, 1]$, such that $x_j = x_e + \frac{\Delta x_i}{2} (\xi_j + 1)$. The solution within the element can now be approximated as a Lagrange polynomial expansion, i.e.:

$$u(x, t) \approx u^e(\xi, t) = \sum_{i=0}^p u_i^e(t) \ell_i(\xi) \quad (2.29)$$

where $\ell_i(\xi)$ are the Lagrange polynomials which take up the value of 1 at ξ_j and 0 for $\xi_{k \neq j}$:

$$\ell_i(x) = \prod_{j=0}^p \frac{(\xi - \xi_j)}{(\xi_i - \xi_j)} \quad (2.30)$$

It is clear that $u^e(\xi, t)$ is a polynomial of degree p .

The flux is defined as $f(u^e)$. Since, in the general case, the flux is not a linear function of u^3 , it can be approximated by Eq. 2.31:

$$f(u) \approx f_e^D(\xi, t) = \sum_{i=0}^p f(u_i^e) \ell_i(\xi) \quad (2.31)$$

where the super-index D denotes that this flux is discontinuous across the boundaries of each element Ω_e . This discontinuous flux is, again, a polynomial of order p . Since this flux definition is discontinuous across the interface, some reconstruction procedure is necessary to get a continuous flux, as it must be if the scheme is to be conservative. As highlighted by Huynh in his paper [58] there are some requirements that the corrected function, which we denote as f^C where C denotes corrected (or continuous), must comply with the following requirements:

- $f^C(x)$ must be of degree $p + 1$ so that $\partial f^C / \partial x$ is of degree p

³In the case of Euler equations the flux might not even be a polynomial, e.g., $p = (\gamma - 1)(\rho E - 1/2(\rho u)^2 / \rho)$

- $f^C(x)$ must be close to $f^D(x)$. This is, $\|f^C(x) - f^D(x)\|$ is to be minimized.
- $f^C(x)$ must take on the value of the interface fluxes at the element interface.

To compute the corrected flux, a correction flux function is defined such that $f_e^C(\xi) = f_e^D(\xi) + \gamma_e(\xi)$. Within the FR method, the flux correction function is defined as:

$$\gamma_e(\xi) = [f_0^I - f_e^D(\xi = -1)]g_L(\xi) + [f_p^I - f_e^D(\xi = 1)]g_R(\xi) \quad (2.32)$$

where the super-index I denotes values at the interface so that f_0^I and f_p^I denote the interface flux at $i = 0$ and p , respectively; and g_L and g_R are the left and right correction functions, respectively.

In his original work [58], Huynh experimented with a few different correction functions. It might appear that any function that satisfies the previous criteria would be a suitable correction function, however, not every correction function leads to stable schemes. In particular, based on a Sobolev type energy norm⁴ Vincent *et al.* [60] developed a one-parameter family of functions that they proved to be energy stable. This family of functions recovered the nodal DG and the SD formulation for specific values of the parameter. Their work was further extended in [61] to include multiple parameters. The numerical properties of the FR formulation depend on the this correction functions. Vermeire *et al.* [78] analyzed the numerical properties of these schemes as a function of the family parameters and concluded that the combination of parameters that led to the most accurate scheme is the one that recovers the nodal-based DG scheme which we use in our work. This higher precision comes at the expense of a more restrictive Courant-Friedrich-Levy (CFL) condition [79] (with respect to a SD discretisation, the CFL for the DG case is about 40% lower).

To recover the nodal DG formulation the correction functions take the form of the left and right

⁴ $\|u\| = \int \left(u^2 + c \frac{d^2 u}{dx^2} \right) dx$ Energy stability is ensured if $\frac{d}{dt} \|u\| < 0$

Figure 2.2.: Left- and right Radau polynomials for polynomial order $p = 3$. As it can be seen, the Radau polynomials are of order $p + 1$ ($p + 1$ zeros).

Radau polynomials defined as:

$$\begin{aligned} g_L &= \frac{(-1)^p}{2} (L_p - L_{p+1}) \\ g_R &= \frac{1}{2} (L_p + L_{p+1}) \end{aligned} \quad (2.33)$$

where L_p is the Legendre polynomial of degree p . A representation of the left and right Radau polynomials can be seen in Fig. 2.2. As it can be seen the left and right Radau functions are symmetric with respect to the y -axis, *i.e.*, $g_R(\xi) = g_L(-\xi)$.

At this point the only values left to define are the interface fluxes. These are usually calculated using Riemann solvers [80] of which two⁵ have been implemented in this work: the Roe [81] and the Lax-Friedrich (LF) [82]. This interface fluxes are usually expressed as:

$$\begin{aligned} f_0^I &= f^R(u^{e-1}(\xi = 1), u^e(\xi = -1)) \\ f_p^I &= f^R(u^e(\xi = 1), u^{e+1}(\xi = -1)) \end{aligned} \quad (2.34)$$

where the super-index R denotes here the Riemann solver, which takes the state values at the

⁵Although two Riemann solvers have been implemented only one of them has been used throughout this work, namely, the Roe solver.

discontinuity as input. With the interface fluxes the discretised equation can now be written as

$$\frac{\partial u_j^e}{\partial t} + \frac{d\xi}{dx} \left\{ \sum_{i=0}^p f^D(u_i^e) \frac{d\ell_i}{d\xi}(\xi_j) + [f_0^I - f^D(\xi = -1)] \frac{dg_L}{d\xi}(\xi_j) + [f_p^I - f^D(\xi = 1)] \frac{dg_R}{d\xi}(\xi_j) \right\} = 0 \quad (2.35)$$

which can now be integrated in time. Extension of the method to multiple dimensions is summarized in Appendix A.

2.2.1. Summary of the FR method

For the sake of clarity the FR method is summarised here:

1. An analytical form for u is defined using Lagrange polynomials of degree p : $u^e(\xi, t) = \sum_{i=0}^p u_i^e(t) \ell_i(\xi)$
2. Discontinuous fluxes, f_e^D , of degree p are constructed: $f_e^D(\xi, t) = \sum_{i=0}^p f(u_i^e) \ell_i(\xi)$
3. Interface fluxes, f^I , are calculated using an approximate Riemann solver.
4. A correction flux function, $\gamma(x)$, is constructed using as correction function polynomials of degree $p + 1$.
5. The derivatives of the continuous flux function, f^C , are calculated
6. The solution is advanced in time

The reconstruction procedure is also summarised in a graphical manner in Fig. 2.3. In this figure, the computational domain is split up in three elements 1 to 3. Each element has four solution points, *i.e.*, polynomials of degree $p = 3$ will be used. The values for each of these solution points are used to calculate the fluxes at these points, which are denoted by the circular symbols in the figure. These fluxes are then interpolated using Lagrange polynomials giving the discontinuous flux function represented by the black line. With the values at the interfaces of each of the elements, the interface flux can be calculated and it is represented by the squared symbols. Finally, the continuous flux function (the red line in the figure) can be constructed. The discontinuity between the first and second elements has been deliberately made large so that

Figure 2.3.: Graphical representation of the Flux Reconstruction method. Domain split in three elements. $\circ$ denote fluxes at solution points. Black solid lines denotes discontinuous flux function. $\square$ denotes interface fluxes. Red solid line denotes reconstructed continuous flux function.

it is clearly appreciated that the continuous flux function does not recover the original fluxes. As it can be seen in the third of the elements, this difference between continuous and original fluxes is barely visible as the discontinuity decreases, which is the case in well resolved cases.

2.2.2. Treatment of dissipation terms

Viscous dissipation terms of the Navier-Stokes equations involve second derivatives of the primitive variables. The present representation of the solution and flux functions does not allow for second derivatives without loss of accuracy. These terms require special treatment as introduced by Bassi and Rebay [55] in 1997. This approach, known as the BR2, is presented briefly in this section.

We consider a conservation equation with viscous terms such as the Navier-Stokes equation:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{F} - \mathbf{F}_v) = 0 \quad (2.36)$$

where the viscous fluxes are of the form $\mathbf{F}_v = \mathbf{F}_v(\mathbf{U}, \nabla \mathbf{U})$ which are a function of the unknown

vector $\mathbf{U}$ and its derivatives. To remove the dependency of the viscous fluxes on the derivatives of $\mathbf{U}$, these are replaced by an auxiliary variable $\mathbf{Q}$ so that the coupled system of equations to be solved is

$$\begin{aligned}\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{F} - \mathbf{F}_v(\mathbf{U}, \mathbf{Q})) &= 0 \\ \mathbf{Q} - \nabla \mathbf{U} &= 0\end{aligned}\tag{2.37}$$

In practice, the system can be solved in different steps. First, the auxiliary variables are solved and, then, the inviscid and viscous fluxes can be computed. In [55] the process of solving the system of equations 2.37 is described using a Galerkin approach. In essence, the process consists of achieving second derivatives while maintaining the order of polynomials. The procedure would be as follows

1. Compute common $\mathbf{U}$ values at the element interfaces as average value at the interface.
2. Make $\mathbf{U}$ continuous by applying the correction functions to $\mathbf{U}$. This achieves a distribution of polynomial order $p + 1$
3. Compute derivatives of the corrected $\mathbf{U}$ function and used them to compute the discontinuous flux function.
4. Compute interface viscous fluxes as the average of the viscous terms at the interface.
5. Correct viscous fluxes and derivate.

This is further exemplified with a one-dimensional example. Let us consider the one-dimensional case in which the conservation law is of the form

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left[f(u) - f_v \left(u, \frac{\partial u}{\partial x} \right) \right] = 0\tag{2.38}$$

The auxiliary variable q could be calculated with

$$q \approx q^D = \frac{\partial u^e}{\partial x} = \sum_{i=0}^p u_i^e \frac{d\ell_i}{dx}(x)\tag{2.39}$$

where the superscript D has been used to indicate that the auxiliary variable resulting from the direct derivation of the variable u is discontinuous. Following the flux reconstruction approach

the auxiliary variable would be made continuous with

$$q^C = \sum_{i=0}^p u_i^e \frac{d\ell_i}{dx}(x) + [u_L^I - u_0^e] \frac{dg_L}{dx}(x) + [u_R^I - u_p^e] \frac{dg_R}{dx}(x) \quad (2.40)$$

where the interface (or numerical) values are calculated with $u_L^I = (u_p^{e-1} + u_0^e) / 2$ and $u_R^I = (u_p^e + u_0^{e+1}) / 2$. The continuous viscous flux, f_v^C , can now be calculated

$$f_v^C = \sum_{i=0}^p f_v(u_i^e, q_i^C) \ell_i(x) + [f_v^L - f_v(u_0^e, q_0^C)] g_L(x) + [f_v^R - f_v(u_p^e, q_p^C)] g_R(x) \quad (2.41)$$

The interface viscous fluxes f_v^R and f_v^L at the right and left of the element, respectively, are calculated using the mean value of the flux at the interface, *i.e.*,

$$\begin{aligned} f_v^L &= \frac{1}{2} [f_v^e(u_0^e, q_0^C) + f_v^{e-1}(u_p^{e-1}, q_0^C)] \\ f_v^R &= \frac{1}{2} [f_v^e(u_p^e, q_p^C) + f_v^{e+1}(u_0^{e+1}, q_p^C)] \end{aligned} \quad (2.42)$$

The continuous viscous flux can now be derived

$$f_v^C = \sum_{i=0}^p f_v(u_i^e, q_i^C) \frac{d\ell_i}{dx}(x) + [f_v^L - f_v(u_0^e, q_0^C)] \frac{dg_L}{dx}(x) + [f_v^R - f_v(u_p^e, q_p^C)] \frac{dg_R}{dx}(x) \quad (2.43)$$

2.3. Temporal integration

The semi-discrete form of the governing equations can be written as

$$\frac{d\mathbf{u}}{dt} = \mathbf{R}(\mathbf{u}), \quad (2.44)$$

where $\mathbf{u}$ is a vector containing all the solution points and $\mathbf{R}$ the residual vector that in the general case is a non-linear function of $\mathbf{u}$. Equation 2.44 needs to be integrated in time to advance the solution from time t_n to time $t_{n+1} = t_n + \Delta t$ where Δt is the time-step. There are a myriad of time integration schemes which can be classified in implicit or explicit and present very different properties.

Time integration methods are better understood by thinking in the linearised form of Eq. 2.44

$$\frac{d\mathbf{u}}{dt} = \left(\frac{\partial \mathbf{R}}{\partial \mathbf{u}} \right)_0 \mathbf{u} \quad (2.45)$$

Assuming that the eigenvalues, λ_i and mode-shapes of $(\partial \mathbf{R} / \partial \mathbf{u})_0$ are available, which is never the case except in simple model problems, the solution of problem 2.45 can be regarded as the evolution of N independent equations of the form

$$\frac{dw_i}{dt} = \lambda_i w_i$$

where N is the number of solution points. The eigenvalues λ_i are in general complex, and w_i are the modal variables. To avoid an overloading of the nomenclature, in what follows we will use as model equation $u_t = \lambda u$, in the understanding that this is just one of the many equations that are integrated in time.

In general, the time derivative is expressed as a function of the solution at previous time-steps and the time-step itself, *i.e.*, $du/dt = \phi(\mathbf{u}^{n+1}, \mathbf{u}^n, \mathbf{u}^{n-1}, \dots, \Delta t)$. In implicit schemes, the right-hand-side in Eq. 2.44 is evaluated at t_{n+1} so the resulting scheme is of the form $\phi(\mathbf{u}^{n+1}, \mathbf{u}^n, \mathbf{u}^{n-1}, \dots, \Delta t) = \mathbf{R}(\mathbf{u}^{n+1})$ which is implicit in $\mathbf{u}^{n+1}$. For instance, the implicit Euler scheme is of the form

$$\frac{u^{n+1} - u^n}{\Delta t} = \lambda u^{n+1} \quad (2.46)$$

where the right-hand-side of Eq. 2.44 is taken to be a linear function of u , *i.e.*, $R(u^{n+1}) = \lambda u^{n+1}$ with λ a complex number. For this simple case the equation can be readily solved for u^{n+1} , *i.e.*, $u^{n+1} = u^n / (1 - \lambda \Delta t)$. The amplification factor for this case results in $G = \left| \frac{u^{n+1}}{u^n} \right| = 1 / (1 - \lambda \Delta t)$ where it is clear that $|G| < 1$ as long as $|\lambda \Delta t| < 1 - e^{I\theta}$ ⁶. This means that as long as λ has negative real part the scheme is stable for any Δt . This makes implicit schemes very attractive since the time-step can be made as big as wanted without the scheme becoming unstable. In unsteady simulations, then, the only limit to the time-step would be the temporal

⁶Here the inequality $|G| < 1$ has been solved in the complex plane with $I = \sqrt{-1}$ and θ being an angular coordinate ($re^{I\theta} = r \cos \theta + Ir \sin \theta$).

error which, for a q^{th} -order temporal scheme, would grow as Δt^q . Despite the desirable properties of implicit schemes, in practice, the implicit system requires large amounts of memory to be solved.

Explicit time integration, on the other hand, are of the form $\phi(u^{n+1}, u^n, u^{n-1}, \dots, \Delta t) = R(u^n)$ so, as long as the time-derivative function is explicit on u^{n+1} the whole system is explicit on u^{n+1} . This is convenient since now no inversion is required. *i.e.*, it is computationally inexpensive. This comes, however, with some limitations. If we consider, for instance, a forward Euler scheme, Eq. 2.44 would result in

$$\frac{u^{n+1} - u^n}{\Delta t} = \lambda u^n \quad (2.47)$$

with a linear right-hand-side function ($R(u^{n+1}) = \lambda u^n$) and where the explicit relation is readily found $u^{n+1} = (1 + \lambda \Delta t) u^n$. For this case the amplification factor results in $G = \frac{u^{n+1}}{u^n} = 1 + \lambda \Delta t$ where it is easy to see that $|\lambda \Delta t| < e^{I\theta} - 1$. If λ is real negative, then $0 < \Delta t < -2/\lambda$ which imposes a limit on the maximum time-step.

Despite their larger cost, implicit schemes have the potential to yield a speed-up factor with regards to explicit schemes. As mentioned earlier, this speed-up is limited in unsteady simulations due to accuracy restrictions. Due to this restriction and the additional complexity of its implementation an explicit solver has been implemented instead.

Runge-Kutta (RK) methods are in themselves a whole family of time integrators and can be either explicit or implicit depending on the coefficients of the different steps [83]. In RK methods the time-step is divided in stages (or quadrature points) which serve to integrate the equation. This is expressed as follows. Equation 2.44 is to be integrated numerically with the quadrature formula

$$u^{n+1} = u^n + \Delta t \sum_{i=1}^m b_i R(u(t_n + \tau_i \Delta t)) \quad (2.48)$$

where m is the number of stages and the values of u^m at the intermediate steps are calculated as

$$u^m = u(t_n + \tau_i \Delta t) = u^n + \Delta t \sum_{j=1}^s a_{ij} R(u(t_n + \tau_j \Delta t)) \quad (2.49)$$

which in general is an implicit system. Coefficients a_{ij} , b_i , and τ_j can be chosen to achieve

a certain level of accuracy, stability or other constraints for a given number of stages. Thus, different RK schemes can be obtained by choosing different values of the coefficients a_{ij} , b_i , and τ_j for a particular number of stages, m . Implicit and explicit schemes can be recovered by particular choices of these coefficients [83].

In the present work, an explicit four-step RK scheme has been chosen to march the solution in time. The motivation behind this choice is twofold. On the one hand, we have not been able to develop a quick implementation of an implicit time-integration scheme able to provide large time-steps while maintaining integration accuracy and efficiency simultaneously. On the other hand, explicit schemes are inexpensive and simple and, to date, outperform the computationally more complex implicit schemes. The scheme is as follows

$$\begin{aligned}
 u^1 &= u^n \\
 u^2 &= u^n + \frac{\Delta t}{2} R(u^1) \\
 u^3 &= u^n + \frac{\Delta t}{2} R(u^2) \\
 u^4 &= u^n + \Delta t R(u^3) \\
 u^{n+1} &= u^n + \frac{\Delta t}{6} [R(u^1) + 2R(u^2) + 2R(u^3) + R(u^4)]
 \end{aligned} \tag{2.50}$$

where the numerical superscripts denote values at that RK sub-step. The amplification factor of this scheme is given by

$$G = 1 + \lambda \Delta t + \frac{1}{2!} (\lambda \Delta t)^2 + \frac{1}{3!} (\lambda \Delta t)^3 + \frac{1}{4!} (\lambda \Delta t)^4. \tag{2.51}$$

Contour plots of the amplification factor can be seen in Fig. 2.4. The scheme is fourth-order accurate, *i.e.*, integration error converges as $\mathcal{O}(\Delta t^4)$ and the maximum dimensionless time step, $\lambda \Delta t$ is slightly below 3.0. For consistency, unsteady simulations require that temporal discretization errors are not of higher magnitude than those of spatial discretization, otherwise all the efforts made to achieve a high-order spatial discretisations are spoiled by the temporal integration. If both spatial and temporal error are of the same order, then it would be required that the error convergence rate of the temporal scheme would be at least that of the spatial

Figure 2.4.: Contour plots of amplification factor for the implemented four-step Runge-Kutta method.

discretisation. The maximum time-step imposed by stability constraints in the strongly non-uniform grids required to mesh complex configurations is so low that the temporal errors remain much lower than that of spatial discretization even for sixth-order spatial discretisations.

2.4. Numerical properties

Several researchers have contributed to the understanding of the numerical properties of these schemes [39, 84, 51]. In [39] Moura et al. give the dispersion and damping for several orders of the interpolation polynomial for the DG formulation. In their work, the dispersion and damping characteristics of the scheme are further analysed to establish the suitability of the method for ILES or QDNS of turbulent flows. In this respect, they give the effective shortest wave length that can be resolved for a given polynomial order (shortest wave-lengths suffer from numerical dissipation). The effective shortest resolved wave-length is in essence a mesh criterium, *i.e.*, the maximum mesh spacing is determined by the length of the shortest wave that one wishes to compute for a given accuracy.

In this section the linear stability and the numerical properties of the discretization are discussed. First, the one-dimensional linear wave equation is discussed followed by a discussion of the impact of the diffusion terms.

In the context of linear equations, the numerical properties of the spatial discretization depend on the real and imaginary parts of the discrete spatial operator which in this class of schemes have to be expressed in matrix form necessarily,

$$\frac{d\mathbf{u}}{dt} = A\mathbf{u} \quad (2.52)$$

which has solutions of the form $u(t) = \sum_i u_i \exp(\lambda_i t)$ where λ_i represent the eigenvalues of the discretization matrix A . If λ_i has negative real part, the component of the solution associated to the correspondent eigenvector (in the eigenvector coordinates) will decrease over time and will vanish. If it had a positive real part, the solution would grow over time. The previous is said to be stable whereas the latter is said to be unstable. A component associated to an eigenvalue with zero real part oscillates indefinitely within the domain until it is convected out of it. The real and imaginary part of these eigenvalues are also used to determine the dissipation and dispersion of the numerical discretization.

2.4.1. Discretization of the 1D linear advection equation

The linear advection equation with periodic boundary conditions is considered first. The linearised form of equation 2.1 can be expressed as

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0 \quad (2.53)$$

which has periodic solutions of the form $u = \exp(Ik(x - ct))$ where $I = \sqrt{-1}$, k is the wave number, and c is the propagation speed. In what follows spatially periodic numerical solutions to this equation of the form

$$\mathbf{u}^e = \hat{\mathbf{u}}_k(t) \exp(Ikx) \quad (2.54)$$

are sought, where the vector $\mathbf{u}^e$ contains the SPs of the element e , and the vector $\hat{\mathbf{u}}_k$ the Fourier coefficients. If the mesh is uniformly spaced with element size Δx , the x global coordinate can be written in terms of the computational internal coordinate ξ for each element as

$x^e(\xi) = e\Delta x + (\xi + 1)\Delta x/2$ which for the element $e - 1$ would be $x^{e-1}(\xi) = x^e(\xi) - \Delta x$. Extending the numerical solution 2.54 to the previous element $e - 1$ one would get $\mathbf{u}^{e-1} = \hat{\mathbf{u}}_k \exp[Ik(x - \Delta x)] = \mathbf{u}^e \exp[-Ik\Delta x]$. Taking this into consideration and considering that for the wave equation the interface flux at the left interface results in $f_L = \frac{1}{2}c(u_p^{e-1} + u_0^e) - \frac{c}{2}(u_0^e - u_p^{e-1}) = cu_p^{e-1}$ the discretisation 2.35 results in

$$\frac{du_j^e}{dt} + \frac{2c}{\Delta x} \left[\sum_{i=0}^p u_i^e \frac{d\ell_i}{d\xi}(\xi_j) + (u_p^{e-1} - u_0^e) \frac{dg_L}{d\xi}(\xi_j) \right] = 0 \quad (2.55)$$

where $u_j^e = u^e(x_j)$ and where it has also been considered that there is not contribution from the right since information travels left-to-right. The above equation can be cast in matrix form as

$$\frac{d\mathbf{u}^e}{dt} + \frac{2c}{\Delta x} (\mathbf{S}_0 \mathbf{u}^e + \mathbf{S}_{-1L} \mathbf{u}^{e-1}) = 0 \quad (2.56)$$

where $\mathbf{u}^e = [u_0^e \dots u_p^e]^T$ is the vector with the solution at each of the SPs of element e , and $\mathbf{S}_0$ and $\mathbf{S}_{-1}$ are $(p+1) \times (p+1)$ square matrices containing the contribution of the element e and $e - 1$ respectively. Taking into account that $\mathbf{u}^e = \hat{\mathbf{u}}_k e^{Ikx}$, and injecting this expression in Eq. 2.56 it is readily obtained that

$$\frac{d\hat{\mathbf{u}}_k}{dt} = -\frac{2c}{\Delta x} (\mathbf{S}_0 + \mathbf{S}_{-1L} e^{-Ik\Delta x}) \hat{\mathbf{u}}_k = -\mathbf{S} \hat{\mathbf{u}}_k \quad (2.57)$$

where the matrix $\mathbf{S}_0$ contains information both from the discontinuous flux interpolation process, $\mathbf{S}_{00}$, and the corrected flux coming from the Riemann solver, $\mathbf{S}_{0,-1L}$,

$$\mathbf{S}_0 = \mathbf{S}_{00} + \mathbf{S}_{0,-1L} = \begin{bmatrix} \frac{d\ell_0(\xi_0)}{d\xi} & \frac{d\ell_1(\xi_0)}{d\xi} & \dots & \frac{d\ell_{p-1}(\xi_0)}{d\xi} & \frac{d\ell_p(\xi_0)}{d\xi} \\ \frac{d\ell_0(\xi_1)}{d\xi} & \frac{d\ell_1(\xi_1)}{d\xi} & \dots & \frac{d\ell_{p-1}(\xi_1)}{d\xi} & \frac{d\ell_p(\xi_1)}{d\xi} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{d\ell_0(\xi_{p-1})}{d\xi} & \frac{d\ell_1(\xi_{p-1})}{d\xi} & \dots & \frac{d\ell_{p-1}(\xi_{p-1})}{d\xi} & \frac{d\ell_p(\xi_{p-1})}{d\xi} \\ \frac{d\ell_0(\xi_p)}{d\xi} & \frac{d\ell_1(\xi_p)}{d\xi} & \dots & \frac{d\ell_{p-1}(\xi_p)}{d\xi} & \frac{d\ell_p(\xi_p)}{d\xi} \end{bmatrix} - \begin{bmatrix} \frac{dg_L(\xi_0)}{d\xi} & 0 & \dots & 0 & 0 \\ \frac{dg_L(\xi_1)}{d\xi} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{dg_L(\xi_{p-1})}{d\xi} & 0 & \dots & 0 & 0 \\ \frac{dg_L(\xi_p)}{d\xi} & 0 & \dots & 0 & 0 \end{bmatrix} \quad (2.58)$$

while the matrix $\mathbf{S}_{-1L}$ contains only information from the upstream cell in the flux correction process

$$\mathbf{S}_{-1L} = \begin{bmatrix} 0 & 0 & \cdots & 0 & \frac{dg_L(\xi_0)}{d\xi} \\ 0 & 0 & \cdots & 0 & \frac{dg_L(\xi_1)}{d\xi} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \frac{dg_L(\xi_{p-1})}{d\xi} \\ 0 & 0 & \cdots & 0 & \frac{dg_L(\xi_p)}{d\xi} \end{bmatrix} \quad (2.59)$$

The matrix of the spatial operator $\mathbf{S} = \mathbf{S}_0 - \mathbf{S}_{-1}e^{Ik\Delta x}$ has $p + 1$ families of eigenvalues, $\lambda_i(\hat{k})$, which depend on the reduced wave number $\hat{k} = k\Delta x$.

2.4.1.1. Numerical properties of the 1D linear equation

The eigenvalues of $\mathbf{S}$ are used to examine the numerical properties of the FR discretisation. These eigenvalues are compared against the exact solution to determine the discretisation goodness. The exact solution of the 1D linear wave equation 2.53 subject to periodic boundary conditions is

$$\frac{d\hat{u}_k}{dt} = -Ick\hat{u}_k \quad (2.60)$$

which has the exact eigenvalue $\lambda_{\text{exact}} = -Ick$. By comparing the eigenvalues of the discretised equation against the exact solution the numerical properties can be evaluated. Figure 2.5⁷ plots the value of the imaginary part of the exact and discretised equations for $p = 3$ and $p = 5$ along with the exact relation and the relation for a centred differences scheme. Since there is a larger number of points (or degrees of freedom) for each high-order element length, Δx , it is instinctively obvious that they should resolve higher frequencies than lower order ones. This makes it necessary to scale the eigenvalues with the number $p + 1$. This is, the numerical properties are examined in a “per degree of freedom” basis which makes a fairer comparison.

As it can be seen in Fig. 2.5, the exact eigenvalue is linear with the wave-number k . The eigenvalue of the discretised equation, however, goes linear up to values of $k\Delta x/(p + 1) \approx 1.5$ and then it departs from the analytical relation. Since the dispersion relation is associated with the

⁷This plot is usually referred to as the phase velocity plot or dispersion relation plot.

propagation speed this means that only wave numbers up to this value will be resolved without being dispersed, *i.e.*, with the correct propagation speed. Although barely appreciable in the figure, the dispersion line for $p = 5$ runs much closer to the exact relation up to $k\Delta x/(p+1) \approx 1.5$ meaning that better accuracy is to be expected per degree of freedom for higher order solution. This effect will manifest itself in a numerical example in Chapter 5.

The differences in wave-number resolution of different schemes is depicted in Fig. 2.6. In this figure different waves are plotted with different wave-numbers in relation to the element size, Δx . At the left, a single fourth-order element ($p = 3$) is considered with a bad-resolved wave with wave-number $k\Delta x/(p+1) = \pi$ (dashed line) and a well-resolved wave with wave-number $k\Delta x/(p+1) = \pi/2$ (solid line) whereas at the right, four equally-spaced second-order elements for a central-differences scheme is considered with bad-resolved waves with wave-numbers $k\Delta x = \pi$ (dashed line) and $k\Delta x = \pi/2$ (solid line). As it can be seen, one fourth-order element the size of four second-order elements is capable of resolving wave-lengths that are four second-order elements cannot.

Figure 2.7 shows the real part of the eigenvalues of the discretised equation 2.57 for $p = 3$ and $p = 5$. As it can be seen, there is little (almost negligible) dissipation up to values of $k\Delta x/(p+1) \approx 1.5$. Again, since the exact solution has no dissipation, this means that only wave numbers up to this value will be resolved accurately. As with dispersion characteristics, the sixth-order dissipation line is closer to the x -axis near the $k\Delta x/(p+1) \approx 1.5$ region than the fourth-order case. In Chapter 5 a case will be presented where this effect can be better appreciated.

It is seen in both Fig. 2.5 and 2.7 that the numerical solution has very good behaviour up until $k\Delta x/(p+1) \approx 1.5$. In comparison against standard lower-order methods such as the central differences method, the present FR behaves more accurately per degree of freedom. This can be seen in Fig. 2.5 where the dispersion relation of a second-order central difference scheme has been included. Unlike for the higher-order method, the behaviour of the central differences scheme starts to worsen for values of $k\Delta x \approx 0.5$.

Figure 2.5.: Dispersion relation of the 1D linear wave equation discretised with the FR scheme with $p = 3$ and $p = 5$. The exact relation and the second-order central differences scheme have been added for reference.

Fourth-order element

Centred differences

Figure 2.6.: Schematic representation of waves of different wave-number in relation to the element size Δx . Left: single fourth-order element ($p = 3$) with representation of the bad-resolved wave-number $k\Delta x/(p+1) = \pi$ (dashed line) and well-resolved wave-number $k\Delta x/(p+1) = \pi/2$ (solid line). Right: four equally-spaced second-order mesh elements for central differences with bad-resolved wave-number $k\Delta x = \pi$ (dashed line) and $k\Delta x = \pi/2$ (solid line).

Figure 2.7.: Dissipation of the 1D linear wave equation discretized with the FR scheme with $p = 3$ and $p = 5$.

2.4.1.2. Stability

The stability of the numerical scheme is performed for the particular time integrator discussed earlier, this is an explicit four-step fourth-order Runge-Kutta (RK) scheme.

Because the stability regions of temporal integration schemes depend on the complex variable $\lambda \Delta t$, in the following, the eigenvalues of the discretized equations will always be multiplied by Δt so that they can be plotted against the stability region. By defining the CFL number as $CFL = c\Delta t/\Delta x$, it can be seen that the eigenvalues of matrix $\mathbf{S}$ in Eq. 2.57 are scaled with $2CFL$.

Figure 2.8 plots the eigenvalues of Eq. 2.57 for the fourth order discretization, $p = 3$, against the stability region of the RK4 scheme for a CFL number of $CFL = 0.12$, which is the limit CFL number found by Huynh in his work [58] for this same scheme. It can be seen that the eigenvalues are indeed inside the stability area of the temporal scheme for this CFL number.

2.4.1.3. Discussion about the eigenvalues of the discretized equation

The discretised spatial operator of the linear wave equation 2.57 has $p + 1$ different eigenvalues per wave-number, k . This raises the question of which of these eigenvalues should be considered to analyse the stability and the numerical properties of the scheme. Some insight into what these eigenvalues and their associated eigenvectors mean is given next.

Figure 2.8.: Footprint of the dimensionless spatial eigenvalues, $\lambda\Delta t$, of the 1D linear wave equation for $p = 3$ and $CFL = 0.12$ (red line) and the stability limit, $|G| = 1$, of a four-step RK scheme (black line).

Different interpretations have been given to the eigenmodes associated to each of the eigenvalues of matrix $\mathbf{S}$ [39, 85, 51]. In summary, these interpretations agree with the following reasoning. Each solution with wave-number k can be projected onto the space of the eigenvectors of matrix $\mathbf{S}$ (see example in Fig. 2.9). Since each eigenvector has a different eigenvalue associated to it, each of the components of this projection will have a different numerical behaviour. An example of this can be seen in Fig. 2.9 where a single harmonic has been projected onto the eigenvector space in a discretization with three nodes per cell ($p = 2$); and their associated eigenvalues have been placed on the complex plane along with the stability region of a four-step fourth-order RK scheme. As it can be seen in this figure, most of the energy associated with the solution is carried by the first eigenmode whereas the remaining two eigenmodes carry little energy compared with the total energy. Additionally, it can be seen that the eigenvalue associated to the energy carrying eigenmode has very little dissipation (small negative real part) so that it will be integrated in time more accurately. The other two, however, have large levels of dissipation (specially the third eigenmode) and so they will be damped in a few time-iterations. If the initial solution was a higher harmonic most of the energy would be contained in the second and third modes. These have higher dissipation rates and so after some time-iterations their energy would be dissipated and only the little energy carried by the first mode would survive. The former case would be said to be a resolved wave-length whereas the latter would be said to be

Figure 2.9.: Projection of a single harmonic ($k\Delta x \approx \frac{\pi}{6}$) onto the eigenvector space for a third-order discretization ($p = 2$).

an under-resolved wave-length.

2.4.2. The 1D linear advection-diffusion equation

The analysis of the 1D case is extended here to the linear convection-diffusion equation to study the role of diffusion in stability. The model equation reads:

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} \quad (2.61)$$

where ν is a constant viscosity coefficient. Though in general viscous terms are small they play a relevant role in particular in thin regions such as boundary layers and wakes.

The convective term in Eq. 2.61 is modelled exactly in the same way as in Eq. 2.55, so only modelling of the dissipative term is left. This will be done following the strategy of Bassi and Rebay [55] explained previously.

To discretise the first derivative of u , first u is reconstructed to make it continuous. In this case,

the first derivative for the i th node in the element has the form:

$$\frac{du_j^e}{dx} = \frac{2}{\Delta x} \left[\sum_{i=0}^p u_i^e \frac{d\ell_i}{d\xi}(\xi_j) + (u_R^I - u_p^e) \frac{dg_R}{d\xi}(\xi_j) + (u_L^I - u_0^e) \frac{dg_L}{d\xi}(\xi_j) \right] \quad (2.62)$$

Unlike convection, diffusion is not directional so that both the correction from the left and right elements are kept in Eq. 2.62. The left and right interface values in 2.62 (u_L^I and u_R^I) are defined following [55], that is averaging u at the interface

$$u_R^I = \frac{1}{2} (u_0^{e+1} + u_p^e), \text{ and } u_L^I = \frac{1}{2} (u_p^{e-1} + u_0^e) \quad (2.63)$$

Substituting 2.63 in 2.62:

$$\begin{aligned} \frac{du_j^e}{dx} = \frac{2}{\Delta x} & \left[\sum_{i=0}^p u_i^e \frac{d\ell_i}{d\xi}(\xi_j) + \frac{1}{2} (u_0^{e+1} - u_p^e) \frac{dg_R}{d\xi}(\xi_j) \right. \\ & \left. + \frac{1}{2} (u_p^{e-1} - u_0^e) \frac{dg_L}{d\xi}(\xi_j) \right] \end{aligned} \quad (2.64)$$

which can be cast in matrix form as:

$$\frac{d\mathbf{u}^e}{dx} = \frac{2}{\Delta x} \left[\mathbf{S}_{00} - \frac{1}{2} (\mathbf{S}_{0,-1L} + \mathbf{S}_{0,+1R}) \right] \mathbf{u}^e + \frac{1}{2} \mathbf{S}_{1,-1R} \mathbf{u}^{e+1} + \frac{1}{2} \mathbf{S}_{-1,+1L} \mathbf{u}^{e-1} \quad (2.65)$$

Following the approach of section 2.4.1 and seeking for periodic solutions of the form $\mathbf{u}^e(x) = \hat{\mathbf{u}}_k e^{Ikx}$

$$\frac{d\hat{\mathbf{u}}_k}{dx} = \frac{2}{\Delta x} \left\{ \left[\mathbf{S}_{00} - \frac{1}{2} (\mathbf{S}_{0,-1L} + \mathbf{S}_{0,+1R}) \right] + \frac{1}{2} (\mathbf{S}_{1,-1R} e^{I\hat{k}} + \mathbf{S}_{-1,+1L} e^{-I\hat{k}}) \right\} \hat{\mathbf{u}}_k = \frac{2}{\Delta x} \mathbf{D} \hat{\mathbf{u}}_k$$

The second derivative follows the same logic as the first one so that we can express it as:

$$\begin{aligned} \frac{d^2 u_j^e}{dx^2} = \frac{2}{\Delta x} & \left[\sum_{i=0}^p \frac{du_i^e}{dx} \frac{d\ell_i}{d\xi}(\xi_j) + \left(\left(\frac{du}{dx} \right)_R^I - \frac{du_p^e}{dx} \right) \frac{dg_R}{d\xi}(\xi_j) \right. \\ & \left. + \left(\left(\frac{du}{dx} \right)_L^I - \frac{du_0^e}{dx} \right) \frac{dg_L}{d\xi}(\xi_j) \right] \end{aligned} \quad (2.66)$$

Figure 2.10.: Effect of the $Re_{\Delta x}$ on the fourth-order discretisation eigenvalues for $CFL_{inv} = 0.12$. The dashed black line encloses the stability region of the RK scheme.

As in equation 2.62 the values at the interface are taken as the mean value between the two sides of the interface, that is, the relations 2.63 are applied also to the first derivatives. This is equivalent to applying twice the same spatial operator and, therefore, can be cast in matrix form in the same manner as equation 2.65:

$$\frac{d^2 \hat{\mathbf{u}}_k}{dx^2} = \left(\frac{2}{\Delta x^2} \mathbf{D} \right)^2 \hat{\mathbf{u}}_k \quad (2.67)$$

The Fourier transformed version of the semi-discrete form of the continuous equation 2.61 becomes as

$$\frac{d \hat{\mathbf{u}}_k}{dt} = -\frac{2c}{\Delta x} \left(\mathbf{S}_0 - \mathbf{S}_{-1L} e^{Ik\Delta x} \right) \hat{\mathbf{u}}_k + \frac{4\nu}{\Delta x^2} \mathbf{D}^2 \hat{\mathbf{u}}_k = \frac{2c}{\Delta x} \left[-\left(\mathbf{S}_0 - \mathbf{S}_{-1L} e^{Ik\Delta x} \right) + \frac{2}{Re_{\Delta x}} \mathbf{D}^2 \right] \hat{\mathbf{u}}_k \quad (2.68)$$

where the element-based Reynolds number, $Re_{\Delta x}$, has been defined as $Re_{\Delta x} = c\Delta x/\nu$.

Figure 2.10 shows how the cell Reynolds number affects the eigenvalues of the 1D advection equation (case $Re_{\Delta x} = \infty$) for the fourth-order discretization ($p = 3$). As the cell Reynolds number decreases, the real part of the eigenvalues grows towards more negative values,. As it can be seen, this can push the eigenvalues out of the stability region. A further scaling is,

Figure 2.11.: Scaled fourth-order discretisation eigenvalues for different values of the $Re_{\Delta x}$. The CFL used for each $Re_{\Delta x}$ follows the expression $CFL = CFL_{inv} \left(1 + \frac{2\sigma_{max}}{Re_{\Delta x}^2} \right)^{-1}$. The dashed black line encloses the stability region of the RK scheme.

therefore, necessary to keep them within the stability area. The proposed scaling is as follows

$$CFL = CFL_{inv} \left(1 + \frac{2\sigma_{max}}{Re_{\Delta x}^2} \right)^{-1} \quad (2.69)$$

where CFL_{inv} denotes the maximum CFL used in inviscid simulations and σ_{max} denotes the maximum of the eigenvalues of matrix D^2 in 2.68 (which are all real positive). The scaled eigenvalues are depicted in Fig. 2.11 where it is seen that they all fall back into the stable region. The maximum stable time-step follows from the previous definition 2.69

$$\Delta t = CFL_{inv} \frac{\Delta x}{c} \left(1 + \frac{2\sigma_{max}}{Re_{\Delta x}^2} \right)^{-1} \quad (2.70)$$

2.5. Discretization of the 1D Euler equations

2.5.1. The one-dimensional Euler equations

The eigenvalue analysis has been extended to the one-dimensional linearized Euler equations which can be written in conservative form as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0$$

with

$$\mathbf{U} = \begin{Bmatrix} \rho \\ \rho u \\ \rho E \end{Bmatrix}; \mathbf{F} = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ u(\rho E + p) \end{Bmatrix} \quad (2.71)$$

where ρ , u and E are the density, velocity and total energy of the fluid; and p is the pressure which is related to ρ , u and E through the perfect gas equation $p = (\gamma - 1)\rho(E - \frac{1}{2}u^2)$ with γ the ratio of specific heats. To perform the stability analysis the Euler equations are linearized around a steady state uniform solution $\mathbf{U}_0$ such that $\mathbf{U} = \mathbf{U}_0 + \mathbf{u}(x, t)$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{A}_0 \frac{\partial \mathbf{u}}{\partial x} = 0 \quad (2.72)$$

where the Jacobian $\mathbf{A}_0 = (\partial \mathbf{F} / \partial \mathbf{U})_0$ is evaluated at state $\mathbf{U}_0$, and

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 1 & 0 \\ \frac{\gamma-3}{2}u_0^2 & (3-\gamma)u_0 & \gamma-1 \\ -\gamma u_0 E_0 + (\gamma-1)u_0^3 & \gamma E_0 - \frac{3}{2}(\gamma-1)u_0^2 & \gamma u_0 \end{bmatrix} \quad (2.73)$$

Matrix $\mathbf{A}_0$ can be diagonalised as $\Lambda = \mathbf{T}^{-1} \mathbf{A}_0 \mathbf{T}$ where Λ is a diagonal matrix whose elements are the eigenvalues of matrix $\mathbf{A}_0$; and $\mathbf{T}$ a matrix whose columns are the associated right eigenvectors. Taking the previous into account, the system of equations 2.72 can be rewritten

as

$$\frac{\partial \mathbf{w}}{\partial t} + \Lambda \frac{\partial \mathbf{w}}{\partial x} = 0 \quad (2.74)$$

with $\mathbf{w} = \mathbf{T}^{-1}\mathbf{u}$. This transformed system of equations is diagonal and each equation is an independent wave equation. Even though the diagonalised form of the Euler equations (Eq. 2.74) will not be used in the computation of the numerical eigenvalues it helps to understand the behaviour of the Euler equations. As it has previously been mentioned the one-dimensional Euler equations reduce to three wave equations with propagation speeds u_0 , $u_0 - a_0$ and $u_0 + a_0$ where u_0 is the convection velocity and a_0 is the speed of sound. The characteristic variables associated to each of these velocities are the entropy and downstream and upstream acoustic waves.

2.5.2. The Riemann problem

Since they will be needed in the discretization of the linearized Euler equations, some comments about the interface fluxes are made here. As mentioned in Section 2.2, the interface fluxes are taken to be a solution of the Riemann problem. Out of the several Riemann solutions [80] those by Roe [81] and LF [82] are considered here for this analysis. In the simulations performed within this thesis, only the Roe solver has been used since it is slightly less dissipative (as it will be shown).

The general form of these solutions of the Riemann problem is of the form

$$\mathbf{F}^I = \frac{1}{2} (\mathbf{F}^R + \mathbf{F}^L) - \frac{|\mathbf{A}_0|}{2} (\mathbf{U}^R - \mathbf{U}^L) \quad (2.75)$$

where the superscripts, I , R and L stand for quantities at the Interface, to the Right and to the Left of the element, respectively. Matrix $|\mathbf{A}_0|$ is constructed based on the absolute values of the eigenvalues of $\mathbf{A}_0$. Depending on which of the two solvers is considered, two different methods of constructing matrix $|\mathbf{A}_0|$ will be used.

For a Roe-type solver, matrix $|\mathbf{A}_0|$ is constructed as $|\mathbf{A}_0| = \mathbf{T}|\Lambda|\mathbf{T}^{-1}$ where $|\Lambda|$ is the diagonal matrix containing the absolute value of the eigenvalues of $\mathbf{A}_0$ in its diagonal, $|\Lambda| =$

$\text{diag}(|u_0| \ |u_0 - a_0| \ |u_0 + a_0|)$. For a LF type solver the matrix $|\mathbf{A}_0|$ is constructed as $|\mathbf{A}_0| = \mathbf{T}|\mathbf{\Lambda}_{max}|\mathbf{T}^{-1}$ where $|\mathbf{\Lambda}_{max}|$ is the diagonal matrix containing the maximum eigenvalue of $|\mathbf{A}_0|$ in its diagonal, $|\mathbf{\Lambda}_{max}| = (|u_0| + |a_0|)\mathbf{I}$. Each of the Riemann solvers yields different eigenvalue footprints on the $\lambda \triangle t$ plane.

2.5.3. Numerical discretization 1D

It can be seen that equation 2.72 is equivalent to equation 2.53 in 1D. In an analogous fashion the corrected flux function can be expressed as

$$\mathbf{F}(\xi) = \sum_{i=0}^p \mathbf{A}_0 \mathbf{u}_i \ell_i(\xi) + [\mathbf{F}_L^I - \mathbf{A}_0 \mathbf{u}_{i=0}] g_L(\xi) + [\mathbf{F}_R^I - \mathbf{A}_0 \mathbf{u}_{i=p}] g_R(\xi) \quad (2.76)$$

where by substituting the value of the interface value 2.75 and deriving with respect to the spatial coordinate becomes

$$\begin{aligned} \frac{d\mathbf{F}_j}{d\xi} = \sum_{i=0}^p \mathbf{A}_0 \mathbf{u}_i \frac{d\ell_i(\xi_j)}{d\xi} + \frac{1}{2} (\mathbf{A}_0 + |\mathbf{A}_0|) (\mathbf{u}^L - \mathbf{u}_{i=0}) \frac{dg_L(\xi_j)}{d\xi} \\ + \frac{1}{2} (\mathbf{A}_0 - |\mathbf{A}_0|) (\mathbf{u}^R - \mathbf{u}_{i=p}) \frac{dg_R(\xi_j)}{d\xi} \end{aligned} \quad (2.77)$$

Here, again, we consider periodic solutions of the perturbations of the type $\mathbf{u}^e = \hat{\mathbf{u}}^k \exp(ikx)$ so that the relation with the element to the left and right is $\mathbf{u}^{e-1} = \mathbf{u}^e \exp(-Ik\triangle x)$ and $\mathbf{u}^{e+1} = \mathbf{u}^e \exp(Ik\triangle x)$, respectively, and substituting in 2.77

$$\begin{aligned} \frac{dF_j}{d\xi} = \sum_{i=0}^p \mathbf{A}_0 \mathbf{u}_i \frac{d\ell_i(\xi_j)}{d\xi} + \frac{1}{2} (\mathbf{A}_0 + |\mathbf{A}_0|) (\mathbf{u}_{i=p} \exp(-Ik\triangle x) - \mathbf{u}_{i=0}) \frac{dg_L(\xi_j)}{d\xi} \\ + \frac{1}{2} (\mathbf{A}_0 - |\mathbf{A}_0|) (\mathbf{u}_{i=0} \exp(Ik\triangle x) - \mathbf{u}_{i=p}) \frac{dg_R(\xi_j)}{d\xi} \end{aligned} \quad (2.78)$$

which can be cast in matricial form when considered for all solution points of the element. The discretized Eq. 2.72 is finally written as

$$\frac{d\{\mathbf{u}^e\}}{dt} = - \left([\mathbf{A}_0] [\mathbf{D}] + \frac{1}{2} [\mathbf{A}_0 + |\mathbf{A}_0|] [C_L] + \frac{1}{2} [\mathbf{A}_0 - |\mathbf{A}_0|] [C_R] \right) \{\mathbf{u}^e\} \quad (2.79)$$

Figure 2.12.: Eigenvalues of the discretized 1D Euler equations for $M = 0.3$ for two different Riemann solvers: Roe-type solver (red dots) and Lax-Friedrich (black dots). Black dashed line encloses the stable region of the RK4 scheme.

where $[\mathbf{A}_0]$ and $[\mathbf{A}_0 \pm |\mathbf{A}_0|]$ are block-diagonal matrices containing $\mathbf{A}_0$ and $\mathbf{A}_0 \pm |\mathbf{A}_0|$ in the diagonal, respectively; matrix $[\mathbf{D}]$ is the discontinuous derivative matrix and $[C_L]$ and $[C_R]$ are the left and right correction matrices, respectively.

Figure 2.12 shows the numerical eigenvalues of the discretized Euler equations in 1D for a Mach number $M = u_0/a_0 = 0.3$ and the two Riemann solvers considered. The CFL number with which these eigenvalues have been scaled is defined analogously as with the wave equation, i.e., $\text{CFL} = (a_0 + |u_0|) \Delta t / \Delta x$. Each family of eigenvalues has been associated with the different characteristic variables, i.e. entropy and up- and downstream running acoustic waves.

It can be seen that when using a Roe-type solver the three distinct families of eigenvalues appear with a behavior identical to that of the 1D case which means that this method recovers the diagonal system 2.74. Because the same CFL number has been used for all three equations and each has a different propagation speed, the eigenvalues are scaled differently.

For an LF solver the families of eigenvalues are distorted with respect to the 1D case, except for the downstream acoustic wave family which coincides with that family of the Roe solver. Further, the entropy family splits into two curves, with more dissipative eigenvalues for the least resolved wave-lengths.

In practice, these differences have little effect on stability since the eigenvalues with the largest negative real part are still at some distance from the stability limit. The time-step is, therefore, not affected by the type of Riemann solver used. The LF Riemann solver, however, has higher level of dissipation in the entropy and upstream running acoustic modes.

3. Method Implementation

Contents

3.1. Navier-Stokes equations	57
3.2. Hardware and Software architecture	61
3.3. Implementation of the Operators	64
3.4. Parallel use of multiple GPUs	72
3.5. Explicit treatment of round-off errors	77

In Chapter 2 the FR method and its numerical properties were introduced in the context of the linear wave equation, with and without dissipation, and the Euler equations. In this chapter, the Navier-Stokes equations are presented along with details of their implementation on Graphics Processor Units. Several aspects of this implementation are described in this chapter.

3.1. Navier-Stokes equations

The differential form of the Navier-Stokes equations is:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} = \frac{\partial \mathbf{F}_v}{\partial x} + \frac{\partial \mathbf{G}_v}{\partial y} + \frac{\partial \mathbf{H}_v}{\partial z} \quad (3.1)$$

being $\mathbf{U} = (\rho \ \rho u \ \rho v \ \rho w \ \rho E)^T$ the conservative variables,

$$[\mathbf{F} \ \mathbf{G} \ \mathbf{H}] = \begin{bmatrix} \rho u & \rho v & \rho w \\ \rho u^2 + p & \rho uv & \rho uw \\ \rho uv & \rho v^2 + p & \rho vw \\ \rho uw & \rho vw & \rho w^2 + p \\ u(\rho E + p) & v(\rho E + p) & w(\rho E + p) \end{bmatrix} \quad (3.2)$$

the inviscid fluxes and

$$[\mathbf{F}_v \ \mathbf{G}_v \ \mathbf{H}_v] = \begin{bmatrix} 0 & 0 & 0 \\ \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \tau_{yy} & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \tau_{zz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x & u\tau_{xy} + v\tau_{yy} + w\tau_{yz} - q_y & u\tau_{xz} + v\tau_{yz} + w\tau_{zz} - q_z \end{bmatrix} \quad (3.3)$$

the viscous fluxes. The fluid is assumed to behave as a perfect gas, thus,

$$E = C_v T + \frac{1}{2} (u^2 + v^2 + w^2) \quad (3.4)$$

where C_v is the specific heat at constant volume. The pressure can be expressed as a function of the conservative variables:

$$p = (\gamma - 1) \left[\rho E - \frac{1}{2\rho} \left((\rho u)^2 + (\rho v)^2 + (\rho w)^2 \right) \right] \quad (3.5)$$

where $\gamma = C_p/C_v$ is the relation between the specific heats, at constant pressure and constant volume. The viscous stress tensor expression in Einstein notation is

$$\tau_{ij} = \mu (\partial_i v_j + \partial_j v_i) - \frac{2}{3} \mu (\nabla \cdot \mathbf{v}) \delta_{ij} \quad (3.6)$$

and the heat flux vector is expressed using the Fourier's law: as

$$\mathbf{q} = -k\nabla T \quad (3.7)$$

In the preceding equations, Eq. 3.6 and 3.7, the constants μ and k are the laminar viscosity and the conductivity, respectively. The laminar viscosity is temperature dependent and is given by the Sutherland's law

$$\mu = \frac{1.458 \cdot 10^{-6} T^{3/2}}{T + 110.4} \quad (3.8)$$

and the thermal conductivity is

$$k = \frac{\mu C_p}{Pr} \quad (3.9)$$

where the Prandtl number for air is approximately 0.71.

3.1.1. Navier-Stokes equations in the FR Context

In this section, the implementation of the NS equations within the FR framework is described. The fluid domain is discretised using hybrid unstructured grids [86, 87, 88]. For each grid cell a series of SPs are defined at which the state variable $\mathbf{U}$ is evaluated. As explained in Chapter 2, the solution within the cell is approximated using Lagrange polynomials ℓ_i of order p . For convenience the spatial dimensions (x, y, z) are mapped onto standard element of coordinates (ξ, η, ζ) , and a transformation between them is defined using the Jacobian matrix

$$\mathbf{J} = \begin{bmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{bmatrix} \quad (3.10)$$

and its inverse

$$\mathbf{J}^{-1} = \begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix}. \quad (3.11)$$

Equation 3.1 is then expressed as

$$|\mathbf{J}| \frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \tilde{\mathbf{F}}}{\partial \xi} + \frac{\partial \tilde{\mathbf{G}}}{\partial \eta} + \frac{\partial \tilde{\mathbf{H}}}{\partial \zeta} = 0 \quad (3.12)$$

where

$$\begin{aligned} \tilde{\mathbf{F}}(\mathbf{U}) &= |\mathbf{J}| [\xi_x (\mathbf{F} - \mathbf{F}_v) + \xi_y (\mathbf{G} - \mathbf{G}_v) + \xi_z (\mathbf{H} - \mathbf{H}_v)] \\ \tilde{\mathbf{G}}(\mathbf{U}) &= |\mathbf{J}| [\eta_x (\mathbf{F} - \mathbf{F}_v) + \eta_y (\mathbf{G} - \mathbf{G}_v) + \eta_z (\mathbf{H} - \mathbf{H}_v)] \\ \tilde{\mathbf{H}}(\mathbf{U}) &= |\mathbf{J}| [\zeta_x (\mathbf{F} - \mathbf{F}_v) + \zeta_y (\mathbf{G} - \mathbf{G}_v) + \zeta_z (\mathbf{H} - \mathbf{H}_v)] \end{aligned} \quad (3.13)$$

and $|\mathbf{J}|$ is the determinant of $\mathbf{J}$. As explained in Chapter 2 an approximate Riemann solver, as proposed by Roe [81], is used to evaluate the common convective fluxes at the interface. The common viscous flux is evaluated using an average of the primitive variables and gradients at both sides of the interface [55]. The Gauss-Lobatto points have been used for the hexahedral elements. The triangle point distributions of Warburton [89] are employed for triangular prisms.

The fact that the SP at the interface between elements are coincident allows for the evaluation of the flux without any further operations.

The gradients of velocity, pressure and temperature needed for the computation of the viscous fluxes are expressed as a function of the derivatives in the computational coordinates and the Jacobian:

$$\frac{\partial \mathbf{V}}{\partial x_i} = (\mathbf{J}^{-1})^T \frac{\partial \mathbf{V}}{\partial \xi_i} \quad (3.14)$$

where $x_i = x, y, z$, and $\xi_i = \xi, \eta, \zeta$; and $\mathbf{V} = (\rho \ u \ v \ w \ p)^T$ is the vector of primitive variables. As mentioned in Chapter 2 the common value at the flux points in this case is the average value of the primitive variables at both sides of the element.

3.2. Hardware and Software architecture

Commodity GPU clusters have been used for a long time [90] because, despite their drawbacks, they present a clear advantage in terms of cost over other types of hardware. Commodity GPUs are substantially cheaper than GPUs specifically designed for computing. Cost reduction in commodity GPUs is obtained with a less tight quality control and removing in the chips functionality oriented to avoid or control computing errors. On the other hand, GPUs were designed to process graphics which is commonly done in single precision. Commodity GPUs are, hence, single precision hardware what means that its efficient use requires single operations. As it will be shown later on in this chapter, this leads to round-off errors which have to be dealt with.

Unlike CPUs, GPUs have thousands of processor cores (usually called threads) with very fast memory bandwidth. Though considerably slower than those of CPUs, *i.e.*, less operation counts per clock-cycle, the large number of processors can increase the overall operation count over that of a CPU. However, to take advantage of these large number of cores the operations must be independent of each other, *i.e.*, one operation must not depend on the result of the previous operation. Fortunately, this is usually the case in most operations in CFD. For instance, one does not need to know the x -derivative of the density in the node i to compute this derivative in the node $i + 1$. At the same time, this implies that the efficient usage of GPUs is heavily dependent on the programming paradigm.

Similar to primary memory and cache memory in CPUs, GPUs have three levels of memory with different access bandwidths. The largest memory on the GPU is called global memory. This memory has highly varying access bandwidth depending on the way it is accessed. When each of the threads performing a given task access global memory in an ordered manner (thread 1 accesses memory location 1, thread 2 memory location 2, etc.) very high memory access bandwidths (close to the peak bandwidth) can be achieved. However, when these accesses are unordered or have a stride (thread 1 accesses memory location 1, thread 2 memory location 1 plus the stride, etc.) the bandwidth decreases dramatically¹. The second level of memory is the local memory and is shared between the threads of a group. This memory sits physically closer

¹<https://docs.nvidia.com/cuda/cuda-c-best-practices-guide/index.html#memory-optimizations>

to the processors, and it has a much higher memory access bandwidth than the global memory. This memory, unlike the global memory, is scarce. The smallest memory of the three types is the register memory which is only visible to the thread.

As it will be explained in later sections, because of the decay of global memory access bandwidth for unordered accesses, data arrays are sorted in specific ways and the functions to be executed on GPU designed to access these data in an ordered manner while making the most of the scarce local memory. This is critical for an efficient implementation of the method.

The size of the problems tackled in this work are far greater than those which can be handled with a single GPU. This makes it necessary to partition the computational domain in multiple sub-domains that can be handled by individual GPUs. In order for the process to be efficient, these partitions need to run independently of each other and communicate the necessary information. The data from the neighbouring partitions are transferred by an MPI implementation through an Infiniband network.

The hardware architecture on which the code has been run can be seen in Fig. 3.1. The cluster has a master node (head node in the figure) that is used to manage the queue and the available resources. This node is not used for computation and is connected to the computing nodes (nodes 1, 2, 3... in Fig. 3.1) which are connected among them with an Infiniband network. There are a total of twenty computing nodes. Each of the computing nodes has two CPUs (Intel Xeon E5-2620 @2.1GHz) connected to ten GPUs (NVIDIA GeForce GTX 1080 Ti 11GB GDDR5X) via PCI-e bus. A total of 200 GPUs were, then, available in this particular cluster at ITP Aero. The global memory of the GPUs is of 20GB each while the local memory (that shared between threads) is 48KB.

The implementation of the FR method has been made over the existing software infrastructure of a second-order unstructured code known as Mu^2s^2T [91, 90]. This code is written using a mixture of C++ and OpenCL [92], which is C-99 language with some extensions to express parallelism. When running on multiple GPUs (which is always the case in the problems tackled in this work) each MPI² process lives in a CPU processor and controls one GPU. This might

²The MVAPICH library has been used: <https://mvapich.cse.ohio-state.edu/>

Figure 3.1.: Schematic of the hardware architecture where the simulations have been run. Top: configuration of the configuration. Bottom: internal architecture of the computing nodes.

not be optimal in terms of communications because even if the GPUs are connected to the same CPU, data transfer between these GPUs is performed using MPI communications.

The resulting solver can run on a CPU alone, or a combination of CPU and any many-core hardware that can support OpenCL. In the present work the simulations are performed using a combination CPUs and GPUs.

Within this setup, every operation of the flux reconstruction method is performed on the GPU through OpenCL functions called kernels. The order in which this functions are executed and the transfers of data from one MPI process to another is determined by the C++ code running on the CPU core controlling the GPU. In this manner, the CPU does not perform any calculation but will dictate the order in which operations in the GPU are carried out. For instance, when an iteration is finished, the CPU code will check that all the MPI processes are on the same line and that no errors have occurred, it will check if data is to be dumped onto a file and then start a new iteration. At the new iteration it will call the kernel to compute discontinuous derivatives of the primitive variables and communicate sub-domain boundary data to other processes. When this data has been communicated, the kernel to compute the correction to the discontinuous derivatives of the primitive variables will be called.

As it can be seen in the simplified example above, the CPU code does not perform any computation with domain data (primitive variables, conservative variables, etc.). These are left to the GPU kernels which can perform them at a much faster pace.

3.3. Implementation of the Operators

The different functions needed to implement the FR scheme have been designed to maximise the memory access bandwidth and the number of operations per memory access. This is very important because, as it will be discussed, the time that it takes to execute these functions is bounded by memory access time. This means that, once the data are accessed, the best strategy is to perform as many operations as possible with that data before writing them back into the global memory. This is the reason why storing data in local memory is so important. It serves as a storage area where data can be accessed at much higher bandwidths than it would otherwise be if that data were to be accessed in global memory.

The formulation presented in section 3.1 to integrate the NS equations can be summarised in the following steps:

1. Evaluate the discontinuous derivatives of the primitive variables. This is a matrix-vector product, where the matrix elements are the derivatives of the Lagrange polynomials for each solution point: $d\ell_i/d\xi_j$ ($\mathbf{S}_{00}$ in Eq. 2.58). For instance, the derivative of u along ξ in ξ_j in a 1D element is calculated as $\left. \frac{\partial u}{\partial \xi} \right|_{\xi_j} = \sum_{i=1}^{N_{SP}} \left. \frac{d\ell_i}{d\xi} \right|_{\xi_j} u_i$
2. Correct the derivative of the primitive variables with the correction functions and the average value of the primitive variables at the face points. This is a matrix-vector product where the matrix is given by the left and right Radau functions (hexahedra) or by the lifting coefficients (see Appendix A). The vector is made up by the differences between the primitive variables at both sides of the face points. For instance, the correction of the derivative of u along ξ in ξ_j in a 1D element is calculated as $(u^L - u_0^e) \frac{dg_L}{d\xi}(\xi_j) + (u^R - u_p^e) \frac{dg_R}{d\xi}(\xi_j)$

3. Transform the gradient of primitive variables to physical coordinates (Eq. 3.14). For the derivative of u along x in a 2D element $\frac{\partial u}{\partial x} = \frac{\partial \xi}{\partial \eta} \frac{\partial u}{\partial \xi} + \frac{\partial \eta}{\partial \eta} \frac{\partial u}{\partial \eta}$
4. Compute the fluxes for each solution point (equations 3.2 and 3.3).
5. Evaluate the discontinuous derivatives of both viscous and convective fluxes. This is analogous to what is done in step 1 except now the fluxes are derived instead of primitive variables. $\frac{\partial \mathbf{F}}{\partial \xi} = \sum_{i=1}^{N_{SP}} \left. \frac{d\ell_i}{d\xi} \right|_{\xi_j} \mathbf{F}_i$
6. Compute the correction to the fluxes with the values at the element interfaces. This is analogous to step 2 except now the vector contains the interface flux (computed using a Riemann solver) instead of the mean value of primitive variables. $(\mathbf{F}^L - \mathbf{F}_0^e) \frac{d\eta_L}{d\xi}(\xi_j) + (\mathbf{F}^R - \mathbf{F}_p^e) \frac{d\eta_R}{d\xi}(\xi_j)$
7. Enforce boundary conditions.
8. Advance to the next stage of the RK scheme and go to step 1.

Most of the operations in the FR algorithm are self-contained within the cell and only require information from its neighbouring cells to compute the corrections of the gradient and the fluxes. This fact is convenient from a parallel implementation standpoint: processors can fetch data from memory, perform all needed operations and write results back into memory without needing communication. As it will be explained this is particularly efficient in massively parallel processors such as a GPU.

The previous steps have been translated into individual OpenCL kernels except for steps 2 and 3 which are combined in a single kernel to minimise global memory accesses. Each kernel is executed simultaneously by individual threads operating with different data.

For clarity in how these kernels operate an example is given next. We consider the discontinuous derivatives of the primitive variables. When this function is invoked on the CPU process, the GPU is invoked subsequently to execute the associated kernel. This kernel is, then, copied to a number of threads of the GPU. These threads are arranged in groups and each group is associated with an element so that the group contains as many threads as solution points. In the particular case of the discontinuous derivative of the primitive variables, each thread is

associated with a particular solution point and it will do the following:

1. Load a line of the matrix of the Lagrange derivative coefficients ($\mathbf{S}_{00}$ in Eq. 2.58) associated with the solution point and store it in local memory. .
2. Load the primitive variable of the solution point and store it in local memory.
3. Wait until all the threads of the group have reached this point.
4. Compute the derivative of the primitive variable by performing the associated matrix vector multiplication operating over elements in local memory.
5. Store the derivative in global memory and repeat the process from 2 with the following primitive variable.

As it can be seen the coefficients are only loaded once and stored in local memory so that they can be reused for the derivation of each primitive variable. Storing variables in local memory allows for unordered memory accesses at high rates. This has two effects: *i*) avoids strided memory access which results in very low bandwidth (further comments are made in the following sections) and *ii*) lower the number of global memory accesses. Accessing global memory is considerably more expensive than accessing local memory, and is to be avoided. As mentioned, this intensive use of local memory is instrumental to achieve high efficiencies.

3.3.1. Data arrangement for maximum achievable bandwidth

As mentioned in the previous section, the threads executing the kernels need to access data in global memory. As it turns out, the way in which they access global memory is critical in overall code efficiency. There are two logical data array structures that can be considered. The first one is to order the variables of each node contiguously, this is

$$[\rho_0, \rho u_0, \rho v_0, \rho w_0, \rho E_0, \dots, \rho_N, \rho u_N, \rho v_N, \rho w_N, \rho E_N] \quad (3.15)$$

where it has been considered that there are $N = N_e \cdot N_{SP}$ nodes in the mesh where N_e is the number of elements and N_{SP} is the number of solution points per element.. The problem with

this data structure is that, if the kernel is to compute the derivative of a given variable (*e.g.* the density), the threads need to access array elements that are far apart. This is called a strided memory access and is highly detrimental in terms of memory access bandwidth as mentioned in the previous section. This leads to the second data array structure (which is the structure chosen in the present work)

$$[\rho_0, \dots, \rho_N, \rho u_0, \dots, \rho u_N, \dots, \rho E_0, \dots, \rho E_N] \quad (3.16)$$

where it is seen that the first variable for each of the solution point is put first, then the second variable, and so on. The solution points of each element are consecutive in this array, *i.e.*, the first N_{SP} densities correspond to the first element and so forth. In this manner all threads of the kernel access variables in a consecutive order. These accesses coalesce in a single clock cycle yielding optimal bandwidth.

By optimal bandwidth it is understood the maximum achievable bandwidth. This may not be the peak bandwidth but the measured in benchmarks such as STREAM [93], which is around 80% of the peak value. Thus, if the peak bandwidth of the GeForceGTX1080Ti is 484 GB/s, the maximum measured bandwidth is around 380 GB/s.

3.3.2. Evaluation of the discontinuous derivatives

As explained in Chapter 2 flux and variable derivatives can be expressed as matrix-vector operations. These matrix-vector operations can be handled by linear algebra libraries like cuBLAS [94] in CUDA or ViennaCL [95] in OpenCL. However, several authors [96, 97] have compared the performance of such libraries against specifically tailored OpenCL/CUDA kernels for the specific data layout and the latter generally outperform the former.

As mentioned earlier the execution of the kernels can be made more efficient by using local GPU memory. For the chosen hardware this memory is small with a current maximum size of 48 kB. This means that, even though using this memory is crucial for efficient matrix-vector operations, it needs to be used carefully to avoid overflowing it. Figure 3.2 shows in a schematic way the

Figure 3.2.: Sketch of data arrangement and discontinuous derivative operations on the GPU architecture. Global data is copied in local memory. Each thread performs its corresponding matrix-vector operation for each of the variables (a generic variable $\phi_i = \phi(\xi_i)$ is used in the figure for compactness).

memory structure of the implementation on GPUs. As it is seen in the sketch, global data is copied to the local memory of the group. Each group is responsible for one single element, so that only data for that particular element is stored in local memory along with the derivatives of the Lagrange polynomials, $d\ell_i(\xi_j)/d\xi$ ($\mathbf{S}_{00}$ in Eq. 2.58). The matrix-vector operations can then be performed accessing only local memory.

To evaluate the discontinuous derivative at point i of the generic variable ϕ , *i.e.* $\sum_i \phi_i \frac{d\ell_i(\xi_j)}{d\xi}$, (steps 1 and 5 in Section 3.3), a group of threads is executed for each cell, Fig. 3.2. For each group the following steps are taken:

1. Fetch the data from global memory and copy it in the shared memory arrays. Since there are no repeated nodes between cells, the data access pattern is regular and coalesced memory accesses are possible, achieving the optimal data transfer rate (as mentioned in the previous section). The measured bandwidth is $\sim 350 \text{ GB/s}$, near the 75% peak value for the GPU.
2. Each thread evaluates its corresponding row of the matrix-vector product for each vari-

able, $\sum_i \phi_i \frac{d\ell_i(\xi_{j=\text{thread}})}{d\xi}$. These memory accesses are not ordered, but since the data are already contained in the shared memory, this is no longer penalising the efficiency of the method.

3. Write the result back to the corresponding global memory position. The data storage is also done with an optimal data arrangement.

Although the basic steps for computing the discontinuous derivatives of the primitive variables and the fluxes (steps 1 and 5 in Section 3.3) are the same there is one difference between them. This is, the primitive variables are derived with respect to each computational coordinate whereas the fluxes are only derived with respect to one. Additionally, the viscous and convective fluxes are combined so that only the difference is derived, this is, $\mathbf{F} - \mathbf{F}_v$.

These kernels are memory bound, *i.e.*, the cost of accessing the data and writing the results is in general much larger than the cost of executing the kernel operations. For a fourth-order discretization, $p = 3$, 90% of the time is spent accessing or writing data.

Local memory requirements

The amount of data stored in the local memory depends on the type of element. For hexahedra elements, the derivatives in each direction are done with the derivatives of the Lagrange polynomials and so the same matrix $\mathbf{S}_{00}$ can be used in each direction. This is of size $(p+1)^2$ for polynomial order p so that the total number of data stored in local memory is $(p+1)^2 + (p+1)^3$ where the second term corresponds to the variable stored in local memory. For prisms (see Appendix A), there would be $(p+1)^2(p+2)/2$ solution points to be stored along with $(p+1)^2$ derivatives of the Lagrange polynomials used for the derivatives along the direction perpendicular to the triangular base and $(p+1)^2(p+2)^2/4$ for the derivatives parallel to the triangular base. Considering that the local memory is limited to 48KB in our particular hardware, this imposes a limit on the polynomial order that can be run. When operations are carried out in single precision this limit is $p = 13$ whereas for double precision the polynomial order limit would be $p = 10$.

3.3.3. Evaluation of derivative corrections

The derivative correction kernels (steps 2 and 6 in Section 3.3) use data from adjacent cells to compute common values at the interfaces. Accessing data from neighbouring cells will rarely coalesce so that these memory accesses will be significantly slower in comparison to accessing data from the same cell. Some authors [97] make use of a dedicated kernel that pre-computes interface values so that they can be accessed at the cell level. However, this pushes these expensive data accesses off to a different kernel and incurs into an additional memory read and write. Simpler kernels result from this approach.

In this work, simpler kernels are given up in favour of a reduced number of memory accesses which results in more complicated kernels. Once the common values are computed, they are kept in a shared or local memory array, and then re-used to evaluate the correction for all cell points. As explained above, that allows re-using the shared data among threads at virtually no cost, and also writing the corrected derivatives with optimal memory access.

As with the discontinuous derivatives kernels, in these kernels the threads are organised in groups which correspond with a particular element so there are as many threads as solution points. Since these kernels perform two distinct functions the operations are grouped in two different loops. The first loop computes the common values at the interfaces and the second loop corrects the discontinuous derivatives.

The first loop runs over the face nodes of the element. Since there are points that are on multiple faces the size of the loop is larger than the total number of points of the element. For this reason, some of the threads are required to perform the following actions more than once:

1. Copy data of the face point from global memory into local memory.
2. Copy data of the neighbouring node from global memory into local memory.
3. Compute common values. This is the mean of the primitive variable or the common flux for steps 2 and 6, respectively.
4. Compute the jump in primitive variables (if step 2) or flux (if step 6) and store it into local memory.

When the this loop over the boundary nodes is finished the second loop over the solution points is started. This contains the following actions

1. Access the correction matrix elements in global memory needed to perform the correction of the solution point.
2. Multiply the previous correction elements by the jump in primitive variable (step 2) or flux (step 6) already stored in local memory.
3. Access discontinuous derivative in global memory, add the correction term and save it back in global memory.

As it can be seen, since the correction kernels compute the common values at the interfaces the kernels are more complicated. The global data accesses of the first loop are strided accesses and are therefore slow as explained in section 3.2. Global accesses of the second loop are ordered and can be done at a higher bandwidth. The measured bandwidth for these kernels is 330 GB/s for the hexahedra and 282 GB/s for the prisms, which is 68% and 58% of the peak memory bandwidth.

The flux correction kernel (step 6 in Section 3.3) requires a larger amount of local memory. This is because to compute the common flux it is required to store in local memory primitive variables, gradients of primitive variables, the Jacobi matrix and the common fluxes for each of the face points. These are 34 values for each of the face points. For hexahedra elements this imposes a limit to the polynomial order of $p = 8$ for single precision, and $p = 6$ for single precision.

3.3.4. Kernels time breakdown

The percentage of the time spent on each of the kernels enumerated earlier in section 3.3 is represented in Fig. 3.3 for a 28^3 sixth-order hexahedra element mesh. This mesh was used to run the Taylor-Green vortex case [98] which is a benchmark case described in Chapter 5.

As it can be seen, the correction of the fluxes (kernel 6) is the most time consuming step, followed by the discontinuous derivation of the fluxes (kernel 5), the Runge-Kutta time-stepping

Figure 3.3.: Percentage of time spent executing each of the 8 kernels enumerated in section 3.3 for the Taylor-Green vortex case with a 28^3 sixth-order hexahedra elements.

(kernel 8) and then the gradient correction and the transformation to the physical plane (steps 2 and 3), that are performed in a single kernel to reduce the number of global memory accesses. Then the convective and viscous flux evaluation (step 4), and finally the derivation of the primitive variables (step 1). The boundary conditions are generally applied over a reduced sub-set of the mesh points and do not account for more than a few percentage points. In this case, only periodic boundary conditions are applied consuming around 1% of the total time.

It is noticeable that the Runge-Kutta consumes a significant part of the total time. This is due to the use of single precision arithmetic. In that scenario, care must be taken to avoid accumulating round-off errors in the summation that is produced at every step of the Runge-Kutta method. Therefore, we have employed the Kahan summation algorithm (see section 3.5) to minimise the summation round-off. The algorithm involves two extra arrays with the size of the conservative variables, that have to be accessed and updated every Runge-Kutta step. It must be emphasised once again that for mesh order $p = 4$ every kernel is memory bound, *i.e.*, the majority of the kernel execution time is spent loading and storing data from the global memory.

3.4. Parallel use of multiple GPUs

In addition to the fine grained parallelism natural to the use of the GPUs, when the problem size is large enough and it does not fit into a single GPU, multiple GPUs are used in parallel.

Figure 3.4.: Example of pipelined communication. The communication of a single message (top) is split into smaller messages (bottom) so communication between elements can happen concurrently.

The domain is decomposed using the parallel domain decomposition library ParMeTiS [99] and each GPU is controlled by a single MPI process running on a CPU core.

In this situation, some data residing on the GPU memory must be sent to the GPU memory in charge of computing the adjacent domain. That poses a serious limitation, because the data transfer rate from the origin GPU to the CPU is given by the PCI-e bus at 16 GB/s with an effective bandwidth around 12 GB/s. These data must be, then, sent through the Infiniband network via Message Passing Interface (MPI) with a transfer rate for the Infiniband network of around 22.5 GB/s (90% of the peak value of 25 GB/s for the 100 Gb/s two-port Infiniband³) and finally it is sent back to the destination GPU through the PCI-e at 12 GB/s. The effective GPU to GPU bandwidth is then

$$\frac{1}{BW_{GPU \leftrightarrow GPU}} = \frac{2}{BW_{GPU \leftrightarrow CPU}} + \frac{1}{BW_{CPU \leftrightarrow CPU}} \quad (3.17)$$

which, for large enough messages, results in an effective GPU to GPU bandwidth of 4.7 GB/s, which is lower than the minimum of the PCI-e and Infiniband bandwidths. The following subsection deals with strategies by which this severe bottleneck can be minimised.

3.4.1. Increase of data transfer bandwidth

One of the techniques used to improve communication efficiency is the use of pinned, or page-locked, memory to perform the data transfers between the CPU and the GPU. The GPU cannot access CPU memory directly which is paged by default. Hence, when the GPU needs to access CPU memory, the GPU driver must first allocate a temporary pinned array on the CPU. The CPU can then transfer data to this pinned array so that the GPU can access it. The pinned memory technique consists of allocating CPU memory directly into the pinned memory so that the allocation and transfer from paged memory to pinned memory is avoided. With this, a data transfer rate of 12 GB/s through the PCI-e bus is obtained. Otherwise the effective bandwidth is almost halved.

Another technique is communication pipe-lining. The process is depicted in Fig. 3.4 schematically. If the full message was to be sent in full, it first needs to be sent to the CPU, then communicated via MPI to another CPU and finally sent to the destination GPU. By dividing the message into smaller pieces, the MPI communication of the first piece starts as soon as the first GPU to CPU transfer finishes, and it proceeds while the second piece of GPU to CPU data transfer is carried out. The second piece of CPU to CPU data exchange is executed simultaneously to the first CPU to GPU data transfer and the third GPU to CPU one, etc. By overlapping the three communication operations the data transfer rate is increased. For message sub-divisions larger than 1MB bandwidth improvements between 50% and 100% (depending on the PCI-e bus traffic) have been measured⁴. For the large cases run within this thesis, the typical size of a message is of about 25MB which is split up into messages of 2-4MB (10 to 6 subdivisions).

3.4.2. Overlap of communication and computation

The cost of communication between domain partitions can be hidden if it is overlapped with the computations. This is possible because the elements of each sub-domain are organised in two groups: a group containing only the internal elements to the sub-domain, and a second group

³Because we have two ports we can achieve 100Gb/s in each direction, hence we would have 200Gb/s or 25GB/s.

⁴https://mvapich.cse.ohio-state.edu/performance/pt_to_pt/

Figure 3.5.: Overlap of computation and communication. The $GPU \rightarrow CPU$ communication, the $CPU \rightarrow CPU$ MPI communication, and the $CPU \rightarrow GPU$ data transfer proceed simultaneously with the kernel executions. As long as the total communication time t_{comm} remains lower than $t_{interiorCells}$, the extra-cost of running parallel simulations will be $t_{gatherScatter} + (1 - GPU_{eff})t_{frontierCells}$, where GPU_{eff} is the parallel efficiency of the GPU, which is below unity when the number of kernel threads is small.

including only elements with faces on the sub-domain frontier. This is very convenient since the interior elements do not depend on information from other sub-domains and can, hence, be computed concurrently with the communication process. This is schematically explained in Fig. 3.5. As long as the time required by the three-step communication strategy explained above, t_{comm} , is less than the time spent computing interior elements, $t_{interiorCells}$, the communication cost is negligible.

Overlapping communications and computation can only be achieved by making the communications non-blocking. MPI non-blocking communications that allow the simultaneous launch of GPU kernels are, thus, used. The CPU is left in charge of managing the communications during the code execution. GPU to CPU data transfers also happen concurrently with the computation. This is done by creating three independent execution queues on the CPU: one controlling the GPU to CPU data transfer, another for the CPU to GPU, and the last one to control the kernels execution.

Despite the time that it takes to compute the interior elements, $t_{interiorCells}$, being large in comparison to that taken to compute the frontier elements, the communication time could be larger than $t_{interiorCells}$. It is, then, necessary to establish a checkpoint before the computation of the frontier elements to make sure that the communications among the sub-domains have finished.

Figure 3.6.: Parallel speed-up for different number of mesh partitions. The mesh corresponds to the linear cascade case described in Chap. 6 for Reynolds number $Re = 10^5$ and reduced frequency $f_r = 1$. The mesh consists of 183 million points. In the cases presented above, the GPU memory usage varies between 92% when partitioned in 10 sub-domains to 10% at 80 sub-domains.

3.4.3. Application to a representative case

The efficacy of the overlapping technique can be seen in Fig. 3.6 for the linear cascade case of Chap. 6. This is the test case used for comparing against experiments and is, therefore, the most representative case in the context of this work. In particular, the data presented in Fig. 3.6 corresponds to the case with a Reynolds number of $Re = 10^5$ and reduced frequency $f_r = 1$. The mesh consists of 755k sixth-order, $p = 5$, prisms and 410k sixth-order hexahedra totalling 183M points. The time per time-step is obtained averaging the last 200 time-steps of a simulation of 1,000 time-steps. In this manner, it is ensured that the kernel set-up time, I/O operations and pre-processing steps are avoided.

As it can be seen in Fig. 3.6 the speed-up is very close to the ideal one up to 40 partitions. Up to this point, the differences between the actual speed-up and the ideal one is due to small load unbalances among the partitions, *i.e.*, some partitions can have a slightly higher number of nodes than others. Taking into consideration that the maximum mesh size that can be handled by each of the GPUs used in this work is 20M nodes, with 40 partitions the GPU capacity is at about 20%. When dividing the domain into 80 subdomains, the speed-up deteriorates further, although still is very good considering that the average size of the sub-domain, in this case,

is about 2 million nodes (roughly 10% of the GPU maximum capacity). For this number of partitions, the time for communications becomes of the order of the time required to compute the inner cells of the subdomain, *i.e.*, $t_{comm} \sim t_{interiorCells}$. As will be discussed in Chap. 6 the simulations object of this thesis are decomposed in around 30 and 40 partitions and are, therefore, in the region of ideal speed-up scaling.

3.5. Explicit treatment of round-off errors

One of the novelties presented in this work is that the developed high-order Navier-Stokes solver is executed in single-precision GPUs. This is very convenient since the available hardware infrastructure to perform the simulations is based on single-precision commodity GPUs. Running in double-precision incurs in simulations that are much more costly. There are two considerations to be made in this regard. Running in double-precision doubles the memory requirements in the first place. Since the kernels used here are memory bounded, as explained previously, the kernel computing time should double at least as well. On the other hand, the double-precision computations have to be conducted on single-precision hardware which means that these operations also take longer to run. In our implementation, the overall penalty factor is around 2.5, this is, it is 2.5 times more expensive to simulate in double precision than it takes to run it in single.

It is well known that single-precision computations are prone to encounter round-off errors. Within the course of this work, one of the operations which is particularly affected by round-off errors is time-stepping. Round-off errors have also been encountered when computing the time-averaged solution although this does not affect the simulation itself, rather the post-processed time-averaged solution.

Fine meshes, as the ones used in this work, along with the restrictive CFL condition imposed by stability considerations, unavoidably lead to small time-steps. Time advancement is strongly affected by round-off errors as a result.

Let us consider the structure of a generic time-stepper, for instance the stage of a Runge-Kutta

Algorithm 3.1 Pseudo-code of a Kahan summation algorithm as used in our RK implementation.

```

1.  function updateU(RHS,U,dt , corrections )
2.      for node = 0 to RHS.length do
3.          // correction of RHS*dt
          corrRHSdt = RHS[node]*dt - corrections [node]

          // time advance U
4.          newU = U[node] + corrRHSdt

          // new correction digits
5.          corrections [node] = (newU - U[node]) - corrRHSdt

6.          U[node] = newU
7.      next node
8.  return

```

scheme,

$$\begin{array}{ccc}
 u_j^{m+1} & = & u_j^m + \alpha \Delta t R_j(u^n) \\
 u_c & & u_c \quad \text{CFL} \left(\frac{\Delta x_{min}}{a_0 + u_0} \right) \hat{k} u_c \\
 \downarrow & & \downarrow \\
 \mathcal{O}(1) & & \mathcal{O}(1) \quad \mathcal{O} \left(\text{CFL} \frac{\Delta x_{min}}{\Delta x_{max}} \frac{M_0}{1 + M_0} \hat{k} \right)
 \end{array} \tag{3.18}$$

where the solution at the node j , u_j , is advanced from the m -th RK stage to the $m+1$ with a pseudo-time $\alpha \Delta t$, and where α is a constant of order one which depends on the numerical scheme and is, therefore, irrelevant for the purpose of this discussion; Δt is the time step. The second line in Eq. 3.18 represents the order of magnitude of the equation while the last line is their non-dimensional relative value. CFL is the global Courant-Friederich-Lewis number, $\text{CFL} = (u_0 + a_0) \Delta t / \Delta x_{min}$ and is of order 10^{-1} for our choice of discretisation and where u_0 , a_0 , Δx_{min} are the reference velocity, speed of sound and minimum element size, respectively. When making order of magnitudes non-dimensional in Eq. 3.18, u_0 has been taken as reference velocity and Δx_{max} , the maximum elements length, has been taken as reference length. $M_0 = u_0/a_0$ is the reference Mach number. The relative value of the residual multiplied by Δt with respect the characteristic value of the variable is $\tilde{R}_j = \Delta t R_j(u^m)/u^m \sim \hat{k} \text{CFL} \frac{\Delta x_{min}}{\Delta x_{max}} \frac{M_0}{1 + M_0}$ where time has been made non-dimensional with, $\tilde{t} = t \Delta x_{max}/u_0$. The reduced wave-number is $\hat{k} =$

Figure 3.7.: Mesh of the cylinder case used to validate the Kahan summation algorithm with a close-up around the cylinder.

$k\Delta x_j$ while k is the wave-number of the considered wave.

As it can be seen three are the main parameters that affect the round-off error in the simulation, namely, CFL , $\frac{\Delta x_{\min}}{\Delta x_{\max}}$, and $\frac{M_0}{1+M_0}$. In our case, we have a $\text{CFL} \sim 10^{-1}$ due to the discretisation. The ratio of minimum to maximum mesh sizes has typical values of 10^{-2} or lower; and the simulations of interest in this work have been made at $M_0 = 0.1 - 0.2$. All together, before factoring in the residual, there is a factor already of 10^{-4} . For well resolved wave-lengths the reduced wave-number $\hat{k}$ is small so that the combined effect can reach values below round-off error very quickly.

The time-averaged solution is calculated as the solution is advanced in time. To compute the mean a counter is used to store the number of samples of the mean and an accumulator is used to store the solution. As an example, for a quantity ϕ , the accumulator Φ would be calculated as

$$\begin{aligned} N^{n+1} &= N^n + 1 \\ \Phi^{n+1} &= \Phi^n + \phi^{n+1} = \sum_{i=1}^n \phi^i + \phi^{n+1} \end{aligned} \quad (3.19)$$

where N represents the total number of samples and the superscript is used to differentiate between time instants. The mean of the quantity ϕ at instant $n+1$ would be calculated as $\bar{\phi}^{n+1} = \Phi^{n+1}/N^{n+1}$. Due to the fact that the time step is very small, the number of iterations required to get meaningful results is very large. This means that the accumulator in Eq. 3.19, Φ^n , can become significantly large compared to ϕ what will saturate the summation. This is, ϕ/Φ^n , will have a magnitude which is of the order of the round-off error.

Round-off errors can be significantly reduced by using the Kahan summation algorithm [100]. A pseudo-code of the Kahan summation algorithm can be seen in Fig. 3.1 as applied to the time-advancement step of the Runge-Kutta scheme. In this example, the arrays 'RHS', 'U', and 'corrections' are input to the function along with the time-step, 'dt'. The 'corrections' array stores all the low-digit corrections of every node and variable from the previous time-steps (the example considers a single variable only). In this example the loop in line 2 is over all the nodes in the arrays. In line 3, the corrected right-hand-side value multiplied by the time-step is computed and subtract the low-order digits stored in the 'corrections' array. It needs to be noted that it is required that the values in 'RHS' be multiplied by 'dt' so that the rest of the algorithm is consistent. In line 4, the variable 'U' in the next time-step is computed whereas in line 5 the correcting low-order digits for the following time-step is computed. It is clear that if one had infinite precision the result of the operations in line 5 would be zero. This, however, gives the low-order digits do not added in the summation but that will taken into account in the next iteration. Even though the Kahan summation algorithm cannot replace a double-precision summation it does improve significantly the accuracy of the operations.

As can be seen, running in single precision arithmetic comes at the expense of increasing the memory (new array 'corrections') and the number of operations. For instance, in the same case of Fig. 3.3, the time spent in the RK steps increases 40% by using the Kahan summation. Fortunately though, this needs only be done in the RK steps and not for every single array in the code. In the example of Fig. 3.3 the percentage of time spent in the RK steps is 14%, up from 10% when the Kahan algorithm is not used. Overall, running in single-precision is more efficient than running in double-precision.

Effect of the Kahan summation in the RK

To show the round-off error effect on the simulations a small case has been run with and without the Kahan summation algorithm in the RK method. The case consists of the two-dimensional flow around a cylinder at Reynolds number based on the diameter of $Re = 120$ and a Mach number of $M = 0.1$. The computational domain is rectangular with inlet, outlet and periodic

Figure 3.8.: Entropy snapshot and contour lines of a cylinder simulation after 160k time-steps with a time-step of $\Delta\tilde{t} = 2.2 \times 10^{-4}$. Left: single-precision without Kahan summation. Middle: double-precision simulation without Kahan summation. Right: single-precision simulation using Kahan summation.

Figure 3.9.: Entropy snapshot and contour lines of a cylinder simulation after 160k time-steps with a time-step of $\Delta\tilde{t} = 2.2 \times 10^{-5}$. Left: single-precision without Kahan summation. Middle: double-precision simulation without Kahan summation. Right: single-precision simulation using Kahan summation.

boundary conditions and it is discretised with 5,835 fourth-order cells (triangular and quadrangular) which gives a total of 63k nodes. The mesh can be seen in Fig 3.7. The cylinder surface is meshed with 100 QUAD elements with 4 elements across the boundary layer which is a very fine mesh for a laminar case such as this. The maximum time-step is, hence, limited by these small elements. The simulation is started with uniform initial conditions and run for 160k time-steps using a time-step of $\Delta t = 4.4 \times 10^{-7}$ seconds (or $\Delta\tilde{t} = 2.2 \times 10^{-4}$ in non-dimensional form) for both with and without Kahan summation. The time-step is made non-dimensional with the reference free-stream velocity, $u_0 = 40\text{ m/s}$, and the maximum cell size, $\Delta x_{\max} = 0.08\text{ m}$.

It could be thought that the time-step for this case should not pose a problem since it is above the single-precision error. Figure 3.8 shows a snapshot of entropy field after the 160k time-steps which physically corresponds to $tD/U_\infty \simeq 35$. As it can be seen in Fig 3.8 (left), there are considerable differences with the double-precision solution (middle). The first noticeable difference is that in the single-precision simulation without the Kahan summation the entropy

field away from the wake of the cylinder is very noisy. These entropy disturbances are not generated at the inlet but rather appear within the domain. This is mostly due to the fact that this is a low-Mach number case ($M = 0.1$) in the incompressible regime. As a consequence, the sum $\partial(\rho u)/\partial x + \partial(\rho v)/\partial y$ is always small and when multiplied by the time-step become even smaller making the total contribution of the product close or below the single-precision accuracy in Eq. 3.18. This is further aggravated by the fact that outside the wake and the boundary layer the derivatives of the momentum and energy equations are also small, *i.e.*, perturbation wave-numbers are small, resulting in very small residuals. A side effect of the perturbed field is that the wake behind the cylinder sees a non perfectly-symmetric field so that the Karman street vortex seems to develop faster.

Running the simulation with the Kahan summation algorithm solves the problem as it can be seen in the in Fig. 3.8 (right). Both, the simulation using single-precision with Kahan summation and the simulation using double-precision without the Kahan summation yield almost identical solutions. Therefore, it is concluded that it is possible to run our simulations using single-precision arithmetic without a significant loss of accuracy in the solution in shorter simulation times and smaller memory requirements than double-precision simulations.

The experiment was repeated with a time-step ten times lower than the previous one, *i.e.*, $\Delta t = 4.4 \cdot 10^{-8}$ seconds (or $\Delta \tilde{t} = 2.2 \cdot 10^{-5}$ in non-dimensional form) and running ten times more time-steps, *i.e.*, 1.6M iterations, than in the previous case. The results can be seen in Fig. 3.9. As it can be seen in the entropy snapshot of the simulation without the Kahan summation in the RK steps the noise completely dominates the field. Even with this smaller time-step the Kahan summation is able to reproduce the solution obtained using double-precision.

4. Boundary Conditions

Contents

4.1. Introduction	83
4.2. Boundary conditions in FR	85
4.3. Wall boundary condition	85
4.4. Inlet/Outlet boundary conditions	87

4.1. Introduction

While the numerical solver plays a vital part in the resolution of the NS equations the boundary conditions are what define the problem and must, hence, be properly modelled to get meaningful solutions. Some boundary conditions are easy to implement, this is the case of, for instance, periodicity. However, there are some others which have been thoroughly researched.

Inlet and outlet Boundary Conditions (BCs) in internal aerodynamics are particularly difficult to define and to implement. One of the problems of these conditions is that the variables to be imposed are combinations of conservative or primitive variables (such as the inlet swirl angle or total temperature) which complicates the algebra. The second problem is that it is not always possible to locate the BCs away from the region of interest so that the reflections that occur at the BCs cannot be damped, especially in high-order low-dissipation solvers, and may significantly influence the result of the problem being studied.

Figure 4.1.: Schematic of the element boundary flux configuration for element i . Left: element i is a boundary element. Right: element i is an internal element.

The problem of the inlet/outlet BCs has been tackled in the past by several researchers. Giles [101] laid down the theoretical basis for the implementation of non-reflecting boundary conditions. In his work different conditions were tackled such as the 1D unsteady BCs, 2D single-frequency BCs, 2D steady BCs and the approximate 2D unsteady BCs. The work of Poinso and Lele [102] expanded the work done by Thompson [103] and considered viscous terms which are supplied as additional terms in the BC evolution equation. These latter conditions are referred to as the Navier-Stokes Characteristic BCs (NSCBCs). The conditions proposed by Poinso and Lele were further developed for 3D flows by Lodato [104]. In his work, Lodato demonstrated that 3D terms were required in fully 3D outflow conditions to avoid flow distortion at the outlet.

In turbomachinery problems, inlet boundary conditions for the mean flow are usually specified in terms of total pressure, P_t , total temperature, T_t , and flow angles. This choice is not mathematically based on a characteristic analysis of the steady problem but is frequently used as well in turbomachinery design. Odier et al. [105, 106] extended the concept developed by Poinso and Lele [102] to express the inlet boundary condition as a function of total pressure, P_t , total temperature, T_t , radial angle, α_r , and swirl angle, α_θ .

In the present chapter the formulation of the boundary conditions implemented for this work is detailed. The particularities of the implementation in the context of the FR method are described and some simple examples are presented.

4.2. Boundary conditions in FR

As mentioned in Chapter 2 elements “receive” information from neighbouring cells through the interface fluxes used in the correction procedure. Ideally, one would like to control boundary elements through this received interface fluxes because this would not require modifying the formulation of the boundary elements. A schematic of this situation can be seen in Fig. 4.1. It is seen that, regardless of whether the element is internal to the domain or a boundary element, the situation for the element does not change, *i.e.*, one just replaces the fluxes of the external face by a boundary flux.

The challenge is then to find out the boundary flux needed to correctly impose the BC. For instance, in the case of periodic boundary condition in a Cartesian coordinate system, it is clear that the boundary flux of one element will be the same as of the periodic element at its corresponding face.

4.3. Wall boundary condition

The wall boundary condition can be imposed in weak form by specifying the following convective and viscous fluxes

$$[\mathbf{F}^w \mathbf{G}^w \mathbf{H}^w] = \begin{bmatrix} 0 & 0 & 0 \\ p & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & p \\ 0 & 0 & 0 \end{bmatrix}, \quad \text{and} \quad [\mathbf{F}_v^w \mathbf{G}_v^w \mathbf{H}_v^w] = \begin{bmatrix} 0 & 0 & 0 \\ \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \tau_{yy} & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \tau_{zz} \\ k \frac{\partial T}{\partial x} & k \frac{\partial T}{\partial y} & k \frac{\partial T}{\partial z} \end{bmatrix} \quad (4.1)$$

where the superscript v denotes viscous fluxes. These fluxes are then used directly to compute the Riemann fluxes in Eq. 2.75 at the element boundary coincident with the wall. The problem with imposing the wall BCs in weak form is that the nodes of the boundary element coincident with the wall reach zero velocity asymptotically with high-resolution meshes. For this particular

BC, the solution at the element level is modified after the application of the BC so that zero velocity is enforced at the wall nodes in each of the RK stages. Although this is against the philosophy of the method, this is a direct and simple way to ensure zero-velocity at all times.

The solid walls in the simulations of this thesis have been modelled as adiabatic. To impose the wall to be adiabatic, the gradients of the primitive variables ρ and p are modified so that the wall-normal gradient of the temperature is zero at the wall. This is required since the temperature is not a primitive variable in our discretization. The gradients of the density and pressure are related to those of the temperature through the state equation:

$$\left(\frac{\partial \rho}{\partial n}\right)_w = \frac{1}{C_v(\gamma-1)T_w} \left(\frac{\partial p}{\partial n}\right)_w \quad (4.2)$$

where the subindex w denotes values evaluated at the wall and where it has already been taken into account that $(\partial T / \partial n)_w = 0$. Relation 4.2 is used to modify the wall-normal gradient of the density. This is done by first removing the wall-normal component of the density and then applying the previous relation, this is,

$$(\nabla \rho)_w = \nabla \rho - (\nabla \rho \cdot \mathbf{n}) \cdot \mathbf{n} + \frac{1}{C_v(\gamma-1)T_w} \left(\frac{\partial p}{\partial n}\right)_w \mathbf{n} \quad (4.3)$$

where $\mathbf{n}$ is the wall normal vector and $\nabla \rho$ is the density gradient before the boundary condition is applied. In relation 4.3 the first two terms on the right-hand-side represent the density gradient in the plane tangent to the wall whereas the last term is the wall-normal contribution that ensures that $(\partial T / \partial n)_w = 0$.

In summary, in order to apply the adiabatic wall boundary condition:

1. Density gradient is modified with Eq. 4.3,
2. Wall viscous and convective fluxes are calculated with Eq. 4.1
3. Velocities are set to zero at the element level
4. These wall fluxes are later used in the correction step of the fluxes derivatives (step 6 in section 3.3) through the Riemann fluxes 2.75

4.4. Inlet/Outlet boundary conditions

As mentioned in the introduction, inlet/outlet boundary conditions are particularly critical and difficult to implement in internal aerodynamic problems. There are two main difficulties with inlet and outlet boundary conditions in unsteady non-linear simulations. The first is related to the reflectivity of the boundary condition whereas the second is related to the ability of the boundary condition to allow time and spatial variations of the variables over the imposed mean values.

Reflections are to be avoided in these simulations. The main reason being that reflections can remain in the domain for very large times due to the fact that the current numerical method has very low numerical dissipation. These reflections can be highly damaging, especially in areas of high sensitivity to external perturbations. This is the case, for instance, of the boundary layer at the leading edge or the inflectional shear layer. In these cases, the spurious perturbations caused by reflections can give rise to earlier transition and reattachment changing the nature of the flow and losses entirely.

In order to reduce the size of the computational domain and its associated computational cost, boundary conditions are generally relatively close to the geometry studied. In these cases, imposing the boundary condition on every node of the boundary might distort the flow near the geometry being studied. This is the case of the present work where the inlet is only half axial chord upstream of the leading edge of the profile studied. In this case, imposing the angle on every node of the inlet keeps the flow from adapting to the profile changing the pressure distribution over the profile. In these cases, it is important to allow variations of the flow at the boundary while imposing the desired values at the mean level. This situation is exemplified in Fig. 4.2 for an outlet which shows that the pressure contours remain unaltered for the simulation with the larger domain (top) and the outlet BC close to the trailing edge of the profile (bottom). This is only possible because pressure distributions over the mean imposed values are allowed. At the same time, time variations must also be allowed in order to reduce the reflectivity of the boundary.

Figure 4.2.: Pressure contours over a NACA0012 profile. Top: extended domain. Bottom: short domain. Pressure is allowed to vary in the y -direction so that the BC does not affect the pressure over the profile.

The boundary condition implementation in this work addresses these issues. These conditions are based on the Poinso and Lele's ideas [102] and the detailed derivation for the inlet by [106]. The general theory behind this method will be presented next along with a summary of the algorithm employed. The particularities of the inlet and outlet conditions are dealt with in later subsections.

4.4.1. NSCBC theory

The implementations of non-reflecting boundary conditions in hyperbolic systems of equations rely on the analysis of the characteristic lines. While achieving a diagonal system for the study of the characteristic lines may be simple for the 1D Euler equations, this is no longer possible for the 2D Euler equations or the general Navier-Stokes case. In the approach presented here, the characteristics are analysed in the direction perpendicular to the boundary and treated as a 1D problem. For simplicity, the boundary is assumed to be perpendicular to the x -axis ($x = \text{const.}$), which is also the case in all the simulations presented here. In the The Navier-Stokes equations can be rewritten in a convenient for for the definition of the boundary conditions as:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} + \frac{\partial \mathbf{D}_i}{\partial x_i} = 0 \quad (4.4)$$

where $\mathbf{D}_i$ represents here the diffusive terms ($\mathbf{F}_v \mathbf{G}_v \mathbf{H}_v$) and $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{H}$ represent the convective fluxes. Equation 4.4 can be expressed in terms of the primitive variables, $\mathbf{V} = (\rho u v w p)^T$, as

$$\frac{\partial \mathbf{V}}{\partial t} + \left(\mathbf{P}^{-1} \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \mathbf{P} \right) \frac{\partial \mathbf{V}}{\partial x} + \mathbf{P}^{-1} \left(\frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} + \frac{\partial \mathbf{D}_i}{\partial x_i} \right) = 0 \quad (4.5)$$

where $\mathbf{P} = (\partial \mathbf{U} / \partial \mathbf{V})$ is the Jacobian of the transformation between conservative and primitive variables. In this equation the matrix $\mathbf{A}_v = \mathbf{P}^{-1} (\partial \mathbf{F} / \partial \mathbf{U}) \mathbf{P}$ can be expressed as $\mathbf{S} \mathbf{\Lambda} \mathbf{S}^{-1}$ where the matrix $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues of matrix $\mathbf{A}_v$.

The following vector is defined

$$\mathcal{L} = \mathbf{\Lambda} \mathbf{S}^{-1} \frac{\partial \mathbf{V}}{\partial x} = \begin{Bmatrix} \mathcal{L}_1 \\ \mathcal{L}_2 \\ \mathcal{L}_3 \\ \mathcal{L}_4 \\ \mathcal{L}_5 \end{Bmatrix} = \begin{Bmatrix} (u-a) \left(\frac{\partial p}{\partial x} - \rho a \frac{\partial u}{\partial x} \right) \\ u \left(a^2 \frac{\partial \rho}{\partial x} - \frac{\partial p}{\partial x} \right) \\ u \left(\frac{\partial v}{\partial x} \right) \\ u \left(\frac{\partial w}{\partial x} \right) \\ (u+a) \left(\frac{\partial p}{\partial x} + \rho a \frac{\partial u}{\partial x} \right) \end{Bmatrix} \quad (4.6)$$

which represents the amplitude time-variations of the one-dimensional characteristic variables. Within the FR scheme the above relations are very straightforward to compute since spatial derivatives are available for every node at all times which is a consequence of solving the NS equations in differential form. Waves associated with different characteristic variables have different propagation speeds. As it can be seen, the waves associated with $\mathcal{L}_1$ and $\mathcal{L}_5$ propagate at velocities $u - a$ and $u + a$, respectively; whereas waves associated with $\mathcal{L}_2$ to $\mathcal{L}_4$ have propagation speed u . This means that for a subsonic inlet BC only $\mathcal{L}_1$ is travelling outside of the domain whereas there are four incoming waves. Analogously, for a subsonic outlet, four waves will be leaving the computational domain whereas only one will be travelling inwards. In practice, what this means, is that waves travelling out can be computed from values within the domain whereas waves travelling in will need to be imposed.

It is further defined the vector $\mathbf{d}$

$$\mathbf{d} = \mathbf{S}\mathcal{L} = \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \end{Bmatrix} = \begin{Bmatrix} \frac{1}{a^2} [\mathcal{L}_2 + \frac{1}{2}(\mathcal{L}_5 + \mathcal{L}_1)] \\ \frac{1}{2\rho a} (\mathcal{L}_5 - \mathcal{L}_1) \\ \mathcal{L}_3 \\ \mathcal{L}_4 \\ \frac{1}{2}(\mathcal{L}_5 + \mathcal{L}_1) \end{Bmatrix} \quad (4.7)$$

The Navier-Stokes equations can then be written using the primitive variables (Eq. 4.5) and the vector $\mathcal{L}$ as

$$\frac{\partial \mathbf{V}}{\partial t} + \mathbf{S}\mathcal{L} + \mathbf{P}^{-1} \left(\frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} + \frac{\partial \mathbf{D}_i}{\partial x_i} \right) = 0 \quad (4.8)$$

The above expression, together with Eq. 4.7, is useful to derive expressions for the boundary conditions as a function of $\mathcal{L}_i$. The Navier-Stokes equations in conservative variables and compact form can be written as

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{P}\mathbf{d} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} + \frac{\partial \mathbf{D}_i}{\partial x_i} = 0. \quad (4.9)$$

to highlight the dependence with $\mathcal{L}_i$ through the vector $\mathbf{d}$. Provided that an appropriate set of $\mathcal{L}_i$ can be obtained to represent the behaviour of the boundary condition, the time-integration of Eq 4.9 provides the values of the conservative variables at the boundary.

Ignoring viscous and transverse terms in Eq. 4.8 results in the local one-dimensional inviscid (LODI) system [102]

$$\frac{\partial \mathbf{V}}{\partial t} + \mathbf{S}\mathcal{L} = 0 \quad (4.10)$$

which in the linear case can be written in the diagonal form

$$\frac{\partial \mathbf{W}}{\partial t} + \mathcal{L} = 0 \quad (4.11)$$

where $\mathbf{W} = \mathbf{S}^{-1}\mathbf{V}$. The LODI system of equations is very convenient to write relations of primitive variables and the time-variations of characteristic variables. While the diagonal sys-

tem 4.11 is useful to explain the behaviour of the boundary condition in the 1D case. For instance, in a subsonic outlet only the characteristic variable w_1 goes into the domain. In a perfectly non-reflecting boundary condition, w_1 would always be constant since no information should enter the domain which means that $\mathcal{L}_1 = 0$. Though perfectly non-reflecting this condition is of little interest in our simulations since there is no way of imposing the desired outlet condition, $p_2 = p^*$ where p^* is the desired outlet pressure. In practice, the condition $\mathcal{L}_1 = 0$ will be relaxed so that it is a function of the desired imposed state, $\mathcal{L}_1 = \mathcal{L}_1(p, p^*)$.

What is explained next are ways to compute sets of $\mathcal{L}_i$ compatible with the inlet/outlet boundary conditions that are useful for the simulation of turbomachinery rows. The general process for computing the conservative variables at the inlet/outlet boundaries can be summarised in the following steps:

1. Obtain an estimate of the ingoing $\mathcal{L}_i$ as a function of the variables to be imposed at that boundary. For instance, in a subsonic outlet where the static pressure is imposed, functional relations of the form $\mathcal{L}_1 = \mathcal{L}_1(p)$ are needed. In a subsonic inlet, the sought relations are of the form $\mathcal{L}_j = f_j(P_t, T_t, \alpha_r, \alpha_\theta)$ for $j = 2, \dots, 5$. In the present work, these relations are obtained by considering the LODI system 4.10.
2. The outgoing $\mathcal{L}_i$ are computed making use of Eq. 4.6 using only information from the domain, *i.e.* they do not depend on the boundary conditions imposed at the boundary.
3. Once all $\mathcal{L}_i$ are obtained, vector $\mathbf{d}$ can be computed with Eq. 4.7, and the conservative variables at the boundaries can be calculated for the next stage of the RK using Eq. 4.9.

As it can be seen the inlet/outlet BCs defined in this manner require marching in time a modified version of the Navier-Stokes equations at the boundaries. The evolution in time of conservative values in Eq. 4.9 allows the boundary condition to drift off the desired values [106]. This drift is necessary for the condition to be at least partially non-reflecting, *e.g.*, if a pressure wave parallel to the outlet boundary condition the outlet-averaged pressure will necessarily increase as it travels across the boundary condition. However, means to bring the mean values at the boundaries to the desired values are explained in the following sections. Odier *et al.* [105] discuss the effect of this drift and the ability of these BCs to deal with transient problems.

The following subsections describe the procedure to compute estimates of the ingoing $\mathcal{L}_i$ for subsonic inlet and outlet conditions.

4.4.2. Subsonic outlet BC

In a subsonic outlet there is a single incoming wave in the domain, therefore, four of the five characteristic waves, *i.e.*, $\mathcal{L}_2$ to $\mathcal{L}_5$, are computed with domain information using Eq. 4.6. As noted earlier, the discretised NS equations in a FR framework are expressed in differential form and so, spatial derivatives of primitive variables are available at every point. This means that calculating the outgoing $\mathcal{L}_2$ to $\mathcal{L}_5$ using Eq. 4.6 is direct. Only $\mathcal{L}_1$ is to be determined (step 1). Further, in turbomachinery, the static pressure is usually specified at the outlet. The fifth equation of the LODI system 4.10 relates the static pressure to characteristic amplitude time-variations as

$$\frac{\partial p}{\partial t} + \frac{1}{2}(\mathcal{L}_5 + \mathcal{L}_1) = 0 \quad (4.12)$$

which gives a relation between $\mathcal{L}_1$ and p . As mentioned in the previous section, this relation is used to obtain estimates of the ingoing $\mathcal{L}$. If one fixes the outlet pressure $\partial p / \partial t = 0$ equation 4.12 reduces to $\mathcal{L}_1 = -\mathcal{L}_5$ which results in a perfectly reflecting boundary. A less reflecting boundary condition can be obtained by removing the dependency on the outgoing wave¹ so that $\mathcal{L}_1 = -2\partial p / \partial t$ which indicates that pressure fluctuations will induce reflections. It has to be noted that the condition 4.12 is applied to every point of the boundary individually so that the condition $\mathcal{L}_1 = -2\partial p / \partial t$ will reflect based on the pressure fluctuations of each of the nodes in the boundary. Since we are interested in imposing just the mean pressure so that pressure variations across the boundary are allowed, a more suitable estimate of $\mathcal{L}_1$ is [102]

$$\mathcal{L}_1 = \sigma_p (\bar{p} - p_2^*) \quad (4.13)$$

¹For the Euler equations, removing the dependency of $\mathcal{L}_1$ on $\mathcal{L}_5$ is mathematically necessary since otherwise first-order spatial derivatives would be the boundary condition of an equation with only first-order spatial derivatives which is not possible.

where $\bar{p}$ is the current spatially averaged static pressure; p_2^* is the target mean static pressure at the exit; and σ_p is the static pressure relaxation coefficient which is the inverse of a time. As it can be seen, condition 4.13 provides $\mathcal{L}_1 = 0$ for ingoing waves with $\bar{p} = p_2^*$ which is perfectly non-reflecting. For perturbations with $\bar{p} \neq p_2^*$ the boundary will reflect upstream pressure waves. This is the penalty that has to be paid for forcing the mean value of p which is required in order to define the mean field. In our implementation $\sigma_p = (1 - M^2) a/L$ where L is a reference length, typically the aerofoil chord. It can be thought of $1/\sigma_p$ as the characteristic time that it takes for the boundary condition to recover the target value. The following sub-section explains this boundary condition in further detail for the one-dimensional case.

The estimate Eq. 4.13 completes step 1 whereas for step 2 we only need to recover the definition of Eq. 4.6 for the outgoing waves

$$\begin{aligned}\mathcal{L}_2 &= u \left(a^2 \frac{\partial \rho}{\partial x} - \frac{\partial p}{\partial x} \right) \\ \mathcal{L}_3 &= u \left(\frac{\partial v}{\partial x} \right) \\ \mathcal{L}_4 &= u \left(\frac{\partial w}{\partial x} \right) \\ \mathcal{L}_5 &= (u + a) \left(\frac{\partial p}{\partial x} + \rho a \frac{\partial u}{\partial x} \right)\end{aligned}\tag{4.14}$$

which are computed with information available in the domain, as mentioned.

4.4.2.1. Analysis for a 1D subsonic outlet

To understand what the above boundary condition means in physical terms, the one-dimensional linearised inviscid case is explored. For the purpose of the following linearised analysis we consider perturbation of the primitive variables over the mean state: $p' = p - p_2$, $u' = u - u_0$

and $p' = p - p_0$ so that the LODI system² can be written as

$$\begin{aligned} \frac{\partial p'}{\partial t} - \rho_0 a_0 \frac{\partial u'}{\partial t} + \mathcal{L}_1 &= 0 \\ a_0^2 \frac{\partial p'}{\partial t} - \frac{\partial p'}{\partial t} + \mathcal{L}_2 &= 0 \\ \frac{\partial p'}{\partial t} + \rho_0 a_0 \frac{\partial u'}{\partial t} + \mathcal{L}_5 &= 0 \end{aligned} \quad (4.15)$$

which can be cast in terms of the characteristic variables as

$$\begin{aligned} \frac{\partial w_1}{\partial t} + \mathcal{L}_1 &= 0 \\ \frac{\partial w_2}{\partial t} + \mathcal{L}_2 &= 0 \\ \frac{\partial w_5}{\partial t} + \mathcal{L}_5 &= 0 \end{aligned} \quad (4.16)$$

where $a_0^2 = \gamma p_0 / \rho_0$, $w_1 = p' - \rho_0 a_0 u'$, $w_2 = a_0^2 p' - p'$ and $w_5 = p' + \rho_0 a_0 u'$. As mentioned in the previous section, only the ingoing wave, w_1 , is to be computed at the boundary. If $\mathcal{L}_1 = 0$ the equation for the ingoing wave reduces to $\partial w_1 / \partial t = 0$ the solution of which is $w_1 = w_{1,0}$ so that if it was initially zero then $w_1 = 0 \forall t$. This means that no incoming wave is generated at the outlet, *i.e.*, the outlet boundary condition is perfectly non-reflecting. Combining the first and fifth characteristic variables we find that $p' = (w_1 + w_5) / 2$. Since w_5 is, in general, not constant, this means that the pressure varies and, in the general case, will be able to drift off the desired value $p' = 0$ or $p = p_2$.

In order to avoid the pressure from drifting away, the condition 4.13 is used where the pressure is allowed to have fluctuations p' in times of the order of $1/\sigma_p$, *i.e.*, $\mathcal{L}_1 = \sigma_p p'$. By substituting in the first equation of the system 4.16 the equation for the reflected ingoing wave can be obtained.

$$\frac{\partial w_1}{\partial t} + \sigma_p \frac{w_1 + w_5}{2} = 0 \quad (4.17)$$

It is then clear that this boundary condition is indeed reflective. The amount of reflection of this condition can be computed using this equation. We analyse the reflection by assuming that we have a periodic outgoing wave of the form $w_5 = \hat{w}_5 \exp(I\omega t)$ where, without loss of generality,

²The equations for velocity components in the perpendicular directions are removed in the 1D case. The numbering of the 3D case is kept for consistency.

$\hat{w}_5$ is assumed to be real. Since this is a linear system, the response will also be periodic so that we can seek solutions of the form $w_1 = \hat{w}_1 \exp(I\omega t)$. Substituting in the previous equation we obtain the solution

$$\frac{\hat{w}_1}{\hat{w}_5} = \frac{-1 + ISt}{1 + St^2} \quad (4.18)$$

where the Strouhal number St as been defined as $St = 2\omega/\sigma_p$ and represents the ratio of characteristic time-variations of reflected ingoing waves ($1/\sigma_p$) to the time-variations of outgoing waves ($1/\omega$). If the time variations of the outgoing perturbations have frequencies of the order of σ_p or much lower, *i.e.*, $St \sim \mathcal{O}(1)$ or $St \ll 1$, the ration $\Re(\hat{w}_1)/\hat{w}_5$ will be of $\mathcal{O}(1)$ and so there will be large reflections in the outlet. However, for frequencies of the outgoing waves much larger than σ_p , *i.e.* $St \gg 1$, the ratio $\hat{w}_1/\hat{w}_5$ tends to zero and so the reflection will be small. In practice, this means that σ_p needs to be chosen considerably lower than the frequency of the flow features which are intended to be accurately solved. For turbulent eddies arriving at the boundary $\omega \sim u/L_{eddy}$ whereas, following [102] $\sigma_p = (1 - M^2)a/L$ so that $St \sim \frac{2uL}{(1-M^2)aL_{eddy}}$ which for low subsonic flows ($M \sim 0.1 - 0.3$) can be approximated as $St \sim M \frac{L}{L_{eddy}}$ which is large as long as the ratio of reference length to eddy length remains large.

The previous analysis gives an idea of the reflectiveness of the outlet boundary condition used here in the 1D case. In a 2D case, the situation would be analogous as long as the outgoing perturbation is 1D. In the general 3D condition the behaviour of the boundary condition is, in general, less reflective since just mean values of the exit pressure are enforced in Eq. 4.13 to estimate $\mathcal{L}_1$. In fact large perturbations with zero mean over the imposed pressure would not be reflected.

4.4.3. Subsonic inlet BC

The subsonic inlet boundary condition is conceptually equivalent to that of the outlet case except the algebra becomes more involved. In order to exemplify the process of deriving inlet boundary conditions a simplified example for the 1D linearised Euler case where ρ and u are imposed is first discussed.

In a subsonic 1D inlet two waves go into the domain and only one wave goes out. This is, the evolution of $\mathcal{L}_2$ and $\mathcal{L}_5$ is to be found ($\mathcal{L}_3$ and $\mathcal{L}_4$ are zero since no transverse velocity components are considered) which constitutes step 1. For simplicity in the analysis we consider perturbations over the mean values, $\rho' = \rho - \rho_0$, $u' = u - u_0$ and $p' = p - p_0$, so that the first and second equations of the linearised LODI system 4.10 can be written as

$$\begin{aligned}\frac{\partial \rho'}{\partial t} + \frac{1}{a_0^2} \left(\mathcal{L}_2 + \frac{\mathcal{L}_5 + \mathcal{L}_1}{2} \right) &= 0 \\ \frac{\partial u'}{\partial t} + \frac{1}{2\rho a_0} (\mathcal{L}_5 - \mathcal{L}_1) &= 0\end{aligned}\tag{4.19}$$

As happens in the outlet, if $\partial u / \partial t = 0$ and $\partial \rho / \partial t = 0$, then $\mathcal{L}_5 = \mathcal{L}_1$ and $\mathcal{L}_2 = -\mathcal{L}_1$ which is a perfectly reflecting boundary condition. As explained previously, $\mathcal{L}_2$ and $\mathcal{L}_5$ cannot depend on $\mathcal{L}_1$ so that the estimates for $\mathcal{L}_2$ and $\mathcal{L}_5$ become

$$\begin{aligned}\mathcal{L}_2 &= -a_0^2 \frac{\partial \rho'}{\partial t} + \rho_0 a_0 \frac{\partial u'}{\partial t} \\ \mathcal{L}_5 &= -2\rho_0 a_0 \frac{\partial u'}{\partial t}\end{aligned}\tag{4.20}$$

In order to obtain relations of the form $\mathcal{L}_i = \mathcal{L}_i(\rho, u, \rho_0, u_0)$ time derivatives are substituted by $\partial u / \partial t = \sigma_u(u - u_0)$ and $\partial \rho / \partial t = \sigma_\rho(\rho - \rho_0)$ so that

$$\begin{aligned}\mathcal{L}_2 &= -a_0^2 \sigma_\rho \rho' + \rho_0 a_0 \sigma_u u' \\ \mathcal{L}_5 &= -2\rho_0 a_0 \sigma_u u'\end{aligned}\tag{4.21}$$

which complete step 1. As opposed to the outlet case where only one equation had to be considered, in the subsonic inlet case the outgoing characteristics are coupled. The primitive variables can be written in terms of the characteristic variables as

$$\begin{aligned}\rho' &= \frac{1}{a_0^2} \left(w_2 + \frac{w_1 + w_5}{2} \right) \\ u' &= \frac{1}{2\rho_0 a_0} (w_5 - w_1)\end{aligned}\tag{4.22}$$

so that the characteristic variables equations 4.16 are expressed as

$$\begin{aligned}\frac{\partial w_2}{\partial t} - \sigma_p w_2 - w_1 \frac{\sigma_p + \sigma_u}{2} + w_5 \frac{\sigma_u - \sigma_p}{2} &= 0 \\ \frac{\partial w_5}{\partial t} - \sigma_u \frac{w_5 - w_1}{2} &= 0\end{aligned}\quad (4.23)$$

which describes the time-evolution of the ingoing waves w_2 and w_5 as a function of the outgoing wave w_1 and highlights the coupled nature of the inlet boundary condition. Since the above is a linear system, if we consider an outgoing wave of the form $w_1 = \hat{w}_1 \exp(I\omega t)$ where, without loss of generality, $\hat{w}_1$ is considered to be real; the ingoing waves will be of the form $w_2 = \hat{w}_2 \exp(I\omega t)$ and $w_5 = \hat{w}_5 \exp(I\omega t)$ and the system has the solution

$$\begin{aligned}\frac{\hat{w}_2}{\hat{w}_1} &= -\frac{1 + \frac{\sigma_p}{\sigma_u}}{St_u^2 + \left(2\frac{\sigma_p}{\sigma_u}\right)^2} \left(2\frac{\sigma_p}{\sigma_u} + ISt_u\right) - \frac{1 - \frac{\sigma_p}{\sigma_u}}{St_u^2 + \left(2\frac{\sigma_p}{\sigma_u}\right)^2} \frac{(1 + ISt_u) \left(2\frac{\sigma_p}{\sigma_u} + ISt_u\right)}{1 + St_u^2} \\ \frac{\hat{w}_5}{\hat{w}_1} &= \frac{1 + ISt_u}{1 + St_u^2}\end{aligned}\quad (4.24)$$

where $St_u = 2\omega/\sigma_u$. It can be seen that the response of the inlet to the outgoing waves is now dependent on two parameters, namely St_u and σ_p/σ_u which can make this solution more difficult to interpret. For simplicity the ratio of times σ_p/σ_u is usually taken to $\sigma_p/\sigma_u = 1$ [106] so that the above solution behaves analogously to that of the outlet, this is, for $St \gg 1$ the ratios $\hat{w}_2/\hat{w}_1 \approx 0$ and $\hat{w}_5/\hat{w}_1 \approx 0$ and non-reflecting boundary condition is recovered. This is to say that outgoing perturbations with characteristic times $(1/\omega)$ much shorter than those of the inlet $(1/\sigma_i)$ are not reflected whereas perturbations with characteristic times of the order of the characteristic time of the inlet response will be reflected.

The boundary condition is completed with step 2 which gives the value of $\mathcal{L}_1$ as

$$\mathcal{L}_1 = (u - a) \left(\frac{\partial p}{\partial x} - \rho a \frac{\partial u}{\partial x} \right) \quad (4.25)$$

which is computed the values already available in the domain.

Subsonic inlet in turbomachines

Inlet boundary conditions in turbomachinery are usually imposed in terms of total pressure, P_t , total temperature, T_t , and flow swirl and radial angles, α_θ and α_r , respectively. This means that the functional $\mathcal{L}_i = f_i(P_t, T_t, \alpha_r, \alpha_\theta)$ with $i = 2, \dots, 5$ involves more complicated algebra than the subsonic inlet presented above. In addition, the relations $\mathcal{L}_i = f_i(P_t, T_t, \alpha_r, \alpha_\theta)$ are not linear which complicates the analysis of the inlet response to outgoing perturbations. Despite these differences the procedure to impose the inlet boundary conditions in turbomachinery flows remains the same. The relations of these variables with the primitive variables is

$$\begin{aligned} \frac{P_t}{p} &= \left(1 + \frac{\gamma-1}{2} M^2\right)^{\frac{\gamma}{\gamma-1}} \\ \frac{T_t}{T} &= \left(1 + \frac{\gamma-1}{2} M^2\right) \\ \tan \alpha_\theta &= \frac{v}{\sqrt{u^2 + w^2}} \\ \tan \alpha_r &= \frac{w}{u} \end{aligned} \quad (4.26)$$

where M is the Mach number, $M^2 = (u^2 + v^2 + w^2)/a^2$. By differentiating these equalities 4.26 with respect to time and after some algebra one gets

$$\begin{aligned} \frac{1}{\gamma} \left(\frac{1}{P_t} \frac{\partial P_t}{\partial t} - \frac{1}{p} \frac{\partial p}{\partial t} \right) &= M \left(1 + \frac{\gamma-1}{2} M^2\right)^{-1} \frac{\partial M}{\partial t} \\ \frac{1}{\gamma-1} \left(\frac{1}{T_t} \frac{\partial T_t}{\partial t} - \frac{1}{T} \frac{\partial T}{\partial t} \right) &= M \left(1 + \frac{\gamma-1}{2} M^2\right)^{-1} \frac{\partial M}{\partial t} \\ (u^2 + w^2) \tan \alpha_r \frac{\partial \alpha_\theta}{\partial t} &= v \frac{\partial v}{\partial t} \cos^2 \alpha_\theta - \left(u \frac{\partial u}{\partial t} + w \frac{\partial w}{\partial t} \right) \sin^2 \alpha_\theta \\ u (1 + \tan^2 \alpha_r) \frac{\partial \alpha_r}{\partial t} &= \frac{\partial w}{\partial t} - \tan \alpha_r \frac{\partial u}{\partial t} \\ M \frac{\partial M}{\partial t} &= \frac{1}{a^2} \left(u \frac{\partial u}{\partial t} + v \frac{\partial v}{\partial t} + w \frac{\partial w}{\partial t} \right) - \frac{1}{2} \frac{M^2}{T} \frac{\partial T}{\partial t} \end{aligned} \quad (4.27)$$

Using the relations 4.27 and 4.10 it is possible to compute time-derivatives of the variables of interest (P_t , T_t , α_θ and α_r) as function of $\mathcal{L}_i$. The reader is referred to [106] for the details about

the algebraic operations. The final expressions are stated here for reference

$$\begin{aligned}
 \mathcal{L}_5 &= \frac{1}{F_4 F_2 + F_5 F_1} \left[F_1 \frac{\partial T_t}{\partial t} + F_2 \frac{\partial P_t}{\partial t} + \frac{\gamma}{\gamma-1} \frac{P_t}{C_p T_t} F_2 F_3 + \frac{1}{C_p} F_1 F_3 - (F_2 F_6 + F_1 F_7) \mathcal{L}_1 \right] \\
 \mathcal{L}_2 &= \frac{1}{F_2} \frac{\partial T_t}{\partial t} + \frac{F_3}{C_p F_2} - \frac{F_5}{F_2} \mathcal{L}_5 - \frac{F_7}{F_2} \mathcal{L}_1 \\
 \mathcal{L}_3 &= -\frac{2v}{\sin(2\alpha_\theta)} \frac{\partial \alpha_\theta}{\partial t} - \frac{v}{\sqrt{u^2 + w^2}} \left[\frac{uw}{\cos^2 \alpha_r} \frac{\partial \alpha_r}{\partial t} - (u + w \tan \alpha_r) \frac{\mathcal{L}_5 - \mathcal{L}_1}{2\rho a} \right] \\
 \mathcal{L}_4 &= \frac{\mathcal{L}_5 - \mathcal{L}_1}{2\rho a} \tan \alpha_r - \frac{u}{\cos^2 \alpha_r} \frac{\partial \alpha_r}{\partial t}
 \end{aligned} \tag{4.28}$$

where

$$\begin{aligned}
 F_1 &= \frac{u^2 + v^2 + w^2}{2} \frac{\gamma}{\gamma-1} \frac{P_t}{\rho C_p T_t} \\
 F_2 &= \frac{T_t}{\rho} - \frac{u^2 + v^2 + w^2}{2\rho C_p} \\
 F_3 &= v\mathcal{L}_3 + w\mathcal{L}_4 \\
 F_4 &= -\frac{\rho a P_t}{2} \frac{1}{p} + \frac{\gamma}{\gamma-1} \frac{P_t}{2C_p T_t} \left[(\gamma-1) \frac{u^2 + v^2 + w^2}{a} - u \right] \\
 F_5 &= -\frac{\gamma-1}{2} \frac{T_t}{a} + \frac{1}{2C_p} \left[(\gamma-1) \frac{u^2 + v^2 + w^2}{a} - u \right] \\
 F_6 &= -\frac{\rho a P_t}{2} \frac{1}{p} + \frac{\gamma}{\gamma-1} \frac{P_t}{2C_p T_t} \left[(\gamma-1) \frac{u^2 + v^2 + w^2}{a} + u \right] \\
 F_7 &= -\frac{\gamma-1}{2} \frac{T_t}{a} + \frac{1}{2C_p} \left[(\gamma-1) \frac{u^2 + v^2 + w^2}{a} + u \right]
 \end{aligned} \tag{4.29}$$

As in the outlet case, if P_t, T_t, α_r and α_θ are fixed, i.e., $\partial/\partial t = 0$, the ingoing waves $\mathcal{L}_2$ to $\mathcal{L}_5$ are expressed as a function of $\mathcal{L}_1$ which result in reflecting boundary conditions. A better estimate of the ingoing waves is given by removing the explicit dependency on $\mathcal{L}_1$ in the previous relations which results in [106]

$$\begin{aligned}
 \mathcal{L}_5 &= \frac{1}{F_4 F_2 + F_5 F_1} \left[F_1 \frac{\partial T_t}{\partial t} + F_2 \frac{\partial P_t}{\partial t} + \frac{\gamma}{\gamma-1} \frac{P_t}{C_p T_t} F_2 F_3 + \frac{1}{C_p} F_1 F_3 \right] \\
 \mathcal{L}_2 &= \frac{1}{F_2} \frac{\partial T_t}{\partial t} + \frac{F_3}{C_p F_2} - \frac{F_5}{F_2} \mathcal{L}_5 \\
 \mathcal{L}_3 &= -\frac{2v}{\sin(2\alpha_\theta)} \frac{\partial \alpha_\theta}{\partial t} - \frac{v}{\sqrt{u^2 + w^2}} \left[\frac{uw}{\cos^2 \alpha_r} \frac{\partial \alpha_r}{\partial t} - (u + w \tan \alpha_r) \frac{\mathcal{L}_5}{2\rho a} \right] \\
 \mathcal{L}_4 &= \frac{\mathcal{L}_5}{2\rho a} \tan \alpha_r - \frac{u}{\cos^2 \alpha_r} \frac{\partial \alpha_r}{\partial t}
 \end{aligned} \tag{4.30}$$

which indicate that there will be reflections as long as the time-derivatives of $P_t, T_t, \alpha_r, \alpha_\theta$ are different than zero. These conditions are applied at each of the nodes so that the reflections de-

pend on the local values of the time derivatives. Since we are interested in imposing $P_t, T_t, \alpha_r, \alpha_\theta$ in a mean sense only, a better estimate of the ingoing $\mathcal{L}_i$ is made following the logic of the outlet, i.e.,

$$\frac{\partial \phi}{\partial t} = \sigma_\phi (\bar{\phi} - \phi^*) \quad (4.31)$$

where ϕ is any of $P_t, T_t, \alpha_r, \alpha_\theta$; $\bar{\phi}$ is its space average in the boundary; ϕ^* is the target value; and σ_ϕ is the relaxation coefficient for variable ϕ . These estimates used in the relations of Eq. 4.30 complete step 1. The same relaxation coefficient as that taken for the outlet boundary condition has been chosen for $P_t, T_t, \alpha_r, \alpha_\theta$. As with the outlet boundary condition, this will lead to certain level of reflection depending on the frequency of the outgoing perturbations in relation to σ_ϕ . Outgoing perturbations of $P_t, T_t, \alpha_r, \alpha_\theta$ with zero spatial mean at the inlet will cause $\mathcal{L}_2$ to $\mathcal{L}_5$ to be zero and the boundary will be perfectly non-reflecting.

As with the subsonic inlet presented before, the outgoing wave is calculated with Eq. 4.6 as

$$\mathcal{L}_1 = (u - a) \left(\frac{\partial p}{\partial x} - \rho a \frac{\partial u}{\partial x} \right). \quad (4.32)$$

which completes step 2.

Figure 4.3 shows the sequence of a pressure wave leaving the computational domain. The top simulation in Fig. 4.3 has been carried out without the relaxation coefficient, $\sigma_i = 0$, whereas the bottom row simulation is done with the relaxation coefficients on, $\sigma_i \neq 0$. It can be seen that the top row is able to “absorb” the outgoing pressure wave without any reflection. The simulation at the bottom, however, reflects a planar wave. This is because the BC is trying to maintain the mean total pressure at the desired level. Since the correction to the derivatives due to changes in mean values are applied to all points of the BC at the same time a planar reflection wave is created. Although this does look like a major drawback, in practice, this is not a major concern, since these reflections are of a much lower magnitude than the outgoing waves. The example given in Fig. 4.3 is, indeed, a limiting case of a pressure wave caused during the start-up process of the simulation when the initial condition does not fulfil the wall BC which is enforced during the first time-step.

Figure 4.3.: Time sequence of a circular pressure wave leaving the computational domain after an impulsive start. Effect of the relaxation parameters on the inlet BC. Top: without relaxation coefficients ($\sigma_i = 0$). Bottom: with relaxation coefficients ($\sigma_i \neq 0$).

5. Test Cases

Contents

5.1. Introduction	103
5.2. Inviscid 2D vortex in uniform flow	103
5.3. Blasius boundary layer	108
5.4. Taylor-Green vortex	111
5.5. Turbulent channel simulation	114

5.1. Introduction

In this chapter test cases are presented in increasing order of complexity to make sure that the implemented numerical scheme produces the expected results. These test cases span from simple laminar cases to a fully turbulent case to ensure that the relevant aspects of turbine aerodynamics can be captured accurately.

5.2. Inviscid 2D vortex in uniform flow

The first of the test cases is that of the inviscid 2D vortex transport problem in a uniform flow. This problem is defined in Wang *et al.* [50] and aims at testing the capability of the numerical scheme to preserve vorticity in an inviscid flow configuration. This is of vital importance in

turbulence resolving simulations such as LES where the correct transport of vortical structures is a must.

The problem is defined as a two-dimensional vortex superimposed over a compressible uniform flow. The vortex in non-dimensional variables is defined as follows

$$\begin{aligned}\delta\tilde{u}(x,y) &= -\beta\tilde{y}\exp\left(-\frac{\tilde{r}^2}{2}\right) \\ \delta\tilde{v}(x,y) &= \beta\tilde{x}\exp\left(-\frac{\tilde{r}^2}{2}\right) \\ \delta T(x,y) &= \frac{\gamma-1}{2}(\beta M_\infty)^2 \exp(-\tilde{r}^2)\end{aligned}\tag{5.1}$$

so that the initial conditions are

$$\begin{aligned}\tilde{u}_0(x,y) &= 1 + \delta\tilde{u}(x,y) \\ \tilde{v}_0(x,y) &= \delta\tilde{v}(x,y) \\ \tilde{T}_0(x,y) &= 1 - \delta\tilde{T}(x,y) \\ \tilde{\rho}_0(x,y) &= (\tilde{T}_0(x,y))^{1/(\gamma-1)} \\ \tilde{p}_0(x,y) &= \tilde{\rho}_0(x,y)\tilde{T}_0(x,y)\end{aligned}\tag{5.2}$$

where the non-dimensional variables are defined as

$$\tilde{x} = \frac{x-x_c}{R}; \tilde{y} = \frac{y-y_c}{R}; \tilde{u} = \frac{u}{U_\infty}; \tilde{v} = \frac{v}{U_\infty}; \tilde{T} = \frac{T}{T_\infty}; \tilde{\rho} = \frac{\rho}{\rho_\infty}; \tilde{p} = \frac{p}{p_\infty}; \tilde{r}^2 = \tilde{x}^2 + \tilde{y}^2\tag{5.3}$$

with $p_\infty = \rho_\infty R_{\text{gas}} T_\infty$; U_∞ is the velocity of the unperturbed flow (assumed to be aligned with the x axis) so that $M_\infty = U_\infty / \sqrt{\gamma R_{\text{gas}} T_\infty}$, β is the intensity of the vortex, x_c and y_c , the coordinates of the vortex centre, and R its radius. The exact solution of this problems corresponds to a translation of the vortex centre with velocity U_∞ , this is, $x_c(t) = x_{c0} + U_\infty t$. It is clear that the lower the Mach number the more challenging from a numeric standpoint since more time-steps are needed for a convective perturbation, such as the vorticity, to move across the domain since in an explicit scheme the time-step stability limit is controlled by the sound speed and not by U_∞ . For reference, Fig. 5.1 shows the vertical component of the velocity, δv , described above,

Figure 5.1.: Contour plots of the vertical velocity component of the Euler vortex in uniform flow (left) and linear mesh used for the simulation (right).

which is indicative of the vorticity for this case. All the boundaries in the domain are periodic which allows to test the method without interference of the boundary condition implementation. A Mach number of $M_\infty = 0.05$ has been chosen for this experiment as this is a difficult case for a compressible solver. Figure 5.1 (right) shows the linear¹ mesh used for this case. The same experiment was conducted with triangular elements with equivalent results. The simulation is run so that the vortex runs for 40 through-flows, where the through-flow is defined as $T = L/U_\infty$, where L is the length of the domain. Since L is around 8 times the size of the diameter (distance between the maximum and minimum of the velocity), the simulation is run for about 300 vortex diameters. As it can be seen in Fig. 5.1 the length L is discretised with 14 elements which means that the vortex is simulated with about 2 elements in each direction.

Figure 5.2 shows the vertical velocity, v , for a cross section at $y = 0.5L$ (across the middle of the vortex) for the exact solution, and the numerical solutions after 40 through-flows for two different polynomial orders, $p = 3$ (fourth-order solution) and $p = 5$ (sixth-order solution). As it can be seen, there is no appreciable dissipation of the vortex as it is convected for the sixth-order solution. For this discretization order the vortex is represented with 11 solution points in each direction and it is, hence, very well resolved. Both the analytical and the numerical solution lay one on top of the other. When considering a fourth order solution, however, the solution is

¹The linear mesh shows the elements used.

Figure 5.2.: Vertical velocity across the mid section of the vortex for the exact and numerical solution after 40 through-flows for different polynomial orders: $p = 1$, $p = 3$ and $p = 5$.

dissipated by about 22% after running over 40 domain lengths (300 vortex diameters). For this case the vortex is discretised in 7 points in each direction which is still a well resolved case. In fact, for this case $k\Delta x/(p+1) \approx 1$ which should be well resolved (see Fig. 2.5 and 2.7). This highlights the fact that even small dissipation can add up to substantial effects when run over very long periods of time. For comparison, the solution for a second-order approximation using the FR method has been included in the figure. As it can be seen, for this order of accuracy, the solution has been almost completely dissipated.

Although this is not a fair comparison between the different orders of accuracy due to the fact that the number of points is indeed changing for each order it does show the capabilities of the method to transport vortices for large distances (up to 300 diameters) with minimum dissipation for resolutions in the order of $k\Delta x/(p+1) \approx 1$.

A fairer comparison would be to compare the solution of the $p = 3$ simulation with a $p = 5$ simulation with the same number of nodes. This is presented in Fig. 5.3. For this comparison the number of elements across the side of the domain has been reduced to 8 so that the total number of nodes across each direction for a $p = 5$ discretization is 41 (physical nodes) which is close to the number of physical nodes in the previous case for a $p = 3$ discretization (43 physical nodes). As it can be seen in the figure, even with a slightly lower number of nodes the $p = 5$ case has suffered from less dissipation. It is clear that, for a given number of nodes, increasing

Figure 5.3.: Vertical velocity across the mid section of the vortex after 40 flow-through (300 vortex diameters) for a discretization $p = 3$ and $p = 5$ with very similar number of nodes across the domain (43 for the $p = 3$ mesh and 41 for the $p = 5$ mesh).

the order of the solution yields more accurate solutions.

Since the exact solution for this case is available the total error of integration can be readily compute and the order of accuracy checked. To do this a number of meshes have been created and the code run for different polynomial orders. For each mesh the vortex is run for 10 flow-throughs (75 vortex diameters) and the solution is compared against the exact solution. The error can be computed as

$$\varepsilon = \sqrt{\frac{\sum_i^N (u(x, y) - u_i^{\text{exact}}(x, y))^2}{N}} \quad (5.4)$$

where N is the number of nodes of the domain and u_i^{exact} is the exact solution. Figure 5.4 shows the error of the vertical component of the velocity for all the cases considered. As it can be seen, two polynomial orders have been considered, namely $p = 3$ and $p = 5$, and for each polynomial order 4 different meshes are run. The x -axis shows the number of elements used across each direction. As it can be seen, the error follows the expected line of convergence indicating that indeed the spatial discretization scheme is of the desired order of accuracy.

Another important aspect that is worth commenting on is that of the time-integration accuracy. The results shown in Fig. 5.4 have been obtained with the maximum time-step allowed by stability. Given that the time-integration scheme used is fourth-order, it would not be surprising if the time-integration error would dominate in the sixth-order approximation. As it can be

Figure 5.4.: Convergence of error with mesh refinement for the Euler vortex case for two different polynomial orders, $p = 3$ and $p = 5$.

seen, however, the error follows a -6 slope in logarithmic coordinates. It is noted, then, that the time-integration error is much lower than the spatial discretization error, at least for polynomial orders as high as 5.

In summary, it has been demonstrated that the present scheme is able to transport fairly resolved vortices over large distances (300 vortex diameters) with minimum dissipation and marginal resolution. It has also been seen that, for a given number of nodes, using higher order polynomials yields more accurate results. Finally, it has been shown that the order of convergence is the expected one given by the polynomial order and that the temporal error is not dominant and much lower than that of the spatial discretisation.

5.3. Blasius boundary layer

This test case is intended to test the implementation of the viscous terms as well as the accuracy of the viscous solutions. The order of convergence of the cases is also seen here. The test case consists of a flat plate on a uniform flow without angle of attack. A schematic of this test case can be seen in Fig. 5.5 (left). The Mach number has been chosen to be $M = 0.2$ which is considerably higher than in the previous case although still low enough as to consider the flow as incompressible and recover the Blasius boundary layer solution. One of the meshes used for

Figure 5.5.: Schematic of the flat plate test case (left) and one of the meshes used in the study (right).

the present experiments is shown in the same figure (right). Different levels of refinement for the bottom side were tried to investigate the resolution effect on the boundary layer solution. The mesh refinement is done by doubling the number of points in the boundary layer at each level of refinement while keeping the mesh it departed from. In this manner, the finer mesh contains the nodes of all the courser meshes. The coarser mesh is designed to have 4 elements across the boundary layer at 80% of the length of the plate downstream of the leading edge of the plate. The initial mesh is stretched near the wall.

The numerical solution was compared against the Blasius solution. The thin layer Navier-Stokes approximation of a incompressible flow is obtained assuming that the characteristic length in the stream-wise direction, x , is much longer than in the normal direction, y , and hence $(\partial/\partial x) \ll (\partial/\partial y)$ and $v \ll u$. In this case, the continuity equation and the momentum equations in the x and y direction reduce respectively to [107]:

$$\begin{aligned} \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0 \\ u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} &= -\frac{1}{\rho} \left(\frac{\partial p}{\partial x} \right)_e + \nu \frac{\partial^2 u}{\partial y^2} \\ \frac{\partial p}{\partial y} &= 0 \end{aligned} \quad (5.5)$$

Blasius proposed a similarity solution in the case of external constant free flow, $dU_e/dx = 0$ or $(\partial p/\partial x)_e = 0$, and introduced the following self-similar variables

$$\eta = y \sqrt{\frac{U_e}{\nu x}} \text{ and } \psi = \sqrt{2\nu U_e x} f(\eta) \quad (5.6)$$

where ψ is the stream function such that $u = \psi_y$ and $v = -\psi_x$. Substituting in the momentum equation in the x -direction yields the Blasius equation

$$2\frac{d^3f}{d\eta^3} + f\frac{d^2f}{d\eta^2} = 0 \quad (5.7)$$

which is an ordinary differential equation. Figure 5.6 shows the solution of the previous equation ($u/U_\infty = df/d\eta$) along with the numerical solution for different meshes. For each mesh resolution the solution is always displayed at the same location along the x -axis, at $x = 0.8$ (80% of the plate length downstream of the leading edge of the plate). It can be seen that the Blasius boundary layer solution is recovered accurately for all mesh resolutions tried. Further, it is seen that four fourth-order elements are enough to recover the Blasius boundary layer solution.

Since different meshes with different mesh refinement have been used it is possible to analyse the order of convergence with decreasing element size for a given polynomial order. Taking the finest mesh solution as the reference solution an error function is defined as

$$\varepsilon = \sqrt{\frac{\sum_i^N (u(\eta_i) - u_i^{\text{ref}}(\eta_i))^2}{N}} \quad (5.8)$$

where u^{ref} is the reference solution and η_i are coordinates of coarse mesh which are all contained in the reference mesh (most refined mesh). Nodes internal to the element have not been considered. Table 5.1 shows the errors for each case against the finest mesh solution along with the order of convergence. The mesh size in the y -direction in non-dimensional form, η , is included for the first element off the wall. To compute the convergence order (κ) the error, ε , is assumed to be proportional to the mesh spacing, Δy , so that $\varepsilon = a\Delta y^\kappa$. The order of convergence can be readily computed with two sets of error and mesh spacing as $\kappa = (\log \varepsilon_1 - \log \varepsilon_2) / (\log \Delta \eta_1 - \log \Delta \eta_2)$. As it can be seen the order of convergence of the error between mesh 1 and 2 is 1.72 which is considerably lower than the expected convergence order whereas the order of convergence between meshes 2 and 3 is 3.5 which is closer to the expected value of 4. The low convergence order between meshes 1 and 2 could be due the effect of the leading edge of the flat plate at which low mesh resolution can lead to differences in the

Figure 5.6.: The Blasius boundary layer solution (solid line) is compared against the results of the simulations for different meshes (solid dots) with polynomial degree $p = 3$. For clarity in the plot only solutions at the elements boundaries have been plotted.

Table 5.1.: Summary of the errors and convergence order of the Blasius solution.

mesh #	$\Delta\eta$	error	Order of convergence
1	1.64	2.27×10^{-3}	-
2	0.78	6.29×10^{-4}	1.72
3	0.38	5.07×10^{-5}	3.5

pressure and velocity fields.

5.4. Taylor-Green vortex

The Taylor-Green vortex case [98] in three dimensions has been used here to assess the implementation of viscous terms which expands the two-dimensional validation of the Blasius boundary layer. This test case consists of an initial vortex distribution with a zero-mean velocity that, for large enough Reynolds numbers, becomes unstable breaking them down into smaller and smaller eddies. Figure 5.7 shows the initial vorticity (left) and vorticity after 10 characteristic times (defined below) (Figure 5.7 right) where it is seen that vortical scales have

Figure 5.7.: Iso-surfaces of vorticity at $t = 0$ (left) and $t = 10t_c$ (right) for the Taylor-Green vortex at $Re = 1600$ for polynomial order $p = 3$ and a mesh with 128 elements across each direction.

decreased. As the vortices become smaller, viscosity effects become more dominant, and vortices are dissipated. The domain is a cube of edge length πL periodic in all 6 faces. As with the Euler vortex case, this is also an ideal case since no boundary conditions need to be imposed. The initial condition is given by

$$\begin{aligned}
 \frac{u}{U_0} &= \sin \tilde{x} \cos \tilde{y} \cos \tilde{z} \\
 \frac{v}{U_0} &= -\cos \tilde{x} \sin \tilde{y} \cos \tilde{z} \\
 \frac{w}{U_0} &= 0 \\
 \frac{p - p_0}{\rho_0 U_0^2} &= \frac{1}{16} [\cos 2\tilde{x} \cos 2\tilde{y}] [\cos 2 + 2] \\
 \rho &= \frac{P}{RT_0} \\
 T &= T_0
 \end{aligned} \tag{5.9}$$

where the sub-index 0 denotes an initial value and $\tilde{x} = x/L$, $\tilde{y} = y/L$ and $\tilde{z} = z/L$. The Reynolds number of this case is defined as $Re = \rho_0 U_0 L / \mu$ and has been set to 1600. The characteristic convective time is defined as $t_c = L / U_0$. The domain has been meshed with 128 fourth-order

Figure 5.8.: Time evolution of the kinetic energy dissipation.

hexahedra in each direction and run on 80 GPUs. With this setup, 20 characteristics times were computed in 70 minutes. The vorticity at $t = 0$ can be seen in Fig. 5.7 left.

The total kinetic energy dissipation rate is defined as

$$-\frac{dE_k}{dt} = -\frac{d}{dt} \left(\frac{1}{\rho_0 \Omega} \int_{\Omega} \rho \frac{\mathbf{v} \cdot \mathbf{v}}{2} d\Omega \right) \quad (5.10)$$

where Ω is the volume of the domain. Figure ?? compares the total kinetic energy dissipation rate obtained with the present solver against that of the spectral DNS data provided in <https://how5.cenaero.be>. As it can be seen the dissipation increases steadily up to time $t/t_c \simeq 9$. During this time the initial vortex distribution becomes unstable with vortices breaking down into smaller ones hence steadily increasing the rate of energy dissipation. Since there is not a source of kinetic energy in this case the dissipation rate has to peak at a value and then decrease from there as the flow comes to a halt. As it can be seen in the figure, the present code is able to predict the dissipation rate throughout the simulation very accurately. The peak of kinetic energy dissipation is slightly under-predicted but overall agreement is excellent. It is then demonstrated that the viscous terms of the equations have been properly implemented.

Figure 5.9.: Turbulent channel case configuration.

5.5. Turbulent channel simulation

Unlike in the previous case, the present section deals with wall-bounded turbulence. Among the several wall-bounded turbulent flows that could be analysed, probably the simplest of them in terms of geometric simplicity is the fully turbulent channel. The fully developed turbulent channel flow case has been used as a validation case for discontinuous Galerkin like methods by other authors, *e.g.* Carton and Hillewaert [108]. This case represents the flow between two parallel walls at a distance 2δ from each other. Figure 5.9 shows the domain configuration. The mean flow is aligned with the x -direction, the lower and upper walls are at $y = 0$ and $y = 2\delta$ respectively and z is the span-wise direction. Periodicity is assumed in the stream- and span-wise directions so that a forcing term representing the stream-wise pressure gradient is added to the equations to set the mean flow. This is an ideal test-case since no inlet or outlet boundary conditions need to be specified hence significantly reducing the number of uncertainties.

To keep the cost of the simulations low we have run the lowest Reynolds number out of the DNS data in [1], *i.e.*, $Re_\tau = 180$. A second simulation at $Re_\tau = 325$ has been run using the same mesh to analyse the sensitivity of the results to mesh resolution. The mesh is rather coarse and consists of 41 equally spaced fourth-order elements in each direction, which translates to 4.4M nodes (1.9M physical points, *i.e.*, removing repeated nodes). This case was partitioned in 40 sub-domains so that it could run 70 flow-throughs (700,000 iterations) in four hours. The Mach number of the simulation has been limited to $M = 0.1$ so that it can be considered as incompressible. The friction Reynolds number is defined as $Re_\tau = \rho u_\tau \delta / \mu$ where the friction velocity, u_τ , is defined as $u_\tau^2 = \tau_w / \rho$ where τ_w is the shear stress at the wall, $\tau_w = \mu \frac{\partial u}{\partial y}$, and the physical computational box is $L_x = 2\pi\delta$, $L_y = 2\delta$ and $L_z = \pi\delta$ (or $L_x^+ = 1090$, $L_y^+ = 347$ and

Figure 5.10.: Left: mean velocity profile in wall units for $Re_\tau = 180$ and $Re_\tau = 325$ along with the log law $u^+ = 1/0.41 \log(y^+) + 5.0$ (dashed line). For clarity the DNS solution at $Re_\tau = 550$ (green line), the iLES solution at $Re_\tau = 325$ (circles) and an additional log-law curve have been shifted upwards with $\Delta u^+ = 5$. Right: comparison of the root-mean-square values of the velocity fluctuations for $Re_\tau = 180$. $\circ$: Mu^2s^2T values; Solid lines: DNS from Lee and Moser[1].

$L_z^+ = 545$ in wall units) in the x , y and z directions, respectively, which is smaller than that used by Lee and Moser in [1] ($L_x = 8\pi\delta$, $L_y = 2\delta$ and $L_z = 3\pi\delta$).

Figure 5.10 (left) compares the mean velocity profile obtained with the high-order solver against the DNS results [1]. Our results are in remarkable good agreement with the DNS data, especially considering the fact that a smaller computational box has been used. It is also worth noting that the first element spans across the inner and buffer regions² (element within $y^+ \leq 8.48$) and its first collocation point is located at $y^+ = 2.34$ which is above the standard recommended values ($y^+ < 2$) [109]. Minor differences with the DNS data can be seen in the logarithmic region most likely due to the size of the computational box, as mentioned above.

A simulation with $Re_\tau = 325$ was run using the same computational box. Running a higher Reynolds number makes the box size more suitable but makes the mesh resolution even coarser. Figure 5.10 (left) also shows the results obtained along with the DNS results for $Re_\tau = 550$ (for clarity of the figure the results have been shifted in the positive direction with a $\Delta u^+ = 5$). In this case the first element, again, spans across the inner and buffer sub-layers and its first

²The buffer region spans between the viscous and the logarithmic sub-layers.

inner point of this element off the wall is at $y^+ = 4.19$, i.e., almost out of the viscous sub-layer ($y^+ \lesssim 5$). As seen, even with this coarser mesh the mean velocity profile follows the log-law and compares well with the DNS results considering these are at a higher Re_τ which shift the logarithmic region slightly [1].

Figure 5.10 (right) shows the root-mean-square of the velocity fluctuations. Again here excellent agreement between the DNS [1] and our iLES is seen despite the use of a computational box with marginal resolution. It is evident from this test case that the present numerical method is capable of delivering accurate results of turbulent flows even with marginal resolution.

6. The Validation Case

Contents

6.1. The experimental setup	117
6.2. The numerical setup	120

As mentioned in Chapter 1 one of the issues keeping HO methods from being widely adopted by the engineering community is the lack of trust towards them driven by the fact that the validation efforts made so far are against either low quality or incomplete sets of data. In this work an effort is made to validate the implemented method and to demonstrate the capabilities of HO methods in the context of LES and QDNS simulations. The validation has been chosen to be of interest for LPT design, namely, the Reynolds lapse rate phenomenon. The objective of this case is, hence, twofold: validate our numerical method and advance in the understanding of what is possible with these methods for LES/QDNS simulations.

The present chapter is organized as follows. First, a description of the experimental setup and details about the numerical cases is made. Next, the assumptions and hypothesis made in order to define the numerical setup are explained.

6.1. The experimental setup

The validation case has been chosen so that high-quality experimental data is available. The reference experimental data correspond to a generic research low-speed linear LPT cascade designed by ITP Aero and tested at the Fluid Dynamics Laboratory of the Universidad Politcnica

de Madrid (UPM). The linear cascade is made up of five blades and two tail boards with an aspect ratio of 3.5. The inlet angle is set at $\alpha_1 = 32^\circ$ and the total turning is about 90° . The aerofoil is highly loaded and the suction peak is at about 30% of the axial chord c_x .

The test facility operates at very low Mach numbers (depending on the Reynolds number the Mach number may vary between 0.02 and 0.08), so that the flow can be considered incompressible.

The experimental facility operates as an open return, impulse wind tunnel. The approaching flow has a 495x240 mm rectangular cross-section. Prior to the test section, conditioning elements are included to ensure the proper quality of the cascade approaching flow with a measured free inlet turbulence below 0.2%¹.

The rig was operated in a closed test-cell that can be isotropically seeded with oil steam, enabling laser based velocimetry diagnostics. For that purpose, the test section is built in perspex. A mechanism was included to allow unsteady studies where the linear cascade was exposed to upstream generated wakes shed from uniformly spaced bars, which were displaced by a computer controlled electric motor. The moving bars mechanism followed a design similar to other existing facilities [17]. Flow diagnostics available in the experimental facility include wall pressure characterization, inlet to outlet total pressure loss measurements, and temporally resolved anemometry through hot-wire and Laser Doppler Velocimetry (LDV).

Figure 6.1 shows an schematic of the experimental configuration. Both inlet and outlet measurement planes are indicated with green dots. At the inlet a coarse LDV traverse is made. At the outlet fine traverses with a pitot probe as well as with LDV are made across the wakes of two pitches: that of the central aerofoil and the contiguous one. Although not seen in the figure, there is a pitot probe upstream of the inlet plane. Pressure tapings are placed on both the suction and pressure side of the central aerofoil. In the figure, V_B indicates the velocity of the wake-generating bars.

¹This result is obtained by filtering out low frequencies from a hot-wire signal placed before the cascade. These low frequencies are associated to wind-tunnel cycle oscillations and it is common practice to filter them out in order to produce inlet free stream turbulence.

Figure 6.1.: Schematic of the experimental set-up.

6.1.1. Considered cases

Tests were carried at an array of Reynolds numbers and reduced frequencies. The tested Reynolds number are $Re = 5 \times 10^4$, 7.5×10^4 , 10^5 , 1.5×10^5 , 2×10^5 and 2.5×10^5 and the reduced frequencies $f_r = 0.05$, 0.5 and 1 . The Reynolds number is defined here as

$$Re = \frac{\rho S v_2}{\mu} \quad (6.1)$$

where S is the perimetral chord, and v_2 is the isentropic outlet velocity; and the reduced frequency is defined as

$$f_r = f \frac{S}{v_2} \quad (6.2)$$

where f is the bar-passing frequency.

Note that a reduced frequency of zero, $f_r = 0$, is avoided here. This is because in that state wind-tunnel instabilities develop freely making the results useless. The reduced frequency of $f_r = 0.05$ stabilizes the wind-tunnel and it is run as equivalent of the steady state. This hypothesis is valid as long as the time between bar-passing remains sufficiently large as to not affect the laminar separation processes. This is discussed in the next section 6.2.2.

Figure 6.2.: Location of the boundary layer traverse measurement stations along with the absolute value of the friction coefficient of the steady $Re = 10^5$ case.

6.1.2. Boundary layer measurement stations

Figure 6.2 shows the locations of the boundary layer traverse measurements. The friction coefficient for the steady $Re = 10^5$ case has also been put in the picture for reference. These locations help to understand the evolution of the boundary layer. For steady cases, station *a* is before separation, locations *b*, *c*, and *d* are near the reattachment point and station *e* is at the trailing edge. The traverse stations denoted with letters *a*, *b*, *c*, *d* and *e* correspond to the axial locations $x/c_x = 0.56$, $x/c_x = 0.79$, $x/c_x = 0.84$, $x/c_x = 0.89$, and $x/c_x = 0.99$, respectively.

6.2. The numerical setup

In this section the numerical setup and the hypothesis made with regards to the experimental setup are discussed. First, the meshes used for the simulations are described. After this, the hypothesis made and their validity are dealt with. The computational cost and an analysis of the convergence of the different cases are discussed last.

A selection of the experimental cases has been made for the numerical simulations. This down-

selection of cases was necessary in order to be able to run with a modest amount of computational resources. Yet these are very significant in an industrial context where designers are always hungry of such resources. The steady state, $f_r = 0.05$, had to be chosen since, despite it being extremely difficult to run (see, for instance, [45]) due to its high sensitivity to incoming perturbations, it is easier to understand and validate due to the lower number of physical phenomena involved. The unsteady cases $f_r = 0.5$ and $f_r = 1$ were also chosen since it is important, from a design point of view, to understand the effect of the incoming wakes on the cascade aerodynamics. The down-selection is, then, done on the number of Reynolds cases. The chosen Reynolds number cases were $Re = 5 \times 10^4$, 10^5 and 2×10^5 . These Reynolds numbers were chosen so that the different trends of loss against Reynolds number (in logarithmic scale) can be captured. In this sense, the loss of the $Re = 5 \times 10^4$ and 2×10^5 cases lay on the two different regions of loss vs. Re curve whereas the loss of the $Re = 10^5$ case lays in between the two regions.

6.2.1. Computational grids

Unstructured meshes were constructed using hexahedra and prisms of triangular base. The grids were generated extruding a 2D mesh of the domain a number of planes in the span-wise direction.

Three different meshes were used for the three different Reynolds number steady simulations. The main motivation behind using different meshes is efficiency. This is because turbulent scales are smaller in the $Re = 2 \times 10^5$ case than in the $Re = 5 \times 10^4$ case so that using the same mesh would imply that the latter would be much more resolved than the first. Additionally, the most unstable wavelength of the secondary instabilities increases as the Reynolds number decreases, so that lower Reynolds number simulations require a larger span than those at higher Reynolds numbers. Using the same mesh would require having the resolution of the $Re = 2 \times 10^5$ case with the span of the $Re = 5 \times 10^4$ resulting in extremely large meshes.

Figure 6.3 displays the meshes for steady and steady $Re = 10^5$ cases along with details of the mesh around the profile and detail of the trailing edge. It can be seen that the mesh features

Figure 6.3.: Linear mesh topology (internal nodes to element not shown) used for steady and unsteady $Re = 10^5$ cases with a close-up of the mesh around the profile and trailing edge detail. Black lines in the detail define the second-order mesh and the red lines connect the high-order internal nodes.

a refined mesh near the wall. Additional elements were added to the region where the bubble appears in order to capture the KH breakdown process as well as in the wake region. The domain and the wake region were extended considerably downstream of the trailing edge because the wake measurements were done at an axial station located 57% of the axial chord (c_x) downstream of the trailing edge. The mesh was further extended another 43% c_x downstream of the measurement station in order to minimize the effect on the measurements of the reflections of the outlet BC.

In order to build the meshes, a second order unstructured mesh is created first, and then the internal points of each element added. In curved boundaries, such as the aerofoil, their parametric representation is used to evaluate the actual position of the inner point in the surface and then the increments in the position with respect the nominal one, which assumes that the boundaries of the element are straight lines, are translated to the rest of the inner points using radial basis functions. An example of the resulting high-order mesh around the TE can be seen in the detail of Fig. 6.3.

A summary of the meshes used in the steady analyses can be seen in Tab. 6.1. In the table

Table 6.1.: Summary of the meshes used in the steady simulations for each Reynolds number.

$Re \times 10^{-3}$	50	100	200
Span ($\%c_x$)	40	20	15
z -elements	75	75	100
Comp. points (M)	150.5	122.5	337
Physic. points (M)	58.9	48	132
Δx^+	< 5	< 8	< 12
Δy^+	< 1	< 1	< 1
Δz^+	< 5	< 5	< 5

Table 6.2.: Summary of the meshes used in the unsteady simulation for each Reynolds number.

$Re \times 10^{-3}$	50	100	200
Span ($\%c_x$)	30	30	15
z -elements	35	50	60
Comp. points (M)	188.8	250.2	454.2
Physic. points (M)	102.1	135.2	245.9
Δx^+	< 4	< 8	< 10
Δy^+	< 1	< 1	< 1
Δz^+	< 3	< 7	< 7

the simulated span as a percentage of the axial chord, the number of high-order elements in the span direction, the number of computational points, the number of physical points and the mesh spacing in wall units are displayed for all three meshes. It is seen in the table that, for every Reynolds number, the values of grid spacing in wall units is below the recommended values by Piomelli and Chasnov in [42] for wall-resolved LES ($\Delta y^+ < 2$, $\Delta x^+ \simeq 50 - 150$ and $\Delta z^+ \simeq 15 - 40$). Larger values of the spacing in wall units were found at the fore side of the aerofoil due to the very high acceleration of the flow and a very thin boundary layer. That is, however, a purely laminar region and so no additional elements were needed.

Unlike the in steady cases, the meshes for unsteady cases need to be extended upstream in order to make room for the passing bars. An example of such meshes can also be seen in Fig. 6.3. The outlet measurement station is the same as with the steady cases and it has also been marked in the figure.

Table 6.1 is a summary of the meshes used in the unsteady cases. As can be seen, the span is the same for the $Re = 5 \times 10^4$ and $Re = 10^5$ cases. This is because in unsteady cases the natural transition mechanism is bypassed and the breakdown of the span-wise KH vortices is forced by

the perturbations introduced by the turbulent spots. It suffices, then, for the span-wise spacing to be large enough to accommodate the formation of the longitudinal streaks.

The mesh spacing in the wall normal direction is $\Delta y^+ = 1$ for all the simulated cases in the boundary layer turbulent region, whereas the stream- and span-wise spacing are $\Delta x^+ \simeq \Delta z^+ < 10$ for all the simulations which are smaller than the minimum values recommended by Piomelli and Chasnov in [42]. The resulting meshes are, then, fine enough for wall-resolved LES.

6.2.2. Steady state hypothesis

The validity of considering the $f_t = 0.05$ case as the steady case is discussed here. Figure 6.4 shows the experimental time variations of turbulent intensity at two different locations. At the inlet (before the bars), it is seen that the inlet free stream turbulence intensity is indeed very low ($<1\%$)². This very low turbulent intensity is maintained after the bars in the gap region (between bars and cascade) where a spike in turbulence intensity is seen due to the passing of the bar wake.

As it can be seen, the cascade is only under the effect of the wake for about 10% of the time. After the cascade, the turbulent intensity is slightly decreased due to the flow acceleration. The peak associated with the wake of the bar has also decreased and its width increased due to distortion within the passage. Even with this distortion, the aerofoil surface is only exposed to the turbulent wake less than the 15% of the time.

Whether the $f_r = 0.05$ case can be approximated by a steady case depends on whether the bubble is re-established fast enough with respect to the wake passing frequency. Figure 6.5 shows the energy spectra of the steady $Re = 10^5$ case at different locations within the suction side boundary layer. It can be seen that there is a clear peak in energy at reduced frequency $f_r \simeq 5$ which is about 100 times larger than the bar-passing frequency.

The steady assumption is believed to be a good approximation of the low reduced frequency of 0.05 since the roll-up frequency of the bubble is much higher than that of the incoming wakes.

²This value is indeed larger than the nominal value of 0.2%. This is because, unlike for the nominal value, here, this is a LDV signal without post-processing, i.e., here low frequencies have not been filtered out.

Figure 6.4.: Experimental turbulence intensity as a function of time measured at different axial locations.

The good results of the numerical simulations would also suggest that this is, indeed, the case.

6.2.3. Unsteady cases considerations

The experimental ratio of bar pitch to blade pitch is very close to one (1.07 for the $f_r = 0.5$) but not exactly. Simulating the exact ratio would, therefore, require a large domain in order to include several blades. Since the bar pitch is very close to that of the cascade we approximate the former to be the same as the latter, allowing the simulation of only one pitch. This approximation, however, modifies the flow parameter slightly. In our simulations, a flow parameter of $\phi = 1.17$ is achieved which is slightly higher than the experimental one ($\phi = 1.1$). The higher reduced frequency of $f_r = 1$ is achieved by doubling the number of bars within the pitch.

In the experimental setup, the incoming wakes are generated by cylindrical rods moving in the pitch-wise direction. To avoid the extra cost associated with the meshing of the rods and the sliding plane interpolation algorithm, immersed boundary conditions have been implemented in the code. This technique has been successfully employed by other authors [23] based on the method of Goldstein et al. [110].

Figure 6.5.: Non-dimensional energy spectra for the $Re = 10^5$ case within the boundary layer at different stream-wise stations. The dashed line stands for the free turbulence energy decay rate ($E \sim f^{-5/3}$).

6.2.4. Stream-tube definition

Although the aspect ratio of the linear cascade is relatively high (3.5), the secondary flows created at the end-walls induce a blockage in the passage which gives rise to a radial contraction of the streamlines. This is a frequent situation in which the mid-section of the aerofoils is not affected by the secondary flow, but the effect of the stream-tube contraction needs to be retained.

The axial velocity in a straight linear cascade with straight end-walls and incompressible flow is constant in the absence of secondary flows. Since the velocity and the angles are measured at the inlet and the exit of the cascade the contraction of the stream-tube can be derived from the measurements. The stream-tube is assumed to vary linearly between the leading edge and trailing edge of the aerofoil and constant outside of the aerofoil passage.

Considering the stream-tube to vary only in the axial direction (x -direction) the modified governing equations read [111]:

$$\frac{\partial U}{\partial t} + \nabla \cdot (\mathbf{F}_{inv} - \mathbf{F}_{visc}) = S \quad (6.3)$$

where, here, $\mathbf{F}_{inv}$ and $\mathbf{F}_{visc}$ stand for the inviscid and viscous fluxes of the Navier-Stokes equations, respectively, and S is the stream-tube correction which is of the form:

$$S = p \frac{\partial h}{\partial x} \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T - \frac{1}{h} \frac{\partial h}{\partial x} F \quad (6.4)$$

where p is the static pressure, h is the height of the stream-tube and F is the inviscid flux in the x -direction.

Figure 6.6 shows the pressure coefficient for the steady case of $Re = 10^5$ with (solid line) and without (dashed line) the stream-tube correction. It is seen that the stream-tube moves the suction peak slightly upstream and increases its intensity, i.e., the pressure at the suction peak is lower than that of the case without the stream-tube. The flow is, then, subject to a greater pressure gradient which moves the onset of the laminar separation and the associated transition-reattachment points upstream with respect to the case without stream-tube.

It is worth noting that the definition of the stream-tube has not been tuned to match the profile

Figure 6.6.: Pressure coefficient distribution for different flow conditions (steady $Re = 10^5$).

pressure distribution but it has been derived from the axial velocity measurements upstream and downstream of the cascade. This stream-tube correction has been applied to every case run.

6.2.5. Mach number effects

In order to reduce the computational effort in the compressible solver, the simulations in this work has been performed at an exit Mach number of 0.2, which is sufficiently small to consider the flow as incompressible. In order to assess the impact of the change of the Mach number on the results another simulation at $M_{exit} = 0.1$ and $Re = 10^5$ has been performed.

Figure 6.6 compares the pressure coefficient distributions for two different exit Mach numbers, namely $M_{exit} = 0.2$ (solid line) and $M_{exit} = 0.1$ (dotted line), for the $Re = 10^5$ steady case. As it can be seen, the lower Mach number case presents a slightly higher pressure coefficient along the suction side between the leading edge and the suction peak. Downstream the suction peak no major differences are seen except at the reattachment point, where a slightly earlier reattachment in the low Mach number case can be noticed. Differences between the two cases are, however, minor and the nominal exit Mach number of 0.2 is deemed acceptable, which approximately cuts the computational costs by a factor of two.

6.2.6. Convergence

The statistical convergence of the flow was investigated in order to ensure that the results presented in this thesis are meaningful. Due to their different nature, two different approaches have

Figure 6.7.: Convergence history of RMS velocity peak in the boundary layer of station e (circles) and the wake measurement position (squares) for the steady $Re = 10^5$

Figure 6.8.: Convergence history of RMS velocity of the station e boundary layer (a) and the wake measurement position (b) as a function of the number of flow-through times. The insets correspond to the zones containing the largest velocity fluctuations. ($Re = 10^5$)

been used to examine convergence for the steady and unsteady cases.

Figure 6.7 displays the convergence history of the peak value of the Root Mean Square (RMS) of the velocity at station e and in the wake measurement plane as a function of the number of flow-through for the $Re = 10^5$ steady case. For the steady cases the error, ε , is defined as:

$$\varepsilon = \frac{RMS - RMS_f}{RMS_f} \quad (6.5)$$

where RMS_f is the root mean square of the velocity in the last iteration. Since the first four flow-through were disregarded for the statistics the first point of the plot was obtained after five flow-through times, and the error at this time is about 5%, which is already low. It can be noticed how the error nicely converges, and in the last flow-through, the variation in the peak RMS at station e and the wake is less than 0.5%. The difference in RMS between running 10 or 15 flow-through times is estimated at 2%. We will see that the errors between the numerical and experimental results in terms of RMS are greater than 2% and therefore, this error cannot be attributed to a lack of convergence of the statistics.

Figure 6.8 plots the profiles of velocity RMS at the station e and the wake as a function of time. The profiles obtained for the longest times accumulate all the previously existing statistical data. It can be noticed how the RMS profiles nicely converge to a limit value being the regions with the highest fluctuations the latest to converge.

In the case of the unsteady simulations, a limited amount of bar-passing periods are run which requires checking for statistical convergence. In order to study the convergence, we define RMS_t as

$$RMS_t = \sqrt{\frac{\sum_{i=0}^{N_t} (V_i - \bar{V}^t)^2}{N_t}} \quad (6.6)$$

where N_t is the number of time-steps taken up to time t , V_i is the velocity at time-step i and $\bar{V}^t$ is the mean velocity up to time t . Taking this definition into account it is possible to define the error as

$$\text{error} = \frac{RMS_t - RMS_{t-T_w}}{RMS_{\text{final}}^{\max}} \quad (6.7)$$

Figure 6.9.: Convergence of the RMS of velocity fluctuations at different locations across the computational domain for the $Re = 10^5$ cases. Top: $f_r = 0.5$ cases; bottom: $f_r = 1$ cases. y denotes local vertical distance to the wall.

where RMS_{final}^{max} is the maximum of the RMS at the considered section where the error is being looked at. Using the maximum of the RMS for each section avoids artificially large errors in RMS at locations with very low RMS such as the very near wall region and the outer field.

This definition of error allows to check how the RMS is changing between wake-passings. For an infinite number of wake-passings the error defined in this manner should approach zero. Figure 6.9 shows, for both reduced frequencies, $f_r = 0.5$ and $f_r = 1$, the time-evolution of errors at different locations within the suction side boundary layer and at the exit measuring station. For all cases, it can be seen that the error between the last wake-passings is below 2% which is lower than the difference between numerical and experimental values, as will be seen in the chapter dedicated to results. The numerical solutions are considered to be sufficiently converged in the context of this thesis.

Table 6.3.: Summary of the computational cost and resources in steady cases.

$Re \times 10^{-3}$	No. GPUs	Initial/Total flow-through	Time-steps $\times 10^{-6}$ / Elapsed time (h)
50	30	3/10	5.7/215
100	30	4/18	6.4/188
200	40	3/7.1	3.4/174

6.2.7. Computational cost

For all steady cases, each simulation was initiated with a RANS solution and then run with the high-order solver for a number of flow-through times until an equilibrium (quasi-steady) solution is achieved. This stage lasts for about 3-4 flow-throughs (see Tab. 6.3) and the statistics are not recorded since they might be contaminated by the start-up transient. Then, the simulations are restarted and statistics compiled for about 10 flow-through times depending on the case. Table 6.3 provides the actual numbers for the results presented in this work. Spatial averaging in the span-wise direction was also carried out to speed-up convergence of the mean flow and the statistics of the turbulence. It is noted here that the amount of flow-through times needed to converge statistics differs from case to case. At higher Reynolds numbers smaller turbulent scales are present, and these evolve faster than the larger ones observed at lower Reynolds numbers (frequencies of the turbulent fields tend to be higher at higher Reynolds numbers). This means that more eddy turnover cycles per flow-through time take place at higher than at lower Reynolds numbers.

For the calculations presented here a cluster of GPUs was used. This cluster is made up of 20 nodes with 10 NVIDIA GeForce GTX1080Ti in each node. Nodes are inter-connected with a network of two-port Infiniband at 100 Gb/s per port. A summary of the computational resources used in the simulations is given in Tab. 6.3. Typically, about 30-40 GPUs per simulation have been used to keep computational resources at a modest level which is essential in an industrial environment. The total elapsed time of the simulations is less than one week, statistics and mean global data are continuously monitored. Although case dependent, the elapsed time per flow-through with these computational resources is about half a day.

As with the steady cases, each unsteady simulation was initiated with a RANS solution. The

Table 6.4.: Summary of the computational cost and resources in unsteady cases.

$Re \times 10^{-3}$	f_r	# GPUs	Bar-passing periods (total)	Iters $\times 10^{-6}$ / Comp. time (h)
50	0.5	30	13.8	3.7/191.3
	1	20	24	3.2/224.4
100	0.5	30	14	3.4/217.8
	1	30	35.8	3.1/207.1
200	0.5	40	14.7	6.7/552
	1	40	29.2	6.6/548

simulation was then run with the high-order solver for four bar-passing periods in the case of high reduced frequency ($f_r = 1$) and two bar-passing periods in the case of low reduced frequency ($f_r = 0.5$). From this point, statistics were computed for ten bar-passing periods in the case of $f_r = 0.5$ and twenty bar-passings for the $f_r = 1$ case.

Table 6.4 summarizes the computational resources used for the present simulations. The number of GPUs varies from case to case with the lowest number of resources allocated to the $Re = 5 \times 10^4$ case with $f_r = 1$. i.e., 20 GPUs running 334.3 hours. The most expensive cases were the $Re = 2 \times 10^5$ ones, as expected, with an elapsed time of 552 and 548 hours for the cases of $f_r = 0.5$ and $f_r = 1$, respectively.

7. Results

Contents

7.1. Steady results	135
7.2. Unsteady results	147

In this chapter the results from the validation case are presented. The results of the steady cases are presented first followed by the unsteady results. In each section the general flow features are described and then the results commented. Concluding remarks are made last.

7.1. Steady results

In order to be able to predict the performance of an LPT the fundamental physics involved needs to be captured accurately. As mentioned in Chapter 6, at the Reynolds number considered, the suction side boundary layer experiences a laminar separation due to the adverse pressure gradient giving rise to a shear layer close the wall. This shear layer associated with the bubble rolls up due to KH instabilities. The vortices that form from the roll-up (KH vortices) deform as they travel downstream and eventually breakdown into turbulence. This process generates noise that radiates out into the domain. This process is illustrated in figure 7.1 which shows the iso-surfaces of Q-criterion coloured by the entropy levels at the Trailing Edge (TE) region. The pressure fluctuation field is plotted in the background. It is evident from this picture that the numerical settings generated for the analysis are adequate and that the code is able to reproduce,

Figure 7.1.: Iso-surfaces of Q -criterion coloured with the entropy level at the aft of the aerofoil. Pressure fluctuation field is plotted in the rear plane (steady $Re = 5 \times 10^4$).

at least qualitatively, the separation-transition-reattachment phenomena which are taking place in the suction of the aerofoil.

7.1.1. Time-averaged results

Figure 7.2 shows the pressure coefficient for different Reynolds numbers along with the experimental results. As can be seen in the figure, this aerofoil is highly loaded with the suction peak located around $x/c_x = 0.3$. At the considered Reynolds numbers, there is a laminar separation bubble and reattachment over the suction side. The bubble region can be identified by the flattening of the pressure distribution in the diffusion zone whereas the reattachment is associated with the steep pressure jump right downstream of the bubble. It can be seen that despite the uncertainties associated with the incoming turbulence and reduced frequency the match between the experimental and numerical results is fairly good which is a clear indication that all the hypotheses which had been made are correct, at least in the first approximation.

A more precise way of determining the separation and reattachment locations is by looking at the computed wall friction coefficients displayed in Fig. 7.3. The separation point can be readily identified by seeking the point where the C_f of the suction side intersects first the x -axis. For all the simulated Reynolds numbers this happens around $x/c_x = 0.6$. The reattachment point is given by the point at which the C_f changes sign again close to the trailing edge. It is seen that

Figure 7.2.: Experimental and numerical steady pressure coefficient distributions for different Reynolds numbers. Locations of boundary layer traverse measurements are indicated with letters a to e .

Figure 7.3.: Steady friction coefficient along the aerofoil for different Reynolds numbers. The suction side is denoted with SS and the pressure side with PS.

the reattachment for the $Re = 5 \times 10^4$ occurs very close to the trailing edge, around $x/c_x = 0.95$.

As expected, the length of the separation bubble as well as the reattachment position are Reynolds number dependent. The higher the Reynolds number the shorter the bubble. It should be noted that, for $Re = 5 \times 10^4$, the reattachment point is very close to the TE. This implies firstly, that a not fully developed turbulent boundary layer extends over a small fraction of the aerofoil surface. Secondly, that the thickness of the boundary is very large (comparable to the aerofoil thickness) and the standard scaling of the losses with $Re^{-1/2}$ associated to laminar boundary layers may be lost.

The agreement between simulations and experiments in Fig. 7.2 is very good. Some differences are observed close to the leading edge (LE). For all the simulated Reynolds numbers there were minor differences around the reattachment point, as it seen in figure 7.2. In general, the numerical results show a slightly shorter and thinner bubble than that of the experiments. This variation in the reattachment point induces variations in the loading which ultimately shows up in the front part of the aerofoil. These differences are possibly due to the receptivity of the transition point to the incoming turbulence which has not been retained in the simulations. Overall, the differences with experiments are minor and the trend of the pressure distribution with the Reynolds number is well captured.

7.1.2. Boundary layer analysis

In order to determine whether the BL is well captured, numerical results are compared against experimental BL traverse data obtained using hot-wire anemometry. Figures 7.4, 7.5 and 7.6 plot the profiles of the mean absolute value of the velocity and the root-mean-square (RMS) of fluctuations of the same for the three simulated Reynolds numbers at different streamwise-locations over the suction side of the aerofoil (locations *a* to *e* specified in Chapter 6 also seen in Fig. 7.2).

Since hot-wires only measure the modulus of the velocity in the normal plane to the wire the numerical values presented here have been obtained by time-averaging the absolute value of the

Figure 7.4.: Boundary layer profiles of velocity (top) and root-mean-square of fluctuations of the absolute value of the velocity (bottom) at several stream-wise locations for the steady $Re = 5 \times 10^4$ case. $\circ$ experimental data; — numerical results.

Figure 7.5.: Boundary layer profiles of velocity (top) and root-mean-square of fluctuations of the absolute value of the velocity (bottom) at several stream-wise locations for the steady $Re = 10^5$ case. $\circ$ experimental data; — numerical results.

Figure 7.6.: Boundary layer profiles of velocity (top) and root-mean-square of fluctuations of the absolute value of the velocity (bottom) at several stream-wise locations for steady $Re = 2 \times 10^5$ case. $\circ$ experimental data; — numerical results.

$x-y$ velocity components to mimic what hot-wires measure. The fluctuations presented here are then the RMS variations of the $x-y$ velocity modulus. Note that since the measurement probe has a finite length, some spanwise averaging of the absolute velocity RMS is to be expected. This span-averaging of the RMS of the velocity depends on the wire length measured in wall units, as reported in [112]. Ng et al. [112] observed that to reproduce the fluctuations of a flat plate turbulent boundary layer experimentally, the hot-wire length, ℓ , measured in wall units, had to be $\ell^+ < 20$. In the particular case presented here, the length of the wire in wall dimensions is $\ell^+ = 17, 34$ and 63 for the $Re = 5 \times 10^4, 10^5$ and 2×10^5 cases, respectively. This means that according to [112], the hot-wire is expected to under-predict the actual RMS values of the velocity between 10% and 30%.

Results for the $Re = 5 \times 10^4$ case in Fig. 7.4 are described first. At station *a*, the BL is still very thin, laminar, and attached. Due to the adverse pressure gradient, the laminar BL undergoes separation and a rapid growth of the BL thickness is observed in station *b* through to *d*. The roll-up of the shear layer associated with the separated region increases the velocity fluctuations

which grow in magnitude as they develop downstream of the separation. This is seen in the RMS distributions of stations b to d . The maximum in velocity fluctuations occurs shortly after the reattachment point as observed by McAuliffe et al. [13]. In this case, as discussed earlier, the reattachment point occurs very near the TE of the aerofoil. It is evident that in location e in Fig. 7.4 the flow has already reattached. Additionally, the velocity fluctuations in this station are the largest of the series in the same figure indicating that this location is, indeed, very close to the reattachment point which is the location where the largest fluctuations are expected [113]. Differences between experiments and simulations are noticeable from station b . The bubble growth seems to be higher in the experiments when compared to the simulations. Station e highlights the difference in the reattachment location. While a rather developed turbulent BL has already been developed in the simulation at this location, the experiments show a profile closer to reattachment. The RMS of fluctuations of the absolute value of the velocity in the blade-to-blade plane is in good agreement with experimental data. In this case slightly lower values of the RMS of the fluctuations are observed upstream of station c and slightly higher downstream of it, the latter most likely due to experimental span-wise average of the lower turbulent structures [112]. The peaks in RMS fluctuations are closer to the wall in the simulations which is consistent with a thinner laminar separation bubble. At station b two peaks in the RMS of the absolute value of the velocity can be identified both, in the simulations and the experiments. The furthest from the wall is associated to the roll-up vortices of the shear layer whereas the inner peak is associated to the already developing near-wall turbulence. This behaviour had already been observed in the DNS simulations of flat plates subject to an adverse pressure gradient (see for instance Fig. 11 (left) in [13] for more details).

Figure 7.5 shows the evolution of the BL for the $Re = 10^5$ case at the same locations as in the previous case. Here, again, the BL at station a is attached and laminar. It is seen that separation has already occurred at station b while at station d the flow has already reattached. The velocity profiles at these stations are perfectly captured by the simulations. Station c is very close to the reattachment point and so small differences in the reattachment location can result in big differences in BL profiles. The fact that the largest velocity fluctuations are at station d

in the series in Fig. 7.5 is consistent with the observations in [13, 113] that the largest values of fluctuations are shortly after reattachment. At this point, transition has occurred and a turbulent BL starts to develop. At the last station, *e*, a fully turbulent BL has already developed which can be noticed by a fuller velocity profile typical of fully turbulent BLs. The agreement between numerical and experimental results is excellent for the mean velocity profiles. Velocity fluctuations in the inner region of the BL differ significantly between experiments and simulations, especially after location *c*. This is, again, due to experimental span-wise filtering of small turbulent structures [112].

Only one of the stations of the case of $Re = 2 \times 10^5$ falls within the laminar separation region in Fig. 7.6. At station *a* separation has not yet occurred whereas in *b* a very thin separated region can be seen. In location *c* the flow has already reattached and a fully turbulent BL develops downstream towards the TE. The largest level of velocity fluctuations are seen in station *b* indicating that this location is very close to the reattachment point. Excellent agreement in the mean velocity profile is seen in this case. Differences between simulations and experiments are also seen here for the RMS of velocity fluctuations. As in the previous case, this is due to experimental span-wise filtering of small turbulent structures [112].

It is concluded that the match between the numerical and experimental results is very good. The matching of velocity fluctuations was good in locations where the most fluctuations happens in the $x - y$ plane, e.g., KH vortex dominated regions. As turbulence develops, smaller structures appear also in the span-wise direction. Due to the finite length of the hot-wire, velocity fluctuations were somewhat filtered which resulted in a bad matching between experimental and numerical data.

7.1.3. Wake data

Wake data is measured 57% of the axial chord downstream of the TE. These data are summarized in Fig. 7.7. From top to bottom Fig. 7.7 shows the experimental data (for the central aerofoil, $\circ$, and the adjacent one, $\square$) along with numerical results for pitch-wise distributions of total-pressure loss (non-dimensionalised by peak loss at $Re = 10^5$), outlet angle (offset by

a reference angle), velocity deficit (non-dimensionalised by a reference deficit) and RMS of axial- and pitch-wise velocity fluctuations for the simulated Reynolds numbers (lower to higher from left to right).

In terms of the total-pressure defect distribution, it is seen that the width of the defect is smaller with increasing Reynolds numbers, as it was expected, since not only the bubble is smaller at higher Reynolds numbers but also the BL is thinner. The match between experiments and simulations is remarkably good, especially for the higher Reynolds number cases. The higher mismatch between experiments and simulations for $Re = 5 \times 10^4$ might be due to a higher uncertainty of the measurements as evidenced by the larger differences between the results for two consecutive channels (circles and squares in the figure). It is to be noted here that measurements at this Reynolds number are very challenging due firstly, to the very low dynamic pressures, and secondly, the larger uncertainty in the periodicity created by the instabilities associated to the presence of big oscillating bubbles in the suction side of the aerofoil which can couple among them. For more details about how LDV measurements are conducted the reader is referred to [114].

Differences are observed for the angle distributions across the simulated Reynolds numbers. The most accurate results are obtained for the higher Reynolds numbers. The mean angle across the pitch is, however, in very good agreement with the experimental results.

Velocity distributions are in very good agreement with the experiments across the range of simulated Reynolds numbers. As with the total pressure deficit, it is also seen here that the width of the velocity defect diminishes with increasing Reynolds number.

The RMS of the axial- and pitch-wise velocity fluctuations present some differences between experiments and numerical results. In this case the offset that exists away from the centre of the wake is due to the fluctuations introduced by the upstream wakes and free inlet turbulence. The agreement is remarkably good for the higher Reynolds number cases. Again, differences are observed for the $Re = 5 \times 10^4$ case although velocities are lower in the experimental case so that larger errors are to be expected in this case.

Figure 7.7.: Summary of steady wake traverses data at measuring station ($57\%c_x$ downstream of the TE) for all three Reynolds numbers ($Re = 5 \times 10^4$, 10^5 and 2×10^5 , from left to right). From top to bottom: Total pressure loss, outlet angle, velocity and RMS of velocity fluctuations in axial- and pitch-directions. Solid line: numerical results; circles: experimental data for the central aerofoil of the cascade; squares: experimental results for the aerofoil above the central one.

Figure 7.8.: Steady exit angle as a function of the Reynolds number. Experimental results for the instrumented airfoil (black line) and the contiguous one (red line). . . . : dotted line notates the bracket of the measurement errors.

7.1.4. Cascade Exit Angle and Losses

The characterization of the total pressure loss and the turning of the flow is of utmost importance for aerodynamic designers. Figure 7.8 shows the flow exit angle, offset by a reference angle, at different Reynolds numbers for both the experiments and simulations. It is seen that the turning of the cascade increases (more negative values of the outlet angle, α_2) with decreasing Reynolds numbers. At low Reynolds numbers, the laminar separation bubble is larger giving rise to an increase in the blade loading. Another way of looking at this effect is by noting that it is equivalent to an increase of the effective curvature of the suction side as a result of the laminar separation bubble which promotes an increase in the flow turning. This increase is accurately predicted by the simulations as can be seen in Fig. 7.8. The numerical results are well within the uncertainties of the experiments which have been included in the plot as dotted lines. The estimated error in the measurement of the exit angle associated with the measurement chain is about $\pm 0.2^\circ$. The largest difference between the simulations and the experiments occurs for the $Re = 2 \times 10^5$ case.

Figure 7.8 also includes the exit angle of the blade adjacent to the instrumented one. This is another measure of the experimental uncertainty which in this case is higher for the higher Reynolds number. The exit angle at this Reynolds number could be considered to be within tolerances too. The errors associated with the lack of periodicity of the cascade should be added to those of the measurement chain.

Figure 7.9.: Total-pressure loss coefficient scaled with the coefficient at $Re = 10^5$ ($\omega^* = \omega/\omega_{Re=10^5}$) as a function of the Reynolds number.

Figure 7.9 represents the total pressure loss coefficient defined as:

$$\omega = \frac{P_{t1} - P_{tmo}}{P_{t1} - p_2}$$

where P_{t1} , P_{tmo} and p_2 denote mean total pressure at inlet; mixed-out total pressure at the exit, and the exit static pressure, respectively, for both, the experiments and simulations.

It can be noticed that there is a change in the losses trend with the Reynolds number between $Re = 5 \times 10^4$ and $Re = 10^5$. To better appreciate this change in the trend, the result of a simulation at $Re = 2.5 \times 10^4$ has been added as well as the curves $\omega \propto Re^{-1}$ and $\omega \propto Re^{-1/2}$. The existence of two different regimes in the separation bubble of a flat plate had already been observed. The simulations agree very well with the experimental results. The somewhat larger difference between experiment and simulation at $Re = 5 \times 10^4$ was expected since when compared against the experiment, a larger portion of the suction side was under attached flow conditions, as a result of the slightly shorter predicted bubble. The error in the loss coefficient associated with the measurement chain is about $\pm 2.5\%$ [114]. The additional error associated with the lack of periodicity between the different passages is estimated in about $\pm 2.5\%$. However, this error is very difficult to estimate, especially at the lowest Reynolds numbers where a slight change in the operating point of the cascade can give rise to large passage-to-passage variations. The symbols corresponding to the experiments in Fig. 7.9 are plotted with brackets but are hardly seen since these are much smaller than the variations of the losses with the Reynolds number.

7.1.4.1. Physical Interpretation

As it has been mentioned before, total pressure losses present a change in their trend with the Reynolds number at low Reynolds numbers which is associated with a change in the relative weight of the different loss-generating mechanisms. At the highest Reynolds numbers ($Re = 2 \times 10^5$) the BL is very thin, and the laminar separation bubble is short and thin as well. In these cases, the boundary layer approximation is fulfilled over the entire aerofoil surface so that, the most relevant loss-generating mechanism is laminar viscous friction and the losses proportional to $\omega \propto Re^{-1/2}$. However, at very low Reynolds numbers ($Re = 0.5 \times 10^5$), the bubble thickness becomes comparable to the aerofoil thickness, and the thin layer approximation of the Navier-Stokes equations, which is ultimately responsible for the scaling of the losses with $Re^{-1/2}$, is not valid any more. The flow within and about the bubble is believed to dominate loss production. The flow pattern is similar to that found about bluff bodies at low Reynolds numbers, which exhibit a large and closed re-circulation region. At these very low Reynolds numbers, the scaling with the losses is $\omega \propto Re^{-1}$, likewise that seen for bluff bodies at low Reynolds numbers. In this regime, predicting losses and their physical mechanisms is challenging since it requires to predict the length and thickness of the separation bubble accurately [14].

7.2. Unsteady results

Unlike in steady flow conditions, when the incoming wakes generated by the upstream bars enter into the aerofoil's cascade, these are stretched and deformed as they pass through the passage. The turbulence associated with the incoming wake can penetrate the boundary layer in areas of high boundary layer receptivity, i.e., the leading edge [115]. This generates longitudinal streaks on the suction side which travel downstream and are ultimately responsible for the appearance of turbulent spots [13, 115] which promote an earlier breakdown of the KH vortices. Figure 7.10 (top) shows iso-surfaces of Q-criterion ($Q = 500,000$) coloured by the Mach number (the background depicts the instantaneous entropy field). It can be seen that as the wake is stretched about the leading edge of the profile, turbulent vorticity aligns along the

Figure 7.10.: Top: iso-surfaces of Q-criterion coloured by Mach number with entropy field in the background. Bottom: velocity fluctuation (the darker the colour the more negative the fluctuation) with entropy field in the background. Case: $Re = 5 \times 10^4$, $f_r = 1$. Images are deformed for confidentiality reasons.

Figure 7.11.: Time-averaged profile pressure coefficient, c_p , for all the tested cases. $\circ$: Steady experimental data; $---$: steady numerical data; $\square$: unsteady experimental data; $-$: unsteady numerical predictions. The arrows mark the locations where the boundary layer profiles have been measured.

flow direction. Figure 7.10 (bottom) depicts the field of velocity fluctuations near the suction side, which clearly shows the presence of the aforementioned longitudinal streaks that lead to an earlier transition to turbulence further downstream. The physical phenomena involved in the flow are, hence, well captured by our numerical setup.

7.2.1. Time-averaged pressure coefficient

The pressure coefficients are compared against experimental data in Fig. 7.11. The pressure coefficients of the steady cases have also been included for both experimental and numerical data to highlight the differences between both regimes. In the steady case, the presence of the separation bubble is clearly seen by a flattening of the suction-side c_p curve followed by a sharp pressure recovery at the reattachment. It can be observed that the lower the Reynolds number the longer the steady separation bubble. Moreover, the reattachment point is very close to the trailing edge at the lowest Reynolds number of $Re = 5 \times 10^4$, whereas for the highest Reynolds number case, of $Re = 2 \times 10^5$, the reattachment point is around $x/c_x = 0.8$.

As explained in the previous section, the presence of the wakes promotes the appearance of

turbulent spots and earlier breakdown of the KH vortices thus shortening the bubble. In addition to these effects on the boundary layer, which can also be observed in a steady case with increased inlet turbulence, there is also the negative jet impinging on the suction side [4] which accelerates/decelerates the local free-stream velocity as the wake migrates across the passage. This acceleration/deceleration of the local free-stream velocity changes the separation and reattachment locations so that the separation bubble not only is shorter with respect to the steady case but it also changes its location during the wake passing.

For the Reynolds number of $Re = 5 \times 10^4$ and the reduced frequency $f_r = 0.5$ case the flattening of the suction side c_p can still be seen indicating the presence of a rather long bubble for a large portion of the bar-passing period. At the reduced frequency of $f_r = 1$, the flattening of the suction side c_p is barely visible indicating that the mean separation bubble is shorter than in the case of $f_r = 0.5$. Additionally, the pressure recovery is less pronounced, indicating that the reattachment location varies with the bar phase [5].

A similar behaviour is observed for $Re = 10^5$. Again, a slight flattening of the c_p distribution can be seen for the lower reduced frequency, $f_r = 0.5$, followed by a gradual pressure recovery which indicates that the reattachment point varies periodically with the bar passing. For the higher reduced frequency of $f_r = 1$ the rate of pressure recovery is almost constant along the diffusive part of the suction surface indicating that the size of the bubble has been further decreased.

For the highest Reynolds number, $Re = 2 \times 10^5$, the time-averaged separation bubble disappears for both reduced frequencies resulting in an almost-constant pressure recovery rate in the rear part of the aerofoil.

For all cases the numerical results agree very well with the experimental measurements. Some minor differences appear near the suction peak, due to small differences between the measured and predicted angle of attack.

Figure 7.12: Phase-averaged skin friction coefficient. Top row: steady case. Middle row: $f_r = 0.5$. Bottom row: $f_r = 1$. White continuous lines enclose areas of negative skin friction. The dashed lines indicate the location of the boundary layer transition measurement stations.

7.2.2. Phase-averaged skin friction coefficient

The skin friction coefficient provides further insight to better understand the dynamic behaviour of the separation bubble and the boundary layer. Figure 7.12 shows the phase-averaged skin friction coefficient contours as a function of the suction side longitudinal length, s , and time, t . In this figure the friction coefficient is defined as $c_f = \tau_w / (1/2 \rho_{is} V_{is}^2)$ where ρ_{is} and V_{is} are the local isentropic density and velocity, respectively; and $\tau_w = \mu(\partial u / \partial y)_{y=0}$ is the tangential stress at the wall. For clarity, three periods are plotted in the vertical axis. The wake-passing period, T_w , has been used to non-dimensionalise time. The friction contours for the steady cases has also been included for reference.

The figure also shows the location of the experimental boundary layer measurement stations a , b , c , d and e . These are marked with vertical dashed lines. The separated regions are enclosed by a white solid line. It can be seen that location a is attached for each of the simulated cases. The general trends inferred in the previous section in relation to the bubble behaviour are confirmed. It is seen that the length of the bubble decreases as the Reynolds number and the reduced frequency increase, as expected, and that its behaviour is highly unsteady. At $Re = 5 \times 10^4$ and $f_r = 0.5$ the bubble is large and there is a reversed flow in each of the computed phases. However, it is seen that the bubble almost disappears at time $t/T_w = 0.25$. For this case, the passing wake induces an instability of the separated region which creates coherent vortices synchronous with the wake similar to those seen by Stieger and Hodson [116] which generate marked traces downstream of reattachment as seen in [117]. These vortices can be seen in Fig. 7.13, which shows the phase-averaged axial velocity perturbation ($v_x^{\text{phase}} - v_x^{\text{mean}}$) over the aft suction side at time $t/T_w = 0.2$. Pockets of positive and negative perturbation velocity can be seen which are typically observed in vortical structures (depicted with a dashed arrow). The vortices are formed before the centre of the wake's negative jet is convected to the separation region. The formation of the vortices may be related to the interaction of the area of high turbulent kinetic energy (greyed area in Fig. 7.13) with the boundary layer. As explained in [24] this region travels faster than the centre of the negative jet. At the higher reduced frequency of $f_r = 1$ these vortices are not seen indicating that either they are not synchronous or they do not

Figure 7.13.: Iso-contours of phase-averaged axial velocity perturbation ($v_x^{\text{phase}} - v_x^{\text{mean}}$) for phase $t/T_w = 0.2$ over the aft of the suction side. The centre of the wake is marked with a dashed line. Dotted lines denote rotation of phase-coherent vortices. A patch of wake high turbulence is marked with a greyed area.

appear.

Figure 7.12 shows regions of low skin friction followed by patches of higher skin friction in the transitional area which are in line with the so-called becalmed regions explained by Schulte and Hodson in [17]. The duration of the becalmed regions is nearly constant in physical time for a constant Reynolds number since it is associated with the recovery time of the bubble but its relative contribution to the mean flow is larger for the higher reduced frequencies. It is also seen here that there are wake-passing phases in which no separated region exists at all. This is true in a greater or lesser degree for all the Reynolds numbers and reduced frequencies presented in this work. This phenomenon can be appreciated in Fig. 7.12 where the regions of negative skin friction associated with the bubble have been separated of the rest in the t - s plot by a white line.

For $Re = 10^5$ the behaviour of the boundary layer is very similar to that seen in the $Re = 5 \times 10^4$ case. As in the previous case, a coherent vortical structure synchronous with the passing wake seems to exist downstream of the separated region for the lower reduced frequency. In comparison with the $Re = 5 \times 10^4$ case, the friction level downstream the bubble is higher here indicating a further developed turbulent boundary layer.

In the $Re = 2 \times 10^5$ case it is seen that the separated region has further decreased for both reduced frequencies. It is also seen that the friction coefficient after reattachment is the highest among the results, indicating a more developed and thinner turbulent boundary layer, as it was expected.

Figure 7.14.: Time-average of two-dimensional velocity modulus (lower horizontal axis) and RMS (upper horizontal axis) of fluctuations of the two-dimensional velocity modulus in the boundary layer for the $Re = 5 \times 10^4$ cases. Top: $f_r = 0.5$; Bottom: $f_r = 1$. $\circ$ experimental velocity; $\triangle$ experimental RMS of velocity fluctuations; — numerical velocity; - - numerical RMS of velocity fluctuations.

7.2.3. Velocity profiles

As with the steady results, the time-averaged state of the boundary layer over the suction side of the profile is studied at locations *a*, *b*, *c*, *d* and *e*. These locations can be seen in Figs. 7.11 and 7.12 as well as in Chapter 6.

Figure 7.14 shows the boundary layer profiles at the five measurement stations for both the numerical results and experimental data. The top row of plots shows the boundary layer state for the $f_r = 0.5$ case, whereas the bottom row shows the boundary layer for the $f_r = 1$ case. Location *a* is attached for both reduced frequencies, as it can be seen by looking at the mean velocity profile. There exists a non-negligible value of velocity fluctuations due to the incoming wakes. These fluctuations are intensified within the boundary layer indicating activity due to the longitudinal streaks. The simulations seem to be under-predicting the level of fluctuations at this location slightly. The agreement is, nevertheless, excellent.

At location *b* a time-averaged separated region can be inferred from a kink in the absolute velocity profile, seen near $y/S = 0.8\%$ for both reduced frequencies. This separation region

Figure 7.15.: Time-average of two-dimensional velocity modulus (lower horizontal axis) and RMS (upper horizontal axis) of fluctuations of the two-dimensional velocity modulus in the boundary layer for the $Re = 10^5$ cases. Top: $f_r = 0.5$; Bottom: $f_r = 1$. $\circ$ experimental velocity; $\triangle$ experimental RMS of velocity fluctuations; — numerical velocity; — — numerical RMS of velocity fluctuations.

Figure 7.16.: Time-average of two-dimensional velocity modulus (lower horizontal axis) and RMS (upper horizontal axis) of fluctuations of the two-dimensional velocity modulus in the boundary layer for the $Re = 2 \times 10^5$ cases. Top: $f_r = 0.5$; Bottom: $f_r = 1$. $\circ$ experimental velocity; $\triangle$ experimental RMS of velocity fluctuations; — numerical velocity; — — numerical RMS of velocity fluctuations.

is also visible, though reduced, at location c (especially for the lowest reduced frequency). In comparison to location b , however, the RMS of the velocity fluctuations is increased within the boundary layer which is an indication of a more turbulent behaviour. Indeed, this location is very close to the reattachment region for both reduced frequencies, where high turbulent fluctuations occur. Since points b and c are in regions of intermittent reattachment and RMS is high at reattachment, small differences in the separation-reattachment cycle can lead to different RMS profiles, which explains the differences between the experimental data and the numerical predictions seen in the RMS profiles near the wall at these locations.

At location c the flow is either separated or reattached, for both reduced frequencies, depending on the passing-wake phase. For the reduced frequency of $f_r = 0.5$ location d is marginally separated for some phases although, during most of the period the flow is attached. For the $f_r = 1$ case, location d is attached during the whole period. At location e the boundary layer exhibits a turbulent-like velocity profile. For all locations and reduced frequencies, the numerical results show an excellent agreement with the experimental data.

Figure 7.15 shows the boundary layer profiles for the $Re = 10^5$ case. It is clear from the plots that the separation bubble in this case is considerably smaller than that of the $Re = 5 \times 10^4$ case for both reduced frequencies. This can also be inferred from Fig. 7.12 where only the station b line cuts through the separation region for both reduced frequencies. Unlike the $Re = 5 \times 10^4$ case, now the level of fluctuation is high from location b to location e . This is not surprising since at location b the boundary layer is in a separation-reattachment cycle and downstream this location a turbulent boundary layer develops. The agreement of the numerical simulations and the experimental data is excellent for all the stations and reduced frequencies, especially so for the time-averaged absolute velocity profile. The RMS of the velocity fluctuations also agrees well, especially at the locations a and b . As the boundary layer develops the numerical simulations seem to over-predict the RMS of the velocity fluctuations. This might be because smaller structures appear which might be filtered in the span direction by the wire due to its finite length, as explained above [112].

Figure 7.16 shows the boundary layer profiles for $Re = 2 \times 10^5$. No mean separation is observed

Figure 7.17.: Distribution of time-averaged total pressure defect distribution at the exit measurement station. Top: $f_r = 0.5$; Bottom: $f_r = 1$. $\square, \triangle$: experimental measurements for central and contiguous pitches; Solid line: numerical data.

for any of the reduced frequencies. The mean velocity profiles agree very well with the experimental data at all the measurement stations. The RMS of the absolute velocity fluctuations for these cases also show differences near the wall, where the fluctuations are largest. This is, as in the previous case, likely to be due to the filtering of the hot-wire of the shortest scales and the highest frequencies. This effect is amplified here since for this Re the turbulent boundary layer contains smaller scales than the previous ones [112].

7.2.4. Wake traverse data

Figure 7.17 shows the numerical and experimental pitch-wise total pressure loss distribution for the different Reynolds numbers. The top row shows the $f_r = 0.5$ cases whereas in the bottom the $f_r = 1$ cases are displayed. Two pitches are measured at the exit measurement station so that two sets of experimental data are presented. This serves to illustrate the level of periodicity of the experimental cascade for each case.

Unlike in steady cases, here some loss production is expected to take place within the passage due to the stretching and mixing of the incoming wake [25]. The maximum of these passage losses across the pitch depends on the flow parameter, ϕ , and the reduced frequency. Additionally, in the present case, we find pitch-wise locations with negative total pressure loss. Several

Figure 7.18.: Distribution of time-averaged velocity distribution at the exit measurement station. Top: $f_r = 0.5$; Bottom: $f_r = 1$. $\square, \triangle$: experimental measurements for central and contiguous pitches; Solid line: numerical data.

Figure 7.19.: Distribution of time-averaged angle distribution at the exit measurement station. Top: $f_r = 0.5$; Bottom: $f_r = 1$. $\square, \triangle$: experimental measurements for the central and contiguous pitches; Solid line: numerical data.

Figure 7.20.: Distribution of time-averaged $u'v'_{\text{rms}}$ distribution at the exit measurement station. Top: $f_r = 0.5$; Bottom: $f_r = 1$. $\square, \triangle$: experimental measurements for central and contiguous pitches; Solid line: numerical data.

reasons make that possible, among them the unsteady work [118, 119].

All plots in the figure have similar peak total pressure losses. For a given reduced frequency, the largest difference among the cases is the wake width, that decreases with increasing Reynolds number. Increasing the reduced frequency widens the base of the aerofoil wake, due to the increase in of the passage losses[23].

For all simulated Reynolds numbers and reduced frequencies, the agreement between numerical and experimental data is very good. The largest differences are encountered in the outer region of the wake profile. It is believed that this may be due to the slight difference in the flow parameter between experiments and simulations. Nevertheless, the agreement is deemed excellent.

Figure 7.18 shows the time-averaged pitch-wise distribution of the velocity defect. Analogous to the total pressure loss distribution, the velocity defect width depends on the Reynolds number and reduced frequency and follows the same trends as the total pressure loss distributions. The numerical and experimental data are in good agreement. Slight differences are also observed in the outer region of the wake which are associated with a small difference in the flow parameter. It is noted here that the total pressure and velocity measurements were made with different measurement techniques and are, therefore, independent from one another.

Figure 7.19 displays the time-averaged distributions of exit angle for all cases, that present very little variations from case to case. This is mostly because the presence of wakes significantly reduces the separation bubble and the flow remains attached during most of the period. The largest differences between the numerical and experimental data occur for the lowest Reynolds number case, $Re = 5 \times 10^4$. However, it must be noted that for this Reynolds number, the periodicity of the cascade is worse than for the others. This can be noticed in the difference in the measured exit angle between two adjacent pitches. Therefore, it is likely that this difference is due to the different set-up between the simulations and the experiments. In general, it seems that the periodicity of the experimental setup worsens slightly for larger reduced frequencies. The numerical predictions are, nevertheless, in very good agreement with the experimental data.

Figure 7.20 shows the time-averaged pitch distribution of RMS of the Reynolds stress $u'v'$ for all Reynolds number and reduced frequencies. This variable gives an idea of the turbulent mixing in the wake. Some offset between numerical and experimental data is observed which might be due to small spurious reflections of the outlet boundary conditions. Despite the observed differences between numerical results and experimental data it is seen that the correct distribution is computed.

7.2.5. Cascade exit angle and losses

Figure 7.21 shows the time and pitch-wise averaged exit angle variation with the Reynolds number and reduced frequency. As mentioned earlier, the low sensitivity of that variable is because the separation bubble is significantly reduced in unsteady cases, making the flow in the suction side of the aerofoil behave as an attached flow for all cases. Only for the lowest Reynolds number the angle changes slightly, especially for the low reduced frequency case, which is the one with the largest separation bubble. The numerical results agree very well with the experimental data, with small differences well within the measurement uncertainty, which is estimated to be about $\pm 0.2^\circ$.

Figure 7.22 shows the time and pitch-wise averaged total pressure loss variation with Reynolds number and reduced frequency in logarithmic scale. The averaged total pressure loss, ω , is

Figure 7.21.: Time and pitch-wise averaged exit angle variation with Reynolds number and reduced frequency.

defined as

$$\omega = \frac{P_{t1}^{\text{mo}} - P_{t2}^{\text{mo}}}{P_{t1}^{\text{mo}} - p_2}$$

where P_{t1}^{mo} is the mixed-out time-averaged total pressure between the wake-generating bars and the cascade, P_{t2}^{mo} is the mixed-out time-averaged total pressure at the exit section and p_2 is the time-averaged exit static pressure. The experimental P_{t1}^{mo} is not measured; it is calculated by means of a bar model that takes into account the drag coefficient as a function of the relative velocity, bar Reynolds number and bar diameter and spacing. The total pressure loss coefficients shown in the figure are scaled with the total pressure loss for the $Re = 10^5$ and $f_r = 0.5$ case.

For the cases presented here, the higher the reduced frequency, the larger the losses, which is in agreement with other experimental results [17]. As it is explained in [17] for a sufficiently high Reynolds number the total pressure loss increases with reduced frequency. Profile losses increase due to a larger fraction of the turbulent boundary layer caused by the earlier separated boundary layer transition, whereas passage losses increase with an increasing number of wakes within the wake (as long as wakes remain discrete [23]). At the lower Reynolds number of $Re = 5 \times 10^4$, the incoming wakes reduce losses with respect to the steady state. This is due to the reduction of the laminar separation bubble size which reduces momentum loss.

For the reduced frequency of $f_r = 1$ a change of trend is clearly seen at $Re = 10^5$ in the experiments which also appears in the steady cases [?]. The reason of this change in slope is yet not clear though it is believed to be in relation to changes in the ratio of turbulent to laminar bound-

Figure 7.22.: Time and pitch-wise averaged total pressure loss variation with the Reynolds number and reduced frequency. The experimental steady loss values have been added for reference. Note the logarithmic scale of the ordinate.

ary layer extension over the suction side. The change in slope can also be seen for $f_r = 0.5$ albeit less pronounced. Very good agreement is seen between numerical and experimental data, especially for the lower Reynolds numbers.

8. Conclusions and Future Work

Contents

8.1. Summary	163
------------------------	-----

8.2. Future work	166
----------------------------	-----

In this chapter a summary of the work carried out for this thesis is summarized. Conclusions about the results obtained are also drawn and some possibilities for future work are commented on.

8.1. Summary

In Chapter 2 the Flux Reconstruction method was introduced for advection and advection-diffusion equations. The stability analysis of FR schemes was analysed. Emphasis was given on the required scaling for stable simulations so that the largest possible time-step can be given. This is essential for efficient simulations.

Viscous effects were considered in the analysis of the wave equation in order to understand their effect on the eigenvalues. It was seen that the effect is dependent on the cell Reynolds number, $Re_{\Delta x}$, and for values of $Re_{\Delta x} < 20$ the eigenvalues become unstable if the appropriate scaling is not applied.

The 1D analysis was extended to the Euler equations. The influence of two different Riemann solvers (namely the Roe and LF solvers) on the numerical eigenvalues was analysed. Whereas the LF solver had more dissipation potential, this had no effect on stability.

Both triangular and quadrangular elements were considered in the stability analysis. It was seen that triangular elements have slightly higher limiting CFL number condition whereas rectangular elements behave like 1D elements.

The scaling of the CFL that it is here proposed takes all the above into consideration and so the maximum time-step can be computed as

$$\Delta t = \text{CFL}_{1D} \frac{\Delta y}{c(1+M)} \left(1 + \frac{1}{\text{AR}}\right)^{-1} \left(1 + \frac{2\sigma_{\max}}{Re_{\Delta x}^2}\right)^{-1}$$

Due to the small time-steps resulting from stability conditions and as a consequence of running single precision arithmetic it is necessary to deal with round-off errors. In order to do so the Kahan summation algorithm has been implemented in the RK step operations.

Chapter 3 presented the details of the implementation of the FR method for the solution of the Navier-Stokes equations for its execution on a cluster of utility GPU. As it was explained, data is arranged so that it can be accessed in coalesced manner which provides the highest bandwidth. Further, these kernels make use of thread group/block shared memory which is used to store cell data for high-speed access.

In order to take advantage of parallel computing, communication between processes were carefully designed. Strategies for a more efficient data transfer from GPU to GPU are required due to the lower bandwidth. One such strategy is communication pipe-lining which increases effective bandwidth by overlapping the three steps of the communication process. Additionally, overlapping communication and computation allows to minimize the overhead of parallel communications.

In chapter 4 the implementation of the boundary conditions was detailed. The basic methodology for the implementation of boundary conditions within the FR framework was described first. After this the wall BC was described. It was noted that, in order for the wall nodes to comply with the zero-velocity condition at each time-step, the variables at these nodes are corrected directly in the cell, which goes against the FR philosophy.

It was noted the difficulty of imposing inlet/outlet BC. In the present work the NSCBC were

implemented. The basic formulation of these BC with their particularities were described. Emphasis was made in the necessary correction so that mean desired values are imposed. Imposing mean values produces planar reflections though these are of much lower magnitude than the perturbation that caused it.

In order to check the correct implementation of the method and its capabilities several test cases were run and explained in Chapter 5. These test cases tackled different aspects of numerical simulations. The 2D Euler vortex case made it clear that the code is able to transport vortices with no dissipation which is crucial when dealing with turbulent flows. The Blasius boundary layer case showed that viscous terms were implemented correctly and that laminar boundary layers can be accurately captured with as little as four fourth-order elements across the boundary layer. The 2D circular cylinder simulation showed that the correct unsteady behaviour can be simulated with the present implementation. For this case the Strouhal number of the wake was measured and compared against experimental values. The last test case was that of the periodic fully turbulent channel. This case is the only of the test cases that involves turbulence. Mean velocity profiles as well as root mean square values of the velocity components were compared against DNS data with very good agreement.

In Chapter 6, the results of numerical simulations at various Reynolds numbers and reduced frequencies of a low-speed LPT linear cascade were compared against a high-quality set of experimental data.

A summary of the results is presented next:

1. The combined effect of Reynolds number and reduced frequency over time-averaged pressure coefficient and boundary layers is perfectly captured by the numerical setup.
2. The numerical phase-averaged skin friction coefficient for each case was used to explain the behaviour of both time-averaged pressure coefficient and boundary layer velocity and RMS profiles.
3. The effect of the Reynolds number and reduced frequency over the time-averaged total pressure loss, velocity profile, flow angle and shear Reynolds stress at the exit measurement station was very well captured by the present numerical simulations. Mixed-out exit

angle and total pressure loss agree very well with experimental measurements.

The present results confirm the suitability of the proposed numerical method to affordably simulate the steady and unsteady flow around LPT linear cascades, with a degree of matching between measurements and numerical predictions that is well within the experimental measurements accuracy. The present work expands on the understanding of what is possible with compact high-order CFD methods.

This is the first time that a high-fidelity simulation of an LPT low-speed linear cascade with moving bars has been compared against such a complete set of experimental data. Other studies consist of either parametric studies without in-depth experimental data or comparisons against incomplete sets of data.

8.2. Future work

So far the current implementation is only able to cope with steady and unsteady flows with the unsteady forcing generated by immersed bars. Though this approach already provide useful information about the unsteady aerodynamic behaviour it is limited to simulations with pitch ratios of $N : n$ with N the number of immersed bars and n the number of profiles simulated. Because of the additional cost of simulating multiple passages (which can make the whole simulation unfeasible depending on the blade count) n is limited to 1 so, effectively, the possible pitch ratios reduces to configurations $N : 1$. This is generally not representative of Low Pressure Turbine configurations.

Different techniques can be applied so that one ends up simulating one single passage such as phase-lag or the time-inclined method. Unlike with other methods, the time-inclined methodology does not assume a specific frequency. This is of great advantage since no assumptions can be made on frequencies in turbulent simulations.

In the time-inclined method due to Giles [120], the equations are transformed in time so that, in the transformed domain, the solution is periodic in space and time (corochronic periodicity).

In [120], Giles elegantly showed that the relationship between transformed and physical variables is exact when the Euler equations are transformed. The relationship is no longer exact when the Navier-Stokes equations are considered. This can be seen when considering the following scalar advection-diffusion equation

$$\frac{\partial u}{\partial t} + \frac{\partial cu}{\partial x} + \frac{\partial cu}{\partial y} = \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (8.1)$$

which transforms to

$$(1 - \lambda c) \frac{\partial u}{\partial t'} + \frac{\partial cu}{\partial x'} + \frac{\partial cu}{\partial y'} = \nu \left(\frac{\partial^2 u}{\partial x'^2} + \frac{\partial^2 u}{\partial y'^2} \right) - 2\lambda \nu \frac{\partial^2 u}{\partial y' \partial t'} + \lambda^2 \nu \frac{\partial^2 u}{\partial t'^2} \quad (8.2)$$

where λ is the time-inclined transformation parameter. The last two terms in Eq. 8.2 are responsible for this transformation to not be exact/analytical. As it can be seen these terms tend to zero in steady simulations which is why they can be neglected when performing RANS simulations. However, in high-fidelity simulations such as LES these terms might have an impact if ignored. Studying what the impact of ignoring these terms on the solution might be is of essence for these type of simulation.

One possible line of future work from this thesis is to investigate the impact of viscous terms in the time-transformed Navier-Stokes equations. This could enable the simulation of arbitrary blade counts with the simulation of a single passage. This will increase greatly the possible configurations that can be analysed efficiently.

A. Extension of the FR method to multiple dimensions

A.1. Extension of the method by Huynh to 2D (QUADs)

The method by Huynh explained in Chapter 2 can be readily applied to Quadrangular (QUAD) elements by performing 1D operations in each of the independent directions. For completeness, the procedure for doing so is explained in what follows.

As in the 1D case a conservation equation is considered now in two dimensions:

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} + \frac{\partial g(u)}{\partial y} = 0 \quad (\text{A.1})$$

Analogous to the 1D case, in 2D the generic QUAD element is transformed into a standard element with independent coordinates $\xi \in [-1, 1]$ and $\eta \in [-1, 1]$. The transformed equation becomes:

$$J \frac{\partial u}{\partial t} + \frac{\partial \tilde{f}(u)}{\partial \xi} + \frac{\partial \tilde{g}(u)}{\partial \eta} = 0 \quad (\text{A.2})$$

where

$$\begin{aligned} \tilde{f} &= J(\xi_x f + \xi_y g) \\ \tilde{g} &= J(\eta_x f + \eta_y g) \end{aligned} \quad (\text{A.3})$$

with J being the determinant of the Jacobian of the coordinates transformation and ξ_x , ξ_y , η_x and η_y the components of the inverse of the Jacobian. Taking as a reference the 1D flux approx-

imation of the solution the corrected fluxes can be expressed as:

$$\begin{aligned}\tilde{f}(\xi, \eta) = & \sum_{m=0}^p \sum_{n=0}^p \tilde{f}_{mn} \ell_m(\xi) \ell_n(\eta) + \\ & [\tilde{f}_L^I - \tilde{f}(-1, \eta)] g_L(\xi) + [\tilde{f}_R^I - \tilde{f}(1, \eta)] g_R(\xi)\end{aligned}\quad (\text{A.4})$$

$$\begin{aligned}\tilde{g}(\xi, \eta) = & \sum_{m=0}^p \sum_{n=0}^p \tilde{g}_{mn} \ell_m(\xi) \ell_n(\eta) + \\ & [\tilde{g}_D^I - \tilde{g}(\xi, -1)] g_L(\eta) + [\tilde{g}_U^I - \tilde{g}(\xi, 1)] g_R(\eta)\end{aligned}\quad (\text{A.5})$$

where the subscripts L , R , D and U in the interface fluxes denote Left, Right, Down and Up, respectively, and the superscript I stands for value at the interface. The resulting discretised equation reads:

$$\begin{aligned}J \frac{\partial u}{\partial t} + & \sum_{m=0}^p \sum_{n=0}^p \tilde{f}_{mn}^D \frac{d\ell_m}{d\xi}(\xi) \ell_n(\eta) + \\ & \sum_{m=0}^p \sum_{n=0}^p \tilde{g}_{mn}^D \ell_m(\xi) \frac{d\ell_n}{d\eta}(\eta) + \\ & [\tilde{f}_L^I - \tilde{f}(-1, \eta)] \frac{dg_L}{d\xi}(\xi) + [\tilde{f}_R^I - \tilde{f}(1, \eta)] \frac{dg_R}{d\xi}(\xi) + \\ & [\tilde{g}_D^I - \tilde{g}(\xi, -1)] \frac{dg_L}{d\eta}(\eta) + [\tilde{g}_U^I - \tilde{g}(\xi, 1)] \frac{dg_R}{d\eta}(\eta) = 0\end{aligned}\quad (\text{A.6})$$

which can be integrated in time with an appropriate time marching scheme.

A.2. The CPR method for Triangles

Unlike with QUAD elements, the application of the flux reconstruction methods in Triangular (TRIA) elements is not as straightforward. There are different ways to deal with TRIA elements. One possibility is to apply a coordinate transformation to the QUAD element and transform it into a TRIA one [71, 121, 122] (one edge is collapsed to one point). Although this method has the advantage of allowing the same formulation of QUADs to be used on triangular elements it requires special treatment of values at the collapsing edge.

In order to avoid the special treatment of nodes at collapsed edges the flux reconstruction formu-

lation for triangles has to differ from that of quadrilaterals [123, 124]. The need for a different formulation stems from the fact there is not set of two independent coordinates that define the triangle. Both methodologies (the TRIA formulation and the QUAD-to-TRIA transformation) imply additional operations to be performed with respect to the baseline 1D formulation. To the author's knowledge, there is no clear advantage of using one method over the other.

The Lifting Collocation Penalty (LCP) formulation by Wang and Gao [124] naturally extends the FR approach to mixed grids. Due to its simplicity and its consistency with the formulation chosen for quadrilateral elements, this has been the formulation chosen in this work. A summary of the LCP formulation is presented in what follows. Let us consider a two-dimensional hyperbolic conservation law

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0 \quad (\text{A.7})$$

where $\mathbf{F} = (\mathbf{F}_x, \mathbf{F}_y)$. We assume that the computational domain is divided into N non-overlapping elements $\{\Omega_i\}$. Once divided, we follow a Galerkin approach by multiplying Eq. A.7 by a test function w and integrating over the element:

$$\int_{\Omega_i} w \frac{\partial \mathbf{U}}{\partial t} d\Omega + \int_{\partial \Omega_i} w \mathbf{F}(\mathbf{U}) \cdot \mathbf{n} dS - \int_{\Omega_i} \nabla w \cdot \mathbf{F}(\mathbf{U}) d\Omega = 0 \quad (\text{A.8})$$

where the flux terms have been obtained by integrating by parts and applying the Gauss theorem and where $\partial \Omega_i$ is the boundary of the element Ω_i . As in the 1D case, we approximate the state variable and fluxes with

$$\begin{aligned} \mathbf{U} &\approx \mathbf{U}^e = \sum_{i=0}^{N_e} \mathbf{U}_i^e L_i(x, y) \\ \mathbf{F} &\approx \mathbf{F}^e = \sum_{i=0}^{N_e} \mathbf{F}_i^e L_i(x, y) \end{aligned} \quad (\text{A.9})$$

where N_e is the number of solution points within the element e and L_i are Lagrange polynomials of degree p . This approximate solution is required to satisfy

$$\int_{\Omega_i} w \frac{\partial \mathbf{U}^e}{\partial t} d\Omega + \int_{\partial \Omega_i} w \tilde{F}^n(\mathbf{U}^e, \mathbf{U}^{e+}, \mathbf{n}) dS - \int_{\Omega_i} \nabla w \cdot \mathbf{F}(\mathbf{U}^e) d\Omega = 0 \quad (\text{A.10})$$

where the super-index n denotes normal across the interface and where the normal flux at the

interfaces has been replaced by a common Riemann flux:

$$\mathbf{F}(\mathbf{U}) \cdot \mathbf{n} \approx \tilde{F}^n(\mathbf{U}^e, \mathbf{U}^{e+}, \mathbf{n}) \quad (\text{A.11})$$

where $\mathbf{U}^{e+}$ is the solution value at the corresponding node outside the element e . As with the 1D case, it is the common interface flux what is coupling elements together. Integrating by parts again the remaining flux volume integral we obtain:

$$\int_{\Omega_i} w \frac{\partial \mathbf{U}^e}{\partial t} d\Omega + \int_{\Omega_i} w \nabla \cdot \mathbf{F}(\mathbf{U}^e) d\Omega + \int_{\partial\Omega_i} w [\tilde{F}^n(\mathbf{U}^e, \mathbf{U}^{e+}, \mathbf{n}) - F^n(\mathbf{U}^e)] dS = 0 \quad (\text{A.12})$$

The last term in A.12 can be viewed as a penalty term, *i.e.* it penalizes the flux discontinuities at the interface. As explained in [124] a correction function δ of order p is introduced that satisfies the “lifting” operation

$$\int_{\Omega_i} w \delta d\Omega = \int_{\partial\Omega_i} w [\tilde{F}^n] dS \quad (\text{A.13})$$

where $[\tilde{F}^n] = \tilde{F}^n(\mathbf{U}^e, \mathbf{U}^{e+}, \mathbf{n}) - F^n(\mathbf{U}^e)$ is the normal flux difference. Equation A.12 can now be written in terms of the correction function as

$$\int_{\Omega_i} w \left[\frac{\partial \mathbf{U}^e}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}^e) + \delta_i \right] d\Omega = 0 \quad (\text{A.14})$$

since the terms within the squared brackets are polynomials of degree p and the test functions are selected so that there is a unique solution, Eq. A.14 has to be satisfied everywhere within the element:

$$\frac{\partial \mathbf{U}^e}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}^e) + \delta_i = 0 \quad (\text{A.15})$$

where only the correction function δ is so far unknown. As it can be seen, we have recovered a differential form where only δ depends on the weighting functions. In order to determine δ some assumptions need to be made. First, it is assumed that the correction function is of the form $\delta_i = \sum_{j=1}^N \delta_{ij} L_j$. It is also assumed that the flux jump at the interfaces for each face f can also be put as a $[\tilde{F}^n]_f = \sum_j [\tilde{F}^n]_{f,j} L_j$. Finally, it is further assumed that the weighting functions

take the form of Lagrange polynomials, *i.e.*, $w_j = L_j$, which is the approach taken in the DG framework. Equation A.13 takes then the form

$$\int_{\Omega_i} L_k \sum_j L_j \delta_{i,j} d\Omega = \sum_{f \in \partial\Omega_i} \int_f L_k \sum_l L_l [\tilde{F}^n]_{f,l} dS \quad (\text{A.16})$$

which establishes the relationship of δ with the flux jump at the interface. This relationship can be put in terms of the lifting coefficients, α_{jfl} , with which Eq. A.15 becomes

$$\frac{\partial \mathbf{U}_i^e}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}_i^e) + \frac{1}{V_i} \sum_{f \in \partial V_i} \sum_l \alpha_{jfl} [\tilde{F}^n]_{fl} S_f = 0 \quad (\text{A.17})$$

where V_i is the volume of the element. Since the lifting correction function, δ , depends on the weighting function so does the lifting coefficients. The lifting coefficients can be integrated in the standard element and are, hence, constant and independent from the geometry. In the standard element the discretised equation becomes

$$\frac{\partial \mathbf{U}_i^\xi}{\partial t} + \nabla^\xi \cdot \mathbf{F}^\xi(\mathbf{U}_i^\xi) + \frac{1}{V_i^\xi} \sum_{f \in \partial V_i^\xi} \sum_l \alpha_{jfl} [\tilde{F}^n]_{fl}^\xi S_f^\xi = 0 \quad (\text{A.18})$$

where again, here, $J = \frac{\partial(x,y)}{\partial(\xi,\eta)}$ is the Jacobian of the coordinate transformation; and the superscript ξ has been used to denote variables and operators in computational space which are defined as

$$\begin{aligned} \mathbf{U}^\xi &= |J| \mathbf{U}^e \\ \nabla^\xi &= \left(\frac{\partial}{\partial \xi}, \frac{\partial}{\partial \eta} \right) \\ \mathbf{F}^\xi &= |J| (\xi_x \mathbf{F}_x + \xi_y \mathbf{F}_y) \\ \mathbf{F}^\eta &= |J| (\eta_x \mathbf{F}_x + \eta_y \mathbf{F}_y) \end{aligned} \quad (\text{A.19})$$

A.3. Discretization of the 2D linear advection equation

A.3.0.1. Discretization with QUAD elements

The linearised form of equation A.7 is considered for the following discretization

$$\frac{\partial u}{\partial t} + c_x \frac{\partial u}{\partial x} + c_y \frac{\partial u}{\partial y} = 0 \quad (\text{A.20})$$

where the propagation speed $c = (c_x, c_y)$ is at angle $\alpha = \arctan \frac{c_y}{c_x}$ with respect to the x axis.

In order to perform the analysis of the 2D case some assumptions with regards to the mesh are done. We first consider the case of a mesh with uniform rectangular elements of size Δx and Δy in the x and y directions, respectively. In discrete form

$$\begin{aligned} \frac{\partial u_{ij}^e}{\partial t} + \frac{2c_x}{\Delta x} \left\{ \sum_{m=0}^p \sum_{n=0}^p u_{mn}^e \frac{d\ell_m}{d\xi}(\xi_i) \ell_n(\eta_j) + [u_{pj}^{eL} - u_{0j}^e] \frac{dg_L}{d\xi}(\xi_i) \right\} \\ + \frac{2c_y}{\Delta y} \left\{ \sum_{m=0}^p \sum_{n=0}^p u_{mn}^e \ell_m(\xi_i) \frac{d\ell_n}{d\eta}(\eta_j) + [u_{ip}^{eD} - u_{i0}^e] \frac{dg_D}{d\eta}(\eta_j) \right\} = 0 \end{aligned} \quad (\text{A.21})$$

where u^{eL} and u^{eD} denote the solution at the left and below the considered element, respectively. By defining the aspect ratio (AR), as $\text{AR} = \frac{\Delta x}{\Delta y}$ (we assume here Δy to be the short direction so that $\text{AR} \geq 1$) one could write the previous equation in matrix form

$$\frac{\partial \mathbf{u}^e}{\partial t} + \frac{2c}{\Delta y} \left\{ \left[\frac{\cos(\alpha)}{\text{AR}} (D_\xi - C_L) + \sin(\alpha) (D_\eta - C_D) \right] \mathbf{u}^e + \frac{\cos(\alpha)}{\text{AR}} C_L \mathbf{u}^{eL} + \sin(\alpha) C_D \mathbf{u}^{eD} \right\} = 0 \quad (\text{A.22})$$

where the matrices D_ξ , D_η contain the products of the Lagrange polynomials and their derivatives and are used to compute the discontinuous derivatives and C_L , C_D contain the derivatives of the Radau polynomials and are used compute the correction terms due to discontinuities at the interfaces. This is the equivalent of Eq. 2.56 in multiple dimensions. If solutions of the form $u^e(x, y) = u^{k_1 k_2} \exp(ik_1 x) \exp(ik_2 y)$ are sought, and considering that for the element to the left and to the one below one has $u^{eL}(x - \Delta x, y) = \exp(-ik_1 \Delta x) u(x, y)$ and

$u^{eD}(x, y - \Delta y) = \exp(-ik_2 \Delta y) u^e(x, y)$, respectively, the above equation becomes

$$\frac{\partial \mathbf{u}^e}{\partial t} + \frac{2c}{\Delta y} \left\{ \frac{\cos(\alpha)}{\text{AR}} [D_\xi + (\exp(-ik_1 \Delta x) - 1) C_L] + [D_\eta + (\exp(-ik_2 \Delta y) - 1) C_D] \sin(\alpha) \right\} \mathbf{u}^e = 0 \quad (\text{A.23})$$

It can be noted in Eq. A.23 that for $\alpha = 0$ the 1D case scaled with the AR is recovered and for $\alpha = \pi/2$ again the 1D case is recovered. It is seen that what happens in one direction is independent from what happens in the other one.

A.3.0.2. Discretization with TRIA elements

As with the QUAD elements, assumptions need to be made for meshes containing TRIA elements. For simplicity, we consider the mesh of triangular elements to be the result of splitting the QUAD elements by its diagonal. This serves us to use the same periodic relations as in the QUAD case. In order to derive the discretised equations we depart from equation A.18 already written in computational coordinates:

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{J} \left\{ \frac{\partial \mathbf{f}^\xi}{\partial \xi} + \frac{\partial \mathbf{f}^\eta}{\partial \eta} + \frac{1}{V_i^\xi} \sum_{f \in \partial V_i^\xi} \sum_l \alpha_{jfl} [\tilde{\mathbf{f}}]_{fl}^\xi S_f^\xi \right\} \quad (\text{A.24})$$

The first two terms in the Right-Hand-Side (RHS) of equation A.24 are the discontinuous derivatives. These can be put in matrix form as $\frac{c}{\Delta y} (D_\xi + D_\eta) \mathbf{u}$ with both D_ξ and D_η depending on α and AR. The size of these two matrices is $2N_{SP} \times 2N_{SP}$ with N_{SP} the number of solution points per element. Unlike the QUAD case, in TRIA elements the dependence on α and AR cannot be put explicitly the reason being that we cannot separate the terms in the physical directions as in the other case.

The last term in the RHS of equation A.24 is put as a product of two matrices $\frac{c}{\Delta y} \Theta \Lambda \mathbf{u}$. The first matrix, Θ , is the matrix associated to the lifting coefficients which already accounts for the volume (V_i^ξ) and face area (S_f^ξ) factors and is of size $2N_{SP} \times 2 \cdot 3(p+1)$. Finally, matrix Λ determines the jumps across the interface. It is a function of $k_x \Delta x$ and $k_y \Delta y$ and it is of size

Figure A.1.: Eigenvalues for a QUAD element with an AR=10 (green dots) along with the eigenvalues of the 1D case (black dots) and the stability region of a four-steps fourth-order RK time integration scheme (black line).

$2 \cdot 3(p+1) \times 2N_{SP}$. The final discretised form of equation A.24 in matrix form is:

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{c}{\Delta y} (D_\xi + D_\eta + \Theta \Lambda) \mathbf{u} \quad (\text{A.25})$$

A.3.0.3. Eigenvalues of the 2D spatial discretization

The influence of the 2D parameters, α and AR, on the eigenvalues is considered here. Again the eigenvalues are multiplied by the time-step, Δt , so that they can be plotted in the complex $\lambda \Delta t$ plane. The CFL number is here defined as $\text{CFL} = \frac{c \Delta t}{\Delta y}$ since Δy is taken to be the smaller dimension. Since, in general, the 2D CFL, CFL_{2D} , is smaller than in 1D one, CFL_{1D} , the CFL_{2D} will be expressed in terms of CFL_{1D} .

Figure A.2.: Eigenvalues for a QUAD element with an AR=10 (green dots) along with the eigenvalues of the 1D case (black dots) and the stability region of a four-steps fourth-order RK time integration scheme (black line).

Figure A.3.: Eigenvalues for a TRIA element with an AR=1 (green dots) along with the stability region of a four-steps fourth-order RK time integration scheme (black line).

Figure A.4.: Eigenvalues for a TRIA element with an AR=10 (green dots) along with the eigenvalues of the 1D case (black dots) and the stability region of a four-steps fourth-order RK time integration scheme (black line).

Figures A.1 and A.2 show the eigenvalues of the QUAD discretization for AR=1 and AR=10, respectively. A CFL of $\text{CFL}_{2\text{D}} = \text{CFL}_{1\text{D}}/1.1$ for the first and $\text{CFL}_{2\text{D}} = \text{CFL}_{1\text{D}}/2$ for the latter have been used, respectively. It is seen that, with these CFL numbers, the families of eigenvalues scale appropriately and are enclosed by the 1D eigenvalues. Changing the propagation direction, α , the distribution of the eigenvalues also changes although always within the 1D curve.

Equation A.23 itself suggest a scaling of the form

$$\text{CLF}_{2\text{D}} = \text{CFL}_{1\text{D}} \left(1 + \frac{1}{\text{AR}} \right)^{-1} \quad (\text{A.26})$$

which is the scaling used in Fig. A.1 and A.2. This same scaling of the CFL number has been used in the TRIA element cases. Figures A.3 and A.4 show the eigenvalues for TRIA discretization at $\alpha = 45^\circ$. As it can be seen, the eigenvalues also scale well with the proposed scaling A.26 albeit them being slightly more unstable (more negative real part). Unlike with QUAD elements, changing the the propagation direction in these cases reduces the size of the eigenvalues, i.e., they come closer to the origin, taking them away from the unstable region.

Understanding the behaviour of the eigenvalues with varying geometry is important in explicit

simulations, not just because of stability considerations but because it allows us to run with the largest possible time-step. The time-step is linear with the time the simulation needs to run for so it has a direct impact on cost. Taking into account the definition of CFL and the previous scaling the maximum time-step that can be taken without falling out of the stable region is

$$\Delta t = \text{CFL}_{1D} \frac{\Delta y}{c} \left(1 + \frac{1}{\text{AR}} \right)^{-1} \quad (\text{A.27})$$

where $\text{CFL}_{1D} = 0.12$.

Bibliography

- [1] Lee, M., and Moser, R. D., 2015. “Direct numerical simulation of turbulent channel flow up to $re_\tau \approx 5200$ ”. *Journal of Fluid Mechanics*, **774**, pp. 395–415.
- [2] Mayle, R. E., 1991. “The 1991 igt scholar lecture: The role of laminar-turbulent transition in gas turbine engines”. *ASME J. Turbomach.*, **113**.
- [3] Hodson, H. P., and Howell, R. J., 2005. “Bladerow interactions, transition, and high-lift aerofoils in low-pressure turbines”. *Annu. Rev. Fluid Mech.*, **37**, pp. 71–98.
- [4] Coull, J. D., and Hodson, H. P., 2011. “Unsteady boundary-layer transition in low-pressure turbines”. *Journal of Fluid Mechanics*, **681**, pp. 370–410.
- [5] Lou, W., and Hourmouziadis, J., 2000. “Separation bubbles under steady and periodic-unsteady main flow conditions”. *ASME J. Turbomach.*, **122**(4), pp. 634–643.
- [6] Opoka, M. M., Thomas, R. L., and Hodson, H. P., 2008. “Boundary layer transition on the high lift t106a low-pressure turbine blade with an oscillating downstream pressure field”. *Journal of Turbomachinery*, **130**, p. 021009.
- [7] Volino, R. J., 2012. “Effect of unsteady wakes on boundary layer separation on a very high lift low pressure turbine airfoil”. *Journal of Turbomachinery*(1), p. 011011.
- [8] Montomoli, F., Hodson, H., and Haselbach, F., 2010. “Effect of roughness and unsteadiness on the performance of a new low pressure turbine blade at low reynolds numbers”. *Journal of Turbomachinery*, **132**(3), p. 031018.
- [9] Ameri, A., 2018. “Implicit-les simulation of variable-speed power turbine cascade for low free-stream turbulence conditions”. In *ASME Turbo Expo: Power for Land*,

- Sea, and Air Volume 2B: Turbomachinery, American Society of Mechanical Engineers, p. V02BT41A032.
- [10] Cui, J., Rao, V. N., and Tucker, P., 2016. “Numerical investigation of contrasting flow physics in different zones of a high-lift low-pressure turbine blade”. *ASME J. Turbomach.*, **138**(1), p. 011003.
- [11] Hammer, F., Sandham, N. D., and Sandberg, R. D., 2018. “The influence of different wake profiles on losses in a low pressure turbine cascade”. *International Journal of Turbomachinery, Propulsion and Power*, **3**(2), p. 10.
- [12] Kalitzin, G., Wu, X., and Durbin, P. A., 2003. “Dns of fully turbulent flow in a lpt passage”. *International Journal of Heat and Fluid Flow*, **24**(4), pp. 636–644.
- [13] McAuliffe, B. R., and Yaras, M. I., 2010. “Transition mechanisms in separation bubbles under low-and elevated-freestream turbulence”. *ASME J. Turbomach.*, **132**(1), p. 011004.
- [14] Medic, G., and Sharma, O., 2012. “Large-eddy simulation of flow in a low-pressure turbine cascade”. In *ASME Turbo Expo 2012: Turbine Technical Conference and Exposition*, American Society of Mechanical Engineers, pp. 1239–1248.
- [15] Michelassi, V., Wissink, J., Fr-Ograve, J., hlich, and Rodi, W., 2003. “Large-eddy simulation of flow around low-pressure turbine blade with incoming wakes”. *AIAA journal*, **41**(11), pp. 2143–2156.
- [16] Sandberg, R., Pichler, R., and Chen, L., 2012. “Assessing the sensitivity of turbine cascade flow to inflow disturbances using direct numerical simulation”. In *13th International Symposium for Unsteady Aerodynamics, Aeroacoustics and Aeroelasticity in Turbomachinery (ISUAAAT)*.
- [17] Schulte, V., and Hodson, H. P., 1998. “Unsteady wake-induced boundary layer transition in high lift lp turbines”. *ASME J. Turbomach.*, **120**(1), pp. 28–35.
- [18] Jacobs, R., and Durbin, P., 2001. “Simulations of bypass transition”. *Journal of Fluid Mechanics*, **428**, p. Journal of Fluid Mechanics.

- [19] Roberts, S. K., and Yaras, M. I., 2006. “Large-eddy simulation of transition in a separation bubble”. *Journal of fluids engineering*, **128**(2), pp. 232–238.
- [20] Wu, X., and Durbin, P. A., 2001. “Evidence of longitudinal vortices evolved from distorted wakes in a turbine passage”. *Journal of Fluid Mechanics*, **446**, pp. 199–228.
- [21] Germano, M., Piomelli, U., Moin, P., and Cabot, W. H., 1991. “A dynamic subgrid-scale eddy viscosity model”. *Physics of Fluids A: Fluid Dynamics*, **3**(7), pp. 1760–1765.
- [22] Lilly, D. K., 1992. “A proposed modification of the germano subgrid-scale closure method”. *Physics of Fluids A: Fluid Dynamics*, **4**(3), pp. 633–635.
- [23] Michelassi, V., Chen, L.-W., Pichler, R., and Sandberg, R. D., 2015. “Compressible direct numerical simulation of low-pressure turbines—part ii: Effect of inflow disturbances”. *Journal of Turbomachinery*, **137**(7), p. 071005.
- [24] Zhang, X. F., and Hodson, H., 2010. “Effects of reynolds number and freestream turbulence intensity on the unsteady boundary layer development on an ultra-high-lift low pressure turbine airfoil”. *Journal of Turbomachinery*, **132**(1), p. 011016.
- [25] Michelassi, V., Chen, L., Pichler, R., Sandberg, R., and Bhaskaran, R., 2016. “High-fidelity simulations of low-pressure turbines: Effect of flow coefficient and reduced frequency on losses”. *Journal of Turbomachinery*, **138**, p. 111006.
- [26] Pichler, R., Michelassi, V., Sandberg, R., and Ong, J., 2018. “Highly resolved large eddy simulation study of gap size effect on low-pressure turbine stage”. *Journal of Turbomachinery*, **140**(2), p. 021003.
- [27] Mansour, M. L., Konan, S. M., and Goswami, S., 2007. “Prediction of lapse rate in low pressure turbines with and without modeling of the laminar-turbulent transition”. In *ASME Turbo Expo 2007: Power for Land, Sea, and Air*, American Society of Mechanical Engineers, pp. 1189–1197.
- [28] Lyall, M. E., King, P. I., Sondergaard, R., Clark, J. P., and McQuilling, M. W., 2012. “An investigation of reynolds lapse rate for highly loaded low pressure turbine airfoils with forward and aft loading”. *ASME J. Turbomach.*, **134**(5), p. 051035.

- [29] Hura, H. S., Joseph, J., and Halstead, D. E., 2012. “Reynolds number effects in a low pressure turbine”. In ASME. Turbo Expo: Power for Land, Sea, and Air, Volume 8: Turbomachinery, Parts A, B, and C, American Society of Mechanical Engineers, pp. 1121–1130.
- [30] Lesieur, M., and Metais, O., 1996. “New trends in large-eddy simulations of turbulence”. *Annual review of fluid mechanics*, **28**(1), pp. 45–82.
- [31] Vreman, A., 2004. “An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications”. *Physics of fluids*, **16**(10), pp. 3670–3681.
- [32] You, D., and Moin, P., 2007. “A dynamic global-coefficient subgrid-scale eddy-viscosity model for large-eddy simulation in complex geometries”. *Physics of Fluids*, **19**(6), p. 065110.
- [33] Sidharth, G., and Candler, G. V., 2018. “Subgrid-scale effects in compressible variable-density decaying turbulence”. *Journal of Fluid Mechanics*, **846**, pp. 428–459.
- [34] Sayadi, T., and Moin, P., 2012. “Large eddy simulation of controlled transition to turbulence”. *Physics of Fluids*, **24**(11), p. 114103.
- [35] Smagorinsky, J., 1963. “General circulation experiments with the primitive equations: I. the basic experiment”. *Monthly weather review*, **91**(3), pp. 99–164.
- [36] Moin, P., Squires, K., Cabot, W., and Lee, S., 1991. “A dynamic subgrid-scale model for compressible turbulence and scalar transport”. *Physics of Fluids A: Fluid Dynamics*, **3**(11), pp. 2746–2757.
- [37] Lee, J., Choi, H., and Park, N., 2010. “Dynamic global model for large eddy simulation of transient flow”. *Physics of Fluids*, **22**(7), p. 075106.
- [38] Grinstein, F. F., Margolin, L. G., and Rider, W. J., 2007. *Implicit large eddy simulation*, Vol. 10. Cambridge university press Cambridge.
- [39] Moura, R., Sherwin, S., and Peiró, J., 2015. “Linear dispersion–diffusion analysis and its application to under-resolved turbulence simulations using discontinuous galerkin spectral/hp methods”. *Journal of Computational Physics*, **298**, pp. 695–710.

- [40] Ma, P., Lv, Y., and Ihme, M., 2015. “Discontinuous galerkin scheme for turbulent flow simulations”. *Annual Research Briefs, Center for Turbulence Research, Stanford University*(1-12).
- [41] Chapman, D. R., 1979. “Computational aerodynamics development and outlook”. *AIAA journal*, **17**, pp. 1293–1313.
- [42] Hallbäck, M., Henningson, D., Johansson, A., and Alfredsson, P., 2013. *Turbulence and Transition Modelling: Lecture Notes from the ERCOFTAC/IUTAM Summerschool Held in Stockholm, 12–20 June, 1995*. Springer Science & Business Media, ch. Large-Eddy Simulations: theory and applications (by U. Piomelli and J.R. Chasnov), pp. 269–336.
- [43] Piomelli, U., 2008. “Wall-layer models for large-eddy simulations”. *Progress in aerospace sciences*, **44**(6), pp. 437–446.
- [44] Carton de Wiart, C., Hillewaert, K., Lorriaux, E., and Verheylewegan, G., 2015. “Development of a discontinuous galerkin solver for high quality wall-resolved/modelled dns and les of practical turbomachinery flows on fully unstructured meshes”. In ASME Turbo Expo 2015: Turbine Technical Conference and Exposition, American Society of Mechanical Engineers Digital Collection.
- [45] Cassinelli, A., Montomoli, F., Adami, P., and Sherwin, S. J., 2018. “High fidelity spectral/hp element methods for turbomachinery”. In ASME Turbo Expo 2018: Turbomachinery Technical Conference and Exposition, American Society of Mechanical Engineers, pp. V02CT42A020–V02CT42A020.
- [46] Marconcini, M., Pacciani, R., Arnone, A., Michelassi, V., Pichler, R., Zhao, Y., and Sandberg, R., 2018. “Les and rans analysis of the end-wall flow in a linear lpt cascade with variable inlet conditions: Part ii loss generation”. In ASME Turbo Expo 2018: Turbomachinery Technical Conference and Exposition, American Society of Mechanical Engineers, pp. V02BT41A024–V02BT41A024.
- [47] Marty, J., 2014. “Numerical investigations of separation-induced transition on high-

- lift low-pressure turbine using rans and les methods”. *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, **228**(8), pp. 924–952.
- [48] Scillitoe, A. D., Tucker, P. G., and Adami, P., 2018. “Large eddy simulation of boundary layer transition mechanisms in a gas-turbine compressor cascade”. In ASME Turbo Expo 2018: Turbomachinery Technical Conference and Exposition, American Society of Mechanical Engineers, pp. V02AT45A009–V02AT45A009.
- [49] Bolinches-Gisbert, M., Robles, D. C., Corral, R., and Gisbert, F., 2020. “Prediction of reynolds number effects on low-pressure turbines using a high-order iles method”. *Journal of Turbomachinery*, **142**(3).
- [50] Wang, Z. J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H. T., Kroll, N., May, G., Persson, P.-O., van Leer, B., and Visbal, M., 2013. “High-order cfd methods: current status and perspective”. *International Journal for Numerical Methods in Fluids*, **72**(8), pp. 811–845.
- [51] Vincent, P. E., Castonguay, P., and Jameson, A., 2011. “Insights from von neumann analysis of high-order flux reconstruction schemes”. *Journal of Computational Physics*, **230**(22), pp. 8134–8154.
- [52] Sandberg, R. D., Michelassi, V., Pichler, R., Chen, L., and Johnstone, R., 2015. “Compressible direct numerical simulation of low-pressure turbinesâpart i: Methodology”. *Journal of Turbomachinery*, **137**(5), p. 051011.
- [53] Kim, J. W., and Sandberg, R. D., 2012. “Efficient parallel computing with a compact finite difference scheme”. *Computers & Fluids*, **58**, pp. 70–87.
- [54] Ekaterinaris, J. A., 2005. “High-order accurate, low numerical diffusion methods for aerodynamics”. *Progress in Aerospace Sciences*, **41**(3-4), pp. 192–300.
- [55] Bassi, F., and Rebay, S., 1997. “A high-order accurate discontinuous finite element method for the numerical solution of the compressible navier–stokes equations”. *Journal of computational physics*, **131**(2), pp. 267–279.
- [56] Brazell, M. J., Sitaraman, J., and Mavriplis, D. J., 2016. “An overset mesh approach for

- 3d mixed element high-order discretizations”. *Journal of Computational Physics*, **322**, pp. 33–51.
- [57] Fernandez, P., Nguyen, N. C., and Peraire, J., 2017. “The hybridized discontinuous galerkin method for implicit large-eddy simulation of transitional turbulent flows”. *Journal of Computational Physics*, **336**, pp. 308–329.
- [58] Huynh, H. T., 2007. “A flux reconstruction approach to high-order schemes including discontinuous galerkin methods”. In 18th AIAA Computational Fluid Dynamics Conference, p. 4079.
- [59] May, G., and Jameson, A., 2005. “High-order accurate methods for high-speed flow”. In 17th AIAA Computational Fluid Dynamics Conference, p. 5251.
- [60] Vincent, P. E., Castonguay, P., and Jameson, A., 2011. “A new class of high-order energy stable flux reconstruction schemes”. *Journal of Scientific Computing*, **47**(1), pp. 50–72.
- [61] Vincent, P. E., Farrington, A. M., Witherden, F. D., and Jameson, A., 2015. “An extended range of stable-symmetric-conservative flux reconstruction correction functions”. *Computer Methods in Applied Mechanics and Engineering*, **296**, pp. 248–272.
- [62] Vincent, P. E., and Jameson, A., 2011. “Facilitating the adoption of unstructured high-order methods amongst a wider community of fluid dynamicists”. *Mathematical Modelling of Natural Phenomena*, **6**(3), pp. 97–140.
- [63] Witherden, F. D., and Jameson, A., 2017. “Future directions in computational fluid dynamics”. In 23rd AIAA Computational Fluid Dynamics Conference, p. 3791.
- [64] Reed, W. H., and Hill, T., 1973. Triangular mesh methods for the neutron transport equation. Tech. rep., Los Alamos Scientific Lab., N. Mex.(USA).
- [65] Cockburn, B., and Shu, C.-W., 1989. “Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws. ii. general framework”. *Mathematics of computation*, **52**(186), pp. 411–435.
- [66] Kopriva, D. A., and Kolas, J. H., 1996. “A conservative staggered-grid chebyshev mul-

- tidomain method for compressible flows”. *Journal of computational physics*, **125**(1), pp. 244–261.
- [67] Bassi, F., and Rebay, S., 1997. “High-order accurate discontinuous finite element solution of the 2d euler equations”. *Journal of computational physics*, **138**, pp. 251–285.
- [68] Aguiar, A. R., Breviglieri, C., Martini dos Santos, T., Azevedo, J. L. F., and Wang, Z. J., 2017. “High-order meshes for flow simulations with a spectral difference method”. In 23rd AIAA Computational Fluid Dynamics Conference, p. 4294.
- [69] Fortunato, M., and Persson, P.-O., 2016. “High-order unstructured curved mesh generation using the winslow equations”. *Journal of Computational Physics*, **307**, pp. 1–14.
- [70] Persson, P.-O., and Peraire, J., 2006. “Sub-cell shock capturing for discontinuous galerkin methods”. In 44th AIAA Aerospace Sciences Meeting and Exhibit, p. 112.
- [71] Mengaldo, G., De Grazia, D., Moxey, D., Vincent, P. E., and Sherwin, S. J., 2015. “Dealiasing techniques for high-order spectral element methods on regular and irregular grids”. *Journal of Computational Physics*, **299**, pp. 56–81.
- [72] Moura, R. C., Sherwin, S. J., and Peiró, J., 2016. “Eigensolution analysis of spectral/hp continuous galerkin approximations to advection–diffusion problems: Insights into spectral vanishing viscosity”. *Journal of Computational Physics*, **307**, pp. 401–422.
- [73] Coppola, G., Capuano, F., Pirozzoli, S., and de Luca, L., 2019. “Numerically stable formulations of convective terms for turbulent compressible flows”. *Journal of Computational Physics*, **382**, pp. 86–104.
- [74] Kopriva, D. A., and Gassner, G. J., 2014. “An energy stable discontinuous galerkin spectral element discretization for variable coefficient advection problems”. *SIAM Journal on Scientific Computing*, **36**, pp. A2076–A2099.
- [75] Gassner, G. J., Winters, A. R., and Kopriva, D. A., 2016. “Split form nodal discontinuous galerkin schemes with summation-by-parts property for the compressible euler equations”. *Journal of Computational Physics*, **327**, pp. 39–66.

- [76] Karniadakis, G., and Sherwin, S., 1999. *Numerical mathematics and scientific computation*. Oxford University Press.
- [77] Witherden, F. D., and Vincent, P. E., 2014. “An analysis of solution point coordinates for flux reconstruction schemes on triangular elements”. *Journal of Scientific Computing*, **61**(2), pp. 398–423.
- [78] Vermeire, B. C., and Vincent, P. E., 2016. “On the properties of energy stable flux reconstruction schemes for implicit large eddy simulation”. *Journal of Computational Physics*, **327**, pp. 368–388.
- [79] Courant, R., Friedrichs, K., and Lewy, H., 1928. “Über die partiellen differenzengleichungen der mathematischen physik”. *Mathematische annalen*, **100**(1), pp. 32–74.
- [80] Toro, E. F., 2013. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media.
- [81] Roe, P. L., 1981. “Approximate riemann solvers, parameter vectors, and difference schemes”. *Journal of computational physics*, **43**, pp. 357–372.
- [82] Lax, P. D., 1954. “Weak solutions of nonlinear hyperbolic equations and their numerical computation”. *Communications on pure and applied mathematics*, **7**(1), pp. 159–193.
- [83] Butcher, J. C., 1963. “Coefficients for the study of runge-kutta integration processes”. *Journal of the Australian Mathematical Society*, **3**(2), pp. 185–201.
- [84] Vermeire, B., and Vincent, P., 2017. “On the behaviour of fully-discrete flux reconstruction schemes”. *Computer Methods in Applied Mechanics and Engineering*, **315**, pp. 1053–1079.
- [85] Van den Abeele, K., 2009. “Development of high-order accurate schemes for unstructured grids”. *Phd thesis in Vrije Universiteit Brussel*.
- [86] Contreras, J., Corral, R., Fernandez-Castaneda, J., Pastor, G., and Vasco, C., 2002. “Semi-unstructured grid methods for turbomachinery applications”. In *Turbo Expo: Power for Land, Sea, and Air*, Vol. 3610, pp. 947–955.

-
- [87] Wang, Z., Quintnl, J., and Corral, R., 2017. “Accelerating advancing layer viscous mesh generation for 3d complex configurations”. *Procedia engineering*, **203**, pp. 128–140.
- [88] Burgos, M. A., Chia, J. M., Corral, R., and López, C., 2010. “Rapid meshing of turbomachinery rows using semi-unstructured multi-block conformal grids”. *Engineering with Computers*, **26**(4), pp. 351–362.
- [89] Warburton, T., 2006. “An explicit construction of interpolation nodes on the simplex”. *Journal of Engineering Mathematics*, **56**(3), pp. 247–262.
- [90] Corral, R., Gisbert, F., and Pueblas, J., 2017. “Execution of a parallel edge-based navier-stokes solver on commodity graphics processor units”. *International Journal of Computational Fluid Dynamics*, **31**(2), March, pp. 93–108.
- [91] Gisbert, F., Bolinches-Gisbert, M., Pueblas, J., and Corral, R., 2018. “Efficient implementation of flux reconstruction schemes for the simulation of compressible viscous flows on graphics processing unigs”. In Tenth International Conference on Computational Fluid Dynamics (ICCFD10), no. ICCFD10-307.
- [92] Khronos Group, 2011. Opencl 1.2 specification. <https://www.khronos.org/registry/OpenCL/specs/opencl-1.2.pdf>, November.
- [93] Deakin, T., Price, J., Martineau, M., and McIntosh-Smith, S., 2016. “GPU-STREAM v2.0: Benchmarking the achievable memory bandwidth of many-core processors across diverse parallel programming models”. In ISC High Performance 2016 International Workshops, ExaComm, E-MuCoCoS, HPC-IODC, IXPUG, IWOPH, P3MA, VHPC, WOPSSS, M. Taufer, B. Mohr, and J. M. Kunkel, eds., Springer, pp. 489–507.
- [94] NVIDIA, 2018. cublas. <https://docs.nvidia.com/cuda/cublas>, June.
- [95] Rupp, K., Tillet, P., Rudolf, F., Weinbub, J., Morhammer, A., Grasser, T., JÄEngel, A., and Selberherr, S., 2016. “Viennacl—linear algebra library for multi- and many-core architectures”. *SIAM Journal on Scientific Computing*, **38**(5), pp. S412–S439.
- [96] Klöckner, A., Warburton, T., Bridge, J., and Hesthaven, J., 2009. “Nodal discontinuous

- Galerkin methods on graphics processors”. *Journal of Computational Physics*, **228**(21), pp. 7863 – 7882.
- [97] Castonguay, P., Williams, D., Vincent, P., Lopez, M., and Jameson, A., 2011. “On the Development of a High-Order, Multi-GPU Enabled, Compressible Viscous Flow Solver for Mixed Unstructured Grids”. In 20th AIAA Computational Fluid Dynamics Conference, no. 2011-3229.
- [98] de Wiart, C. C., Hillewaert, K., Duponcheel, M., and Winckelmans, G., 2014. “Assessment of a discontinuous Galerkin method for the simulation of vortical flows at high Reynolds number”. *International Journal for Numerical Methods in Engineering*, March.
- [99] Karypis, G., Schloegel, K., and Kumar, V., 2003. Parmetis, parallel graph partitioning and sparse matrix ordering library, version 3.1, August 15.
- [100] Kahan, W., 1965. “Further remarks on reducing truncation errors, commun”. *Assoc. Comput. Mach*, **8**, p. 40.
- [101] Giles, M., 1988. *Non-reflecting boundary conditions for the Euler equations*. Computational Fluid Dynamics Laboratory, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology.
- [102] Poinso, T. J., and Lele, S. K., 1992. “Boundary conditions for direct simulations of compressible viscous flows”. *Journal of computational physics*, **101**(1), pp. 104–129.
- [103] Thompson, K. W., 1987. “Time dependent boundary conditions for hyperbolic systems”. *Journal of computational physics*, **68**(1), pp. 1–24.
- [104] Lodato, G., 2008. “Tridimensional boundary conditions for direct and large-eddy simulation of turbulent flows. sub-grid scale modeling for near-wall region turbulence”. Theses, INSA de Rouen, Dec.
- [105] Odier, N., Poinso, T., Duchaine, F., Gicquel, L., and Moreau, S., 2019. “Inlet and outlet characteristics boundary conditions for large eddy simulations of turbomachinery”.

In ASME Turbo Expo 2019: Turbomachinery Technical Conference and Exposition, American Society of Mechanical Engineers Digital Collection.

- [106] Odier, N., Sanjosé, M., Gicquel, L., Poinso, T., Moreau, S., and Duchaine, F., 2019. “A characteristic inlet boundary condition for compressible, turbulent, multispecies turbomachinery flows”. *Computers & Fluids*, **178**, pp. 41–55.
- [107] Prandtl, L., 1905. “Über flüssigkeitsbewegung bei sehr kleiner reibung. verhandlungen des iii. internationalen mathematiker-kongresses, heidelberg”. *LPGA*, **2**, pp. 575–584.
- [108] Carton de Wiart, C., and Hillewaert, K., 2015. “A discontinuous galerkin method for implicit les of moderate reynolds number flows”. In 53rd AIAA Aerospace Sciences Meeting, p. 0055.
- [109] Hallböck, M., Henningson, D., Johansson, A., and Alfredsson, P., 2013. *Turbulence and Transition Modelling: Lecture Notes from the ERCOFTAC/IUTAM Summerschool Held in Stockholm, 12–20 June, 1995*, Vol. 2. Springer Science & Business Media.
- [110] Goldstein, D., Handler, R., and Sirovich, L., 1993. “Modeling a no-slip flow boundary with an external force field”. *Journal of computational physics*, **105**(2), pp. 354–366.
- [111] Giles, M., 1991. Unsflo: A numerical method for the calculation of unsteady flow in turbomachinery. Tech. rep., Cambridge, Mass.: Gas Turbine Laboratory, Massachusetts Institute of Technology.
- [112] Ng, H., Monty, J., Hutchins, N., Chong, M., and Marusic, I., 2011. “Comparison of turbulent channel and pipe flows with varying reynolds number”. *Experiments in fluids*, **51**, pp. 1261–1281.
- [113] Serna, J., and Lázaro, B., 2015. “On the bursting condition for transitional separation bubbles”. *Aerospace Science and Technology*, **44**, pp. 43–50.
- [114] Lázaro, B. J., González, E., Cadrecha, D., Antoranz, A., and Parra, J., 2017. “Low reynolds number response of high efficiency, intermediate pressure compressor profiles”. In ASME Turbo Expo 2017: Turbomachinery Technical Conference and Exposition, American Society of Mechanical Engineers, pp. V02AT39A008–V02AT39A008.

- [115] Saric, W. S., Reed, H. L., and Kerschen, E. J., 2002. “Boundary-layer receptivity to freestream disturbances”. *Annual review of fluid mechanics*, **34**(1), pp. 291–319.
- [116] Stieger, R., and Hodson, H., 2004. “The transition mechanism of highly loaded low-pressure turbine blades”. *Journal of Turbomachinery*, **126**, pp. 536–543.
- [117] Stieger, R., Hollis, D., and Hodson, H., 2004. “Unsteady surface pressures due to wake-induced transition in a laminar separation bubble on a low-pressure cascade”. *Journal of Turbomachinery*, **126**, pp. 544–550.
- [118] Rose, M. G., Jenny, P., Gier, J., and Abhari, R. S., 2013. “Experimentally observed unsteady work at inlet to and exit from an axial flow turbine rotor”. *Journal of Turbomachinery*, **135**(6), p. 061017.
- [119] Rose, M. G., and Marx, M., 2014. “Unsteady work transfer within a turbine blade row passage”. *Journal of Turbomachinery*, **136**(9), p. 091001.
- [120] Giles, M. B., 1990. “Stator/rotor interaction in a transonic turbine”. *Journal of Propulsion and Power*, **6**(5), pp. 621–627.
- [121] Watkins, J. E., Romero, J., and Jameson, A., 2016. “Multi-gpu, implicit time stepping for high-order methods on unstructured grids”. In 46th AIAA Fluid Dynamics Conference, p. 3965.
- [122] Romero, J., and Jameson, A., 2016. “Extension of the flux reconstruction method to triangular elements using collapsed-edge quadrilaterals”. In 54th AIAA Aerospace Sciences Meeting, p. 1825.
- [123] Castonguay, P., Vincent, P. E., and Jameson, A., 2012. “A new class of high-order energy stable flux reconstruction schemes for triangular elements”. *Journal of Scientific Computing*, **51**(1), pp. 224–256.
- [124] Wang, Z., and Gao, H., 2009. “A unifying lifting collocation penalty formulation including the discontinuous galerkin, spectral volume/difference methods for conservation laws on mixed grids”. *Journal of Computational Physics*, **228**(21), pp. 8161–8186.