A spacetime framework for aerodynamics of complex motions

Imanol Flamarique Ederra
Department of Aerospace Engineering
University of Bristol

A dissertation submitted to the University of Bristol
in accordance with the requirements of the degree of
Doctor of Philosophy in the Faculty of Engineering.

2019
Abstract

A two-dimensional spacetime framework is presented to solve unsteady aerodynamics problems as an alternative to conventional approaches for complex unsteady problems involving large deformations or topological change. Some examples of problems that the spacetime method can cope with seamlessly are store separation, slat and flap deployment, spoiler deflection or rotor-stator configurations. It avoids methods such as Chimera or overset grids, or even re-meshing, by the use of a finite-volume approach both in space and time. The simulation of unsteady problems of dimension \( N \) is effectively done as the simulation of steady problems of dimension \( N+1 \). Hence, both the geometry and its motion are defined by a spacetime mesh in an \( N+1 \) dimensional space. The use of an arbitrary Lagrangian-Eulerian formulation along with a geometric conservation law are also avoided by the spacetime formulation. Moreover, it is a conservative method both in space and time. Therefore, it is very suitable for the solution of time-periodic problems.

The finite-difference approach used for the time integration in conventional methods based on an ALE formulation uses directionally biased schemes since the solution is only known at previous time levels. In contrast with this, the use of a central-difference scheme in spacetime yields non-physical transient solutions as a consequence of pressure waves travelling backwards in time. The search for a more realistic time stencil has led to the formulation of one hybrid (central-difference in space, upwind in time) and two upwind stencils. Initially, most of the work has been done based on an Euler solver. Then, a RANS formulation has also been implemented with the Spalart-Allmaras one-equation turbulence model.

Several two-dimensional unsteady aerodynamics problems have been computed with the different formulations and compared with the central-difference scheme. In particular, the following problems are presented in this work: a one-dimensional periodic piston problem and one with a rapid change of direction; the shock tube problem; a two-dimensional isentropic Euler vortex transport problem; a periodic pitching NACA-0012 aerofoil at different flight conditions; a simple flap deflection; a slat and slotted flap deployment; a spoiler deployment; an investigation of adverse lift due to rapidly deploying spoiler; a full landing case with a combination of slat, flap and spoiler deployments along with ground effect; a case where aerofoils fly in opposite directions at subsonic and supersonic speeds; and a rotor-stator configuration with infinite relative motions. Moreover, some of the spacetime solutions have been correlated with a couple of analytical solutions and some empirical data.

It has been demonstrated that the use of a central-difference stencil leads to non-physical solutions as a consequence of pressure waves travelling backwards in physical-time, as expected. It has also been proved that upwind (e.g. Van Leer, Roe) and hybrid (CSUT = central-difference in space, upwind in time) stencils yield more representative physical solutions and improve the rate of convergence. The benefits derived from the use of an upwind stencil as opposed to a central-difference one are more noticeable in the case of non-periodic problems, especially in fast transients. Unfortunately, upwind stencils are more dissipative and, as implied by the results of the isentropic Euler vortex transport problem, they did not seem to achieve as high a temporal accuracy as the central-difference counterpart. The potential for very efficient time-accurate simulations through the spacetime method has been demonstrated by the use of a variable time-step size across the spatial domain.
Abstract

It is possible to use small time-steps in the neighbourhood of the geometry, where big gradients occur, whilst retaining very large time-steps far away in the farfield, where the solution remains almost constant throughout the whole simulation. The versatility and broad applicability of the spacetime method to almost any kind of unsteady problems have been shown by the simulation of a wide range of problems involving complex boundary motions. Large relative motions or topological changes in the geometry are simulated with ease by the use of a spacetime formulation, which avoids the use of an arbitrary Lagrangian-Eulerian (ALE) formulation in combination with a geometric conservation law (GCL). The solver did not need any modifications to cope with any of the problems presented here which proves its potential for highly automated CFD simulations. This could, in turn, speed up the design cycle of industrial applications.
Acknowledgments

First of all, I would like to thank Drs. Rendall, Gaitonde and Jones for their unconditional support, encouragement and guidance throughout my PhD. They have always had motivational words for me and have given me good advices. All this work would have not been possible without them. I would also like to thank Prof. Allen for his assistance in mesh generation and advice in form of yearly reviews.

Secondly, I would like to dedicate a few lines to all my office colleagues who made my time as a PhD student much more enjoyable, with those curry nights and Friday beers. In particular, I am very grateful to my friends Alessandro, Will, Alex and Matt who have been like a second family during my time in Bristol and with whom I have shared very good moments in and outside office hours. I will never forget our bike rides, our ski holiday, our trip to Denver for the AIAA Aviation Conference or, even, our lunch and dinner times. My gratitude also goes to my friends in Pamplona, Madrid and, more recently, also Braunschweig.

Finally, I would like to thank my family and relatives. I am especially grateful towards my parents who gave me a good education and have been very supportive throughout this whole time. I am also thankful because they have always encouraged me to pursue my goals regardless of how difficult they might seem to be. Last but not least, a special gratitude goes to my girlfriend Laura. She has always believed in me and has encouraged me every time I needed it. I am very grateful for all the moments we have shared together because they have given me the energy I needed to finalise this work.

The work presented here received financial support from Innovate UK as part of the Enhanced Fidelity Transonic Wing (EFT) project led by Airbus.
Author’s Declaration

I declare that the work in this dissertation was carried out in accordance with the regulations of the University of Bristol. The work is original except where indicated by special reference in the text and no part of the dissertation has been submitted for any other degree.

Any views expressed in the dissertation are those of the author and in no way represent those of the University of Bristol.

The dissertation has not been presented to any other University for examination either in the United Kingdom or overseas.

Signed: .......................................................... ..........................................................

Date: .......................................................... ..........................................................
# Contents

- **Title Page** i
- **Abstract** iii
- **Acknowledgments** v
- **Author’s Declaration** vii
- **Contents** ix
- **List of Figures** xiii
- **List of Tables** xix
- **List of Symbols** xxi

## 1 Introduction

1.1 Background .......................................................... 1
1.2 Arbitrary Lagrangian-Eulerian formulation and geometric conservation law ........ 3
1.3 Existing methods for moving and deforming meshes .................................... 4
   1.3.1 Mesh motion .................................................. 5
   1.3.2 Chimera or overset grids .................................. 12
   1.3.3 Sliding grids ............................................... 13
   1.3.4 Immersed boundary method ................................ 13
   1.3.5 Cartesian cut-cell grids .................................. 14
   1.3.6 Re-meshing .................................................. 14
   1.3.7 Meshless or meshfree algorithms .......................... 16
1.4 Spacetime .............................................................. 19
1.5 Outline of thesis ...................................................... 24
## 2 Discretisation methods

2.1 Introduction ........................................................................... 27

2.2 Spatial discretisation of the physical equations ....................... 28

2.2.1 Integral formulation of the Navier-Stokes equations .............. 28

2.2.2 Second order central-difference scheme ............................... 30

2.2.3 Second order upwind scheme .......................................... 34

2.3 Spatial discretisation of the Spalart-Allmaras turbulence model ... 44

2.4 Spacetime discretisation of the physical equations .......... 46

2.4.1 Integral formulation of the Navier-Stokes equations .............. 46

2.4.2 Second order central-difference scheme ............................... 50

2.4.3 Second order upwind scheme .......................................... 53

2.4.4 Hybrid central-difference in space, upwind in time ............... 62

2.5 Spacetime discretisation of the Spalart-Allmaras turbulence model 62

2.6 Boundary conditions for the spatial discretisation ............. 64

2.6.1 Solid boundaries .......................................................... 65

2.6.2 Farfield boundaries ...................................................... 67

2.7 Boundary conditions for the spacetime discretisation ......... 71

2.7.1 Solid boundaries .......................................................... 71

2.7.2 Farfield boundaries ...................................................... 75

2.7.3 Time boundaries .......................................................... 75

## 3 Solution method: time integration and equivalence to ALE

3.1 Introduction ........................................................................... 77

3.2 Explicit methods .................................................................... 78

3.2.1 Explicit fourth-order Runge-Kutta method ............................ 79

3.3 Implicit methods ................................................................... 80

3.3.1 Implicit Runge-Kutta methods .......................................... 81

3.4 Steady-state problems ........................................................ 82

3.5 Unsteady or time-accurate problems ..................................... 83

3.6 Spacetime problems ............................................................ 83

3.7 Comparison between an arbitrary Lagrangian-Eulerian formulation and the spacetime framework 84

3.7.1 Arbitrary Lagrangian-Eulerian formulation .......................... 84

3.7.2 Cell-centered finite-volume spacetime framework ............... 86

3.7.3 Vertex-centered finite-volume spacetime framework .......... 90
4 Stability analysis

4.1 Introduction ................................................. 93

4.2 One-dimensional convective-diffusive equation ................. 95

4.3 Spacetime formulation of the two-dimensional Navier-Stokes equations .............................................. 99

4.3.1 Inviscid terms with a second-order central-difference discretisation ................................................. 106

4.3.2 Inviscid terms with a second-order upwind discretisation ................................................................. 109

4.3.3 Viscous terms with a second-order central-difference discretisation ...................................................... 112

4.3.4 Estimation of the CFL condition ................................ 119

5 Results .................................................................. 121

5.1 Introduction .......................................................... 121

5.1.1 Periodic semi-infinite piston ........................................ 122

5.1.2 Motion of piston with a sharp change of direction .......... 124

5.2 Shock tube problem .................................................. 126

5.3 Two-dimensional isentropic Euler vortex ......................... 128

5.4 Oscillatory and transient pitching NACA 0012 ................. 131

5.4.1 AGARD R-702(3E3) Case 1 ...................................... 132

5.4.2 AGARD R-702(3E3) Case 3 ...................................... 134

5.4.3 AGARD R-702(3E3) Case 5 ...................................... 137

5.4.4 A hybrid stencil (CSUT) ........................................... 140

5.4.5 Negative Spalart-Allmaras model ........................... 142

5.5 Simple flap ............................................................ 147

5.5.1 Investigation of the artificial dissipation in time .......... 151

5.6 Slat and slotted flap .................................................. 152

5.7 Spoiler .................................................................. 155

5.8 Investigation of adverse lift ......................................... 158

5.9 Landing case ............................................................ 162

5.10 Problems with large relative motions ......................... 166

5.10.1 Aerofoils flying in opposite directions ....................... 166

5.10.2 Rotor-stator problem ............................................. 169

6 Conclusions and future work ................................... 171

6.1 Conclusions ............................................................ 171

6.2 Future work ............................................................. 174

A Arbitrary Lagrangian-Eulerian formulation .................... 177
1.1 Examples of fluid flow applications: (a) internal combustion engine (source: https://www.konstrukter.cz/), (b) stator rotor configuration (source: http://www.pfmanllc.com/engineering-design/), and (c) flow around a convertible car, the Jaguar F-type (source: Masters thesis by Imanol Flamarique Ederra) .................................................. 2

1.2 Comparison between the one-dimensional moving mesh in an arbitrary Lagrangian-Eulerian formulation and the equivalent two-dimensional spacetime mesh .................................................. 3

1.3 Mesh deformation for rotated flap using (a) tensional spring analogy and (b) linear elasticity analogy. Extracted from Nielsen et al. .......................................................... 6

1.4 Transfinite interpolation on an L-shaped domain: (a) bad parameterisation yields non-univalent map and (b) good parameterisation yields univalent map. Extracted from Gordon et al. ............... 8

1.5 Bilinear blend yielding (a) a non-univalent map with overspill and (b) the equivalent univalent map through the imposition of auxiliary constraints. Extracted from Gordon et al. ............... 8

1.6 Delaunay triangulation on quadrilateral domain using (a) standard Delaunay triangulation (undesired Delaunay triangles are marked) and (b) Laplace-Delaunay smoothing of the triangulation in (a). Extracted from Field ................................. 9

1.7 Smoothing based on the biharmonic operator compared to the equivalent one based on the Laplace operator: (a) initial (undeformed) mesh, (b) deformed mesh via the Laplace operator, and (c) deformed mesh via the biharmonic operator. Extracted from Helenbrook ............................................. 10

1.8 Viscous mesh deformation of a pitching NACA 0012 aerofoil through radial basis functions (RBF): (a) with a deformation $\Delta \alpha = +5$ deg, (b) detailed view of figure (a) at trailing edge, (c) with a deformation $\Delta \alpha = -5$ deg, (d) detailed view of figure (c) at trailing edge .......................................................... 11

1.9 Chimera grids examples for (a) a moving rotor (extracted from Pomin et al.) and (b) a deploying flap case (extracted from Renzoni et al.) .......................................................... 12

1.10 Interface between adjacent blocks in sliding grid planes: (a) planar interface with equal size cells at either side, (b) planar interface with different cell sizes at either side and (c) curved interface with different cell sizes at either side .......................................................... 13

1.11 Multi-element NHLP-2D aerofoil in takeoff configuration using Cartesian cut-cell grids: (a) overall view of the surrounding mesh, (b) detailed view of slat deflected $\delta_s = 25$ deg, and (c) detailed view of flap deflected $\delta_f = 2$ deg. Extracted from Kidron et al. .......................................................... 15

1.12 Velocity field at $Re = 22,000$ for the intake stroke (infinitely long) of an internal combustion engine using Cartesian cut-cell grids: (a) mean velocity and (b) instantaneous velocity. Extracted from Meinke et al. .......................................................... 15

1.13 Store separation problem using re-meshing. Cutting planes through the Cartesian volume discretisation at different time levels. Extracted from Murman et al. .......................................................... 16
1.14 Solution of the ONERA M6 wing subject to twisting deformation submerged in a freestream flow at $M_\infty = 0.84$ and an angle of attack $\alpha = 3.06$ deg. The Euler equations for fluid motion have been discretised on a cloud of points via a meshfree method as depicted for three sections of the wing. Also, pressure coefficient $C_p$ distributions along these three planes are provided for a coarse, fine and coarse adapted clouds of points. Extracted from Ortega et al. ......... 17

1.15 Solution at the interface of overset grids via a meshless or meshfree method. Extracted from Katz ......... 17

1.16 Solution of free-surface flow after the break of a dam and impact against a vertical wall through the method of smoothed particle hydrodynamics (SPH). Depicted pressure fields at three different time-steps (time increases from top to bottom). Extracted from Colagrossi et al. ......... 18

1.17 Spacetime representation of a two-dimensional NACA 0012 aerofoil oscillating periodically in a pitching motion. The third dimension represents physical time $t$ and is to be integrated along with the spatial coordinates through the use of a spacetime finite-volume mesh. ......... 19

1.18 Illustration of the spacetime mesh used by Thompson et al. The choice of orthogonal planes in the time direction $t$ limits the applicability of this method since more complex motions such as topological changes in the geometry require a coupled integration of spacetime volumes, i.e. via the use of fully unstructured grids. ......... 20

1.19 Solution of a two-dimensional store separation problem in a crossflow at $M_\infty = 0.3$ through a finite-volume spacetime method. Mach contours are depicted at several time slices. Extracted from Rendall et al. ......... 21

1.20 Solution of two aerofoils pitching in tandem through a discontinous Galerkin spacetime method. Depicted entropy contours and mesh at three different time slices. Extracted from Wang et al. ......... 22

2.1 Structured (left) and unstructured (right) meshes ......... 31

2.2 Riemann problem at one-dimensional cell faces due to piecewise linear solution (discontinuous at interfaces). ......... 34

2.3 Slope limiters ......... 44

2.4 Example of spacetime mesh: pitching NACA 0012 aerofoil ......... 49

2.5 Momentum balance: time fluxes (left) and spatial fluxes (right) ......... 54

2.6 Smoothing applied on first-order upwind fluxes to avoid issues with flow aligned faces ......... 64

2.7 Representation of ghost or halo cells at the boundary of an unstructured mesh ......... 65

2.8 Projection of normal velocity $u_n = \mathbf{v} \cdot \mathbf{n}$ along Cartesian coordinates $x$ and $y$ ......... 68

2.9 Spacetime mesh for one-dimensional cell expanding between two consecutive time levels ......... 72

2.10 A wall’s rigid body motion with a segment element shrinking between two consecutive time levels ......... 75

3.1 One-dimensional moving mesh in arbitrary Lagrangian-Eulerian framework, equivalent to the problem formulated in spacetime given in Figure 3.2 ......... 85

3.2 One-dimensional moving mesh in a cell-centered spacetime framework, equivalent to the problem formulated in ALE given in Figure 3.1 ......... 86

3.3 Comparison between the arbitrary Lagrangian-Eulerian framework and the spacetime counterpart for constant time-step size $\Delta t$ and volumes $V_j$ throughout simulation time ......... 90

3.4 One-dimensional moving mesh in a vertex-centered spacetime framework, equivalent to the problem formulated in ALE given in Figure 3.1 ......... 91
4.1 Region of stability of fourth-order four-stages explicit Runge-Kutta scheme (shaded area) eigenvalue spectra for conditions \(4.1\) and \(4.2\). .......................................................... 94

4.2 Region of stability of explicit fourth-order Runge-Kutta method (shaded area) and eigenvalue spectrum of equation \(u_t + au_x = \alpha u_{xx}\) discretised via a central-difference scheme (region inside dashed green curve). .......................................................... 99

4.3 Schematic representation of characteristic lines in spacetime. If \(\xi = t\) then \(\xi_t = 1\) and all eigenvalues are \(\frac{dt}{dt^2} = \xi_t = 1\). .......................................................... 108

4.4 Region of stability of fourth-order four-stages explicit Runge-Kutta scheme (shaded area) and circle of radius \(R = 1.39\) centered at \(z = -1.39\) (green dashed line). .............................................. 113

5.1 Spacetime mesh for one-dimensional periodic semi-infinite piston. .......................................................... 122

5.2 Pressure contours for one-dimensional periodic semi-infinite piston: central-difference (left) and upwind (right). .......................................................... 123

5.3 Comparison of central-difference and upwind results to piston theory .......................................................... 123

5.4 Spacetime mesh (left) used on moving one-dimensional finite piston with a sharp change of direction, and pressure contours plot (right). .......................................................... 124

5.5 Comparison of pressure distribution throughout time at a location \(x_0\) between central-difference and upwind for piston with sharp change of direction. .......................................................... 125

5.6 Sod’s shock tube problem layout with initial conditions \((t = 0)\) at both sides of the membrane, left \(L\) and right \(R\). .......................................................... 126

5.7 Sod’s shock tube problem: comparison of density (top), pressure (center) and Mach (bottom) distributions at time \(t = 0.2\) between theory and CFD results via three different stencils (JST, Van Leer and Roe). .......................................................... 127

5.8 Pressure contours plot for two-dimensional isentropic Euler (inviscid) vortex transport problem: initial solution (top), JST (second from top), Van Leer (second from bottom) and Roe (bottom). .......................................................... 129

5.9 Pressure contours plot comparison between theoretical solution and CFD results for two-dimensional isentropic Euler (inviscid) vortex transport problem at \(t = 10\): JST (top), Van Leer (center) and Roe (bottom). .......................................................... 130

5.10 Example of spacetime geometry for pitching NACA 0012. Meshes are constructed by stacking up two-dimensional grid planes in the \(t\) direction. An O-grid is used in the inviscid case (left) and a C-grid is used in the viscous case (right). .......................................................... 131

5.11 \(C_p\) distribution plots for pitching NACA 0012 with amplitude \(\Delta \alpha = 2.41\) deg at \(M_\infty = 0.6\), \(\alpha_0 = 2.89\) deg and \(k = \frac{\alpha_0 \omega_c}{2U_\infty} = 0.0808\). Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 1. .......................................................... 133

5.12 \(C_N\) (left) and \(C_m\) (right) coefficients for pitching NACA 0012 with amplitude \(\Delta \alpha = 2.41\) deg at \(M_\infty = 0.6\), \(\alpha_0 = 2.89\) deg and \(k = \frac{\alpha_0 \omega_c}{2U_\infty} = 0.0808\). Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 1. .......................................................... 134

5.13 \(C_p\) distribution plots for pitching NACA 0012 with amplitude \(\Delta \alpha = 2.44\) deg at \(M_\infty = 0.6\), \(\alpha_0 = 4.86\) deg and \(k = \frac{\alpha_0 \omega_c}{2U_\infty} = 0.0810\). Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 3. Continues in Figure 5.14 .......................................................... 135

5.14 Continuation from Figure 5.13. \(C_p\) distribution plots for pitching NACA 0012 with amplitude \(\Delta \alpha = 2.44\) deg at \(M_\infty = 0.6\), \(\alpha_0 = 4.86\) deg and \(k = \frac{\alpha_0 \omega_c}{2U_\infty} = 0.0810\). Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 3. .......................................................... 136
<table>
<thead>
<tr>
<th>Page</th>
<th>Content</th>
</tr>
</thead>
</table>
|5.15| $C_N$ (left) and $C_m$ (right) coefficients for pitching NACA 0012 with amplitude $\Delta \alpha = 2.44$ deg at $M_\infty = 0.6$, $\alpha_0 = 4.86$ deg and $k = \frac{\alpha_0}{\pi M_\infty} = 0.0810$. Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 5. 
|5.16| $C_p$ distribution plots for pitching NACA 0012 with amplitude $\Delta \alpha = 2.51$ deg at $M_\infty = 0.755$, $\alpha_0 = 0.016$ deg and $k = \frac{\alpha_0}{\pi M_\infty} = 0.0814$. Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 5. Continues in Figure 5.17. 
|5.17| Continuation from Figure 5.16. $C_p$ distribution plots for pitching NACA 0012 with amplitude $\Delta \alpha = 2.51$ deg at $M_\infty = 0.755$, $\alpha_0 = 0.016$ deg and $k = \frac{\alpha_0}{\pi M_\infty} = 0.0814$. Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 5. 
|5.18| $C_N$ (left) and $C_m$ (right) coefficients for pitching NACA 0012 with amplitude $\Delta \alpha = 2.51$ deg at $M_\infty = 0.755$, $\alpha_0 = 0.016$ deg and $k = \frac{\alpha_0}{\pi M_\infty} = 0.0814$. Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 5. 
|5.19| $C_p$ distributions for pitching NACA 0012 at $M_\infty = 0.85$ for $\omega t = 30, 120, 210, 300$ deg. 
|5.20| Periodic pitching NACA 0012 aerofoil ($\Delta \alpha = 2.51$ deg, $k = 0.1229$) in a freestream flow at $M_\infty = 0.5$, $Re_\infty = 10^6$ and $\alpha_0 = 9$ deg at phase angles $\omega t = 0, 60, 150, 270$ deg: velocity contours (left) and $C_p$ distributions (right). 
|5.21| $L^2$-norms of density $\rho$ residuals for periodic pitching NACA 0012 aerofoil ($\Delta \alpha = 2.51$ deg, $k = 0.1229$) in a freestream flow at $M_\infty = 0.5$, $Re_\infty = 10^6$ and $\alpha_0 = 9$ deg. 
|5.22| $C_p$ distributions for highly unsteady non-periodic pitching motion of a NACA 0012 aerofoil with amplitude $\Delta \alpha = 3$ deg at $M_\infty = 0.6$, $\alpha_0 = 5$ deg and $k = \frac{\alpha_0}{\pi M_\infty} = 0.25$. Comparison of JST and Van Leer solutions. 
|5.23| Vorticity contours at several time levels during the highly unsteady non-periodic pitching motion of a NACA 0012 aerofoil with amplitude $\Delta \alpha = 3$ deg at $M_\infty = 0.6$, $\alpha_0 = 5$ deg and $k = \frac{\alpha_0}{\pi M_\infty} = 0.25$. 
|5.24| Spacetime mesh for simple flap deflection on a NACA 0012 aerofoil flying at $M_\infty = 0.7$. 
|5.25| Pressure contour plots before the simple flap deflection ($\delta_F = 13.5$ deg, $t = \frac{\alpha_0}{\pi M_\infty} \approx 0.295$) on NACA 0012 flying at $M_\infty = 0.7$: JST (top left), CSUT (top right), Van Leer (bottom left) and Roe (bottom right). 
|5.26| Pressure contour plots during the simple flap deflection ($\delta_F = 13.5$ deg, $t = \frac{\alpha_0}{\pi M_\infty} \approx 0.295$) on NACA 0012 flying at $M_\infty = 0.7$: JST (top left), CSUT (top right), Van Leer (bottom left) and Roe (bottom right). 
|5.27| Pressure contour plots after the simple flap deflection ($\delta_F = 13.5$ deg, $t = \frac{\alpha_0}{\pi M_\infty} \approx 0.295$) on NACA 0012 flying at $M_\infty = 0.7$: JST (top left), CSUT (top right), Van Leer (bottom left) and Roe (bottom right). 
|5.28| $C_p$ distributions before, during and after the flap deflection ($\delta_F = 13.5$ deg, $t = \frac{\alpha_0}{\pi M_\infty} \approx 0.295$) corresponding to those in Figures 5.25, 5.26 and 5.27, respectively, for NACA 0012 flying at $M_\infty = 0.7$. 
|5.29| $L^2$-norms of density residuals for simple flap deflection ($\delta_F = 13.5$ deg, $t = \frac{\alpha_0}{\pi M_\infty} \approx 0.295$) on a NACA 0012 aerofoil flying at $M_\infty = 0.7$. 
|5.30| Pressure contour plots during the simple flap deflection ($\delta_F = 13.5$ deg, $t = \frac{\alpha_0}{\pi M_\infty} \approx 0.295$) on NACA 0012 flying at $M_\infty = 0.7$ via JST: without artificial dissipation in time (left) and with artificial dissipation in time (right). 
|5.31| $C_p$ distributions during the flap deflection ($\delta_F = 13.5$ deg, $t = \frac{\alpha_0}{\pi M_\infty} \approx 0.295$) corresponding to the time-level depicted in Figure 5.30 for NACA 0012 flying at $M_\infty = 0.7$. 
|5.32| Spacetime mesh for a slat and flap deployment on a RAE 2815 aerofoil flying at $M_\infty = 0.23$. 

---

xvi
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.33</td>
<td>Pressure contours plots with velocity streamlines for slat and flap deployment on a RAE 2815 aerofoil flying at $M_\infty = 0.23$: JST (left), Van Leer (centre) and CSUT (right).</td>
</tr>
<tr>
<td>5.34</td>
<td>Convergence residuals for a slat and flap deployment on a RAE 2815 aerofoil flying at $M_\infty = 0.23$.</td>
</tr>
<tr>
<td>5.35</td>
<td>Spacetime mesh for spoiler deployment case.</td>
</tr>
<tr>
<td>5.36</td>
<td>Pressure contours for spoiler deployment on a NACA 0012 aerofoil flying at $M_\infty = 0.25$: JST (left), Van Leer (centre) and CSUT (right).</td>
</tr>
<tr>
<td>5.37</td>
<td>Convergence residuals for spoiler deployment on a NACA 0012 aerofoil flying at $M_\infty = 0.25$.</td>
</tr>
<tr>
<td>5.38</td>
<td>Spoiler configuration (between $0.7c$ and $0.8c$) on a NACA 0012 aerofoil flying at $M_\infty = 0.25$.</td>
</tr>
<tr>
<td>5.39</td>
<td>Spacetime mesh and geometry of the spoiler used in the investigation of the adverse lift.</td>
</tr>
<tr>
<td>5.40</td>
<td>Pressure contours for spoiler deployment in investigation of adverse lift at non-dimensional times $\hat{t} = \frac{U_\infty}{c} = 0, 1.1, 2.2, 3.3, 4.4$ and $10$: JST (left), Van Leer (centre) and Roe (right).</td>
</tr>
<tr>
<td>5.41</td>
<td>Adverse lift on rapidly deploying spoiler ($\frac{U_\infty}{c} = 4.4$ and $\delta_S = 90$ deg) on a NACA 0012 aerofoil flying at $M_\infty = 0.25$ and $\alpha = 0$ deg.</td>
</tr>
<tr>
<td>5.42</td>
<td>Pressure contours for spoiler deployment in investigation of adverse lift at non-dimensional times $t = \frac{U_\infty}{c} = 0, 1.1, 2.2, 3.3, 4.4$ and $10$: JST (left), Van Leer (centre) and Roe (right).</td>
</tr>
<tr>
<td>5.43</td>
<td>Spacetime mesh for landing case, i.e. aerofoil with a slat and flap deployment followed by an increase in its angle of incidence and a spoiler deployment which, in turn, decreases the incidence, all of which happens while approaching to the ground.</td>
</tr>
<tr>
<td>5.44</td>
<td>Pressure contours and streamlines for landing case in a freestream flow at $M_\infty = 0.15$: (a) JST (left), (b) Van Leer (centre) and (c) CSUT (right). Continues in Figure 5.45.</td>
</tr>
<tr>
<td>5.45</td>
<td>Continuation from Figure 5.44. Pressure contours and streamlines for landing case in a freestream flow at $M_\infty = 0.15$: (a) JST (left), (b) Van Leer (centre) and (c) CSUT (right).</td>
</tr>
<tr>
<td>5.46</td>
<td>History of convergence residuals for landing case in a freestream flow at $M_\infty = 0.15$.</td>
</tr>
<tr>
<td>5.47</td>
<td>Spacetime mesh for NACA 0012 aerofoils flying in opposite directions.</td>
</tr>
<tr>
<td>5.48</td>
<td>Pressure contours for aerofoils flying in opposite directions at: (a) $M_\infty = 0.8$ (left) and (b) $M_\infty = 1.6$ (right).</td>
</tr>
<tr>
<td>5.49</td>
<td>History of convergence residuals for NACA 0012 aerofoils flying in opposite directions in a freestream flow at $M_\infty = 0.8$ and $M_\infty = 1.6$.</td>
</tr>
<tr>
<td>5.50</td>
<td>Spacetime geometry for rotor-stator problem. The mesh at $t = \text{constant}$ boundaries is provided.</td>
</tr>
<tr>
<td>5.51</td>
<td>Pressure contours plots for rotor-stator problem at different time-levels.</td>
</tr>
<tr>
<td>A.1</td>
<td>One dimensional representation of particles and mesh motion for Lagrangian, Eulerian and ALE formulations. Extracted from Donea et al.</td>
</tr>
<tr>
<td>B.1</td>
<td>Sample fluid property in a turbulent flow. The total value can be decomposed into a mean value and a fluctuating part.</td>
</tr>
<tr>
<td>C.1</td>
<td>Comparison between the standard and alternative definitions of the modified vorticity $\Omega$, equations (C.5) and (C.17). As proposed by Allmaras et al.</td>
</tr>
</tbody>
</table>
List of Figures

G.1 Cp distribution plots for pitching NACA 0012 with amplitude $\Delta \alpha = 2.41$ deg at $M_{\infty} = 0.6$, $\alpha_0 = 2.89$ deg and $k = \frac{\omega c}{U_{\infty}} = 0.0808$. Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 1. Continues in Figure G.2 ........................................ 213

G.2 Continuation from Figure G.1 Cp distribution plots for pitching NACA 0012 with amplitude $\Delta \alpha = 2.41$ deg at $M_{\infty} = 0.6$, $\alpha_0 = 2.89$ deg and $k = \frac{\omega c}{U_{\infty}} = 0.0808$. Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 1. Continues in Figure G.3 ........................................ 214

G.3 Continuation from Figure G.2 Cp distribution plots for pitching NACA 0012 with amplitude $\Delta \alpha = 2.41$ deg at $M_{\infty} = 0.6$, $\alpha_0 = 2.89$ deg and $k = \frac{\omega c}{U_{\infty}} = 0.0808$. Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 1. Continues in Figure G.4 ........................................ 215

G.4 Continuation from Figure G.3 Cp distribution plots for pitching NACA 0012 with amplitude $\Delta \alpha = 2.41$ deg at $M_{\infty} = 0.6$, $\alpha_0 = 2.89$ deg and $k = \frac{\omega c}{U_{\infty}} = 0.0808$. Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 1. Continues in Figure G.5 ........................................ 216

G.5 Continuation from Figure G.4 Cp distribution plots for pitching NACA 0012 with amplitude $\Delta \alpha = 2.41$ deg at $M_{\infty} = 0.6$, $\alpha_0 = 2.89$ deg and $k = \frac{\omega c}{U_{\infty}} = 0.0808$. Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 1. Continues in Figure G.6 ........................................ 217

G.6 Continuation from Figure G.5 Cp distribution plots for pitching NACA 0012 with amplitude $\Delta \alpha = 2.41$ deg at $M_{\infty} = 0.6$, $\alpha_0 = 2.89$ deg and $k = \frac{\omega c}{U_{\infty}} = 0.0808$. Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 1. ........................................ 218

H.1 Workflow for the simulation of aerodynamics problems in the spacetime presented here .......................................................... 219

H.2 Flowchart of the updating mechanism at each pseudo-time step .................. 222
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Main strengths and weaknesses of the spacetime framework for the solution of unsteady aerodynamics problems with complex boundary motions. Potential solutions are given for each of the weaknesses.</td>
<td>23</td>
</tr>
<tr>
<td>4.1</td>
<td>Calculation of the CFL condition in spacetime based on maximum convective and diffusive pseudo-time step sizes, $\Delta t^<em>_c$ and $\Delta t^</em>_d$, for different numerical schemes.</td>
<td>120</td>
</tr>
<tr>
<td>B.1</td>
<td>Non-dimensionalisation used in RANS equations</td>
<td>190</td>
</tr>
<tr>
<td>C.1</td>
<td>Constant values used in Spalart-Allmaras one equation turbulence model</td>
<td>198</td>
</tr>
<tr>
<td>C.2</td>
<td>Non-dimensionalisation used for turbulent eddy-viscosity</td>
<td>199</td>
</tr>
</tbody>
</table>
List of Symbols

Symbols represent scalar quantities and boldface symbols represent vector and tensor quantities. Reference variables are represented by a zero subscript which are then used to form the non-dimensional quantities.

### Roman Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>Speed of sound</td>
</tr>
<tr>
<td>$a_{\infty}$</td>
<td>Freestream speed of sound</td>
</tr>
<tr>
<td>$A_x$, $A_y$</td>
<td>Projection of face (spacetime) area perpendicular to $x$ and $y$ directions respectively</td>
</tr>
<tr>
<td>$A_t$</td>
<td>Projection of face spacetime area perpendicular to $t$ direction</td>
</tr>
<tr>
<td>$A_{xx}$, $A_{xy}$, $A_{yy}$</td>
<td>Matrices of linearised inviscid fluxes</td>
</tr>
<tr>
<td>$\tilde{A}_n$</td>
<td>Roe matrix for the calculation of fluxes at a cell interface</td>
</tr>
<tr>
<td>$B$, $B'$</td>
<td>Bases of vectorial spaces</td>
</tr>
<tr>
<td>$B_x$, $B_y$</td>
<td>Matrices of linearised inviscid fluxes in $x$ and $y$ directions</td>
</tr>
<tr>
<td>$B_\xi$, $B_\eta$, $B_\zeta$</td>
<td>Matrices of linearised inviscid fluxes in $\xi$, $\eta$ and $\zeta$ directions</td>
</tr>
<tr>
<td>$B_{x\alpha}^{\xi\beta}$, $B_{y\alpha}^{\xi\beta}$, $B_{y\alpha}^{\eta\beta}$</td>
<td>Matrices of linearised viscous fluxes in Cartesian coordinates</td>
</tr>
<tr>
<td>$E_{\alpha\beta}$</td>
<td>Matrices of linearised viscous fluxes for the momentum equations in generalised coordinates, where $\alpha, \beta = \xi, \eta, \zeta$</td>
</tr>
<tr>
<td>$c_{b1}$, $c_{b2}$</td>
<td>Closure constants of the Spalart-Allmaras turbulence model</td>
</tr>
<tr>
<td>$c_{n1}$</td>
<td>Closure constant of the Spalart-Allmaras turbulence model</td>
</tr>
<tr>
<td>$e_p$</td>
<td>Specific heat capacity at constant pressure</td>
</tr>
<tr>
<td>$c_{t1}$, $c_{t2}$, $c_{t3}$, $c_{t4}$</td>
<td>Closure constants of the Spalart-Allmaras turbulence model</td>
</tr>
<tr>
<td>$e_v$</td>
<td>Specific heat capacity at constant volume</td>
</tr>
<tr>
<td>$c_{w1}$, $c_{w2}$, $c_{w3}$</td>
<td>Closure constants of the Spalart-Allmaras turbulence model</td>
</tr>
<tr>
<td>$C$</td>
<td>Update operator of the numerical integration method from solution $U^n$ to $U^{n+1}$</td>
</tr>
<tr>
<td>$C_\theta$</td>
<td>Smoothing switch for face fluxes of Spallart-Allmaras turbulence model</td>
</tr>
<tr>
<td>$C$</td>
<td>Complex space (contains the real space $\mathbb{R}$)</td>
</tr>
<tr>
<td>$C^n$</td>
<td>Class of functions continuous and differentiable $n$ times</td>
</tr>
<tr>
<td>$C^\infty$</td>
<td>Class of functions which have derivatives of all orders (hence, continuity is implied)</td>
</tr>
<tr>
<td>d</td>
<td>Normal distance to solid wall or slip line (in the wake)</td>
</tr>
<tr>
<td>$d_t$</td>
<td>Distance to the trip location in the Spalart-Allmaras turbulence model</td>
</tr>
<tr>
<td>$D_n$</td>
<td>Destruction term of the negative Spalart-Allmaras turbulence model</td>
</tr>
<tr>
<td>$D_k$</td>
<td>Column vector of numerical dissipation at face $k$</td>
</tr>
<tr>
<td>$e$</td>
<td>Internal energy</td>
</tr>
<tr>
<td>$e_{n1}$, $e_t$</td>
<td>Unit vectors in $x_n$ and $x_t$ directions that form a basis of $\mathbb{R}^2$</td>
</tr>
<tr>
<td>$e_{n1}$, $e_{t1}$, $e_{t2}$</td>
<td>Unit vectors in $x_n$, $x_{t1}$, and $x_{t2}$ directions that form a basis of $\mathbb{R}^3$</td>
</tr>
<tr>
<td>$e_{x1}$, $e_{y1}$, $e_{z1}$</td>
<td>Unit vectors in $x$, $y$ and $z$ directions that form a basis of $\mathbb{R}^3$</td>
</tr>
<tr>
<td>$E$</td>
<td>Total energy</td>
</tr>
<tr>
<td>$E_{\infty}$</td>
<td>Freestream total energy</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$f_x, f_y$</td>
<td>Functions used in the estimation of the maximum time-step size for viscous terms in spacetime</td>
</tr>
<tr>
<td>$f_n$</td>
<td>Function of the ratio $\chi$ which modifies the diffusion term in the negative Spalart-Allmaras turbulence model</td>
</tr>
<tr>
<td>$f_i^V$</td>
<td>Tensor notation of volume forces per unit volume in $i$ direction</td>
</tr>
<tr>
<td>$f_{i1}, f_{i2}$</td>
<td>Trip functions used in the Spalart-Allmaras turbulence model</td>
</tr>
<tr>
<td>$f_{i1}$</td>
<td>Ratio between the turbulent eddy viscosity and the Spalart-Allmaras working variable</td>
</tr>
<tr>
<td>$f_{i2}$</td>
<td>Function used in the modified vorticity of the Spalart-Allmaras turbulence model</td>
</tr>
<tr>
<td>$f_w$</td>
<td>Function used in the destruction term of the Spalart-Allmaras turbulence model</td>
</tr>
<tr>
<td>$F$</td>
<td>A generic function of the conserved variables $U$ at one or more time levels</td>
</tr>
<tr>
<td>$F_n$</td>
<td>Column vector of inviscid fluxes in the normal direction calculated with the conserved variables evaluated at both sides of the face, $U^+$ and $U^-$, respectively</td>
</tr>
<tr>
<td>$F_x, F_y$</td>
<td>Column vectors of inviscid fluxes along $x$ and $y$ coordinates in terms of conserved variables $U$</td>
</tr>
<tr>
<td>$F^{x}_n, F^{y}_n$</td>
<td>Column vectors of viscous fluxes along $x$ and $y$ coordinates in terms of conserved variables $U$</td>
</tr>
<tr>
<td>$\tilde{F}_n$</td>
<td>Column vector of one-sided inviscid fluxes along the normal direction to the face in terms of primitive variables $W_n$ (used in flux-vector splitting when $</td>
</tr>
<tr>
<td>$\tilde{F}_n^+, \tilde{F}_n^-$</td>
<td>Column vectors of contributions to the numerical face flux from forward (or positive) and backward (or negative) moving waves in terms of primitive variables evaluated at both sides of the face, $W_n^+$ and $W_n^-$, respectively (used in flux-vector splitting method due to need for a solution to a Riemann problem at the cell faces)</td>
</tr>
<tr>
<td>$F_n$</td>
<td>Column vector of numerical inviscid face fluxes in terms of the normal primitive variables at both sides of the interface, $W_n^+$ and $W_n^-$, which are, in general, different (used in the flux-vector splitting method)</td>
</tr>
<tr>
<td>$g_t$</td>
<td>Closure constant in the trip term of the Spalart-Allmaras turbulence model</td>
</tr>
<tr>
<td>$G$</td>
<td>Amplification factor of the solution between two consecutive steps in a numerical method</td>
</tr>
<tr>
<td>$H$</td>
<td>Spalart-Allmaras fluxes vector</td>
</tr>
<tr>
<td>$H^c, H^d$</td>
<td>Spalart-Allmaras convective and diffusive fluxes vectors, respectively</td>
</tr>
<tr>
<td>$H_{ST}$</td>
<td>Spalart-Allmaras spacetime fluxes vector</td>
</tr>
<tr>
<td>$H^c_{ST}, H^d_{ST}$</td>
<td>Spalart-Allmaras convective and diffusive spacetime fluxes vectors, respectively</td>
</tr>
<tr>
<td>$I$</td>
<td>Unit along the imaginary axis. It is defined as $I = \sqrt{-1} \in \mathbb{C}$</td>
</tr>
<tr>
<td>$\mathbb{I}$</td>
<td>Identity matrix</td>
</tr>
<tr>
<td>$\Im (z)$</td>
<td>Imaginary part of complex number $z \in \mathbb{C}$</td>
</tr>
<tr>
<td>$J$</td>
<td>Determinant of the Jacobian</td>
</tr>
<tr>
<td>$k_m$</td>
<td>Wave number associated with the $m-$th harmonic in a finite Fourier series</td>
</tr>
<tr>
<td>$k_t$</td>
<td>Turbulent kinetic energy per unit volume</td>
</tr>
<tr>
<td>$L_0$</td>
<td>Characteristic/reference length in the problem</td>
</tr>
<tr>
<td>$\hat{l}_i$</td>
<td>Left eigenvectors of Roe matrix $\hat{A}$</td>
</tr>
<tr>
<td>$M$</td>
<td>Matrix of inviscid fluxes</td>
</tr>
<tr>
<td>$M^v$</td>
<td>Matrix of viscous fluxes</td>
</tr>
<tr>
<td>$M_{ST}$</td>
<td>Matrix of inviscid spacetime fluxes</td>
</tr>
<tr>
<td>$M^v_{ST}$</td>
<td>Matrix of viscous spacetime fluxes</td>
</tr>
<tr>
<td>$n$</td>
<td>Unit normal vector</td>
</tr>
<tr>
<td>$n_{ST}$</td>
<td>Spacetime unit normal vector</td>
</tr>
<tr>
<td>$n_x, n_y, n_t$</td>
<td>Unit normal vector components in $x$, $y$ and $t$ directions</td>
</tr>
<tr>
<td>$n_f$</td>
<td>Number of faces at certain cell in the mesh</td>
</tr>
<tr>
<td>$N$</td>
<td>Number of physical dimensions</td>
</tr>
<tr>
<td>$N_i$</td>
<td>Number of neighbours at cell $i$</td>
</tr>
<tr>
<td>$p$</td>
<td>Pressure</td>
</tr>
<tr>
<td>$P_n$</td>
<td>Production term of the negative Spalart-Allmaras turbulence model</td>
</tr>
<tr>
<td>$\tilde{P}_c$</td>
<td>Augmented Prandtl number equal to $\tilde{P}<em>c = \mu Pr</em>{fr}^{\tilde{P}}$</td>
</tr>
<tr>
<td>$P$</td>
<td>Rotation from local coordinate system at each face $(x_n, x_{t1}, x_{t2})$ to global one $(t, x, y)$</td>
</tr>
<tr>
<td>$\tilde{P}$</td>
<td>Transformation from normal primitive $W_n$ variables to conserved variables $U$</td>
</tr>
<tr>
<td>$q_i$</td>
<td>Heat fluxes in tensor notation</td>
</tr>
<tr>
<td>$Q$</td>
<td>Source term of the Spalart-Allmaras one-equation turbulence model</td>
</tr>
<tr>
<td>$Q_H$</td>
<td>Heat fluxes due to chemical reactions</td>
</tr>
<tr>
<td>$Q$</td>
<td>Matrix of right eigenvalues as columns for diagonalization</td>
</tr>
<tr>
<td>$\hat{r}$</td>
<td>Independent variable of function $\zeta(\hat{r})$ in the destruction term of the Spalart-Allmaras turbulence model</td>
</tr>
</tbody>
</table>
## List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_i$</td>
<td>Monitor at cell $i$ used in the definition of the slope limiters</td>
</tr>
<tr>
<td>$r_i$</td>
<td>Right eigenvectors of Roe matrix $\mathbf{A}$</td>
</tr>
<tr>
<td>$\mathbf{R}$</td>
<td>Column vector of residuals</td>
</tr>
<tr>
<td>$\mathbf{R}_{ST}$</td>
<td>Column vector of spacetime residuals</td>
</tr>
<tr>
<td>$R^{+, R^{-}, R^{0}}$</td>
<td>Locally one-dimensional Riemann invariants</td>
</tr>
<tr>
<td>$R_q$</td>
<td>Specific gas constant</td>
</tr>
<tr>
<td>$\mathbb{R}$</td>
<td>Real space</td>
</tr>
<tr>
<td>$\mathbb{R}^n$</td>
<td>Real coordinate space of $n$ dimensions</td>
</tr>
<tr>
<td>$\Re(z)$</td>
<td>Real part of complex number $z \in \mathbb{C}$</td>
</tr>
<tr>
<td>$s$</td>
<td>Entropy</td>
</tr>
<tr>
<td>$s_2, s_4$</td>
<td>Scaling factors in the JST dissipation model</td>
</tr>
<tr>
<td>$S_{ij}$</td>
<td>Strain rate tensor</td>
</tr>
<tr>
<td>$t$</td>
<td>Time, time coordinate</td>
</tr>
<tr>
<td>$t^*$</td>
<td>Pseudo-time</td>
</tr>
<tr>
<td>$t_0$</td>
<td>Characteristic/reference time in the problem</td>
</tr>
<tr>
<td>$\mathbf{t}$</td>
<td>Tangential unit normal vector</td>
</tr>
<tr>
<td>$T$</td>
<td>Temperature</td>
</tr>
<tr>
<td>$T_\infty$</td>
<td>Freestream temperature</td>
</tr>
<tr>
<td>$\mathbf{T}_{U \rightarrow W}$</td>
<td>Matrix associated with transformation from conserved variables $\mathbf{U}$ to primitive variables $\mathbf{W}$</td>
</tr>
<tr>
<td>$\mathbf{T}_P$</td>
<td>Matrix associated with transformation $\mathbf{P}$</td>
</tr>
<tr>
<td>$\mathbf{T}_{\hat{P}}$</td>
<td>Matrix associated with transformation $\mathbf{\hat{P}}$</td>
</tr>
<tr>
<td>$U_\infty$</td>
<td>Freestream velocity</td>
</tr>
<tr>
<td>$u, v$</td>
<td>Components of fluid velocity in $x$ and $y$ directions</td>
</tr>
<tr>
<td>$u_c, v_c$</td>
<td>Components of a control volume boundary velocity in $x$ and $y$ directions</td>
</tr>
<tr>
<td>$u_n, u_t$</td>
<td>Components of fluid velocity in normal $x_n$ and tangential $x_t$ directions in $\mathbb{R}^2$</td>
</tr>
<tr>
<td>$u_n, u_{t_1}, u_{t_2}$</td>
<td>Components of fluid velocity in normal $x_n$ and tangential $x_{t_1}, x_{t_2}$ directions in $\mathbb{R}^3$</td>
</tr>
<tr>
<td>$u_i$</td>
<td>Components of fluid velocity in tensor notation</td>
</tr>
<tr>
<td>$\mathbf{U}_w, \mathbf{V}_w$</td>
<td>Solid wall velocity</td>
</tr>
<tr>
<td>$\mathbf{U}, \mathbf{V}$</td>
<td>Column vector of conserved variables</td>
</tr>
<tr>
<td>$\mathbf{U}^+, \mathbf{U}^-$</td>
<td>Column vector of conserved variables at both sides of an interface</td>
</tr>
<tr>
<td>$\mathbf{V}$</td>
<td>Fluid volume or cell volume</td>
</tr>
<tr>
<td>$V_c$</td>
<td>Control volume</td>
</tr>
<tr>
<td>$V_{ST}$</td>
<td>Spacetime fluid or cell volume</td>
</tr>
<tr>
<td>$\mathbf{v}$</td>
<td>Fluid velocity vector</td>
</tr>
<tr>
<td>$\mathbf{v}_c$</td>
<td>Control volume boundary velocity vector</td>
</tr>
<tr>
<td>$\mathbf{v}_{ST}$</td>
<td>Fluid spacetime velocity vector</td>
</tr>
<tr>
<td>$\mathbf{V}_w$</td>
<td>Solid wall velocity vector</td>
</tr>
<tr>
<td>$\mathbf{W}$</td>
<td>Column vector of primitive variables</td>
</tr>
<tr>
<td>$\mathbf{W}^+, \mathbf{W}^-$</td>
<td>Column vectors of primitive variables evaluated on the side of the face where the forward (or positive) and backward (or negative) moving waves come from, respectively (used in the flux-vector splitting method)</td>
</tr>
<tr>
<td>$x, y$</td>
<td>Cartesian spatial coordinates</td>
</tr>
<tr>
<td>$x_n, x_t$</td>
<td>Normal and tangential coordinates in $\mathbb{R}^2$</td>
</tr>
<tr>
<td>$x_n, x_{t_1}, x_{t_2}$</td>
<td>Normal and tangential coordinates in $\mathbb{R}^3$</td>
</tr>
<tr>
<td>$x_i$</td>
<td>Cartesian spatial coordinates in tensor notation</td>
</tr>
<tr>
<td>$\mathbf{x}$</td>
<td>Spatial location</td>
</tr>
<tr>
<td>$\mathbf{x}_{ST}$</td>
<td>Spacetime location</td>
</tr>
<tr>
<td>$z$</td>
<td>Generic complex number, i.e. $z \in \mathbb{C}$</td>
</tr>
</tbody>
</table>

## Greek Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_0$</td>
<td>Mean angle of attack in a sinusoidal pitching motion</td>
</tr>
<tr>
<td>$\alpha_i$</td>
<td>Coefficients for the calculation of $\Delta \mathbf{U}$ in the Roe flux-difference method</td>
</tr>
<tr>
<td>$\alpha_{ij}$</td>
<td>Coefficients of a Runge-Kutta method defined in their Butcher tableau</td>
</tr>
<tr>
<td>$\beta_i$</td>
<td>Weights of a Runge-Kutta method defined in their Butcher tableau</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$\chi$</td>
<td>Ratio of Spalart-Allmaras working variable and molecular viscosity</td>
</tr>
<tr>
<td>$\delta_{ij}$</td>
<td>Kronecker delta</td>
</tr>
<tr>
<td>$\delta \phi$</td>
<td>Infinitesimal change of quantity $\phi$</td>
</tr>
<tr>
<td>$\partial V$</td>
<td>Closed surface which represents the boundary of volume $V$</td>
</tr>
<tr>
<td>$\Delta \xi, \Delta \eta, \Delta \zeta$</td>
<td>Grid size along generalised coordinates $\xi$, $\eta$ and $\zeta$</td>
</tr>
<tr>
<td>$\Delta x$</td>
<td>Locally one-dimensional grid size</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>Physical time step size</td>
</tr>
<tr>
<td>$\Delta t^*$</td>
<td>Pseudo time step size</td>
</tr>
<tr>
<td>$(\Delta t^<em><em>c)</em>{\text{UW}}, (\Delta t^</em><em>d)</em>{\text{UW}}$</td>
<td>Maximum convective and diffusive time-steps, respectively, for an upwind discretisation</td>
</tr>
<tr>
<td>$(\Delta t^<em><em>c)</em>{\text{CD}}, (\Delta t^</em><em>d)</em>{\text{CD}}$</td>
<td>Maximum convective and diffusive time-steps, respectively, for a central-difference discretisation</td>
</tr>
<tr>
<td>$\Delta \alpha$</td>
<td>Amplitude of oscillation in a sinusoidal pitching motion</td>
</tr>
<tr>
<td>$\Delta U$</td>
<td>Jumps of the conserved variables at an interface in the Roe flux-difference method</td>
</tr>
<tr>
<td>$\Delta F$</td>
<td>Jumps of the normal fluxes at an interface in the Roe flux-difference method</td>
</tr>
<tr>
<td>$\varepsilon^{(2)}$</td>
<td>JST dissipation model switch defined on cells</td>
</tr>
<tr>
<td>$\varepsilon^{(2)}, \varepsilon^{(4)}$</td>
<td>JST dissipation model switches defined on faces</td>
</tr>
<tr>
<td>$\gamma_i$</td>
<td>Nodes of a Runge-Kutta method defined in their Butcher tableau</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Specific heat capacity ratio</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>Thermal conductivity</td>
</tr>
<tr>
<td>$\hat{\kappa}$</td>
<td>Von Karman constant</td>
</tr>
<tr>
<td>$\kappa^{(2)}, \kappa^{(4)}$</td>
<td>JST dissipation model constants</td>
</tr>
<tr>
<td>$\lambda_i$</td>
<td>Eigenvalues</td>
</tr>
<tr>
<td>$\bar{\lambda} \Delta t^*$</td>
<td>Fourier symbol of the spatial (or spacetime) discretisation</td>
</tr>
<tr>
<td>$\Lambda$</td>
<td>Diagonal matrix</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Dynamic viscosity</td>
</tr>
<tr>
<td>$\mu_\infty$</td>
<td>Freestream dynamic viscosity</td>
</tr>
<tr>
<td>$\mu_t$</td>
<td>Turbulent eddy viscosity</td>
</tr>
<tr>
<td>$\hat{\mu}$</td>
<td>Augmented kinematic viscosity equal to $\mu + \mu_t$</td>
</tr>
<tr>
<td>$\nu$</td>
<td>Kinematic viscosity</td>
</tr>
<tr>
<td>$\nu_\infty$</td>
<td>Freestream kinematic viscosity</td>
</tr>
<tr>
<td>$\nu_t$</td>
<td>Turbulent kinematic eddy viscosity</td>
</tr>
<tr>
<td>$\bar{\mu}, \bar{\nu}$</td>
<td>Working variables for Spalart-Allmaras turbulence model</td>
</tr>
<tr>
<td>$\Omega$</td>
<td>Vorticity</td>
</tr>
<tr>
<td>$</td>
<td>\Omega</td>
</tr>
<tr>
<td>$\Omega$</td>
<td>Modified vorticity of the Spalart-Allmaras turbulence model</td>
</tr>
<tr>
<td>$\bar{\Omega}$</td>
<td>Extra term in the modified vorticity of the Spalart-Allmaras turbulence model</td>
</tr>
<tr>
<td>$\omega_t$</td>
<td>Vorticity at the trip in the Spalart-Allmaras turbulence model</td>
</tr>
<tr>
<td>$\phi$</td>
<td>Generic fluid property (e.g. velocity, density, or pressure)</td>
</tr>
<tr>
<td>$\phi_m$</td>
<td>Phase angle associated with the $m$-th harmonic in a finite Fourier series ($\phi_m = k_m \Delta x$)</td>
</tr>
<tr>
<td>$\Psi$</td>
<td>Scaling for the calculation of the time-step for viscous terms in spacetime</td>
</tr>
<tr>
<td>$\varphi_x, \varphi_y, \varphi_t$</td>
<td>Slope limiters on the $x$, $y$ and $t$ directions, respectively</td>
</tr>
<tr>
<td>$\varphi^{\text{VL}}$</td>
<td>Slope limiter proposed by Van Leer</td>
</tr>
<tr>
<td>$\Phi$</td>
<td>Matrix of slope limiters</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Density</td>
</tr>
<tr>
<td>$\rho_\infty$</td>
<td>Freestream Density</td>
</tr>
<tr>
<td>$\sigma_k$</td>
<td>Closure constant of RANS formulation</td>
</tr>
<tr>
<td>$\sigma_{\text{SA}}$</td>
<td>Closure constant of the Spalart-Allmaras turbulence model</td>
</tr>
<tr>
<td>$\sigma_{ij}$</td>
<td>Total or Favre-averaged shear stress tensor</td>
</tr>
<tr>
<td>$\Upsilon_i$</td>
<td>Spectral radius at cell $i$</td>
</tr>
<tr>
<td>$\varsigma$</td>
<td>Function used in the destruction term of the Spalart-Allmaras turbulence model</td>
</tr>
<tr>
<td>$\theta_k$</td>
<td>Spectral radius at face $k$</td>
</tr>
<tr>
<td>$\tau_{ij}$</td>
<td>Viscous shear stress tensor</td>
</tr>
<tr>
<td>$\xi, \eta, \zeta$</td>
<td>General curvilinear coordinates</td>
</tr>
<tr>
<td>$\Xi$</td>
<td>Factor to account for the stretching of the grid in the JST dissipation model</td>
</tr>
</tbody>
</table>
List of Symbols

Superscripts

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bar{\phi}$</td>
<td>Reynolds averaged or time averaged quantity</td>
</tr>
<tr>
<td>$\phi'$</td>
<td>Reynolds fluctuating or time fluctuating quantity</td>
</tr>
<tr>
<td>$\tilde{\phi}$</td>
<td>Favre or density weighted averaged quantity</td>
</tr>
<tr>
<td>$\phi''$</td>
<td>Favre fluctuating quantity</td>
</tr>
<tr>
<td>$\phi^*$</td>
<td>Non-dimensional quantity</td>
</tr>
<tr>
<td>$\phi^+, \phi^-$</td>
<td>Value of a fluid property evaluated on the side of the face where the forward (or positive) and backward (or negative) moving waves come from, respectively (used in the flux-vector splitting method)</td>
</tr>
</tbody>
</table>

Subscripts

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi_\infty$</td>
<td>Fluid property at the freestream</td>
</tr>
<tr>
<td>$\phi_b$</td>
<td>Fluid property at a boundary</td>
</tr>
<tr>
<td>$\phi_h$</td>
<td>Fluid property at a halo or ghost cell</td>
</tr>
<tr>
<td>$\phi_0$</td>
<td>Fluid property at a particular cell, e.g. cell at a boundary</td>
</tr>
<tr>
<td>$\phi_{ST}$</td>
<td>Spacetime fluid property</td>
</tr>
<tr>
<td>$\phi_{l(k)}, \phi_{r(k)}$</td>
<td>Fluid property on the left and right neighbours of face $k$, respectively</td>
</tr>
</tbody>
</table>

Dimensionless Numbers

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_p$</td>
<td>Pressure coefficient</td>
<td>$\frac{p-p_\infty}{\frac{1}{2} \rho_{\infty} U_\infty^2}$</td>
</tr>
<tr>
<td>$k$</td>
<td>Reduced frequency</td>
<td>$\frac{\omega_c}{2U_\infty}$</td>
</tr>
<tr>
<td>$M_\infty$</td>
<td>Freestream Mach number</td>
<td>$\frac{U_\infty}{a_\infty}$</td>
</tr>
<tr>
<td>$Pr$</td>
<td>Prandtl number</td>
<td>$\frac{\mu}{\kappa}$</td>
</tr>
<tr>
<td>$Pr_t$</td>
<td>Turbulent Prandtl number</td>
<td>$\frac{\text{momentum transfer}}{\text{heat transfer}}$</td>
</tr>
<tr>
<td>$Re_\infty$</td>
<td>Freestream Reynolds Number</td>
<td>$\frac{\rho_{\infty} U_\infty L_0}{\mu_{\infty}}$</td>
</tr>
</tbody>
</table>

Abbreviations and Acronyms

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALE</td>
<td>Arbitrary Lagrangian-Eulerian formulation</td>
</tr>
<tr>
<td>AUSM</td>
<td>Advection Upwind Split Method</td>
</tr>
<tr>
<td>CE/SE</td>
<td>Conservation Element and Solution Element</td>
</tr>
<tr>
<td>CFD</td>
<td>Computational Fluid Dynamics</td>
</tr>
<tr>
<td>CFL</td>
<td>Courant-Friedrichs-Lewy number</td>
</tr>
<tr>
<td>CSUT</td>
<td>Hybrid formulation: Central-difference in Space, Upwind in Time</td>
</tr>
<tr>
<td>DES</td>
<td>Detached-Eddy Simulation</td>
</tr>
<tr>
<td>DNS</td>
<td>Direct Numerical Simulation</td>
</tr>
<tr>
<td>DG</td>
<td>Discontinuous-Galerkin method</td>
</tr>
<tr>
<td>FSI</td>
<td>Fluid-Structure Interaction</td>
</tr>
<tr>
<td>FPM</td>
<td>Finite Points Method</td>
</tr>
<tr>
<td>GCL</td>
<td>Geometric Conservation Law</td>
</tr>
<tr>
<td>JST</td>
<td>Numerical dissipation model named after its authors Jameson-Schmidt-Turkel</td>
</tr>
<tr>
<td>LES</td>
<td>Large-Eddy Simulation</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>MUSCL</td>
<td>Monotonic Upstream-centered Scheme for Conservation Laws</td>
</tr>
<tr>
<td>NASA</td>
<td>National Aeronautics and Space Administration</td>
</tr>
<tr>
<td>RANS</td>
<td>Reynolds-Averaged Navier-Stokes formulation</td>
</tr>
<tr>
<td>RK</td>
<td>Runge-Kutta method</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function</td>
</tr>
<tr>
<td>SA</td>
<td>Spalart-Allmaras one-equation turbulence model</td>
</tr>
<tr>
<td>SPH</td>
<td>Smoothed Particle Hydrodynamics</td>
</tr>
<tr>
<td>ST</td>
<td>Space-Time formulation</td>
</tr>
<tr>
<td>TVD</td>
<td>Total Variation Diminishing</td>
</tr>
<tr>
<td>URANS</td>
<td>Unsteady Reynolds-Averaged Navier-Stokes formulation</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background

There exist thousands of applications which involve fluids, ranging from the routine action of opening a tap to get water through to starting our car engine every time we drive (see for example Figure 1.1). Understanding the nature of these applications and being able to predict their behaviour in any circumstances is the key to making efficient and effective designs. Scientists and engineers have addressed the computerised study of fluids over the last few decades and applications can be found in such diverse fields as medicine, aerospace, marine or oil and gas. Particularly important is the research carried out in aerodynamics for applications within the aerospace industry. Problems of interest range from steady state solutions of simple two-dimensional aerofoils to unsteady simulations of full aircraft models. Although they can sometimes be computationally expensive, steady problems are relatively simple and there exist mature techniques and numerical methods to solve them accurately. On the contrary, unsteady problems, where complex boundary motions or topological changes in the geometry happen, can be very intricate and are still an active area of research. The study of interaction between helicopter rotor-blades and a fuselage constitutes a clear example of the complexity of unsteady aerodynamics. Other common complex problems include store separation, flap and spoiler deployment in take-off and/or landing configurations or the transient process that takes place within an internal combustion engine when valves open/close. Finding accurate and efficient solutions to these problems using the most common CFD solvers remains limited by the capability of existing mesh generation/deformation techniques and interpolation algorithms. A different meshing technology needs to be used depending on the problem under consideration which, inevitably, limits the ability to automate simulations and slows down the design cycle of industrial applications.

Although an integration across four-dimensional space-time domain is required to obtain unsteady solutions of the three-dimensional Navier-Stokes equations, conventional methods have traditionally decoupled this process
into two consecutive and different steps: a finite-volume integration in a three-dimensional space and a finite-difference integration in time. An alternative spacetime framework can be formulated so that both integrations, in space and time, are treated similarly through the use of spacetime finite-volumes, hence effectively solving unsteady problems of dimension $N$ as steady problems of dimension $N + 1$. This implies modifying the fluid equations of motion through the divergence theorem to remove temporal derivatives, which are replaced by temporal fluxes instead. The fully conservative nature in space and time as a consequence of the finite-volume approach in spacetime facilitates the solution of problems where cells appear/disappear between consecutive time levels.

Coupling time and space, and effectively solving both as one, may lead to non-physical behaviour when using a central-difference scheme since information can be propagated backwards in time as a consequence of the temporal stencil being used [1]. The solution at a certain time level may be affected by the solution at later times and this is not physically correct. Understanding the direction in which characteristics of the hyperbolic problem transport disturbances across the spacetime domain, particularly in the time direction where waves are unidirectional, is essential to obtain accurate and meaningful solutions. This is therefore focus of this thesis.
1.2  Arbitrary Lagrangian-Eulerian formulation and geometric conservation law

The use of an arbitrary Lagrangian-Eulerian (ALE) formulation is not necessary in a spacetime formulation of the Navier-Stokes equations since time-accurate problems are effectively treated as steady-state problems of one dimension higher, where the extra dimension represents physical-time. As opposed to this, conventional unsteady CFD methods require the transformation of the mesh at every physical-time step in order to accommodate for solid boundary motions/deformations in the geometry. Therefore, time-accurate problems with mesh motions/deformations have traditionally made intensive use of the ALE formulation. This has often been combined with a geometric conservation law (GCL) as a correction for the inaccurate integration of volumes due to the numerical method used, as explained in Appendix A.

Consider the one-dimensional example in Figure 1.2. The segment $\overline{AB}$ at time $t^n$ shrinks and yields the segment $\overline{AB'}$ at time $t^n + \Delta t$. Assuming constant fluid properties ($\rho(t) = \rho_0$ and $u(t) = u_0$), the discretisation of the conservation of mass, given by equation (A.3) in Appendix A, can be written as

$$\frac{1}{\Delta t} \int_{V(t)} \rho d\Omega - \frac{\rho_0 (x_{B'} - x_A) - \rho_0 (x_B - x_A)}{\Delta t} + \rho_0 \left( \frac{x_{B'} - x_B}{\Delta t} \right) - \rho_0 (u_0 - 0) = 0$$

(1.1)

Here, the use of the ALE formulation is essential to account for the motion of mesh node $B$ with velocity $(u_c)_B = \frac{x_{B'} - x_B}{\Delta t}$. The equivalent formulation of this problem in spacetime yields a two-dimensional problem where the conservation of mass on the spacetime element $\overline{ABB'A'}$ (depicted in blue) leads to (refer to equations
\[ \oint_{ST} \left( \rho n_t + \rho u n_x \right) dS = -\rho_0 (x_B - x_A) + \rho_0 u_0 \Delta t + \rho_0 (x_B - x_{B'}) + \rho_0 (x_{B'} - x_A') - \rho_0 u_0 \Delta t = 0 \] (1.2)

Notice that the spacetime normal vectors at the faces of element \( \overline{ABB'}A' \) are given by (see Figure 1.2)

\[
\begin{align*}
\mathbf{n}_{\overline{AB}} &= \begin{cases} n_t \\ n_x \end{cases} = \begin{cases} -1 \\ 0 \end{cases} \\
\mathbf{n}_{\overline{A'B'}} &= \begin{cases} n_t \\ n_x \end{cases} = \begin{cases} 1 \\ 0 \end{cases} \\
\mathbf{n}_{\overline{AA'}} &= \begin{cases} n_t \\ n_x \end{cases} = \begin{cases} 0 \\ -1 \end{cases} \\
\mathbf{n}_{\overline{BB'}} &= \begin{cases} n_t \\ n_x \end{cases} = \begin{cases} \frac{x_{B'} - x_B}{||BB'||} \\ \frac{\Delta t}{||BB'||} \end{cases}
\end{align*} \] (1.3)

Multiplying equation (1.1) by \( \Delta t \) and re-arranging terms it is straightforward to see that both formulations are equivalent. Bear in mind that unlike the ALE formulation, equation (1.1), no consideration of grid velocities has been necessary in the spacetime formulation where grid velocities are intrinsically zero \( (u_c = 0) \) and the mesh remains constant throughout the whole numerical integration. A similar result can be derived for the conservation of momentum and energy. Moreover, the spacetime formulation of the fluid equations of motion does not need a GCL either because the integration of spacetime volumes is always exact, just like the integration of (spatial) volumes in any steady-state simulation.

### 1.3 Existing methods for moving and deforming meshes

As opposed to the spacetime method, the simulation of unsteady aerodynamic problems with conventional CFD methods is restricted to existing techniques that allow volume meshes to accommodate solid surface motions of any type. These techniques include: radial basis functions, Chimera or overset grids and immersed boundary methods, amongst others. They must be used along with an arbitrary Lagrangian-Eulerian (ALE) formulation of the fluid equations and come with limitations in regard to the type of motions they can cope with. In general, mesh deformation techniques can only deal with small movements if a good quality mesh is to be retained after the deformation process. The fact that no cells can appear or disappear due to a fixed connectivity between cells at two consecutive time levels yields distorted cells with high aspect ratios when large body motions are involved. In such cases, re-meshing, i.e. generating a completely new mesh from the geometry at the current time level would be a suitable solution to the problem of low quality meshes. However, this implies using an interpolation method to relate the fluid variables in the new mesh with those in the previous one, hence making a very intensive usage of available computational resources and a possibly ambiguous interpolation.
Interpolating is not a trivial task, can be non-conservative and often demands a degree of ingenuity. A more detailed description of the existing methods and examples of their applicability are provided in the following sections.

1.3.1 Mesh motion

An automatic mechanism is sought in order to create good quality meshes as an adaptation from previous grids, without the user intervention. One of the simplest and most widely used methods is mesh motion whereby grid nodes move as a consequence of the moving domain boundaries. Their simplicity leads to a low computational cost and the fact that the nodal connectivity list remains unaltered through time marching, i.e. there is a one-to-one mapping between cells of two successive time levels, hence enabling the use of the last computed solution directly without the need for interpolation. It is interesting to look at the analysis and formulation of robust mesh motion techniques implemented by Mavriplis et al. [2, 3, 4] or at the more recent but universal (applicable to any grid type) and parallel approach to mesh motion by Allen [5].

Mesh motion is convenient for problems with small deformations provided that they preserve the quality of the grid, i.e. orthogonality, aspect ratio and smoothness. In the case of large displacements or topological changes in the geometry other techniques such as Chimera or overset grids have proved to work better [6], re-meshing being the optimal although simultaneously the most expensive. Below is a brief introduction to some of the well-known mesh motion techniques: spring analogy, transfinite interpolation, Laplacian smoothing and radial basis functions.

Spring analogy

The spring analogy constitutes one of the main mesh deformation techniques in the literature given its popularity [7] [8] [9] [10]. It compares the mesh to a system of tensional and/or torsional springs whereby a displacement of the boundary forces the movement of interior nodes in order for the system to stay in equilibrium. Hooke’s law is applied to each grid node after the deformation of solid boundaries and the resulting force $F_i$ at node $i$ can be worked out as

$$F_i = \sum_{j=1}^{N_i} k_{ij} (\delta x_j - \delta x_i)$$  \hspace{1cm} (1.4)

where $k_{ij}$ is the stiffness of the spring that connects nodes $i$ and $j$, $\delta x_j$ is the displacement of node $j$ and $N_i$ is the number of neighbours at node $i$. Therefore, the tensional spring analogy is cheap to compute provided that the number of elements in the grid is not extremely large. Although the exclusive use of tensional springs
Chapter 1: Introduction

Figure 1.3: Mesh deformation for rotated flap using (a) tensional spring analogy and (b) linear elasticity analogy. Extracted from Nielsen et al. [13]

is usually enough [7], the addition of torsional springs is advisable to minimize the chances of getting distorted meshes [8, 10, 11]. However, this comes at the expense of a greater computational cost. Torsional springs were found to be particularly essential in preserving the positivity of cell volumes when deforming viscous grids [12].

The stiffness $k_{ij}$ of each tensional spring is usually modelled as the inverse of the length of the spring ($k_{ij} \propto 1/|x_j - x_i|$) [10] or the inverse of the square of the length ($k_{ij} \propto 1/|x_j - x_i|^2$) [9] as a workaround to avoid co-incident nodes. Also, depending on whether the equilibrium length is set to zero or greater than zero, tensional springs can be subdivided into segment and vertex springs [12]. The former is more appropriate in the neighbourhood of convex boundaries. Likewise, the stiffness $k_{ij}$ of torsional springs has been defined in several papers [9, 10] as a quantity proportional to $1/\sin^2 \theta$, where $\theta$ is the angle between adjacent springs. Torsional stiffness tends to infinity when the angle between two springs tends to zero, which prevents springs (i.e. edges in the mesh) from crossing one another.

Some authors have successfully used the truss analogy [2, 11] and the linear elasticity analogy [2, 13] to prevent cell elements from inversion, by using the structural equation of a bar element and a modified linear elasticity equation, respectively. An example of the effectiveness of the linear elasticity analogy can be seen when comparing the resulting deformed mesh via the spring analogy and the linear elasticity analogy due to a flap rotation, Figure 1.3.

Transfinite interpolation

Transfinite interpolation [14, 15] is based on the definition of a bilinearly blended interpolant which maps the domain boundaries at a non denumerable number of points, i.e. a curve in a two-dimensional space or a surface in a three-dimensional one. For example, let us assume that there exists a bivariate continuous vector valued function $\mathcal{F}(u,v)$ that maps a simple square $S_0 = [0,1] \times [0,1] \subset \mathbb{R}^2$ onto a more complex geometry such as a C-grid $C_0 \subset \mathbb{R}^2$ for computational aerodynamics, i.e. $\mathcal{F} : S_0 \mapsto C_0$. The goal of transfinite interpolation is to
find a one-to-one (or univalent) mapping $T(u, v)$ such that it matches $F$ on the boundary, i.e. $T : \partial S_0 \mapsto \partial C_0$. Function $T$ is commonly referred to as the transfinite interpolant of $F$ and it only requires the value of map $F$ at the boundaries, as follows (see [14]),

$$T(u, v) = P_u(F) + P_v(F) - P_u(F)P_v(F)$$

(1.5)

where the so-called projectors $P_u$ and $P_v$ are given by a blend of map $F$ at the boundaries of the domain

$$P_u(F) = \varphi_0(u)F(0, v) + \varphi_1(u)F(1, v)$$

(1.6)

$$P_v(F) = \psi_0(v)F(u, 0) + \psi_1(v)F(u, 1)$$

(1.7)

Finally, the simplest form of blending functions $\varphi_0$, $\varphi_1$, $\psi_0$, and $\psi_1$ may be written as the linear functions

$$\varphi_0(u) = 1 - u \quad \varphi_1(u) = u$$

(1.8)

$$\psi_0(v) = 1 - v \quad \psi_1(v) = v$$

(1.9)

Notice that, in practice, the definition of transfinite interpolant $T$ requires the knowledge of $F$ only at a finite number of points along the boundaries of the domain. This technique is very efficient, requiring little computational power, and works well with structured grids provided that boundaries do not get heavily distorted. However, there are examples of bad parameterisations in which the transformation yields one or more invalid elements, as can be seen in Figures 1.4 and 1.5. Two different remedies have been used to fix a bad (non-univalent) parameterisation in the examples provided by Gordon et al. [14]. On the one hand, a re-parameterisation of the boundary segments has been done for the L-shaped domain in Figure 1.4 using a quantity proportional to the arc length as a more sensible parameter. On the other hand, new auxiliary constraints at $u = \text{constant}$ or $v = \text{constant}$ lines have been introduced in Figure 1.5 in order to gain greater control over internal elements of the transfinite interpolation. The latter is usually simpler and still provides great flexibility.
Although transfinite interpolation is usually seen as a mesh generation only method, in fact it can also be used as a means of re-generating meshes with identical connectivity information, which is effectively equivalent to just deforming the existing mesh, in the event of solid boundary motions.

Laplacian smoothing

Although Laplacian smoothing requires an existing mesh it can be regarded as another mesh motion technique since it can be used to improve the overall grid quality after deformation happens due to boundary motions. Sometimes the grid deformation algorithm takes effect only on boundary nodes, leaving all other nodes unchanged, thus yielding low quality cells nearby. Laplacian smoothing is relatively cheap to compute and can recover the original grid quality via finite differences of Laplace equation, as follows

$$x_{i}^{\text{new}} = x_{i}^{\text{old}} + \lambda \Delta x_{i}$$  \hspace{1cm} (1.10)

with the discrete Laplace operator defined here as

$$\Delta x_{i} = \sum_{j} w_{ij} (x_{j} - x_{i}) \hspace{1cm} \sum_{j} w_{ij} = 1$$  \hspace{1cm} (1.11)

where $x_{i}$ represents position of $i$-th node in the mesh and $\lambda$ and $w_{ij}$ are weighting parameters. Looking at the equations it can be concluded that Laplacian smoothing is equivalent to the spring analogy provided that
1.3. Existing methods for moving and deforming meshes

constant spring stiffnesses are used. On the negative side, however, the method does not necessarily improve the mesh. It can sometimes produce negative or inverted elements, i.e. with negative Jacobian, hence Field [16] proposes a modified Laplacian smoothing in order to retain Delaunay triangulation properties. Figure 1.6 shows a poor quality Delaunay triangulation and the result of applying the proposed Laplace-Delaunay smoothing [16] to it, outperforming the standard Laplacian smoothing on Delaunay triangulations. Likewise, Freitag [17] implements an improved and more expensive version of the method called smart Laplacian smoothing which optimizes some mesh quality measure. It can prevent cell inversion by checking whether the new node locations degrade mesh quality.

Biharmonic smoothing

Equivalent to Laplacian smoothing, Helenbrook [18] proposes a smoothing algorithm based on the biharmonic operator for deforming meshes in an arbitrary Lagrangian-Eulerian formulation. The use of fourth-order PDE’s of the form $\nabla^4 x = 0$ allows the imposition of conditions on both the boundary location and the normal mesh spacing, which the Laplace operator fails to achieve. An example of the use of the biharmonic operator is depicted in Figure 1.7 and compared to the equivalent deformation via the Laplace operator.
Figure 1.7: Smoothing based on the biharmonic operator compared to the equivalent one based on the Laplace operator: (a) initial (undeformed) mesh, (b) deformed mesh via the Laplace operator, and (c) deformed mesh via the biharmonic operator. Extracted from Helenbrook [18]

Radial basis functions

Radial basis functions (RBF) [19, 20] are extremely versatile and cope well with fluid-structure interaction and relatively large mesh motion problems [21, 22]. Their value depends only upon the distance to reference points and no connectivity information is required. An interpolation function $s(x)$ is constructed by weighting the contributions of the different reference points $x_i$ via the coefficients $\alpha_i$

$$s(x) = \sum_{i=1}^{N} \alpha_i \phi(||x - x_i||)$$  \hspace{1cm} (1.12)

where $\phi$ are the radial basis functions and $N$ is the total number of reference points. It is possible to modify the region of influence of each point through a simple modification of the norm [21]. Mesh motion techniques that
exploit RBF’s are efficient and inexpensive since the coupling matrix is calculated only once, with no further modifications needed thereafter, and all other operations being simple matrix multiplications. An example of the application of radial basis functions can be found in Figure 1.8 for the deformation of a viscous mesh corresponding to a pitching NACA 0012 aerofoil. The radial basis functions approach copes well with this type of deformation and grid quality is successfully retained after transformation. An optimal selection of the reference points or centres can be obtained effectively through a greedy method \cite{23,24,25} which improves greatly the efficiency by finding redundant centres. Moreover, there exist a very efficient method \cite{26} based on a multi-scale formulation and capable of achieving comparable or even better mesh quality than with the greedy method, at a similar cost. On the negative side, storage requirements for RBF’s may be bigger than that of its direct competitors since it needs to store the coupling matrix between the reference points and all other points in the mesh.
1.3.2 Chimera or overset grids

When complex rotational parts or relative motions [6] are involved, Chimera or overset grids (originally developed for mesh generation, not mesh deformation) become a more reasonable alternative to mesh motion techniques thanks to the use of a separate body-fitted mesh for each of the moving parts. A global volume mesh is also defined in the background and an intersection between meshes is performed to find the interfaces at each time level. This allows splitting of the fluid domain into several regions and discretisation of them independently, effectively transforming a very intricate mesh generation problem into several simpler and smaller ones. In addition, simple yet powerful high-quality structured grids can be used around each of the moving parts, which translates into more efficient and faster fluid solvers and mesh generators [27, 28]. Boundary motions are very much simplified and only a rotation and/or translation of the existing grids is required before the intersection process happens again, hence saving computational effort. Due to their block-based nature, Chimera grids are suitable for parallel computations [29] and represent a reasonable alternative when an adaptive refinement mechanism is to be implemented [27].

Despite the flexibility described above, interpolation algorithms needed at the boundaries of two overlapping grids are usually costly and complex [28], and can introduce numerical errors unless special care is taken to minimize them. They still cannot deal with arbitrary motions such as: aeroelastic problems [29], where a mesh deformation technique is required in addition; or situations involving topological changes with appearing/disappearing cells, which, once again, rely on interpolations of the solution.

Examples of Chimera grids applied to problems with complex geometries can be found in Figure 1.9. Meshes for two different kinds of problems solved with this approach are included: (a) a moving rotor and (b) a deploying flap.

Figure 1.9: Chimera grids examples for (a) a moving rotor (extracted from Pomin et al. [29]) and (b) a deploying flap case (extracted from Renzoni et al. [28]).
1.3.3 Sliding grids

In contrast to Chimera grids, sliding grid planes are based on grids whose boundaries fit together without any overlapping at all, and slide past each other when there exists a relative motion (see Figure 1.10 for some examples).

![Figure 1.10: Interface between adjacent blocks in sliding grid planes: (a) planar interface with equal size cells at either side, (b) planar interface with different cell sizes at either side and (c) curved interface with different cell sizes at either side](image)

An interpolation method must still be implemented in order to communicate flow variables on both sides of the interface \[30, 31\]. A method for the study of helicopter rotor-fuselage interaction is proposed by Steijl et al. \[31\] proving its accuracy and efficiency, provided the mesh size is not too big, but performing poorly under parallel computations. Moreover, limitations regarding the allowable timestep are also important. Rumsey \[30\] showed that a small timestep is crucial for the construction of an interpolation scheme that deals properly with the propagation of acoustic waves across sliding planes. Likewise, the work of Fenwick et al. \[32\] led to the conclusion that, during the maximum timestep, grids should slide no more than the size of one cell if a correct unsteady behaviour is to be captured.

1.3.4 Immersed boundary method

Immersed boundary methods \[33, 34\] can also be a feasible alternative to deal with mesh deformation in unsteady aerodynamics. They rely on a fixed Cartesian grid in the background and the fluid solver is fully responsible for modelling the boundaries, leaving the mesh unaltered. For instance, the presence of a body immersed in a fluid flow may be characterised by a force density \(f_b(x,t)\) given by

\[
f_b(x,t) = \int_F f_b(s,t) \delta(x - X(s,t)) \, ds\tag{1.13}
\]
which is added to the momentum equation yielding

$$\rho \left( \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \nabla \cdot \mathbf{v} \right) = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f}_b$$

(1.14)

A parameterisation \( \mathbf{X} : (s, t) \rightarrow \mathbb{R}^3 \) of the surface \( \Gamma \) has been used to represent the solid boundary of the body, where \( 0 \leq s \leq 1 \) is a parameter and \( t \geq 0 \) is the time. The Dirac delta is three-dimensional, i.e. \( \delta (\mathbf{x}) = \delta (x) \delta (y) \delta (z) \), and the force density exerted by the solid is given by \( \mathbf{F}_b (s, t) \) for the given parameterisation.

However, this means that solid surfaces do not coincide with mesh lines leading to inaccuracy, particularly for compressible flow. Such methods often fail to ensure conservation of mass, momentum or energy in cells cut by a solid boundary and it is therefore not a popular technique across the aerospace industry where compressible aerodynamics demand a high quality representation of boundaries. Moreover, very thin boundary layers cannot be captured sensibly unless either the cell count is high, meaning large and expensive simulations, or anisotropic refinement is used [35]. Immersed boundary methods can be applied to any arbitrary boundary motion problems with relative ease as long as achieving good solid surface representation is not essential. Examples of the successful application include flows past a cylinder [33] and biomedical problems such as heart valve simulation [36].

1.3.5 Cartesian cut-cell grids

Cartesian cut-cell grids [37, 38] also rely on a fixed Cartesian grid in the background. However, in contrast with immersed boundary methods, the grid is actually cut along the boundaries at every time level incurring a relatively higher computational cost, especially for large problems. Conservation laws are successfully accomplished by means of a finite volume integration overcoming the issues that immersed boundary methods had with compressibility. On the other hand, cut-cell grids also require interpolations between consecutive time levels. In addition, due to the underlying fixed Cartesian grid, the achievable accuracy of the solution for the turbulent Navier-Stokes equations cannot be compared to that of other boundary-fitted techniques [37]. Cartesian cut-cell grids have been applied successfully to various problems, from a slat and flap case [37] (see Figure I.11) through to a flow past a cylinder or an internal combustion engine [38], as seen in Figure I.12.

1.3.6 Re-meshing

As outlined before, re-meshing is perhaps the most general way to deal with moving boundary problems as it can deal successfully with any arbitrary motion preserving good quality meshes at all times [39, 40, 41, 42, 43]. Compared to the above methods it is likely to be the most demanding one in terms of computational effort since a whole new mesh has to be generated at every time level. Being capable of dealing with structured grids, it
1.3. Existing methods for moving and deforming meshes

Figure 1.11: Multi-element NHLP-2D aerofoil in takeoff configuration using Cartesian cut-cell grids: (a) overall view of the surrounding mesh, (b) detailed view of slat deflected $\delta_s = 25$ deg, and (c) detailed view of flap deflected $\delta_f = 2$ deg. Extracted from Kidron et al. [37]

Figure 1.12: Velocity field at $Re = 22,000$ for the intake stroke (infinitely long) of an internal combustion engine using Cartesian cut-cell grids: (a) mean velocity and (b) instantaneous velocity. Extracted from Meinke et al. [38]
is with unstructured meshes where the greater gains in efficiency are achieved given the ability to modify only certain regions of the domain. Nevertheless, it is always necessary to work out the connectivity relationship between cells at consecutive time levels and an appropriate interpolation method has to be derived in the event of topological changes such as appearing/disappearing cells, which may introduce numerical errors across the solution. Besides, re-meshing can be non-conservative therefore making it unsuitable when this is a necessity, such as the prediction of shock waves and other phenomena. Re-meshing can be used along with mesh motion techniques alleviating the amount of computational resources required in places where mesh deformation is sufficient. Strictly, it would only be necessary where large motions or topological changes in the geometry happen. Figure 1.13 shows an example of re-meshing for the simulation of a store separation problem.

### 1.3.7 Meshless or meshfree algorithms

Meshless methods arise in an attempt to circumvent what appears to be the main bottleneck for industrial application of numerical techniques, i.e. good quality and automated mesh generation for complex geometries with sharp edges [45]. As seen in Figure 1.14, they replace the traditionally used grid by a dense cloud of points based on which conservation laws can be discretized [44]. Whilst connectivity information is inherently lost, more programming effort is needed compared to traditional mesh-based methods since there is still the need for finding the neighbours which lie within the domain of influence of each node [46], which can be a very time consuming task. Moreover, the generation of optimal distributions of points tailored for each specific problem is not a trivial task [45, 47] and conservation can sometimes be an issue. Katz [45] proposes a novel technique to deal with interfaces in overset or Chimera grids through the solution of the Navier-Stokes equations via a meshless method, as seen in Figure 1.15. This has been traditionally accomplished by the use of costly and intricate interpolation algorithms. The meshless approach promises to improve the efficiency and the accuracy at those interfaces, especially in the vicinity of solution discontinuities and in the case of unsteady simulations.

Amongst others one can outline three main meshfree methods: smoothed particle hydrodynamics, finite points and meshless local Petrov-Galerkin.
1.3. Existing methods for moving and deforming meshes

Figure 1.14: Solution of the ONERA M6 wing subject to twisting deformation submerged in a freestream flow at $M_\infty = 0.84$ and an angle of attack $\alpha = 3.06$ deg. The Euler equations for fluid motion have been discretised on a cloud of points via a meshfree method as depicted for three sections of the wing. Also, pressure coefficient $C_p$ distributions along these three planes are provided for a coarse, fine and coarse adapted clouds of points. Extracted from Ortega et al. [44]

Figure 1.15: Solution at the interface of overset grids via a meshless or meshfree method. Extracted from Katz [45]
Chapter 1: Introduction

Figure 1.16: Solution of free-surface flow after the break of a dam and impact against a vertical wall through the method of smoothed particle hydrodynamics (SPH). Depicted pressure fields at three different time-steps (time increases from top to bottom). Extracted from Colagrossi et al. [48]

Smoothed particle hydrodynamics

Smoothed particle hydrodynamics (SPH) is a particle method and uses a Lagrangian formulation. It is therefore particularly well suited to problems with free-surface flows like the break of a dam [48] depicted in Figure 1.16. At a given location fluid variables are defined by the weighted contributions of all the particles which lie directly within its domain of influence or smoothing distance. Depending on the smoothing distance there exist several types of weighting functions, the so-called kernel functions, such as Gaussian, where all particles yield a contribution regardless of the distance, or cubic splines, where the domain of influence is limited to a maximum distance from the target particle. Monaghan [46] found that a domain of size three times the smoothing length yields good approximation for properties such as density. Conservation of mass, momentum and energy is intrinsically accomplished as particles transport all physical quantities as they move. Determining the optimal domain of influence for each particle can be a very time consuming task [45] and the efficiency of the method is tightly related to this. Monaghan [46] achieved one order of magnitude more efficient computations, especially in large problems, by defining a cell-like data structure that allows a faster access to neighbour information. However, the use of connectivity information plays against the a priori benefits of meshfree methods.

Finite points

Finite points method (FPM) can be included under the category of meshfree methods. Points remain fixed in space independently of the fluid motion, therefore the method is regarded as an Eulerian method. Point collocation techniques are used to discretise the partial differential equations at each subset of points using least
squares fitting. This brings a great flexibility to the method since any complex geometry could potentially be solved by the finite points method [44, 49]. Although any cloud of points is a priori suitable for finite points method the importance of good clouds selection is outlined by Oñate et al. [47].

**Meshless local Petrov-Galerkin**

Another approach to Eulerian meshfree techniques is the local Petrov-Galerkin method [50, 51, 52]. It is based on a weak formulation of the partial differential equations and uses least square fitting techniques to approximate the solution at a cloud of points. The definition of a very simple local underlying grid is still necessary to perform the integrals that arise from the weak formulation.

### 1.4 Spacetime

The spacetime framework offers an alternative conservative simulation approach. It can accommodate topological changes and variable real (or physical) time-steps, but must be appropriately implemented to preserve time accuracy. The novelty introduced is that both integrations, in space and time, are treated similarly through the use of spacetime finite-volumes, hence effectively solving unsteady problems of dimension $N$ as steady problems of dimension $N + 1$. For instance, a two-dimensional aerofoil oscillating periodically in a pitching motion would require the solution of a twisted wing in a spacetime formulation where the third dimension represents time $t$ (see Figure 1.17). This implies modifying the fluid equations of motion through the divergence theorem to remove temporal derivatives, which are replaced by temporal fluxes instead. The fully conservative nature in space and time as a consequence of the finite-volume approach in spacetime, facilitates the solution of problems where cells appear/disappear between consecutive time levels.

![Spacetime representation of a two-dimensional NACA 0012 aerofoil oscillating periodically in a pitching motion. The third dimension represents physical time $t$ and is to be integrated along with the spatial coordinates through the use of a spacetime finite-volume mesh.](image)

Figure 1.17: Spacetime representation of a two-dimensional NACA 0012 aerofoil oscillating periodically in a pitching motion. The third dimension represents physical time $t$ and is to be integrated along with the spatial coordinates through the use of a spacetime finite-volume mesh.

An early spacetime formulation came from Giles [53] in 1988 at a propulsion conference in Boston. The
imposition of periodic boundary conditions in turbomachinery flows is a complex task, especially when the rotor and stator have different pitch values (i.e. distance between blades). By inclining the computational time plane Giles circumvents this issue and transforms the Euler equations so that any stator-rotor pair can be treated with a pitch ratio of 1. At the same time, Hughes et al. [54] applied a spacetime technique to classical elastodynamics problems via the use of a finite element approach with a discontinuous Galerkin (DG) formulation in the time direction. Lowrie et al. [55] build on the previous work for a much more general problem involving hyperbolic conservation laws. Again, they use a discontinuous Galerkin formulation to create a higher order scheme within the spacetime framework. However this results in a computationally expensive method compared to other conventional approaches. Moreover, their work implicitly assumes some regularity on the structure of the grids used, hence invalidating a more general boundary motion approach. Thompson et al. [56] and Ray [57] use a DG formulation to give their own interpretation of the spacetime method. In particular, Thompson et al. [56] present an adaptive spacetime technique that allows refinement and coarsening of the grid and which they define as robust. This robustness comes at a cost, the loss of the general applicability of the method since they retain orthogonal planes in the time direction leaving the time integration fully decoupled from the space integration (see Figure 1.18).

Tsuei et al. [58] successfully apply the spacetime method developed earlier by Chang [59, 60, 61] at NASA to blade row interaction problems in turbomachinery flows. The so-called space-time conservation element and solution element method (CE/SE) is able to predict unsteady flows without any previous assumptions imposed on the solver, by simply considering fluxes both in space and time. They argue that the scope of this new method is large and that a wide range of applications can benefit from it, however their work is focused on turbomachinery applications and no attempts made towards a more general and arbitrary motion. Perhaps
the most general implementation of the spacetime method are the works by Hixon [62, 63] and Golubev et al. [64, 65]. Their method allows for a variable timestep size across the fluid domain and no decoupling is made between temporal and spatial integrations, allowing for higher-order schemes to be used in the time integration. Zwart et al. [66] apply the spacetime formulation to the solution of a breaking dam, although their implementation lacks a general spacetime mesh with varying timestep sizes across the spatial domain. Similarly, Van der Ven [67] applies a conservative adaptive multigrid algorithm under the spacetime framework to investigate an oscillating two-dimensional aerofoil, demonstrating the potential of the method.

More recently, Rendall et al. [68, 69, 70, 1] use a general formulation of the spacetime method and show its capability for simulating complex moving geometries in one and two-dimensional spaces. In their work they compare their solution against analytical results available from piston theory and investigate the problem of a pitching NACA 0012 aerofoil with both structured and unstructured meshes. Due to the use of a central-difference scheme in time they observe a slight change in the behaviour of the pressure distribution with respect to a conventional dual time-stepping solver. They explain this phenomenon with information propagated backwards in time as a consequence of the temporal stencil being used. They also show the ability of the spacetime solver to cope with rotating parts and appearing/disappearing objects in both a stationary gas and moving subsonic and supersonic flows, as well as a store separation problem (see Figure 1.19). More recently, and parallel to the work presented in this thesis, Wang et al. [71] develop a high-order discontinuous Galerkin spacetime formulation for fully unstructured meshes. In fact, like Thompson et al. [50], they still retain orthogonal planes in the time direction but they manage to generate a fully unstructured spacetime mesh between two consecutive time slabs.

Figure 1.19: Solution of a two-dimensional store separation problem in a crossflow at $M_\infty = 0.3$ through a finite-volume spacetime method. Mach contours are depicted at several time slices. Extracted from Rendall et al. [1].
Chapter 1: Introduction

Figure 1.20: Solution of two aerofoils pitching in tandem through a discontinuous Galerkin spacetime method. Depicted entropy contours and mesh at three different time slices. Extracted from Wang et al. [71]

being able to effectively simulate complex motions and topological changes in the geometry. They successfully solve the compressible Navier-Stokes equations for a spinning cross, a pair of NACA 0012 aerofoils pitching in tandem (Figure 1.20) and a spoiler case.

Along with the development of a spacetime framework there is a need for new grid generation techniques. Being able to generate fully unstructured grids in spacetime brings the possibility to refine the timestep size in some areas of the domain while keeping a coarse one in others where temporal resolution is not required. Currently, it is possible to use available three-dimensional grid generators to create two-dimensional unsteady meshes well-suited for the spacetime framework. However, although attempts have been made at research level [72, 71], there is no available technology at industrial level to automatically generate a truly unstructured four-dimensional grid to be used in 3D+\( t \) problems. Note here the use of 3D+\( t \) to refer to a four-dimensional Euclidean space where three dimensions correspond to physical space and the fourth one represents physical time. Although not yet mature, Behr [73] introduces a simple meshing technique that allows unstructured grids to be created, not only for 2D+\( t \) problems but also 3D+\( t \). Likewise, Ungor et al. [74] and, some time later, Abedi et al. [75] have put some efforts towards the development of 2D+\( t \) grids by a mesh-marching technique. Finally, Persson et al. [72, 71] provide a simple yet powerful mesh generator algorithm based on the analogy between a simplex mesh (i.e. the \( n \)-dimensional generalization of a triangle in two-dimensional space or a tetrahedron in three-dimensional space) and a truss structure. They provide a generalization of the algorithm for an \( n \)-dimensional space and showcase its potential via the four-dimensional mesh of a hypersphere (i.e. the four-dimensional generalization of a circle in two-dimensional space or a sphere in three-dimensional space). However, although this represents an important step, a more mature tool needs to be available if the spacetime framework is to be used in 3D+\( t \) at industrial level.
Spacetime framework for unsteady aerodynamics of complex motions

<table>
<thead>
<tr>
<th>Strengths</th>
<th>Weaknesses</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ It is versatile and applicable to a large (potentially unlimited) number of unsteady aerodynamics problems, even those involving topological changes in the geometry (e.g. it copes well with appearing/disappearing cells). No special treatments are necessary across the whole spectrum of time-accurate problems, hence potential for highly automated CFD simulations which could, in turn, speed up the design cycle of industrial applications.</td>
<td>- It is, in general, slower than its conventional counterpart (e.g. dual-time stepping, implicit solver) since it updates the entire time domain at each pseudo-time iteration. Solution: take advantage of the fact that information travels always forwards in time at a constant pseudo-velocity $u^* = \frac{dt^*}{dt} = 1$, as implied from a characteristics analysis, and update the solution behind this wavefront only (similar idea to a space-marching algorithm for supersonic flows).</td>
</tr>
<tr>
<td>+ It ensures the automatic conservation of mass, momentum and energy in both the space and time integrations. Hence, it is particularly well-suited to periodic problems since initial and final states are connected directly via cell interfaces. It is safe to assume that a solution has converged once $l_2$-norms of residuals have dropped beyond an acceptable threshold.</td>
<td>- It may exhibit non-physical behaviour (e.g. pressure waves travelling backwards in time) if a central-difference stencil is used. Solution: use of upwind biased stencils.</td>
</tr>
<tr>
<td>+ It can be very efficient due to its potential for varying physical-time step sizes at different locations. For instance, large time steps may be used in the farfield whilst still retaining small time steps close to the geometry (e.g. aerofoil) where fluid properties (e.g. $\rho, u, p$) change faster.</td>
<td>- It is cumbersome to generate truly unstructured four-dimensional geometries to solve three-dimensional unsteady problems due to the lack of available commercial software. Remark: although it may still be a pre-mature technology, Persson et al. [72, 71] already succeeded in this.</td>
</tr>
<tr>
<td>+ There is no need for intricate and sometimes ambiguous interpolation methods, such as overset or Chimera grids, which may lead to accuracy losses and make a very intensive usage of available computational resources.</td>
<td>- It is hard to couple with structural solvers to perform fluid-structure interaction (FSI) simulations since motions must be prescribed beforehand. Solution: the spacetime mesh may be subdivided into smaller regions, each of them delimited by two $t = constant$ planes at $t_i$ and $t_i + \Delta t_i$, where $\Delta t_i$ is suitable for FSI coupling. A coupling with the structural solver is then possible. A mesh deformation technique would be necessary in this sub-region after a deformation of the geometry.</td>
</tr>
<tr>
<td>+ There exists the possibility to incorporate some of the high-order schemes (e.g. discontinuous Galerkin), historically only used for the spatial discretisation, into the time discretisation.</td>
<td>- It is not straightforward to couple a spacetime solver with a conventional method to solve unsteady aerodynamics problems and take advantage of the strengths of both methods. Solution: see proposal in Sections 3.7.2 and 3.7.3.</td>
</tr>
<tr>
<td>+ It is suitable for shared- and/or distributed-memory parallelisation, just like any other conventional CFD method.</td>
<td></td>
</tr>
</tbody>
</table>

Table 1.1: Main strengths and weaknesses of the spacetime framework for the solution of unsteady aerodynamics problems with complex boundary motions. Potential solutions are given for each of the weaknesses.
1.5 Outline of thesis

The main goals of this thesis are summarised as follows:

- To demonstrate the ability to accurately solve unsteady aerodynamics problems with complex boundary motions through the use of a spacetime formulation.
- To understand the direction in which disturbances travel in spacetime and investigate the suitability of the different stencils.
- To implement and develop a spacetime framework for the simulation of unsteady aerodynamics problems with different stencils.

The research contributions made by the author of this thesis are summarised as follows:

- Writing the entire CFD code used to simulate the problems presented in this work fully from scratch. The code was written in C and a shared-memory parallelisation based on OpenMP® was implemented. Several other tools were developed to generate spacetime meshes from two-dimensional spatial grids.
- Investigating the implications of using a central-difference scheme for the solution of time-accurate problems with a spacetime formulation. Demonstrated that this leads to non-physical solutions as a consequence of pressure waves travelling backwards in physical-time.
- Implementing two upwind stencils in spacetime, namely Van Leer and Roe, and a hybrid one (central in space, upwind in time). Investigating the improvements achieved by the use of a more realistic time stencil.
- Performing a stability analysis for the calculation of the time-step size in the spacetime formulation provided here.
- Simulating a wide range of time-accurate aerodynamics problems involving complex boundary motions and topological change in the geometry. This comprises mesh generation, simulation, post-processing and analysis of results.
- Collaborating with Cranfield University within the Airbus EFT (Enhanced Fidelity Transonic Wing) project by providing support on spacetime simulation for a kinematic optimisation of high-lift wing devices.
- Writing and publishing the conference paper:
The thesis is divided into six chapters and eight appendices. In Chapter 2 an introduction to the discretisation techniques used in conventional CFD methods for unsteady aerodynamics problems is given. This serves as the basis for the discretisation of the governing equations within the spacetime framework. The discretisation of the Navier-Stokes equations is given first, followed by the discretisation of the Spalart-Allmaras turbulence model. Finally, the implementation of boundary conditions is also explained. In Chapter 3 the available solution methods for the integration of the Navier-Stokes equations in time or pseudo-time are discussed. Also, a comparison between an arbitrary Lagrangian-Eulerian formulation and the spacetime method is given. In Chapter 4 a stability analysis is performed for the solution of the fluid equations of motion in spacetime. In Chapter 5 a range of two-dimensional unsteady aerodynamics problems with large boundary motions and topological changes is simulated. An analysis of the suitability of the different stencils is provided for each of the cases presented. Finally, in Chapter 6 a summary of the conclusions drawn from this research and ideas for future work are given. A short introduction to the arbitrary Lagrangian-Eulerian formulation of the equations for fluid motion, historically used in time-accurate problems with mesh motion/deformation, is given in Appendix A. In Appendices B and C a summary of the Navier-Stokes equations and the one-equation turbulence model by Spalart-Allmaras are given, respectively. Appendix D compares the dissipation of an upwind scheme with the numerical dissipation used in central-difference schemes. An explanation of the calculation of spatial and spacetime gradients within the spacetime formulation is given in Appendix E and the derivation of backward second-differences, used in Chapter 3, is given in Appendix F. Appendix G includes the full set of results for test case 1 of the AGARD report R-702(3E3) explained in Chapter 5. Finally, a short description and some implementation details of the code developed for this thesis are given in Appendix H.
Chapter 2

Discretisation methods

2.1 Introduction

A numerical integration of the Navier-Stokes equations, (B.39), (B.40) and (B.41) in Appendix B and the
one-equation Spalart-Allmaras turbulence model, (C.22) or (C.34) in Appendix C requires the discretisation
of the space-time domain where these are to be solved. Historically, this has been accomplished using one of
the three main approaches as follows: finite differences, where a differential (or strong) form of the conservation
laws is solved; finite elements, where an integral (or weak) form of the conservation laws is solved; and finite
volumes, which constitutes a particular case of finite elements, namely when only one integration point exists
per cell (i.e. shape function across each cell is a “hat”).

Conventional unsteady aerodynamics simulations use a finite difference approach for the time integration
whereas a finite volume or finite element method is used for the integration in space. In the spacetime framework, however, the same finite volume or finite element method used for the spatial discretisation is used for the integration of the equations in physical time. Thus, the integration in space and time is performed at once via
the use of spacetime finite volumes/elements. For simplicity and given the scope of the present work, a finite
volume approach has been used here although there exists evidence that a finite element formulation, such as
discontinuous Galerkin method, can be used as well [71]. As outlined above, the finite volume method is just a
particular case of the finite element formulation where there is only one integration point per cell and weight
functions are equal to one. Hence, most (if not all) of the conclusions drawn from this work should be directly
applicable to a finite element formulation of the spacetime framework.

In this chapter, the spatial discretisation of the physical equations for fluid motion in the case of a central-
difference and upwind stencils are described first for comparison with later sections. However, the main goal is

27
the introduction of the spacetime discretisation of the Navier-Stokes equations, Section 2.4, and the standard and negative Spalart-Allmaras turbulence models, Section 2.5. Similarly, the implementation of boundary conditions for a spatial discretisation is given first in Section 2.6 followed by the corresponding implementation of boundary conditions in the spacetime framework in Section 2.7.

Only the two-dimensional formulation is given here because only one- and two-dimensional problems have been studied in the present work. Nevertheless, the derivation of a three-dimensional version of the equations presented in this chapter should be straightforward.

### 2.2 Spatial discretisation of the physical equations

Due to its simplicity compared to a finite element method and the fact that a coupled integration in space and time is key for the spacetime framework, a finite volume method has been used throughout this work to discretise the physical conservation laws. Both central and upwind stencils are described in the following sections.

#### 2.2.1 Integral formulation of the Navier-Stokes equations

The integral form of the Navier-Stokes equations (B.39), (B.40) and (B.41), given in Appendix B for a two-dimensional problem is

\[
\frac{d}{dt} \int_V U d\Omega + \int_V \left( \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} \right) d\Omega = \int_V \left( \frac{\partial F_v^x}{\partial x} + \frac{\partial F_v^y}{\partial y} \right) d\Omega \tag{2.1}
\]

where the volume is denoted by \( V \), the column vector of conserved variables \( U \) is

\[
U = \begin{pmatrix}
\rho \\
\rho u \\
\rho v \\
\rho E
\end{pmatrix}
\tag{2.2}
\]

the column vectors of inviscid fluxes along Cartesian coordinates \( x = x_1 \) and \( y = x_2 \), namely \( F_x \) and \( F_y \),
2.2. Spatial discretisation of the physical equations

respectively, are

\[
F_x = \begin{pmatrix}
\rho (u - u_c) \\
\rho u (u - u_c) + p \\
\rho v (u - u_c) \\
\rho E (u - u_c) + pu
\end{pmatrix}
\quad F_y = \begin{pmatrix}
\rho (v - v_c) \\
\rho u (v - v_c) + p \\
\rho v (v - v_c) + pv
\end{pmatrix}
\]

and the column vectors of viscous fluxes along Cartesian coordinates \( x = x_1 \) and \( y = x_2 \), namely \( F^v_x \) and \( F^v_y \), respectively, are

\[
F^v_x = \begin{pmatrix}
0 \\
\sigma_{xx} \\
\sigma_{xy} \\
u \sigma_{xx} + v \sigma_{xy} + q_x
\end{pmatrix}
\quad F^v_y = \begin{pmatrix}
0 \\
\sigma_{xy} \\
\sigma_{yy} \\
u \sigma_{xy} + v \sigma_{yy} + q_y
\end{pmatrix}
\]

where \( u = u_1 \) and \( v = u_2 \) are the velocity components along Cartesian coordinates \( x \) and \( y \). Notice that the Reynolds transport theorem in equation (A.1) has been used to extract the time derivative from the first integral and the corresponding terms have been introduced within the inviscid fluxes \( F_x \) and \( F_y \). Using the Green-Gauss theorem, also known as divergence theorem, to transform volume integrals into closed surface integrals, equation (2.1) may be re-written as follows

\[
\frac{d}{dt} \int_V U \, d\Omega + \oint_{\partial V} \left[ (F_x - F^v_x) n_x + (F_y - F^v_y) n_y \right] \, dS = 0
\]

where \( n_x \) and \( n_y \) are the Cartesian components of the normal vector \( n \)

\[
n = \begin{pmatrix} n_x \\ n_y \end{pmatrix}
\]

Using the vectors of inviscid and viscous fluxes, given in equations (2.3) and (2.4), as columns of matrices of inviscid and viscous fluxes, \( \mathbf{M}(U) \) and \( \mathbf{M}^v(U, \nabla U) \), respectively,

\[
\mathbf{M}(U) = \begin{bmatrix} F_x \\ F_y \end{bmatrix} \quad \mathbf{M}^v(U, \nabla U) = \begin{bmatrix} F^v_x \\ F^v_y \end{bmatrix}
\]

29
equation (2.5) can be cast in a more compact form as

$$\frac{d}{dt} \int_V U \, d\Omega + \mathbf{R}(U, \nabla U) = 0$$

(2.8)

where the column vector of residuals $\mathbf{R}(U, \nabla U)$ is given by

$$\mathbf{R}(U, \nabla U) \equiv \oint_{\partial V} (\mathbf{M} - \mathbf{M}_v) \, n \, dS$$

(2.9)

The steady state solution of equation (2.8) yields, once the time derivative vanishes,

$$\mathbf{R}(U, \nabla U) = 0$$

(2.10)

### 2.2.2 Second order central-difference scheme

The second order central-difference scheme proposed by Jameson et al. [76] in 1981 is perhaps one of the simplest but most effective discretisation schemes. A dissipative term which consists of a blend of second and fourth-differences via some pressure-based switches is the key to provide stability to the method. It can be used along with acceleration techniques such as local time stepping, based on the local Courant number (CFL), which yields much faster convergence rates to the steady state solution in meshes with very diverse cell sizes. This scheme is currently known within the CFD community as JST, named after its authors. Criticism over the ‘artificial’ dissipation is addressed by Turkel [77] who demonstrated that the JST numerical dissipation on top of a second-order central-difference scheme is equivalent to the use of an upwind biased scheme (see Appendix D).

The semi-discretisation (i.e. discretisation of the spatial domain only) of equation (2.8) using the second order central-difference JST scheme [76, 78] yields, in the case of unstructured grids (see Figure 2.1),

$$\frac{d}{dt} (U_i V_i) + \sum_{k=1}^{n_f} \left\{ [(\mathbf{F}_x - \mathbf{F}_x^0) A_x + (\mathbf{F}_y - \mathbf{F}_y^0) A_y]_k + \mathbf{D}_k \right\} = 0$$

(2.11)

1 JST are the initials of A. Jameson, W. Schmidt and E. Turkel.
2.2. Spatial discretisation of the physical equations

\[ \frac{d}{dt}(U_i V_i) + \sum_{k=1}^{n_f} \left( [(M - M^v) \mathbf{n} A_k] + D_k \right) = 0 \]  

(2.12)

where the subscript \( i \) denotes a cell with volume \( V_i \) and the subscript \( k \) denotes each of its \( n_f \) faces with area \( A_k \). The values \( A_x = A_k n_x \) and \( A_y = A_k n_y \) are the face area projected normal to the \( x \) and \( y \) coordinate axes, respectively. The main differences with respect to the structured case arise as a consequence of the explicit connectivity information required between elements (see Figure 2.1). The indices of the cells on the left and on the right of each face \( k \) in the mesh are denoted by subscripts \( l(k) \) and \( r(k) \), respectively. Therefore, fluxes are worked out as the flux of the average of conserved variables on the left and right cells

\[ (F_{x,y})_k = F_{x,y} \left( \frac{U_{l(k)} + U_{r(k)}}{2} \right) \]

(2.13)

\[ (F^v_{x,y})_k = F^v_{x,y} \left( \frac{U_{l(k)} + U_{r(k)}}{2} \right) \]

or, in matrix form,

\[ (M)_k = M \left( \frac{U_{l(k)} + U_{r(k)}}{2} \right) \]

(2.14)

\[ (M^v)_k = M^v \left( \frac{U_{l(k)} + U_{r(k)}}{2} \right) \]

Alternatively, the average of the fluxes on both sides at face \( k \) may be used

\[ (F_{x,y})_k = \frac{F_{x,y} \left( U_{l(k)} \right) + F_{x,y} \left( U_{r(k)} \right)}{2} \]

(2.15)

\[ (F^v_{x,y})_k = \frac{F^v_{x,y} \left( U_{l(k)} \right) + F^v_{x,y} \left( U_{r(k)} \right)}{2} \]
or, in matrix form,

\[
(M)_k = \frac{M(U_{l(k)}) + M(U_{r(k)})}{2} \quad (M^v)_k = \frac{M^v(U_{l(k)}) + M^v(U_{r(k)})}{2}
\] (2.16)

The numerical dissipation term \( D_k \) at face \( k \) is constructed as follows

\[
D_k = \left[ \varepsilon_k^{(2)} (U_{r(k)} - U_{l(k)}) - \varepsilon_k^{(4)} (\nabla^2 U_{r(k)} - \nabla^2 U_{l(k)}) \right] \Xi_k \vartheta_k
\] (2.17)

where the undivided Laplacian operator can be computed as

\[
\nabla^2 U_i = \sum_{j=1}^{N_i} (U_j - U_i) = -N_i U_i + \sum_{j=1}^{N_i} U_j
\] (2.18)

where \( N_i \) is the number of neighbours at cell \( i \). The local spectral radii \( \vartheta_k \) and \( \Upsilon_i \) are worked out at face \( k \) and cell \( i \), respectively, as follows

\[
\vartheta_k = (|v_k \cdot n| + a_k) A_k
\] (2.19)

\[
\Upsilon_i = \sum_{k=1}^{n_f} \vartheta_k = \sum_{k=1}^{n_f} (|v_k \cdot n| + a_k) A_k
\] (2.20)

where \( n_f \) is the number of faces at cell \( i \), \( a_k \) is the local speed of sound at face \( k \)

\[
a_k = \frac{a_{l(k)} + a_{r(k)}}{2}
\] (2.21)

and \( v_k = \{u_k, v_k\}^T \) is the fluid flow velocity at face \( k \)

\[
v_k = \frac{v_{l(k)} + v_{r(k)}}{2}
\] (2.22)
2.2. Spatial discretisation of the physical equations

Factor $\Xi_k$ is defined in order to account for the stretching of the grid \[78\] as

$$
\Xi_k = \left( \frac{4}{\Upsilon_{l(k)} + \Upsilon_{r(k)}} \right) \cdot \left( \frac{\Upsilon_{l(k)} \Upsilon_{r(k)}}{4\vartheta_k} \right)^{0.3}
$$

(2.23)

Switches $\varepsilon_k^{(2)}$ and $\varepsilon_k^{(4)}$ are functions of local pressure gradients and can be computed as

$$
\varepsilon_k^{(2)} = \frac{\varepsilon_{l(k)}^{(2)} + \varepsilon_{r(k)}^{(2)}}{2}
$$

(2.24)

$$
\varepsilon_k^{(4)} = s_4 \max \left( 0, \kappa^{(4)} - \varepsilon_k^{(2)} \right)
$$

(2.25)

where $\kappa^{(2)} \approx 1$ and $\kappa^{(4)} \approx 0.05$ are closure constants and $\varepsilon_i^{(2)}$ is a switch defined at cell $i$ as follows

$$
\varepsilon_i^{(2)} = \kappa^{(2)} s_2 \left( \frac{\sum_{j=1}^{N_i} (p_j - p_i)}{\sum_{j=1}^{N_i} (p_i + p_j)} \right)
$$

(2.26)

The scaling factor $s_2$ is defined such that the dissipation matches the dissipation of the structured formulation in the case of a quadrilateral cell, i.e. with four neighbours,

$$
s_2 = 2 \left( \frac{N_{l(k)} + N_{r(k)}}{N_{l(k)} N_{r(k)}} \right)
$$

(2.27)

where, again, $N_i$ is the number of neighbours at cell $i$. Finally, scaling factor $s_4$ is defined, in relation to $s_2$, as

$$
s_4 = \left( \frac{s_2}{2} \right)^2
$$

(2.28)

Fluxes are calculated independently of the flow direction, namely they are the same regardless of whether the flow comes from left to right or from right to left. Equation (2.11) results in a second order accurate central-difference scheme provided a good quality mesh \[76\]. Bear in mind that the above implementation of the JST method is only a particular one amongst many different variants that can be found in the specific literature.
2.2.3 Second order upwind scheme

In 1959 Godunov published a first-order accurate monotonicity preserving upwind scheme \cite{79,80}. The assumption that flow variables remain constant across each cell yields a piecewise solution and therefore a Riemann initial-value problem needs to be solved at each cell interface. However, the cost associated with solving this problem is so high that the method did not become popular until the first approximate Riemann solvers arose. These solvers provide reasonably accurate estimates of face fluxes \cite{81} at a fraction of the cost hence making them suitable for upwind biased schemes. Moreover, Godunov’s scheme exhibited numerical dissipation and, under certain circumstances, the solution appeared overly diffuse in the neighbourhood of discontinuities. A series of Godunov-type schemes flourished over the next few years aiming to improve the efficiency and accuracy.

One of the earliest contributions in this direction is the work due to Van Leer \cite{82,83} with his MUSCL scheme (monotonic upstream-centered scheme for conservation laws) where, instead of constant values across each cell, the use of piecewise linear polynomials (see Figure 2.2) opened the door to second-order accurate upwind methods \cite{81,82,84,85}. He proposed a flux-vector splitting where interface fluxes are calculated via consideration of forward and backward moving pseudo-particles according to a velocity distribution \cite{86}. Other important contributions to this statistical approach, sometimes referred to as Boltzmann approach, are the Beam-Warming scheme \cite{87,88} and the work due to Ben-Artzi \cite{89} or Colella \cite{90}. The spurious oscillations caused by the introduction of a high-order interpolation step motivated the use of monotonicity constraints like the slope limiters (different from flux limiters). They restrict the amount that physical quantities can change across one single cell and ensure that the values returned by the reconstruction step are not beyond the mean value in neighbouring cells. Here it is a necessity to mention the total variation diminishing (TVD) technique introduced by Harten \cite{91} as a means of monitoring and damping out the oscillations around shocks. It is based on the conservation of the so-called quantity total variation, defined for hyperbolic conservation laws, which remains constant throughout time except in the presence of shocks where it may decrease \cite{92}.
A series of flux-difference schemes which calculate the physically meaningful fluxes based on the approximate solution to the Riemann problem were also published. Relying on forward and backward moving waves as implied by the characteristics in hyperbolic problems, one can outline here the contributions by Roe [93] and Osher [94].

Finally, in an attempt to retain the efficiency and ease of implementation from flux-vector splitting techniques but, at the same time, achieve the accuracy of Riemann solver-based (or flux-difference) methods, a hybrid approach arose under the name of advection upwind split method, abbreviated as AUSM. The convective term is evaluated by flux-vector splitting whereas the pressure contributions are accounted for via acoustic waves as in the flux-difference method [95, 96].

In the present work, the flux-vector splitting approach by Van Leer [83] and the flux-difference method by Roe [93] is used along with a linear reconstruction step to obtain face values from cell-centre values. The rate of change of fluid properties across each cell is limited by the use of the aforementioned slope limiters. In particular, a slope limiter proposed by Van Leer is the one used in this work. The rest of this section explains these methods for the two-dimensional case of the Navier-Stokes equations.

Unlike Section 2.2.2 where both inviscid and viscous fluxes were discretised via a second-order central-difference scheme, convective terms are now discretised via a second-order upwind scheme whilst diffusive terms are still discretised via central-differences. Therefore, the semi-discretisation (i.e. discretisation of the spatial domain only) of equation (2.8) without any added artificial dissipation yields, in the case of unstructured grids (see Figure 2.1),

$$\frac{d}{dt} (U_i V_i) + \sum_{k=1}^{n_f} \left[ (F_x - F_x^+) A_x + (F_y - F_y^+) A_y \right]_k = 0$$

(2.29)

where the subscript $i$ denotes a cell with volume $V_i$ and the subscript $k$ denotes each of its $n_f$ faces with area $A_k$. The values $A_x = A_k n_x$ and $A_y = A_k n_y$ are the face area projected normal to the $x$ and $y$ coordinate axes, respectively.

As outlined before, in the upwind formulation the use of a piecewise linear solution leads to a discontinuous solution across the faces (see, for instance, interface $C$ in Figure 2.2). In other words, the calculation of inviscid face fluxes, $F_x$ and $F_y$, relies on the solution at both sides of the interface, $U^+$ and $U^-$, which are, in general, different (these correspond to $\phi_{C_{low}}$ and $\phi_{C_{high}}$ at interface $C$ in Figure 2.2). However, the numerical flux at each face must be unique.
Flux-vector splitting method by Van Leer

The flux-vector splitting method developed by Van Leer [83] defines a function \( \tilde{F}_n \left( W^+_n, W^-_n \right) \) of the numerical inviscid fluxes across each face on the normal direction. \( \tilde{F}_n \) is given by contributions from forward (all eigenvalues of \( \frac{\partial \tilde{F}^+_n}{\partial W^+_n} \geq 0 \)) and backward (all eigenvalues of \( \frac{\partial \tilde{F}^-_n}{\partial W^-_n} \leq 0 \)) moving waves, \( \tilde{F}^+_n \) and \( \tilde{F}^-_n \) respectively, i.e.

\[
\tilde{F}_n \left( W^+_n, W^-_n \right) = \tilde{F}^+_n \left( W^+_n \right) + \tilde{F}^-_n \left( W^-_n \right)
\]  

(2.30)

where the subscript \( n \) refers to the direction normal to the face since only the normal velocity component gives non-zero fluxes in the momentum equations. Column vectors \( W^+_n \) and \( W^-_n \) above are the normal primitive variables at both sides of the face calculated via a linear reconstruction step from cell-centre values, as will be explained later. The former corresponds to the side where the forward (or positive) moving wave comes from. The latter corresponds to the side where the backward (or negative) moving wave comes from. The column vector of normal primitive variables in the present two-dimensional case is given by

\[
W_n = \begin{pmatrix}
\rho \\
u_n \\
u_t \\
p
\end{pmatrix}
\]  

(2.31)

where \( u_n \) and \( u_t \) denote the projection of the flow velocity onto normal and tangential directions at the face, respectively. In order to compute these projections, consider transformation \( P : (x, y) \mapsto (x_n, x_t) \) which maps, at every face in the mesh, the global coordinate system to a local one such that \( x_n \) is normal to the face and \( x_t \) is tangential. Let \( T_P \) be the matrix of transformation \( P \). Then, the columns of \( T_P \) are the unit vectors of the basis of the local coordinate system \( B' = \{ e_n, e_t \} \) expressed in terms of the unit vectors of the basis of the global coordinate system \( B = \{ e_x, e_y \} \), as follows

\[
T_P = \begin{bmatrix}
e_n \\
e_t
\end{bmatrix} = \begin{bmatrix}
n_x & n_y \\
n_y & -n_x
\end{bmatrix}
\]  

(2.32)

where \( e_n \) and \( e_t \) correspond to the normal and tangential directions at the face. Hence, \( u_n \) and \( u_t \) may now be
2.2. Spatial discretisation of the physical equations

worked out in terms of matrix \( T_P \) as

\[
\begin{align*}
\begin{bmatrix}
  u_n \\
  u_t
\end{bmatrix} = \begin{bmatrix}
  T_P^{-1}
\end{bmatrix} \begin{bmatrix}
  u \\
  v
\end{bmatrix} 
\end{align*}
\]  

(2.33)

Moreover, let \( \hat{P} : W_n \mapsto U \) be the transformation between normal primitive variables and conserved variables, i.e.

\[
U = \hat{P}(W_n)
\]

(2.34)

Then, the matrix \( T_{\hat{P}} \) of transformation \( \hat{P} \) can be written in terms of matrix \( T_P \) as follows

\[
T_{\hat{P}} = \begin{bmatrix}
  1 & 0 & 0 & 0 \\
  0 & T_P & 0 \\
  0 & 0 & 1
\end{bmatrix} = \begin{bmatrix}
  1 & 0 & 0 & 0 \\
  0 & n_x & n_y & 0 \\
  0 & n_y & -n_x & 0 \\
  0 & 0 & 0 & 1
\end{bmatrix}
\]

(2.35)

A different approach must be followed to compute \( \tilde{F}_n(W_n^+, W_n^-) \) depending on whether the flow is locally subsonic or supersonic. Since only the component of the velocity normal to a face gives non-zero fluxes in the momentum equation, the local normal Mach number \( M_n = u_n / \sqrt{\gamma p/\rho} \) is used to discern between supersonic and subsonic flow.

On the one hand, if the flow is supersonic (\( |M_n| \geq 1 \)) fluxes are given by the properties at the upstream side only. In other words, the double-sided flux function \( \tilde{F}_n(W_n^+, W_n^-) \) is computed in the supersonic range as

\[
\tilde{F}_n = \begin{cases}
  \tilde{F}_n(W_n^+, 0) = \tilde{F}_n(W_n^+) & \text{if } M_n \geq 1 \\
  \tilde{F}_n(0, W_n^-) = \tilde{F}_n(W_n^-) & \text{if } M_n \leq -1
\end{cases}
\]

(2.36)
where the one-sided flux function \( \tilde{F}_n \) is defined as

\[
\tilde{F}_n(W_{\pm}^n) = \begin{cases} 
\rho_{\pm} u_{n\pm}^\pm & \text{if } |M_n| \geq 1 \\
\rho_{\pm} (u_{n\pm}^\pm)^2 + p_{\pm} & \\
\rho_{\pm} u_{n\pm}^\pm & \\
\left(\rho_{\pm}E_{\pm} + p_{\pm}\right) u_{n\pm}^\pm 
\end{cases}
\]

such that

\[
T_p \tilde{F}_n(W_{n\pm}^\pm) = F_x(U_{\pm}^n) n_x + F_y(U_{\pm}^n) n_y
\]

Notice that the superscript “+” refers to waves moving forwards and the superscript “−” to those moving backwards.

On the other hand, if the flow is subsonic (\( |M_n| < 1 \)) information at any interface may be convected from both sides, upstream and downstream. In other words, when \( |M_n| < 1 \) the double-sided flux function \( \tilde{F}_n(W_{n+}^+, W_{n-}^-) \) is computed through the contributions from a forward moving wave, \( \tilde{F}_n^+(W_{n+}^+) \), and a backward moving one, \( \tilde{F}_n^-(W_{n-}^-) \), as

\[
\tilde{F}_n(W_{n+}^+, W_{n-}^-) = \tilde{F}_n^+(W_{n+}^+) + \tilde{F}_n^-(W_{n-}^-) \quad \text{if } |M_n| < 1
\]

where the normal flux functions \( \tilde{F}_n^+ \) and \( \tilde{F}_n^- \) are given \[83\] by

\[
\tilde{F}_n^\pm(W_{n\pm}^\pm) = \pm \frac{\rho_{n\pm}^\pm}{4a_{n\pm}^\pm}\left(\frac{\gamma - 1}{\gamma}u_{n\pm}^\pm + \frac{2a_{n\pm}^\pm}{\gamma}\right)
\]

\[
\left\{ 
\begin{array}{c}
\frac{1}{\gamma - 1}u_{n\pm}^\pm + 2a_{n\pm}^\pm \\
u_{n\pm}^\pm \\
\frac{(\gamma - 1)u_{n\pm}^\pm + 2a_{n+}^\pm}{2(\gamma^2 - 1)} + \frac{(u_{n\pm}^\pm)^2}{2}
\end{array}
\right\}
\]

38
or, in terms of the normal Mach number $M_n$, 

$$
\tilde{F}_n^\pm (W^\pm) = \pm \frac{\rho a^\pm}{4} (M_n^\pm \pm 1)^2 \begin{pmatrix}
1 \\
\frac{a^\pm}{\gamma} \frac{[(\gamma - 1) M_n^\pm \pm 2]}{u_i^\pm} \\
\frac{(a^\pm)^2 [(\gamma - 1) M_n^\pm \pm 2]}{2(\gamma^2 - 1)} + \frac{\left(u_i^\pm\right)^2}{2}
\end{pmatrix}
$$

(2.41)

It is important to notice that $\tilde{F}_n \neq \tilde{F}_n^\pm$. The former are one-sided fluxes in the supersonic case, given by equation (2.37), whereas the latter are the subsonic Van Leer flux functions of the contributions from positive and negative moving waves to the numerical flux, given by equations (2.40)-(2.41). Nevertheless, similarly to $\tilde{F}_n$, Van Leer flux functions $\tilde{F}_n^\pm$ satisfy the following condition

$$
T \hat{P} \tilde{F}_n^\pm (W_n^+, W_n^-) = F_x (U^+) n_x + F_y (U^-) n_y
$$

(2.42)

Bearing in mind the definition of the numerical flux function $\tilde{F}_n$ across the whole Mach range in (2.36) and (2.39), equations (2.38) and (2.42) can be combined to yield

$$
T \hat{P} \tilde{F}_n (W_n^+, W_n^-) A = F_x (U) A_x + F_y (U) A_y
$$

(2.43)

Hence, using (2.43) in equation (2.29) yields

$$
\frac{d}{dt} (U_i V_i) + \sum_{k=1}^{n_f} \left[ T \hat{P} \tilde{F}_n (W_n^+, W_n^-) - (F_x^v (U) n_x + F_y^v (U) n_y) \right] A_k = 0
$$

(2.44)

Equivalently, as done in Section 2.2.2, using the viscous fluxes matrix $M^v$, given by equation (2.7), leads to

$$
\frac{d}{dt} (U_i V_i) + \sum_{k=1}^{n_f} \left[ T \hat{P} \tilde{F}_n (W_n^+, W_n^-) - M^v n \right] A_k = 0
$$

(2.45)

Notice that diffusive term $(F_x^v (U) n_x + F_y^v (U) n_y) A$ or, equivalently, $M^v n A$ are still discretised via a second-order central-difference scheme as in Section 2.2.2.
Flux-difference method by Roe

In the flux-difference method proposed by Roe [93] the fluxes at a face are calculated by means of the conserved variables evaluated at both sides of the interface, $U^+$ and $U^-$, as a combination of forward and backward moving waves. Writing them in terms of normal flux functions, as done for Van Leer fluxes, yields at each face

$$F_n (U^+, U^-) = \frac{1}{2} [F_n (U^+) + F_n (U^-) - \Delta F_n]$$  \hspace{1cm} (2.46)

where the “jump” $\Delta F_n$ of the fluxes across the interface is defined as

$$\Delta F_n = \tilde{A}_n \Delta U$$  \hspace{1cm} (2.47)

The “jump” of conserved variables is the column vector $\Delta U = U^+ - U^-$ and the so-called Roe matrix is given by

$$\tilde{A}_n = \begin{bmatrix}
0 & n_x & n_y & 0 \\
(n - 1) \frac{q}{2} - \hat{u}^2 & n_x - \hat{u} \hat{v} n_y & (3 - \gamma) \hat{u} n_x + \hat{v} n_y & - (n - 1) \hat{v} n_x + \hat{u} n_y & (n - 1) n_x \\
(n - 1) \frac{q}{2} - \hat{v}^2 & n_y - \hat{u} \hat{v} n_x & \hat{v} n_x - (n - 1) \hat{u} n_y & \hat{u} n_x + (3 - \gamma) \hat{v} n_y & (n - 1) n_y \\
(n - 1) \frac{q}{2} - \hat{H} & \hat{u} n_x - (n - 1) \hat{u} \hat{u} n_x & \hat{H} n_x - (n - 1) \hat{u} \hat{v} n_x & \hat{H} n_y - (n - 1) \hat{v} \hat{u} n_y & (n - 1) \hat{u} n_y
\end{bmatrix}$$  \hspace{1cm} (2.48)

where the normal and tangential velocities are defined as

$$\hat{u}_n = \hat{u} n_x + \hat{v} n_y$$  \hspace{1cm} (2.49)

$$\hat{u}_t = \hat{u} n_y - \hat{v} n_x$$  \hspace{1cm} (2.50)

the enthalpy is given by

$$\hat{H} = \hat{E} + \frac{\hat{p}}{\rho}$$  \hspace{1cm} (2.51)
and \( \hat{q} \) is defined as the sum of the square of the velocity components, i.e.

\[
\hat{q} = \hat{u}^2 + \hat{v}^2
\]  

(2.52)

The “hat” notation has been used here to refer to Roe-averaged quantities. Denoting the values of these quantities at each side of the face with superscripts + and −, the Roe averages may be worked out as

\[
\hat{\rho} = \sqrt{\rho^+ \rho^-}, \quad \hat{u} = \frac{\sqrt{\rho^+ u^+ \sqrt{\rho^- u^-}}}{\sqrt{\rho^+} + \sqrt{\rho^-}}, \quad \hat{v} = \frac{\sqrt{\rho^+ v^+ \sqrt{\rho^- v^-}}}{\sqrt{\rho^+} + \sqrt{\rho^-}}, \quad \hat{H} = \frac{\sqrt{\rho^+ H^+ \sqrt{\rho^- H^-}}}{\sqrt{\rho^+} + \sqrt{\rho^-}}
\]  

(2.53)

The right eigenvectors of \( \tilde{A}_n \) can be written as

\[
r_1 = \begin{pmatrix} 1 \\ \hat{u} \\ \hat{v} \\ \hat{q}/2 \end{pmatrix}, \quad r_2 = \begin{pmatrix} 0 \\ n_y \\ -n_x \\ \hat{u}_t \end{pmatrix}, \quad r_{3,4} = \begin{pmatrix} 1 \\ \hat{u} \pm \hat{a} n_x \\ \hat{v} \pm \hat{a} n_y \\ \hat{H} \pm \hat{a} \hat{u}_n \end{pmatrix}
\]  

(2.54)

and are associated with the eigenvalues

\[
\lambda_1 = \hat{u}_n, \quad \lambda_2 = \hat{u}_n, \quad \lambda_{3,4} = \hat{u}_n \pm \hat{a}
\]  

(2.55)

where \( \hat{a} = \sqrt{\gamma \hat{p} / \hat{\rho}} \) is the Roe-averaged speed of sound. The “jump” \( \Delta \mathbf{U} \) of conserved variables can be written in terms of the eigenvectors \( \mathbf{r}_i \), which form a basis, i.e.

\[
\Delta \mathbf{U} = \sum_i \alpha_i \mathbf{r}_i
\]  

(2.56)

If \( \mathbf{l}_i \) are the left eigenvectors of matrix \( \tilde{A}_n \), coefficients \( \alpha_i \) can be calculated from the property of orthogonality of the left and right eigenvectors (\( \mathbf{l}_i^T \mathbf{r}_j = \delta_{ij} \) where \( \delta_{ij} \) is the Kronecker delta) as

\[
\alpha_i = \mathbf{l}_i^T \Delta \mathbf{U}
\]  

(2.57)
or, equivalently, as

\[ \alpha_1 = \gamma - \frac{1}{a^2} \left[ (\hat{H} - \hat{q}) \Delta u_1 + \hat{u} \Delta u_2 + \hat{v} \Delta u_3 - \Delta u_4 \right] \] (2.58)

\[ \alpha_2 = -\hat{u}_1 \Delta u_1 + n_y \Delta u_2 - n_x \Delta u_3 \] (2.59)

\[ \alpha_3 = \gamma - \frac{1}{2a^2} \left\{ \left[ \frac{\hat{q}}{2} \left( 1 + \frac{\hat{u}_n}{\hat{a}} \right) - \frac{\hat{H} \hat{u}_n}{\hat{a}} \right] \Delta u_1 + \left( \frac{\hat{H} n_x}{\hat{a}} - \frac{\hat{q} n_x}{2\hat{a}} - \hat{u} \right) \Delta u_2 + \left( \frac{\hat{H} n_y}{\hat{a}} - \frac{\hat{q} n_y}{2\hat{a}} - \hat{v} \right) \Delta u_3 + \Delta u_4 \right\} \] (2.60)

\[ \alpha_4 = \gamma - \frac{1}{2a^2} \left\{ \left[ \frac{\hat{q}}{2} \left( 1 - \frac{\hat{u}_n}{\hat{a}} \right) + \frac{\hat{H} \hat{u}_n}{\hat{a}} \right] \Delta u_1 - \left( \frac{\hat{H} n_x}{\hat{a}} - \frac{\hat{q} n_x}{2\hat{a}} + \hat{u} \right) \Delta u_2 - \left( \frac{\hat{H} n_y}{\hat{a}} - \frac{\hat{q} n_y}{2\hat{a}} + \hat{v} \right) \Delta u_3 + \Delta u_4 \right\} \] (2.61)

where \( \Delta u_i \) are each of the components of the column vector \( \Delta \mathbf{U} \). Finally, combining equations (2.47) and (2.56), and bearing in mind that \( \bar{A}_{n r i} = \lambda_i r_i \), yields

\[ \Delta \mathbf{F}_n = \sum_i \alpha_i \lambda_i r_i \] (2.62)

Notice that the flux function \( \mathbf{F}_n (\mathbf{U}^+, \mathbf{U}^-) \), defined in equation (2.46), gives the normal contributions of the fluxes across each cell interface. Therefore, it is possible to write

\[ \mathbf{F}_n (\mathbf{U}^+, \mathbf{U}^-) = \mathbf{F}_x (\mathbf{U}) n_x + \mathbf{F}_y (\mathbf{U}) n_y \] (2.63)

Hence, using (2.63) into equation (2.29) leads to

\[ \frac{d}{dt} (\mathbf{U}_i V_i) + \sum_{k=1}^{n_f} \left[ \mathbf{F}_n (\mathbf{U}^+, \mathbf{U}^-) - (\mathbf{F}_x (\mathbf{U}) n_x + \mathbf{F}_y (\mathbf{U}) n_y) \right] k A_k = 0 \] (2.64)

Alternatively, using the viscous fluxes matrix \( \mathbf{M}^v \), given by equation (2.7), yields

\[ \frac{d}{dt} (\mathbf{U}_i V_i) + \sum_{k=1}^{n_f} \left[ \mathbf{F}_n (\mathbf{U}^+, \mathbf{U}^-) - \mathbf{M}^v n \right] k A_k = 0 \] (2.65)
2.2. Spatial discretisation of the physical equations

Extrapolation to face values and slope limiters

In the central-difference scheme the value of the primitive variables at each face is worked out as an average of the values at the neighbouring cells. In the upwind scheme, however, the direction of propagation of some quantities determines whether the values used at the cell interface are taken from the upstream or downstream neighbour cell. First-order methods use the cell-centred value of the neighbour cell as the value at the face. When extra accuracy is required, a second-order correction term may be added to the cell-centred values. These second-order correction terms lead to spurious oscillations which may be avoided by the introduction of the so-called slope limiters $\phi_x$ and $\phi_y$ which limit the rate of change of fluid properties across each cell. If superscripts $\text{f}$ and $\text{c}$ denote the face and cell centres, respectively, face values can be worked out from cell-centre values as

$$W^\text{f} = W^\text{c} + \nabla W^\text{c} \cdot \Phi \cdot (x^\text{f} - x^\text{c})$$

where $x = [x, y]^T$ is the column vector of coordinates, the matrix of slope limiters $\Phi$ is defined as

$$\Phi = \begin{bmatrix} \phi_x & 0 \\ 0 & \phi_y \end{bmatrix}$$

and the matrix of gradients of the primitive variables $\nabla W^\text{c} = \frac{\partial W^\text{c}}{\partial \mathbf{x}}$ is given by

$$\nabla W^\text{c} = \begin{bmatrix} \frac{\partial W_1^\text{c}}{\partial x_j} \\ \frac{\partial W_2^\text{c}}{\partial x_j} \end{bmatrix} = \begin{bmatrix} \frac{\partial W_1^\text{c}}{\partial x} & \frac{\partial W_1^\text{c}}{\partial y} \\ \frac{\partial W_2^\text{c}}{\partial x} & \frac{\partial W_2^\text{c}}{\partial y} \end{bmatrix}$$

The limiters have been chosen to comply with TVD (Total Variation Diminishing) conditions [84] and depend upon the changes of the fluid variables in the neighbourhood of the cell. To account for these changes one can define the monitor $r_i$ at cell $i$ as

$$r^i = \min \left\{ \nabla W_1^\text{c}, \ldots, \nabla W_{N_i}^\text{c} \right\}$$

where $N_i$ is the number of neighbouring cells. In the specific literature there are many available methods for
the computation of the limiters. One of the most well-known slope limiters, and the one used here, is due to
Van Leer [83]

\[ \varphi_{VL} (r) = \frac{r + |r|}{1 + |r|} \quad (2.70) \]

Other slope limiters can be found in Figure 2.3.

### 2.3 Spatial discretisation of the Spalart-Allmaras turbulence model

The integral form of the standard Spalart-Allmaras one-equation model, equation (C.22) in Appendix C, may
be written as

\[ \frac{d}{dt} \int_V \eta \, d\Omega + \int_{\partial V} \mathbf{H}^T \cdot \mathbf{n} \, dS = \int_V Q \, d\Omega \quad (2.71) \]

where the source term \( Q \) is

\[ Q = c_{b1} (1 - f_{l2}) \left( |\Omega| + \frac{1}{Re_{\infty}} \frac{\nu}{d^2} f_{v2} \right) \frac{\nu}{\mu} - \frac{1}{Re_{\infty}} \left( c_{w1} f_w - c_{b1} \frac{\nu}{d^2} f_{l2} \right) \left( \frac{\nu}{d^2} \right) \]

\[ - \frac{1}{\sigma_{SA} Re_{\infty}} \left\{ c_{b2} (\mu + \nu) \frac{\partial}{\partial x_j} \left( \frac{\partial \nu}{\partial x_j} \right) + (1 + c_{b2}) (\nu + \nu) \frac{\partial \nu}{\partial x_j} \frac{\partial \rho}{\partial x_j} \right\} \quad (2.72) \]
2.3. Spatial discretisation of the Spalart-Allmaras turbulence model

and the Spalart-Allmaras fluxes vector $\mathbf{H}$ is

$$\mathbf{H} = \mathbf{H}^c + \mathbf{H}^d$$

(2.73)

with the convective $\mathbf{H}^c$ and diffusive $\mathbf{H}^d$ contributions given by

$$\mathbf{H}^c = \begin{Bmatrix} \overline{u} \\ \overline{v} \end{Bmatrix} \quad \mathbf{H}^d = -\frac{1 + c_{b2}}{\sigma_{SA} Re_{\infty}} (\mu + \overline{n}) \begin{pmatrix} \frac{\partial \overline{\sigma}}{\partial x} \\ \frac{\partial \overline{\sigma}}{\partial y} \end{pmatrix}$$

(2.74)

In the case of the negative version of the Spalart-Allmaras model, equation (C.34), the source term $Q$ is

$$Q = c_{b1} (1 - c_{33}) |\Omega| \overline{\sigma} + \frac{c_{w1}}{Re_{\infty}} \frac{\partial \overline{\sigma}}{\partial x} - \frac{1}{\sigma_{SA} Re_{\infty}} \left( \frac{c_{b2}}{f_n} (\mu + \overline{n} f_n) \frac{\partial \sigma}{\partial x_j} \right)$$

$$+ c_{b2} \left( \mu + \left( f_n + \frac{1}{f_n} \right) \overline{n} \right) \frac{\partial \overline{\sigma}}{\partial x_j} \frac{\partial f_n}{\partial x_j} + \left( 1 + \frac{c_{b2}}{f_n} \right) (\nu + \overline{n} f_n) \frac{\partial \overline{\sigma}}{\partial x_j} \frac{\partial \rho}{\partial x_j}$$

$$+ c_{b2} \left( \mu + \left( f_n + 1 \right) f_n \right) \frac{\partial \overline{\sigma}}{\partial x_j} \frac{\partial f_n}{\partial x_j} + \left( 1 + \frac{c_{b2}}{f_n} \right) (\nu + \overline{n} f_n) \frac{\partial \overline{\sigma}}{\partial x_j} \frac{\partial \rho}{\partial x_j}$$

(2.75)

and the convective $\mathbf{H}^c$ and diffusive $\mathbf{H}^d$ contributions are given in this case by

$$\mathbf{H}^c = \begin{Bmatrix} \overline{u} \\ \overline{v} \end{Bmatrix} \quad \mathbf{H}^d = -\frac{1}{\sigma_{SA} Re_{\infty}} \left( 1 + \frac{c_{b2}}{f_n} \right) (\mu + \overline{n} f_n) \begin{pmatrix} \frac{\partial \overline{\sigma}}{\partial x} \\ \frac{\partial \overline{\sigma}}{\partial y} \end{pmatrix}$$

(2.76)

The semi-discretisation (i.e. discretisation of the spatial domain only) of equation (2.71) in the case of unstructured grids (see Figure 2.1) yields

$$\frac{d}{dt} (\overline{\rho} V_i) + \sum_{k=1}^{n_f} \left[ (\mathbf{H}^c + \mathbf{H}^d) n A \right]_k = Q_i V_i$$

(2.77)

In order to ensure numerical stability of the solution [97], a second order central-difference scheme is used for the diffusive terms $\mathbf{H}^d$ and a first-order upwind scheme for the convective terms $\mathbf{H}^c$, regardless of the stencil used in the discretisation of the Navier-Stokes equations. In other words, the value of the diffusive fluxes at a
face \( k \) in the mesh is given by

\[
(H^d)_k = \frac{(H^d)_{l(k)} + (H^d)_{r(k)}}{2}
\]

where here, again, \( l(k) \) and \( r(k) \) are the left and right neighbours at face \( k \), respectively; and the value of the convective fluxes is given by

\[
(H^c)_k = \begin{cases} 
(H^c)_{r(k)} & \text{if } v \cdot n \geq 0 \\
(H^c)_{l(k)} & \text{if } v \cdot n < 0 
\end{cases}
\]

where \( v \) is the flow velocity and \( n \) is the normal vector at face \( k \) which, by convention, points towards the left neighbour cell.

### 2.4 Spacetime discretisation of the physical equations

Only two-dimensional problems have been computed in this thesis, therefore the following discretisations will consider the case of a three-dimensional spacetime domain (referred to as 2D+t), i.e. two spatial dimensions and time. As explained for the two-dimensional case above, three different stencils have been implemented for spacetime framework: a central-difference and two upwind formulations. Moreover, a hybrid formulation which combines a central-difference stencil for spatial fluxes and an upwind stencil for fluxes in the time direction has also been implemented.

#### 2.4.1 Integral formulation of the Navier-Stokes equations

The integral form of the two-dimensional Navier-Stokes equations is given in equation (2.1). As outlined before, the spacetime framework is based on a coupled integration of the physical equations across the spatial and temporal domain. Therefore, integrating (2.1) between an initial and final time, \( t_0 \) and \( t_F \), and re-arranging terms yields

\[
\int_{t_0}^{t_F} \int_V \left[ \frac{\partial U}{\partial t} + \frac{\partial}{\partial x} (F_x - F_x^v) + \frac{\partial}{\partial y} (F_y - F_y^v) \right] d\Omega dt = 0
\]
2.4. Spacetime discretisation of the physical equations

where $V$ denotes the volume, the column vector of conserved variables $U$ is

$$
U = \begin{bmatrix}
\rho \\
\rho u \\
\rho v \\
\rho E
\end{bmatrix}
$$

(2.81)

the column vectors of inviscid fluxes along Cartesian coordinates $x = x_1$ and $y = x_2$, namely $F_x$ and $F_y$, respectively, are

$$
F_x = \begin{bmatrix}
\rho u \\
\rho u^2 + p \\
\rho uv \\
(\rho E + p) u
\end{bmatrix} \\
F_y = \begin{bmatrix}
\rho v \\
\rho uv \\
\rho v^2 + p \\
(\rho E + p) v
\end{bmatrix}
$$

(2.82)

and the column vectors of viscous fluxes along Cartesian coordinates $x = x_1$ and $y = x_2$, namely $F^v_x$ and $F^v_y$, respectively, are

$$
F^v_x = \begin{bmatrix}
0 \\
\sigma_{xx} \\
\sigma_{xy} \\
\omega \sigma_{xx} + \nu \sigma_{xy} + q_x
\end{bmatrix} \\
F^v_y = \begin{bmatrix}
0 \\
\sigma_{xy} \\
\sigma_{yy} \\
\omega \sigma_{xy} + \nu \sigma_{yy} + q_y
\end{bmatrix}
$$

(2.83)

where $u = u_1$ and $v = u_2$ are the velocity components along Cartesian coordinates $x$ and $y$. Notice the difference between convective fluxes in equation (2.3), corresponding to the arbitrary Lagrangian-Eulerian formulation, and those in equation (2.82), corresponding to the spacetime formulation. In the former case, the spatial mesh moves between successive time-steps whereas, in the latter, the (spacetime) mesh remains fixed throughout (pseudo-)time, hence the different terms.

The key aspect of the spacetime method is the treatment of the time and space integrations identically through the use of four-dimensional spacetime finite-volumes. Within this framework any unsteady problem of dimension $N$ can be effectively solved as another steady problem of dimension $N + 1$. In the current two-dimensional problem, a three-dimensional divergence operator can be defined as (the subscript ST means spacetime)

$$
\nabla_{ST} = \frac{\partial}{\partial t} e_t + \frac{\partial}{\partial x} e_x + \frac{\partial}{\partial y} e_y
$$

(2.84)
which leads to the three-dimensional spacetime formulation of the Navier-Stokes equations as

$$
\int_{V_{ST}} \nabla_{ST} \cdot (M_{ST} - M_{vST}) \, d\Omega_{ST} = 0
$$

or expressed as a closed surface integral through the application of the Green-Gauss (or divergence) theorem

$$
R_{ST} (U, \nabla U) = 0
$$

where the column vector of spacetime residuals is

$$
R_{ST} (U, \nabla U) \equiv \oint_{\partial V_{ST}} (M_{ST} - M_{vST}) \, n_{ST} \, dS_{ST}
$$

the matrix of inviscid spacetime fluxes $M_{ST}$ is

$$
M_{ST} (U, \nabla U) = \left[ \begin{array}{c|c|c|c}
U & F_x & F_y \\
\hline
0 & 0 & 0 & 0
\end{array} \right]
$$

the matrix of viscous spacetime fluxes $M_{vST}$ is

$$
M_{vST} (U, \nabla U) = \left[ \begin{array}{c|c|c|c}
0 & 0 & 0 & 0 \\
0 & F_x^v & F_y^v & 0 \\
0 & 0 & 0 & 0
\end{array} \right]
$$

and the spacetime normal vector $n_{ST}$ is

$$
n_{ST} = \begin{pmatrix} n_t \\ n_x \\ n_y \end{pmatrix}
$$
2.4. Spacetime discretisation of the physical equations

Equation (2.86) can be regarded as the integration of the steady Navier-Stokes equations in (2.10) across a theoretical three-dimensional space and, as such, it can be solved by means of an explicit solver, marching in pseudo-time until the $l_2$-norms of the residuals converge to zero (see equation (3.21) in Section 3.4). In fact, any solution method may be used to solve equation (2.86), not only limited to a dual-time stepping technique, just like any other steady-state simulation. In particular, if desired an implicit solver may take advantage of an unbounded CFL number without a penalty in the time accuracy of the solution.

It is important to bear in mind that the spacetime formulation of the Navier-Stokes equations for fluids motion presented here, based on an $N + 1$ dimensional finite-volume discretisation, is general. It has no limitations in regard to the orientation of faces in spacetime. Here, in contrast with previous attempts to formulate aerodynamics problems in spacetime [56], faces can have any orientation without the need for time-aligned faces in the mesh. The splitting of spatial and temporal fluxes in an arbitrarily oriented face is based on the direction of its normal vector in spacetime, as depicted, for example, in Figure 2.5.

Unlike commonly applied methods, the use of a finite-volume approach for the discretization in time, as well as in space, ensures the automatic conservation of mass, momentum and energy and, more importantly, allows the use of a variable real timestep across the spatial domain without causing a non-physical behaviour of the solution. Notice the potential gain in efficiency over conventional time-stepping techniques due to the fact that a bigger timestep can be used in areas of freestream flow, far away from the perturbations, whilst still retaining sufficiently small timesteps in areas where rapid changes occur. In terms of solution accuracy the spacetime method brings the possibility to incorporate some of the high-order schemes used for spatial discretizations into the temporal dimension [71].

The spacetime framework works well with any arbitrary motion, from big boundary displacements, like a helicopter rotor blade, through to geometric topological changes such as a store separation or a slotted flap deflection. There is no need for further modifications to the solver in any of the former cases, mainly as a consequence of the finite-volume approach in time. All the information related to boundary motions is implicitly given by the spacetime mesh. An example of this is depicted in Figure 2.4 where the pitching movement of a NACA 0012 aerofoil is given by a twisted wing in which the span-wise direction represents the time. Moreover, no connectivity information is required between cells at different time levels since this is implicitly accomplished.

Note that it would not be possible to talk about cells being at a certain time level since different cells span between different time levels.
by the spacetime mesh, therefore allowing for appearing/disappearing cells without the need for interpolation. A summary of the main strengths and weaknesses of the spacetime framework is given in Table 1.1.

2.4.2 Second order central-difference scheme

Adding a derivative in pseudo-time $t^*$ the semi-discretisation (i.e. discretisation of the spacetime domain only, not in pseudo-time) of equation (2.86) using the second order central-difference JST scheme [78] yields, in the case of unstructured grids (see Figure 2.1),

$$
\frac{d}{dt^*} (U_i V_i) + \sum_{k=1}^{n_f} \left\{ \left[ UA_x + (F_x - F^*_x) A_x + (F_y - F^*_y) A_y \right]_k + D_k \right\} = 0
$$

(2.91)

or, using the matrices of spacetime fluxes $M_{ST}$ and $M^v_{ST}$ given in equations (2.88) and (2.89), respectively,

$$
\frac{d}{dt^*} (U_i V_i) + \sum_{k=1}^{n_f} \left\{ \left[ (M_{ST} - M^v_{ST}) n_{ST} A \right]_k + D_k \right\} = 0
$$

(2.92)

where the subscript $i$ denotes a cell with spacetime volume $V_i$ and the subscript $k$ denotes each of its $n_f$ spacetime faces with area $A_k$. The values $A_t = A_k n_t$, $A_x = A_k n_x$ and $A_y = A_k n_y$ are the face area projected normal to the $t$, $x$ and $y$ coordinate axes, respectively. The subscript “ST” has not been used in the volume and areas to avoid overloading the notation. Moreover, matrices $M_{ST}$ and $M^v_{ST}$ are of size $4 \times 3$ because the two-dimensional case is being considered, i.e. the spacetime domain is three-dimensional ($2D+t$). Denoting $l(k)$ and $r(k)$ to the left and right cell indices at a face $k$, respectively, fluxes are worked out as the flux of the average of conserved variables on the left and right cells

$$
(F_{x,y})_k = F_{x,y} \left( \frac{U_{l(k)} + U_{r(k)}}{2} \right) \quad \quad (F^v_{x,y})_k = F^v_{x,y} \left( \frac{U_{l(k)} + U_{r(k)}}{2} \right)
$$

(2.93)

$$
(M)_k = M \left( \frac{U_{l(k)} + U_{r(k)}}{2} \right) \quad \quad (M^v)_k = M^v \left( \frac{U_{l(k)} + U_{r(k)}}{2} \right)
$$

(2.94)

Alternatively, the average of the fluxes on both sides at face $k$ may be used

$$
(F_{x,y})_k = \frac{F_{x,y} (U_{l(k)}) + F_{x,y} (U_{r(k)})}{2} \quad \quad (F^v_{x,y})_k = \frac{F^v_{x,y} (U_{l(k)}) + F^v_{x,y} (U_{r(k)})}{2}
$$

(2.95)
2.4. Spacetime discretisation of the physical equations

\[ (\text{M}_k) = \frac{M (U_{l(k)}) + M (U_{r(k)})}{2} \quad (\text{M}'_k) = \frac{M' (U_{l(k)}) + M' (U_{r(k)})}{2} \] (2.96)

The time fluxes at face \( k \) are computed as

\[ (\text{U}_k) = \frac{U_{l(k)} + U_{r(k)}}{2} \] (2.97)

The numerical dissipation term \( D_k \) at face \( k \) is constructed as follows

\[ D_k = \left[ \varepsilon^{(2)}_k (U_{r(k)} - U_{l(k)}) - \varepsilon^{(4)}_k \left( \nabla^2 U_{r(k)} - \nabla^2 U_{l(k)} \right) \right] \Xi_k \vartheta_k \] (2.98)

where the undivided Laplacian operator can be computed as

\[ \nabla^2 U_i = \sum_{j=1}^{N_i} (U_j - U_i) = -N_i U_i + \sum_{j=1}^{N_i} U_j \] (2.99)

with \( N_i \) being the number of neighbours at cell \( i \). The local spectral radii \( \vartheta_k \) and \( \Upsilon_i \) are worked out at face \( k \) and cell \( i \), respectively, as follows

\[ \vartheta_k = (|v_{ST} \cdot n_{ST}|_k + a_k) S_k \] (2.100)

\[ \Upsilon_i = \sum_{k=1}^{N_i} \vartheta_k = \sum_{k=1}^{N_i} (|v_{ST} \cdot n_{ST}|_k + a_k) S_k \] (2.101)

where \( a_k \) is the local speed of sound at face \( k \)

\[ a_k = \frac{a_{l(k)} + a_{r(k)}}{2} \] (2.102)

and \( (v_{ST})_k = \{1, u_k, v_k\}^T \) is the spacetime fluid flow velocity at face \( k \) which includes a time component and

\[ u_k = \frac{u_{l(k)} + u_{r(k)}}{2} \quad v_k = \frac{v_{l(k)} + v_{r(k)}}{2} \] (2.103)
Chapter 2: Discretisation methods

Factor $\Xi_k$ is defined in order to account for the stretching of the grid as

$$
\Xi_k = \left( \frac{4}{T_{l(k)} + T_{r(k)}} \right) \cdot \left( \frac{Y_{l(k)} Y_{r(k)}}{4\theta_k} \right)^p
$$

(2.104)

where $p = 0.3$ as proposed by [78]. The switches $\varepsilon^{(2)}_k$ and $\varepsilon^{(4)}_k$ are functions of local pressure gradients and can be computed as

$$
\varepsilon^{(2)}_k = \frac{\varepsilon^{(2)}_{l(k)} + \varepsilon^{(2)}_{r(k)}}{2}
$$

(2.105)

$$
\varepsilon^{(4)}_k = s_4 \max \left( 0, \kappa^{(4)} - \varepsilon^{(2)}_k \right)
$$

(2.106)

where $\kappa^{(2)}$ and $\kappa^{(4)}$ are some closure constants and good values for them are $\kappa^{(2)} \simeq 1$, $\kappa^{(4)} \simeq 0.05$. The scaling factors are $s_2 = 2$ and $s_4 = 1$ for triangular meshes, and $\varepsilon^{(2)}_i$ is a switch defined at cell $i$ as follows

$$
\varepsilon^{(2)}_i = \kappa^{(2)} s_2 \left( \frac{\sum_{j=1}^{N_i} (p_j - p_i)}{\sum_{j=1}^{N_i} (p_i + p_j)} \right)
$$

(2.107)

The scaling factor $s_2$ is defined such that the dissipation matches the dissipation of the structured formulation in the case of an hexahedral mesh in spacetime, i.e. with six neighbours,

$$
N_2 = \frac{3 \left( N_{l(k)} + N_{r(k)} \right)}{N_{l(k)} N_{r(k)}}
$$

(2.108)

where, again, $N_i$ is the number of neighbours at cell $i$. Finally, scaling factor $s_4$ is defined, in relation to $s_2$, as

$$
N_4 = \left( \frac{s_2}{2} \right)^2
$$

(2.109)

Fluxes are calculated independently of the flow direction, namely they are the same regardless of whether the flow comes from left to right or from right to left. This central-difference approach in spacetime leads to a non-physical propagation of waves as will be shown later. In an attempt to overcome this issue two upwind stencils are implemented next.
2.4.3 Second order upwind scheme

The integration of the Navier-Stokes equations in space and time using a conventional approach, i.e. through a dual time-stepping technique or an implicit solver (e.g. [(E)S]DIRK), leads to a backward stencil on the time derivatives. In other words, the solution at each time level depends only upon the solution calculated at previous and, at most, current time levels. Within the spacetime framework, however, because the integration in space and time happens simultaneously, in the event of choosing a central-difference scheme (see Section 2.4.2) information may be propagated backwards in time [68, 70, 69, 1]. Namely, whatever happens in the future affects the solution in the past which, obviously, violates the principle of causality. Nevertheless, there are exceptions where, even with a central-difference scheme, the solution obtained is remarkably close to the dual time-stepping counterpart, especially in the case of periodic problems where the notion of future and past events is lost. For the general case, however, a spacetime version of the upwind flux-vector splitting method proposed by Van Leer [83] is implemented in this work aiming to avoid these non-physical phenomena observed with the central-difference formulation. In particular, the two-dimensional formulation (2D+t) is given below.

As in Section 2.2.3, convective terms are now discretised via a second-order upwind scheme whilst diffusive terms are still discretised via central-differences. Therefore, adding a derivative in pseudo-time $t^*$ the semi-discretisation (i.e. discretisation of the spacetime domain only, not in pseudo-time) of equation (2.86) without any added artificial dissipation yields, in the case of unstructured grids (see Figure 2.1),

$$\frac{d}{dt^*} (U_i V_i) + \sum_{k=1}^{n_f} \left[ U A_t + (F_x - F_x^t) A_x + (F_y - F_y^t) A_y \right]_k = 0$$

(2.110)

where the subscript $i$ denotes a cell with spacetime volume $V_i$ and the subscript $k$ denotes each of its $n_f$ spacetime faces with area $A_k$. The values $A_t = A_k n_t$, $A_x = A_k n_x$ and $A_y = A_k n_y$ are the face areas projected normal to the $t$, $x$ and $y$ coordinate axes, respectively. The subscript “ST” has not been used in the volume and areas to avoid overloading the notation.

Following the same approach as in Section 2.2.3, the second-order upwind formulation yields a piecewise linear solution which is discontinuous across the cell faces (see, for instance, interface $C$ in Figure 2.2). In other words, the calculation of inviscid spacetime fluxes, $U$, $F_x$ and $F_y$, relies on the spacetime solution at both sides of the interface, $W^+$ and $W^-$, which are, in general, different (these correspond to $\phi_{C_{low}}$ and $\phi_{C_{high}}$ at interface $C$ in Figure 2.2 of Section 2.2.3). However, the numerical flux at each face must be unique. Therefore, a spacetime version of the flux-vector splitting method developed by Van Leer [83] and the flux-difference method proposed by Roe are implemented hereafter.

Bear in mind that, at every face, spacetime fluxes are split into space and time contributions depending on the direction of the spacetime normal vector, as understood by looking at equation (2.110). See Figure 2.3 for a schematic representation of this.
For inviscid fluxes in time, as implied by a characteristic analysis, a constant pseudo-velocity equal to one can be defined in the time direction, i.e. \( u^* = \frac{dt}{dr} = 1 \). This means that, for a physically meaningful solution, information is always convected only from the upstream cell (or previous in physical time) along the time direction. This corresponds to that with the smallest \( t \) coordinate. Therefore, time fluxes \( \tilde{F}_t(U) = U \) in equation (2.110) must be calculated using the primitive variables evaluated only at the upstream side of the face, i.e. only the forward (or positive) moving wave \( \tilde{F}_t^+ \) (with eigenvalue of \( \partial_t \tilde{F}_t^+ = 1 \geq 0 \)) contributes to the numerical flux in time. Mathematically, time fluxes can be written as

\[
\tilde{F}_t^+ (U^+) = U^+ = \begin{bmatrix}
\rho^+ \\
\rho^+u^+ \\
\rho^+v^+ \\
\rho^+E^+
\end{bmatrix}
\]

(2.111)

where superscript “+” refers to the value of the variable at the side of the face where the forward (or positive) moving wave comes from and is calculated via a linear reconstruction step from cell-centre values, as will be explained later. Notice that the sign of \( \mathbf{n}_{ST} \cdot \mathbf{e}_t \) at each face is used to discern between upstream and downstream cells in the time direction or, in other words, between past and future cells. If \( \mathbf{n}_{ST} \cdot \mathbf{e}_t \leq 0 \) and \( \mathbf{n}_{ST} \) points towards the left cell, then the left side of the face corresponds to the upstream side, and viceversa.

For inviscid fluxes in space, \( \mathbf{F}_x \) and \( \mathbf{F}_y \), information may be convected from both sides, upstream and downstream, if the flow is subsonic but only from the upstream side if the flow is supersonic. Following, their calculation in a spacetime formulation is explained for methods by Van Leer and Roe.
Flux-vector splitting method by Van Leer

The numerical spatial inviscid fluxes across each face on the spacetime normal direction may be given by a function \( \tilde{F}_n (W_n^+, W_n^-) \), which depends on the normal primitive variables at both sides of the interface. These are calculated via a linear reconstruction step from cell-centre values, \( W^+_n \) and \( W^-_n \). Therefore, the definition of \( \tilde{F}_n \) includes the contributions from forward (all eigenvalues of \( \frac{\partial \tilde{F}_n^+}{\partial W_n^+} \geq 0 \)) and backward (all eigenvalues of \( \frac{\partial \tilde{F}_n^-}{\partial W_n^-} \leq 0 \)) moving waves, \( \tilde{F}_n^+ \) and \( \tilde{F}_n^- \) respectively, i.e.

\[
\tilde{F}_n (W_n^+, W_n^-) = \tilde{F}_n^+ (W_n^+) + \tilde{F}_n^- (W_n^-)
\]  

(2.112)

where the subscript \( n \) refers to the direction normal to the face in spacetime since only the normal velocity component gives non-zero fluxes in the momentum equations (see Figure 2.5). \( W_n^+ \) corresponds to the side where the forward (or positive) moving wave comes from. \( W_n^- \) corresponds to the side where the backward (or negative) moving wave comes from. The column vector of normal primitive variables in the present spacetime 2D+\( t \) case is given by

\[
W_n = \begin{pmatrix}
\rho \\
u_n \\
u_{t_1} \\
u_{t_2} \\
p
\end{pmatrix}
\]

(2.113)

where \( \nu_n, \nu_{t_1} \) and \( \nu_{t_2} \) denote the projection of the physical (or real) velocity \( v = \{0, u, v\}^T \) onto a spacetime coordinate system, defined locally at each face, such that axis \( n \) is normal to the face and the other two, \( t_1 \) and \( t_2 \), are tangential. In order to compute these projections, consider transformation \( \mathbf{P} : (t, x, y) \mapsto (x_n, x_{t_1}, x_{t_2}) \) which maps, at every face in the mesh, the global space+time coordinate system to a local spacetime coordinate system such that coordinate \( x_n \) is normal to the face and \( x_{t_1} \) and \( x_{t_2} \) are in-plane directions. Let \( \mathbf{T_P} \) be the matrix associated to this transformation. Notice here the intended distinction between a space+time (\( \mathbb{R}^2 \times \mathbb{R} \)) and a spacetime (\( \mathbb{R}^3 \)) coordinate system. The former is a concatenation of physical space and physical time into one single frame while still keeping space and time coordinates separate. The latter, however, constitutes a real coupling between space and time coordinates such that an increment in one of the coordinates yields increments in both space and time. At this point, an infinite number of different local spacetime coordinate systems (in \( \mathbb{R}^3 \)) may be defined at each face in the mesh. Since the purpose of this projection is the application of Van Leer’s flux-vector splitting method to the spatial fluxes, the spacetime normal vector to the face \( \mathbf{n_{ST}} \) is
Chapter 2: Discretisation methods

taken as the first local direction, i.e.

$$e_n = n_{ST} = \begin{pmatrix} n_t \\ n_y \\ n_x \end{pmatrix}$$ (2.114)

For the second and third local directions there exists an infinite number of possible vectors perpendicular to $e_n$ and between each other such that $e_i \cdot e_j = 0$ for $i \neq j$, as required for the coordinate system to be orthogonal. Nevertheless, for the sake of simplicity, $e_t$ is chosen such that it has a null component in time $t$, i.e. it is perpendicular to vector $e_t$. After normalisation ($|e_t| = 1$) it can be written as follows

$$e_t = \frac{e_n \times e_t}{|e_n \times e_t|} = \begin{pmatrix} 0 \\ \frac{n_y}{\sqrt{n_x^2 + n_y^2}} \\ \frac{-n_x}{\sqrt{n_x^2 + n_y^2}} \end{pmatrix}$$ (2.115)

Notice that the definition of vector $e_t$ (contained within the plane of the face) is given by (2.115) provided that $|e_n \times e_t| \neq 0$. Otherwise, vector $e_t$ is defined such that it is perpendicular to $x$ axis, as follows

$$e_t = \frac{e_n \times e_x}{|e_n \times e_x|} = \begin{pmatrix} \frac{-n_y}{\sqrt{n_x^2 + n_y^2}} \\ 0 \\ \frac{n_x}{\sqrt{n_x^2 + n_y^2}} \end{pmatrix}$$ (2.116)

Assuming $e_t$ is defined by (2.115) (a similar derivation can be made if equation (2.116) is used instead), the third direction may be defined such that the orientation of the local spacetime coordinate system is right-hand oriented or positive, i.e.

$$e_x = \frac{e_n \times e_{t_1}}{|e_n \times e_{t_1}|} = \begin{pmatrix} -\sqrt{n_x^2 + n_y^2} \\ n_x n_t \\ n_y n_t \end{pmatrix}$$ (2.117)

56
2.4. Spacetime discretisation of the physical equations

where, again, a normalisation has been applied so that $||e_t|| = 1$. Matrix $T_P$ of the transformation $P$ can be constructed with the unit vectors of the basis $B' = \{ e_n, e_t, e_t \}$ of the local spacetime coordinate system ($\mathbb{R}^3$) written in terms of the basis $B = \{ e_t, e_x, e_y \}$ of the global space+time coordinate system ($\mathbb{R}^2 \times \mathbb{R}$) as columns, i.e.

$$T_P = \left[ \begin{array}{c} e_n \\ e_t \\ e_t \end{array} \right] = \left[ \begin{array}{ccc} n_t & 0 & -\sqrt{n_x^2 + n_y^2} \\ n_x & n_y & n_x n_t \\ n_y & -n_x & n_y n_t \end{array} \right]$$

(2.118)

And the inverse transformation matrix yields

$$T_P^{-1} = T_P^T = \left[ \begin{array}{ccc} n_t & n_x & n_y \\ 0 & n_x & n_y \\ -\sqrt{n_x^2 + n_y^2} & n_x n_t \end{array} \right]$$

(2.119)

Unless one of the spacetime directions is aligned with the $t$-coordinate, the spacetime coordinates defined locally at the face are not purely spatial and purely temporal but a combination of spatial and temporal coordinates. In other words, the projection of the physical flow velocity $v = \{0, u, v\}^T$, strictly defined in physical space $(x, y)$, onto the local coordinate system yields components in spacetime, as follows,

$$\begin{bmatrix} u_n \\ u_t \\ u_t \end{bmatrix} = T_P^{-1} \begin{bmatrix} 0 \\ u \\ v \end{bmatrix} = \begin{bmatrix} un_x + vn_y \\ vn_x - un_y \sqrt{n_x^2 + n_y^2} \\ n_t (un_x + vn_y) \end{bmatrix}$$

(2.120)

Moreover, let $\hat{P} : W_n \mapsto U$ be the transformation between normal primitive variables in spacetime (in $\mathbb{R}^5$) and conserved variables (in $\mathbb{R}^4$), i.e.

$$U = \hat{P} \left( W_n \right)$$

(2.121)
Then, the matrix $\hat{\mathbf{T}}_\mathbf{P}$ of transformation $\hat{\mathbf{P}}$, of size $4 \times 5$, can be written in terms of the second and third rows of matrix $\mathbf{T}_\mathbf{P}$, given by (2.118), as follows

$$
\hat{\mathbf{T}}_\mathbf{P} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & \mathbf{T}_{P_2} & 0 & 0 & 0 \\
0 & 0 & \mathbf{T}_{P_3} & 0 & 0 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix}
= \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & \frac{n_x}{\sqrt{n_x^2 + n_y^2}} & \frac{n_x n_t}{\sqrt{n_x^2 + n_y^2}} & 0 & 0 \\
0 & -\frac{n_x}{\sqrt{n_x^2 + n_y^2}} & \frac{n_y n_t}{\sqrt{n_x^2 + n_y^2}} & 0 & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
$$

A different approach must be followed to compute $\hat{\mathbf{F}}_n (\mathbf{W}_n^+, \mathbf{W}_n^-)$ depending on whether the flow is locally subsonic or supersonic. Since only the component of the velocity normal to a face gives non-zero fluxes in the momentum equation, the local normal Mach number $M_n = u_n/\sqrt{\gamma p/\rho}$ is used to discern between supersonic and subsonic flow, where $u_n$ is the projection of the physical velocity along the spacetime normal, given by equation (2.120).

On the one hand, if the spatial contribution of the flow is locally supersonic ($|M_n| \geq 1$) fluxes are given by the properties at the upstream side only. In other words, the double-sided flux function $\hat{\mathbf{F}}_n (\mathbf{W}_n^+, \mathbf{W}_n^-)$ is computed in the supersonic range as

$$
\hat{\mathbf{F}}_n = \begin{cases}
\hat{\mathbf{F}}_n (\mathbf{W}_n^+, 0) = \hat{\mathbf{F}}_n (\mathbf{W}_n^+) & \text{if } M_n \geq 1 \\
\hat{\mathbf{F}}_n (0, \mathbf{W}_n^-) = \hat{\mathbf{F}}_n (\mathbf{W}_n^-) & \text{if } M_n \leq -1
\end{cases}
$$

(2.123)

where the one-sided flux function $\hat{\mathbf{F}}_n$ is defined as

$$
\hat{\mathbf{F}}_n (\mathbf{W}_n^+) = \left\{ \begin{array}{ll}
\rho^+ u_n^+ & \\
\rho^+ (u_n^+)^2 + p^+ & \\
\rho^+ u_n^+ t_n^+ & \\
(\rho^+ E^+ + p^+) u_n^+ \\
\end{array} \right.
$$

if $|M_n| \geq 1$

(2.124)

such that

$$
\mathbf{T}_\mathbf{P} \hat{\mathbf{F}}_n (\mathbf{W}_n^+) = \mathbf{F}_x (\mathbf{U}_x^+) n_x + \mathbf{F}_y (\mathbf{U}_y^+) n_y
$$

(2.125)
Notice that the superscript \( + \) refers to waves moving forwards and the superscript \( - \) to those moving backwards.

On the other hand, if the spatial contribution of the flow is locally subsonic (\(|M_n| < 1\)) information at any interface may be convected from both sides, upstream and downstream. In other words, when \(|M_n| < 1\) the double-sided flux function \( \tilde{F}_n(W_n^+, W_n^-) \) is computed through the contributions from a forward moving wave, \( \tilde{F}_n^+ (W_n^+), \) and a backward moving one, \( \tilde{F}_n^- (W_n^-), \) as

\[
\tilde{F}_n(W_n^+, W_n^-) = \tilde{F}_n^+ (W_n^+) + \tilde{F}_n^- (W_n^-) \quad \text{if} \ |M_n| < 1 \tag{2.126}
\]

where the normal flux functions \( \tilde{F}_n^+ \) and \( \tilde{F}_n^- \) are given \[83\] by

\[
\tilde{F}_n^\pm (W_n^\pm) = \pm \frac{\rho_n^\pm}{4a_n^\pm} (u_n^\pm \pm a_n^\pm)^2 \begin{cases} 
1 \\
\frac{\gamma - 1}{\gamma} u_n^\pm \pm \frac{2a_n^\pm}{\gamma} \\
u_n^{t_1} \\
u_n^{t_2} \\
\left[ \frac{(\gamma - 1) u_n^\pm \pm 2a_n^\pm}{2(\gamma^2 - 1)} \right] + \frac{(u_n^{t_1})^2 + (u_n^{t_2})^2}{2}
\end{cases} \tag{2.127}
\]

or, in terms of the normal Mach number \( M_n, \)

\[
\tilde{F}_n^\pm (W_n^\pm) = \pm \frac{\rho_n^\pm a_n^\pm}{\gamma} (M_n^\pm \pm 1)^2 \begin{cases} 
1 \\
\frac{a_n^\pm}{\gamma} [(\gamma - 1) M_n^\pm \pm 2] \\
u_n^{t_1} \\
u_n^{t_2} \\
\left( \frac{a_n^\pm}{\gamma} \right)^2 \left[ \frac{(\gamma - 1) M_n^\pm \pm 2}{2(\gamma^2 - 1)} \right] + \frac{(u_n^{t_1})^2 + (u_n^{t_2})^2}{2}
\end{cases} \tag{2.128}
\]

It is important to notice that \( \tilde{F}_n \neq \tilde{F}_n^\pm. \) The former are one-sided spatial fluxes in the supersonic case, given by equation \(2.124, \) whereas the latter are the subsonic Van Leer flux functions of the contributions from positive and negative moving waves to the numerical spatial flux, given by equations \(2.127-2.128. \) Nevertheless,
similarly to \( \tilde{\bf F}_n \), Van Leer flux functions \( \tilde{\bf F}^\pm_n \) satisfy the following condition

\[
\mathbf{T}_p \, \tilde{\bf F}^\pm_n (\mathbf{W}^\pm_n) = \mathbf{F}_x (\mathbf{U}^\pm) \, n_x + \mathbf{F}_y (\mathbf{U}^\pm) \, n_y
\]  

(2.129)

Bearing in mind the definition of the numerical flux function \( \bar{\bf F}_n \) across the whole Mach range in (2.123) and (2.126), equations (2.125) and (2.129) can be combined to yield

\[
\mathbf{T}_p \, \bar{\bf F}_n (\mathbf{W}_n^+, \mathbf{W}_n^-) A = \mathbf{F}_x (\mathbf{U}) \, A_x + \mathbf{F}_y (\mathbf{U}) \, A_y
\]  

(2.130)

Hence, using equations (2.111) and (2.130) in equation (2.110) leads to

\[
\frac{d}{dt} \left( \mathbf{U}_i \mathbf{V}_i \right) + \sum_{k=1}^{n_f} \left[ \mathbf{U}_n + \mathbf{T}_p \, \bar{\bf F}_n (\mathbf{W}_n^+, \mathbf{W}_n^-) - (\mathbf{F}_x n_x + \mathbf{F}_y n_y) \right] A_k \bigg|_{\mathbf{A}_k} = 0
\]  

(2.131)

or, using the matrix of spacetime viscous fluxes \( \mathbf{M}^v_{\text{ST}} \), given in equation (2.89), yields

\[
\frac{d}{dt} \left( \mathbf{U}_i \mathbf{V}_i \right) + \sum_{k=1}^{n_f} \left[ \mathbf{U}_n + \mathbf{T}_p \, \bar{\bf F}_n (\mathbf{W}_n^+, \mathbf{W}_n^-) - \mathbf{M}^v_{\text{ST}} \mathbf{n}_{\text{ST}} \right] A_k \bigg|_{\mathbf{A}_k} = 0
\]  

(2.132)

where matrix \( \mathbf{M}^v_{\text{ST}} \) is of size \( 4 \times 3 \) because the 2D+t spacetime case is being considered. Notice that diffusive term \( \mathbf{M}^v_{\text{ST}} \mathbf{n}_{\text{ST}} A \) is discretised via a second-order central-difference scheme as in Section 2.4.2.

**Flux-difference method by Roe**

Following Roe [93], explained in Section 2.2.3, spatial fluxes at a spacetime face are calculated by means of the conserved variables evaluated at both sides of the interface, \( \mathbf{U}^+ \) and \( \mathbf{U}^- \), as a combination of forward and backward moving waves. Equations (2.46) to (2.63) are still valid for the calculation of spatial fluxes in spacetime. The only difference relies on the fact that \( n_x \) and \( n_y \) are now the components of the spacetime normal vector along the \( x \) and \( y \) directions. Hence, using equations (2.46) and (2.62) in equation (2.110) leads to

\[
\frac{d}{dt} \left( \mathbf{U}_i \mathbf{V}_i \right) + \sum_{k=1}^{n_f} \left[ \mathbf{U}_n + \mathbf{F}_n (\mathbf{U}^+, \mathbf{U}^-) - (\mathbf{F}_x n_x + \mathbf{F}_y n_y) \right] A_k \bigg|_{\mathbf{A}_k} = 0
\]  

(2.133)
or, using the matrix of spacetime viscous fluxes $\mathbf{M}_{\text{ST}}$, given in equation (2.89), yields

$$
\frac{d}{dt} (U_i V_i) + \sum_{k=1}^{n_f} \left[ U_{ni} + F_n \left( \mathbf{U}^+, \mathbf{U}^- \right) - \mathbf{M}_{\text{ST}} n_{\text{ST}} \right] A_k = 0
$$

(2.134)

where matrix $\mathbf{M}_{\text{ST}}$ is of size $4 \times 3$ because the 2D+$t$ spacetime case is being considered. Notice that diffusive term $\mathbf{M}_{\text{ST}} n_{\text{ST}} A$ is discretised via a second-order central-difference scheme as in Section 2.4.2.

### Extrapolation to face values and slope limiters

The upwind implementation of the spacetime solver is second-order accurate. Face values are reconstructed from cell-centre values via second-order correction terms, as done for the spatial discretisation in equation (2.66). In the case of a spacetime discretisation the column vector of coordinates is $\mathbf{x} = [t, x, y]^T$ and the matrix of slope limiters $\Phi$ includes now a component in the time $t$ direction

$$
\Phi = \begin{bmatrix}
\varphi_t & 0 & 0 \\
0 & \varphi_x & 0 \\
0 & 0 & \varphi_y
\end{bmatrix}
$$

(2.135)

Moreover, the matrix of gradients of the primitive variables $\nabla W^c = \frac{\partial W^c}{\partial \mathbf{x}}$ includes one extra column for the time derivatives

$$
\nabla W^c = \begin{bmatrix}
\frac{\partial W^c_i}{\partial t} & \frac{\partial W^c_i}{\partial x} & \frac{\partial W^c_i}{\partial y}
\end{bmatrix}
\begin{bmatrix}
\frac{\partial \rho}{\partial t} & \frac{\partial \rho}{\partial x} & \frac{\partial \rho}{\partial y} \\
\frac{\partial u}{\partial t} & \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\
\frac{\partial v}{\partial t} & \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \\
\frac{\partial p}{\partial t} & \frac{\partial p}{\partial x} & \frac{\partial p}{\partial y}
\end{bmatrix}
$$

(2.136)

The limiters have been chosen to comply with TVD (Total Variation Diminishing) conditions [84] and depend upon the changes of the fluid variables in the nearby of the cell, as defined by (2.69). Here, again, the slope limiter (2.70) by Van Leer [83] is used (see Figure 2.3).
2.4.4 Hybrid central-difference in space, upwind in time

It was hypothesised that a step further in the development of a spacetime framework would be taking advantage of a central-difference approach in space whilst still upwinding in time. The idea underpinning this new hybrid (CSUT, namely central-difference in space, upwind in time) formulation would allow the strength and robustness of the JST scheme to be retained and, at the same time, achieve more time accurate solutions, comparable to those obtained through the upwind formulation, as a consequence of the use of an appropriate time stencil. A prototype code for this formulation has been written and a small number of test cases considered will be presented.

As done in the upwind case, the spacetime flux at any inclined face in the mesh may be split into space and time contributions. Therefore, equation (2.131) is still applicable in the CSUT. Time fluxes are calculated using equations (2.111)-(2.66) whereas space fluxes are worked out via a central-difference formulation, equation (2.93).

2.5 Spacetime discretisation of the Spalart-Allmaras turbulence model

The integral form of the standard Spalart-Allmaras one-equation model is given in equation (2.71). Again, in the case of the spacetime framework, this needs to be integrated in the time domain along with the integration across the spatial domain. Integrating between an initial and final time, $t_0$ and $t_F$, and using the Green-Gauss theorem to convert the closed surface integral into a volume integral yields

$$
\int_{t_0}^{t_F} \int_V \left[ \frac{\partial \mu}{\partial t} + \nabla \cdot (H_c + H_d) \right] d\Omega dt = \int_{t_0}^{t_F} \int_V Q d\Omega dt \tag{2.137}
$$

where the convective and diffusive fluxes, $H_c$ and $H_d$, and the source term $Q$, both for the standard and negative Spalart-Allmaras models, are given by equations (2.72)-(2.76). As done for the Navier-Stokes equations before, defining a three-dimensional divergence operator for two-dimensional problems in spacetime as (the subscript ST means spacetime)

$$
\nabla_{ST} = \frac{\partial}{\partial t} e_t + \frac{\partial}{\partial x} e_x + \frac{\partial}{\partial y} e_y \tag{2.138}
$$
2.5. Spacetime discretisation of the Spalart-Allmaras turbulence model

leads to the three-dimensional spacetime (2D+\(t\)) formulation of the Spalart-Allmaras model

\[
\int_{V_{ST}} \nabla_{ST} \cdot \left( H^c_{ST} + H^d_{ST} \right) d\Omega_{ST} = \int_{V_{ST}} Q d\Omega_{ST} \tag{2.139}
\]

or using the Green-Gauss theorem again to convert the volume integral into a closed surface integral

\[
\oint_{\partial V_{ST}} \left( H^c_{ST} + H^d_{ST} \right) \cdot n_{ST} dS_{ST} = \int_{V_{ST}} Q d\Omega_{ST} \tag{2.140}
\]

where the spacetime Spalart-Allmaras convective \( H^c_{ST} \) and diffusive \( H^d_{ST} \) fluxes for the standard model are

\[
H^c_{ST} = \begin{cases}
\overline{\mu} \\
\overline{\mu}u \\
\overline{\mu}v 
\end{cases}

H^d_{ST} = -\frac{1 + c_{b2}}{\sigma_{SA} Re_{\infty}} (\mu + \overline{\mu}) \begin{cases}
0 \\
\frac{\partial \sigma}{\partial x} \\
\frac{\partial \sigma}{\partial y}
\end{cases}
\tag{2.141}
\]

and those for the negative model are

\[
H^c_{ST} = \begin{cases}
\overline{\mu} \\
\overline{\mu}u \\
\overline{\mu}v 
\end{cases}

H^d_{ST} = -\frac{1}{\sigma_{SA} Re_{\infty}} \left(1 + \frac{c_{b2}}{f_n}\right) (\mu + \overline{\mu} f_n) \begin{cases}
0 \\
\frac{\partial \sigma}{\partial x} \\
\frac{\partial \sigma}{\partial y}
\end{cases}
\tag{2.142}
\]

The semi-discretisation (i.e. discretisation of the spacetime domain only, not in pseudo-time) of the two-dimensional version of equation (2.140) in the case of unstructured grids (see Figure 2.1) yields

\[
\frac{d}{dt_{ST}} (\overline{\mu} V_i) + \sum_{k=1}^{n_f} [(H^c_{ST} + H^d_{ST}) \cdot n_{ST} A]_k = Q_i V_i \tag{2.143}
\]

Although the diffusive terms can be evaluated either with a central-difference or an upwind scheme, the convective terms should always be evaluated using an upwind biased differencing for stability reasons. A first-order upwind scheme has been used here and a ‘smoothing’ has been applied to avoid issues with flow aligned faces as derived from an internal communication with Prof. Christian Allen. The evaluation of quantities \( \overline{\mu} \) and \( \overline{\mu}v \) at face \( k \) is performed as follows
where the subscripts $l(k)$ and $r(k)$ correspond to the left and right cell neighbours at face $k$, respectively, and $C_\theta$ is a measurement of the alignment of the face with the flow, defined as

$$C_\theta = \left( \frac{v \cdot n}{\max\{||v \cdot n||, 10^{-6}\}} \right)^\frac{1}{\gamma}$$

with the normal vector at each face $n$ pointing towards the left cell. Figure 2.6 depicts the contribution to the flux of cells at either side of a face. Particularly, when the flow is aligned or nearly aligned with the face, i.e. $v \cdot n \simeq 0$, the smoothing allocates the contribution to the flux of each cell in a more natural and continuous way.

### 2.6 Boundary conditions for the spatial discretisation

A set of boundary conditions need to be defined along with the numerical discretisation of the conservation laws, either the Navier-Stokes or the Euler equations. Several approaches exist for the imposition of boundary conditions but, perhaps, one of the most common ones is through the definition of some virtual cells outside the computational domain as depicted in Figure 2.7. These are known as ghost or halo cells and, although they are
2.6. Boundary conditions for the spatial discretisation

used throughout the calculation, they are not part of the solution domain. Appropriate values of the physical quantities ensure a correct approximation of fluxes through boundary faces. In general, one can distinguish between solid and farfield boundaries.

2.6.1 Solid boundaries

Euler equations

The wall boundary condition in an Euler solver means that the normal velocity, relative to a solid face, has to be null at that face, i.e.

\[(v_b - V_w) \cdot n = 0\]  \hspace{1cm} (2.147)

where \(n = \{n_x, n_y\}^T\) is the normal vector, \(v_b = \{u_b, v_b\}^T\) is the flow velocity at the solid boundary and \(V_w = \{U_w, V_w\}^T\) is the wall velocity. Moreover, the tangential velocity component is kept the same as that at the cell center of the neighbouring cell

\[(v_b - v_0) \cdot t = 0\]  \hspace{1cm} (2.148)

where \(t = \{-n_y, n_x\}^T\) is the tangential unit vector and \(v_0 = \{u_0, v_0\}^T\) is the flow velocity at the centre of the neighbouring cell.

Combining equations \(2.147\) and \(2.148\) yields
\[ u_b = u_0 n_y^2 + U_w n_x^2 + (V_w - v_0) n_x n_y \]  \hspace{1cm} (2.149)

\[ v_b = v_0 n_x^2 + V_w n_y^2 + (U_w - u_0) n_x n_y \]  \hspace{1cm} (2.150)

However, as explained above, the imposition of boundary conditions is done through the use of halo cells. Therefore, the value of velocity at the halo cells can be worked out from a central stencil as

\[ u_h = 2u_b - u_0 \]  \hspace{1cm} (2.151)

\[ v_h = 2v_b - v_0 \]  \hspace{1cm} (2.152)

where \( u_h \) and \( v_h \) are the velocity components at the halo cells. Similarly, the imposition of an adiabatic wall means that the gradients of density and pressure across the wall are zero, i.e.

\[ \rho_h = \rho_0 \]  \hspace{1cm} (2.153)

\[ p_h = p_0 \]  \hspace{1cm} (2.154)

where \( \rho_0, \rho_h, p_0 \) and \( p_h \) are the density and the pressure at the inner neighbouring cell and the halo cell, respectively.

**Navier-Stokes equations**

The no-slip condition in a Navier-Stokes solver means that the velocity of the flow at solid walls is the same as the wall velocity. In other words, there is no relative motion between the fluid and the wall

\[ v_b - V_w = 0 \]  \hspace{1cm} (2.155)

Using this condition in equations \(2.151\)–\(2.152\) yields in this case

\[ u_h = 2U_w - u_0 \]  \hspace{1cm} (2.156)

\[ v_h = 2V_w - v_0 \]  \hspace{1cm} (2.157)
2.6. Boundary conditions for the spatial discretisation

Conditions (2.153) and (2.154) for density and pressure are still valid here. If Spalart-Allmaras is used as the turbulence model a condition for the eddy viscosity needs to be imposed as well, namely

$$\bar{\mu}_h = -\bar{\mu}_0$$  (2.158)

2.6.2 Farfield boundaries

If the freestream flow properties are used as the farfield boundary conditions, undesired reflections and non-physical behaviour may arise. As a consequence of the hyperbolic nature of the Navier-Stokes (and Euler) equations, forward and backward moving waves propagate information throughout the domain according to characteristics. In particular, at farfield boundaries certain physical quantities may be convected from inside to outside the domain and, at the same time, other quantities may be convected from outside to inside. Performing a characteristics analysis the following three Riemann invariants are found at any farfield face in the mesh

\[
\begin{align*}
R^0 &= \frac{p}{\rho^\gamma} \quad \text{on } \frac{dx_n}{dt} = u_n \\
R^+ &= u_n + \frac{2a}{\gamma - 1} \quad \text{on } \frac{dx_n}{dt} = u_n + a \\
R^- &= u_n - \frac{2a}{\gamma - 1} \quad \text{on } \frac{dx_n}{dt} = u_n - a
\end{align*}
\]  (2.159-2.161)

where \(p\) is the pressure, \(\rho\) is the density, \(\gamma\) is the heat capacity ratio, \(a\) is the speed of sound and \(u_n\) is the flow velocity normal to the face (see Figure 2.8). Local direction \(x_n\) is normal to the face. These variables remain constant along the specified path lines provided discontinuities do not appear. Combining and re-arranging equations (2.160) and (2.161) yields

\[
\begin{align*}
u_n &= \frac{1}{2} (R^+ + R^-) \\
a &= \frac{\gamma - 1}{4} (R^+ - R^-)
\end{align*}
\]  (2.162, 2.163)

Assuming normal vectors always point outwards towards the outside of the computational domain and bearing in mind that \(u_n = \mathbf{v} \cdot \mathbf{n}\), then \(u_n > 0\) means outflow whereas \(u_n < 0\) means inflow. In the supersonic case, characteristics are convected from the upstream cell, i.e. either from the interior if \(u_n > a\) or from the freestream conditions if \(u_n < -a\). In subsonic flows, however, characteristics may be convected from both sides,
as concluded from (2.159)-(2.161). The transport of the Riemann invariants at any farfield face in subsonic flow ($|u_n| < a$) implies that

\[ (R^+)_b = (R^+)_0 = (u_n)_0 + \frac{2a_0}{\gamma - 1} \]  
\[ (R^-)_b = (R^-)_\infty = (u_n)_\infty - \frac{2a_\infty}{\gamma - 1} \]  

where the subscripts $b$, 0 and \( \infty \) denote boundary, inner cell and freestream conditions, respectively. Using equations (2.164) and (2.165) into (2.162) and (2.163) one can write

\[ (u_n)_b = \frac{(u_n)_0 + (u_n)_\infty}{2} + \frac{2}{\gamma - 1} \cdot \frac{a_0 - a_\infty}{2} \]  
\[ a_b = \frac{\gamma - 1}{2} \cdot \frac{(u_n)_0 - (u_n)_\infty}{2} + \frac{a_0 + a_\infty}{2} \]  

Projecting normal velocity $u_n$ along Cartesian coordinates and finding out the value at the halo cell yields

\[ u_h = 2(u_n)_b n_x - u_0 \]  
\[ v_h = 2(u_n)_b n_y - v_0 \]  

Using the first Riemann invariant $R^0$ given in equation (2.159) and bearing in mind the definition of the speed
of sound in equation (B.47), the density and pressure at a farfield face are

\[
\rho_b = \begin{cases} 
\rho_0 \left( \frac{a_b}{a_0} \right)^{\frac{2}{\gamma}} & \text{if } 0 < u_n < a \\
\rho_\infty \left( \frac{a_b}{a_\infty} \right)^{\frac{2}{\gamma}} & \text{if } -a < u_n < 0 
\end{cases} 
\] (2.170)

\[
p_b = \frac{\rho_b a_b^2}{\gamma} 
\] (2.171)

The corresponding values at the halo cell are given by

\[
\rho_h = 2\rho_b - \rho_0 
\] (2.172)

\[
p_h = 2p_b - p_0 
\] (2.173)

In the case of turbulent solutions, the value of the Spalart-Allmaras viscosity variable \( \mu \) at farfield boundaries take a value of

\[
3\mu_\infty \leq \mu_b \leq 5\mu_\infty 
\] (2.174)

as has been suggested in [98] and [99]. In the current thesis a value of \( \mu_h = 5\mu_\infty \) has been used.

**Giles approach**

An alternative formulation of the non-reflecting farfield boundaries can be found in [100] and has also been implemented here with no noticeable differences in the results. The characteristic variables are calculated as follows

\[
\begin{pmatrix}
\delta v_1 \\
\delta v_2 \\
\delta v_3 \\
\delta v_4
\end{pmatrix} = 
\begin{pmatrix}
1 & 0 & 0 & -1/a_0^2 \\
0 & 0 & 1 & 0 \\
0 & \rho_0/2a_0 & 0 & 1/2a_0^2 \\
0 & -\rho_0/2a_0 & 0 & 1/2a_0^2
\end{pmatrix}
\begin{pmatrix}
\delta \rho \\
\delta u \\
\delta v \\
\delta p
\end{pmatrix} \] (2.175)
where the *delta* variables are taken with respect to the freestream conditions (e.g. $\delta \rho = \rho_0 - \rho_\infty$). The hyperbolic system of equations may be written in terms of these characteristic variables as

$$
\frac{\partial}{\partial t} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} + \begin{bmatrix} u_0 & 0 & 0 & 0 \\ 0 & u_0 & 0 & 0 \\ 0 & 0 & u_0 + a_0 & 0 \\ 0 & 0 & 0 & u_0 - a_0 \end{bmatrix} \frac{\partial}{\partial x} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = 0
$$

(2.176)

This implies that characteristic variables $v_1$ and $v_2$ are constant along the path lines $dn/dt = u_0$, whereas $v_3$ and $v_4$ are conserved along $dn/dt = u_0 + a_0$ and $dn/dt = u_0 - a_0$, respectively. Consequently, the boundary conditions to be imposed at any inflow boundary are

$$
\delta v_1 = 0
$$

(2.177)

$$
\delta v_2 = 0
$$

(2.178)

$$
\delta v_3 = 0
$$

(2.179)

and those at any outflow boundary are

$$
\delta v_4 = 0
$$

(2.180)

The *delta* of the primitive variables at the farfield boundary can be calculated using the following equation

$$
\begin{bmatrix} \delta \rho \\ \delta u \\ \delta v \\ \delta p \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & a_0 / \rho_0 & -a_0 / \rho_0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a_0^2 & a_0^2 \end{bmatrix} \begin{bmatrix} \delta v_1 \\ \delta v_2 \\ \delta v_3 \\ \delta v_4 \end{bmatrix}
$$

(2.181)

And the conditions on the farfield boundaries are

$$
\rho_b = \rho_\infty + \delta \rho
$$

(2.182)

$$
u_b = u_\infty + \delta u
$$

(2.183)
\[ v_b = v_\infty + \delta v \quad (2.184) \]

\[ p_b = p_\infty + \delta p \quad (2.185) \]

Notice that the implementation of boundary conditions has been done through the definition of halo cells. Therefore, using these values in equations (2.151)-(2.152), (2.153) and (2.154) yield the conditions to be imposed on the halo cells.

### 2.7 Boundary conditions for the spacetime discretisation

Similarly to the spatial discretisation, a set of boundary conditions need to be imposed on the physical quantities to ensure a correct approximation of fluxes through boundary faces. As depicted in Figure 2.7 halo or ghost cells are used. In general, one can distinguish between solid and farfield boundaries. However, in the case of spacetime simulations, time boundaries need to be considered as well.

#### 2.7.1 Solid boundaries

**Euler equations**

The implementation of solid boundary conditions in spacetime is slightly different from the approach followed in section 2.6.1 for the spatial discretisation. The normal component of the relative velocity at any solid face has to be null whereas the in-plane (or tangential) components hold the same value as the cell center of the neighbouring cell. Therefore, equations (2.147) and (2.148) are also valid here. The difference relies on the fact that the normal vector \( \mathbf{n} \) accounts only for the spatial discretisation but the normal vector in spacetime \( \mathbf{n}_{ST} = \{n_t, n_x, n_y\}^T \) has a component in time as well. Slicing the 2D+1 spacetime domain at any \( t = \text{constant} \) plane yields a two-dimensional mesh where the direction of the normal vectors at each face corresponds to the spatial components of the spacetime normal vector at that face in the spacetime mesh. Normalizing the spatial components of the spacetime normal vector one can write the normal \( \mathbf{n} \) and tangential \( \mathbf{t} \) vectors at any time as

\[
\mathbf{n} = \left\{ \begin{array}{c} n_x \\ \sqrt{n_x^2 + n_y^2} \\ \sqrt{n_x^2 + n_y^2} \end{array} \right\} \quad \quad \mathbf{t} = \left\{ \begin{array}{c} n_y \\ \sqrt{n_x^2 + n_y^2} \\ \sqrt{n_x^2 + n_y^2} \end{array} \right\} \quad (2.186)
\]
Moreover, the normal component of the wall velocity is given by the spacetime normal vector as demonstrated for the simple one-dimensional problem depicted in Figure 2.9.

![Spacetime mesh for one-dimensional cell expanding between two consecutive time levels](image)

Figure 2.9: Spacetime mesh for one-dimensional cell expanding between two consecutive time levels

Let \( V^n \) be the length (or volume in \( \mathbb{R}^3 \)) of a one-dimensional cell at time \( n\Delta t \), i.e. the length of face \( \overline{AB} \). Assume that one of the endpoints (or faces in \( \mathbb{R}^3 \)) remains stationary (point \( A \)) but the other one moves with velocity \( U_w^\perp \) (point \( B \)), hence its position at time \( n\Delta t + \Delta t \) is a distance \( U_w^\perp \Delta t \) away from its initial position (point \( B' \)). Therefore, the length of the cell at time level \( n+1 \) is

\[
V^{n+1} = V^n + U_w^\perp \Delta t
\]  

(2.187)

or, equivalently,

\[
\frac{\overline{BB'}}{\overline{BB}} \sin \frac{\overline{A'B'B}}{\overline{A'B'B}} = \frac{n_x}{-n_t} \implies \frac{(n+1)\Delta t - n\Delta t}{V^{n+1} - V^n} = \frac{n_x}{-n_t}
\]  

(2.189)

Combining equations (2.187) and (2.189) the velocity of the cell’s endpoint (or face in \( \mathbb{R}^3 \)) may be written in
2.7. Boundary conditions for the spacetime discretisation

terms of the spacetime normal vector components as

\[ U_w^\perp = -\frac{n_t}{n_x} \]  

(2.190)

This same expression can be derived by comparison between mass balance for the one-dimensional problem and
its spacetime counterpart, assuming fluid properties are constant and uniform, i.e. \( \rho (x, t) = \rho_0 \) and \( u(x, t) = u_0 \).
The continuity equation for the one-dimensional problem using the ALE formulation is

\[
\frac{d}{dt} \int_V \rho d\Omega - \int_{\partial V} \rho (u - U_w^\perp) n_x dS = \rho_0 (u_0 - u_0 - 0) = 0 
\]  

(2.191)

Likewise, the continuity equation using the spacetime formulation is

\[
\int_{\partial V_{ST}} \left\{ \rho, \rho u \right\} \left\{ \begin{array} \{n_t} \{n_x} \end{array} \right\} dS_{ST} 
\]

\[
(0 - \rho_0 u_0) \Delta t + (\rho_0 + 0) V^{n+1} + (\rho_0 n_t + \rho_0 u_0 n_x) \xi + (-\rho_0 + 0) V^n = 0 
\]  

(2.192)

where the normal vectors at faces \( \overline{AA}' \), \( \overline{A'B}' \), \( \overline{BB}' \) and \( \overline{AB} \) are given by

\[
\begin{array}{c}
\begin{array} \{n_{AA'}} \{n_{A'B'}} \{n_{BB'}} \{n_{AB}} \\
\{0\} \{1\} \{n_t\} \{n_x\} \{1\} \{0\} \{-1\} \{0\} \\
\end{array}
\end{array} \]  

(2.193)

Dividing by \( \Delta t \), noting that \( \xi n_x = \Delta t \) and re-arranging, equation (2.192) can be written in the form of (2.191) as

\[
\frac{\rho_0 V^{n+1} - \rho_0 V^n}{\Delta t} + \rho_0 (u_0 + \frac{n_t}{n_x}) - \rho_0 (u_0 - 0) = 0 
\]  

(2.194)

where it can be concluded that \( U_w^\perp = -\frac{n_t}{n_x} \) as shown before in equation (2.190). Bearing in mind that \( n_t^2 + n_x^2 = 1 \) for the one-dimensional case, equation (2.190) may be cast to

\[
U_w^\perp = -\frac{n_t}{\sqrt{1 - n_t^2}} 
\]  

(2.195)
Extrapolating this result to a higher dimensional problem yields that the normal wall velocity is

\[ \mathbf{V}_w \cdot \mathbf{n} = -\frac{n_t}{\sqrt{1 - n_t^2}} \quad (2.196) \]

Nothing can be concluded about the in-plane component of the wall velocity from the spacetime normal vector since, in the case of a fully unstructured mesh, no one-to-one correspondence exists between wall nodes at different time positions. One could assume nodes are “fixed” to the wall, but this is somewhat restrictive and inconvenient. Projecting the normal wall velocity into the Cartesian coordinates \( x \) and \( y \) for a two-dimensional problem yields

\[ U_w^\perp = -\frac{n_t}{\sqrt{n_x^2 + n_y^2}} \frac{n_x}{\sqrt{n_x^2 + n_y^2}} = -\frac{n_t n_x}{n_x^2 + n_y^2} \quad (2.197) \]

\[ V_w^\perp = -\frac{n_t}{\sqrt{n_x^2 + n_y^2}} \frac{n_y}{\sqrt{n_x^2 + n_y^2}} = -\frac{n_t n_y}{n_x^2 + n_y^2} \quad (2.198) \]

For the Euler equations only the component of the wall velocities, \( U_w \) and \( V_w \), normal to the face are necessary as can be concluded from equation (2.147). Therefore, combining (2.147), (2.148), (2.186), (2.197) and (2.198), the spacetime wall boundary conditions for inviscid flows are

\[ u_b = \frac{u_0 n_y^2 - v_0 n_x n_y - n_t n_x}{n_x^2 + n_y^2} \quad (2.199) \]

\[ v_b = \frac{-u_0 n_x n_y + v_0 n_x^2 - n_t n_y}{n_x^2 + n_y^2} \quad (2.200) \]

However, boundary conditions are imposed on the halo cells rather than the boundaries themselves. Therefore, equations (2.151)-(2.152) can be used to work out the value at the halo cells. The density and pressure are given by equations (2.153) and (2.154), respectively.

**Navier-Stokes equations**

In a spacetime Navier-Stokes solver the no-slip condition at solid walls requires the input of wall velocities \( U_w \) and \( V_w \) to the solver. The spacetime normal vector only provides information about the normal component of the wall velocity and, as outlined before, nothing can be concluded about the in-plane component unless points are “pinned” to the surface. This can be explained by the fact that there is no necessary one-to-one correspondence between wall nodes at different time positions. In other words, the mesh between two consecutive time levels
2.7. Boundary conditions for the spacetime discretisation

may not be attached to the solid geometry by the exact same nodes as seen, for example, in Figure 2.10 where a segment element in a two-dimensional mesh moves as a consequence of the wall’s rigid body motion. The segment element also shrinks in this case making it impossible to work out a correspondence between nodes at times \( t_0 \) and \( t_0 + \Delta t \).

Therefore, given \( U_w \) and \( V_w \) as input data to the Navier-Stokes equations, the velocities at the solid boundaries can be calculated using equation (2.156). Similarly, the viscosity is given by equation (2.158), and the density and pressure are given by equations (2.153) and (2.154), respectively.

2.7.2 Farfield boundaries

In this work, farfield boundaries in the spacetime solver are oriented such that \( \mathbf{n}_{ST} \cdot \mathbf{e}_t = 0 \). Therefore, they are treated in exactly the same way as in section 2.6.2 for the spatial discretisation.

2.7.3 Time boundaries

This kind of boundary is restricted to spacetime formulations of unsteady CFD problems. For convenience, in the present work spacetime normal vectors of all time boundary faces are such that \( \mathbf{n}_{ST} \cdot \mathbf{e}_t = 1 \) or \( n_t = 1 \) and \( n_x = n_y = 0 \). In other words, they are perpendicular to the time \( t \) direction. They can be either initial, final or periodic boundaries.

Initial time boundary

For the initial time boundary condition it is necessary to calculate the initial steady-state solution. The first \( t = \) constant slice (usually \( t = 0 \)) is used as the initial mesh. Once the solution is obtained it is used to seed the initial solution of the spacetime simulation. Each two-dimensional cell in the initial mesh corresponds to one
three-dimensional halo cell at the initial boundary of the spacetime mesh. The values of the physical properties \((\rho_0, u_0, v_0, p_0 \text{ and } \mu_0)\) on the initial solution are therefore used as the solution at these halo cells, hence imposing the initial condition on the spacetime problem.

**Final time boundary**

The final time boundary condition does not require any special treatment. Time fluxes at a final time boundary may be worked out using the fluid variables at the cell centres that lie on the final time plane. However, it is also possible to extrapolate the cell-centre values to the final faces.

**Periodic time boundary**

In the case of time periodicity a condition on time boundaries is not necessary any more since these are treated in the exact same way as inner spacetime faces. In other words, periodic time boundaries behave like inner spacetime faces (within the fluid domain) where a spacetime cell is attached on each side. Cells on the final boundary are directly connected to cells on the initial boundary. This yields a consistent implementation of time periodic boundaries since they automatically preserve the stencil of the discretisation method used (e.g. central-difference or upwind).
Chapter 3

Solution method: time integration and equivalence to ALE

3.1 Introduction

The semi-discrete form of the Navier-Stokes equations, equation (2.8) in arbitrary Lagrangian-Eulerian formulation or (2.86) in spacetime, need to be integrated in time, either pseudo-time $t^*$ or physical time $t$, as follows

$$\frac{d}{dt} \int_V U d\Omega + R(U) = 0$$

(3.1)

where $U$ is the column vector of conserved variables and $R(U)$ is the column vector of residuals which is specific to the particular discretisation chosen (e.g. finite-volume vs discontinuous-Galerkin, central-difference vs upwind, etc.). There exist several ways to discretise and solve the problem in time as explained in the following sections. In particular, it is important to consider explicit and implicit schemes as this choice has a direct impact on accuracy, stability and convergence of the numerical solution.
3.2 Explicit methods

Explicit schemes [101, 102] are perhaps the most intuitive way to solve problem (3.1) because they use the solution at previous and, at most, the current time level to work out the solution at the next time level, i.e.

\[
U^{n+1} = \mathcal{F}(U^n, U^{n-1}, \ldots, U^{n-m})
\]  

(3.2)

where \( n \) is the current time level and \( m \) is the size of the temporal stencil. For instance, using the well-known forward Euler method (\( m = 1 \)) the integration of (3.1) across volume \( V^n \) at time \( t^n \) yields the solution at the next time level \( U^{n+1} \) as

\[
U^{n+1} = \frac{V^n}{V^{n+1}} U^n - \frac{\Delta t^n}{V^{n+1}} R(U^n)
\]

(3.3)

where \( \Delta t^n = t^{n+1} - t^n \) is the time step size. On the positive side, explicit schemes are, in general, simple and easy to implement and have low memory requirements. However, their main drawback relies on their poor numerical stability. Explicit methods need a much more restrictive constraint on their time-step size compared to implicit ones in order to ensure stability of the numerical solution [103]. A von Neumann analysis yields a condition on the CFL number defined as

\[
\text{CFL} \equiv \frac{\lambda \Delta t}{\Delta x}
\]

(3.4)

where \( \lambda \) represents the largest eigenvalue in the problem or spectral radii and \( \Delta t \) and \( \Delta x \) are the time and space step sizes, respectively. This condition comes from the imposition that the amplitudes \( \hat{u}^n \) of the Fourier modes of the numerical solution do not grow indefinitely (see equation (4.4) in Chapter 4). In other words, the amplification factor \( G \) defined as

\[
G \equiv \frac{\hat{u}^{n+1}}{\hat{u}^n}
\]

(3.5)

satisfies the von Neumann stability condition, namely,

\[
|G| \leq 1
\]

(3.6)
The von Neumann stability condition for the forward Euler method above reads as follows

\[ \frac{CFL_{\text{Euler}}}{\lambda} \leq 1 \quad \Rightarrow \quad \Delta t \leq \frac{\Delta x}{\lambda} \tag{3.7} \]

### 3.2.1 Explicit fourth-order Runge-Kutta method

A very important family of time integration methods are the popular explicit Runge-Kutta schemes \[104, 76, 1\]. As opposed to multistep or multipoint methods, which achieve high-order accurate solutions through the use of the solution at several time steps, Runge-Kutta methods are single-step schemes with multiple number of stages. Different schemes have different number of stages between the current \( n \) and next \( n + 1 \) time levels and, therefore, achieve different degrees of accuracy. This is advantageous since there is no need for an estimated solution at the very first time level in the simulation. The general form of a \( k \)-stage Runge-Kutta method can be written as

\[ U^{n+1} = U^n - \Delta t \sum_{i=1}^{k} \beta_i R \left( U^{(i)} \right) \tag{3.8} \]

where \( \beta_i \) are weights and \( U^{(i)} \) refers to the solution at the \( i \)-th stage of the Runge-Kutta iteration. At each stage, the solution is worked out as

\[ U^{(i)} = U^n - \Delta t \sum_{j=1}^{i-1} \alpha_{ij} R \left( U^{(j)} \right) \quad \forall i \leq k \tag{3.9} \]

where \( \alpha_{ij} \) are weighting coefficients. Different combinations of coefficients \( \alpha_{ij} \) and \( \beta_i \) define different so-called Butcher tableaus which determine the order of accuracy of the Runge-Kutta methods. A Butcher tableau can be expressed as

\[ \begin{array}{c|cccc}
\gamma_1 & \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1N} \\
\gamma_2 & \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2N} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\gamma_N & \alpha_{N1} & \alpha_{N2} & \cdots & \alpha_{NN} \\
& \beta_1 & \beta_2 & \cdots & \beta_N \\
\end{array} \]

where coefficients \( \gamma_i = \sum_j \alpha_{ij} \) constitute the sum of all coefficients \( \alpha_{ij} \) for each stage \( i \). In particular, explicit schemes have a strictly lower triangular matrix \( \alpha_{ij} \) as their Butcher tableau, i.e. all elements on and above
the diagonal are zero ($\alpha_{ij} = 0 \; \forall j \geq i$). Due to its simplicity, the explicit fourth-order accurate Runge-Kutta scheme is a common choice among the CFD community and is also the one used throughout this thesis. It can be written as follows

\begin{align}
U^{(1)} &= U^n \\
U^{(2)} &= U^n - \frac{1}{2} \Delta t^n R^{(1)} \\
U^{(3)} &= U^n - \frac{1}{2} \Delta t^n R^{(2)} \\
U^{(4)} &= U^n - \frac{\Delta t^n}{V^n} R^{(3)}
\end{align}

where $U^n$ is the solution at the current time level $n$ and notation $R^{(i)} = R(U^{(i)})$ for $i = 1, 2, 3, 4$ has been used. The solution at the next time level $n+1$ is

$$U^{n+1} = U^n - \frac{1}{6} \frac{\Delta t^n}{V^n} \left( R^{(1)} + 2R^{(2)} + 2R^{(3)} + R^{(4)} \right)$$

Therefore, its Butcher tableau is

\begin{tabular}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\
\frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\
1 & 0 & 0 & 1 & 0 \\
\hline & \frac{1}{5} & \frac{1}{3} & \frac{1}{3} & \frac{1}{5}
\end{tabular}

A stability analysis for this scheme \cite{105} imposes a limit on the time-step of

$$CFL_{RK4} \leq 2\sqrt{2}$$

\section{3.3 Implicit methods}

Implicit methods \cite{106,104,107}, as opposed to explicit ones, yield an implicit algebraic equation (or system of equations) whereby the solution at the next time level cannot be written explicitly in terms of the solution
at the current and previous time levels, i.e. they must solve an implicit equation or system of equations of the type

\[ F(U^{n+1}, U^n, U^{n-1}, \ldots, U^{n-m}) = 0 \] (3.16)

For instance, problem (3.1) may be discretised using a backward Euler scheme which uses a first-order backward-difference for the time derivative, as follows

\[ U^{n+1} = \frac{V^n}{V_{n+1}} U^n - \frac{\Delta t^n}{V_{n+1}} R(U^{n+1}) \] (3.17)

In contrast with equation (3.3) for the forward Euler method, equation (3.17) is implicit in the unknown \( U^{n+1} \) and cannot be solved directly. An iterative method, also known as marching in pseudo-time, may therefore be used. The procedure is similar to an explicit method. In fact, it is common to use explicit methods to march the solution in pseudo-time. The equation to solve would be

\[ \frac{d}{dt^e} (U^{n+1} V^{n+1}) + F(U^{n+1}, U^n, U^{n-1}, \ldots, U^{n-m}) = 0 \] (3.18)

where function \( F \) may or may not include a discretisation of the physical-time derivatives depending on whether the problem is steady or time-accurate. The ‘steady-state’ solution of (3.18), i.e. the value of \( U^{n+1} \) such that \( \frac{d}{dt^e} (U^{n+1} V^{n+1}) \approx 0 \), is also the solution of equation (3.16). Unlike explicit methods, implicit methods are in general stable and a much more relaxed constraint may be imposed on the time-step size. If a time-accurate solution is sought, i.e. if \( F \) contains the discretisation of the physical-time derivatives, this limitation is, in general, dictated by the need of a realistic and accurate solution rather than to avoid numerical stability issues. Notice that the solution of equation (3.16) is independent of the intermediate solutions that drive equation (3.18) to the ‘steady-state’. Therefore the use of so-called acceleration techniques [104, 107, 108] can speed up convergence without thereby incurring accuracy losses in ‘physical time’ since all intermediate solutions of (3.18) can be disregarded.

### 3.3.1 Implicit Runge-Kutta methods

Implicit Runge-Kutta methods are also very popular within the CFD community, particularly in the solution of time-accurate problems due to their wide range of allowed time step sizes. Diagonally implicit Runge-Kutta (DIRK) schemes are perhaps the preferred choice for applications which require a time-accurate integration
Chapter 3: Solution method: time integration and equivalence to ALE

of the Navier-Stokes equations. Their Butcher tableau is lower triangular and at least one element in their diagonal is non-zero, i.e. \( \alpha_{ij} = 0 \quad \forall j > i \), and, in general, \( \alpha_{ij} \neq 0 \)

\[
\begin{array}{cccc}
\alpha_{11} & 0 & \cdots & 0 \\
\alpha_{21} + \alpha_{22} & \alpha_{21} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
\sum_j \alpha_{Nj} & \alpha_{N1} & \cdots & \alpha_{NN} \\
\beta_1 & \beta_2 & \cdots & \beta_N \\
\end{array}
\]

DIRK schemes represent systems of equations of the form (3.16) where \( F \in \mathbb{R}^{2+n-m} \). A linearisation around solution \( U^{n+1} \) yields

\[
\left( \frac{\partial F}{\partial U} \right)^q \Delta U^{q+1} = -F^q
\]

where function \( F^q = F(U^q, U^n, U^{n-1}, \ldots, U^{n-m}) \), matrix \( \left( \frac{\partial F}{\partial U} \right)^q \) is the Jacobian evaluated with \( U^{n+1} \approx U^q \) and \( \Delta U^{q+1} = U^{q+1} - U^q \). Notice that at the end of this iterative process \( U^{q+1} \to U^q \), hence equation (3.19) yields equation (3.16) and, therefore, \( U^q \approx U^{n+1} \).

### 3.4 Steady-state problems

Steady-state problems, either through an arbitrary Lagrangian-Eulerian framework or with a spacetime framework, reduce to the solution of (2.10) or (2.86), namely

\[
R(U, \nabla U) = 0
\]

In general the residual \( R(U) \) is a non-linear function of the solution \( U \), hence it is sometimes impossible to solve equation (3.20) explicitly. Most times the preferred approach is to march the residuals in pseudo-time \( t^* \) until \( U \) converges to the steady-state solution. Therefore, the following equation is solved

\[
\frac{dU}{dt^*} + R(U, \nabla U) = 0
\]
3.5 Unsteady or time-accurate problems

Unsteady problems may be solved explicitly, i.e. integrating the physical-time explicitly, or implicitly, by marching the space and time discretisation in pseudo-time. If an explicit scheme is used the solution in the simulation represents the evolution of the solution in physical-time. This imposes very restrictive limitations on the allowable time-step size as a consequence of numerical stability issues. If an implicit scheme is used instead, equation (3.18) needs to be solved. The key difference is that function $\mathcal{F}(U)$ now includes the discretisation of the time derivatives as well as the spatial discretisation, and a pseudo-time integration is used to converge the solution at each physical-time step. If a second-order backward-difference is used for the time discretisation, then

$$
\mathcal{F} \left( U^{n+1}, U^n, U^{n-1}, \ldots, U^{n-m} \right) = \frac{3U^{n+1}V^{n+1} - 4U^nV^n + U^{n-1}V^{n-1}}{2\Delta t^n} + R \left( U^n, U^{n-1}, \ldots, U^{n-m} \right) \tag{3.22}
$$

Using (3.22) in (3.18) and solving by means of an explicit scheme in pseudo-time yields the time-accurate solution. This approach is known across the specific literature as dual-time stepping.

3.6 Spacetime problems

The spacetime framework is, in a way, similar to unsteady problems solved via a dual-time stepping technique. The implicit discretisation of time derivatives comes from the spacetime finite-volume approach whereby space and time are discretised together. In this case, function $\mathcal{F}$ in equation (3.18) includes the spacetime discretisation

$$
\mathcal{F} \left( U^{n+1}, U^n, U^{n-1}, \ldots, U^{n-m} \right) = R_{ST} \left( U^{n+1}, U^n, U^{n-1}, \ldots, U^{n-m} \right) \tag{3.23}
$$

where now the superscript $n$ refers to the current pseudo-time level instead of the actual physical-time level, and $R_{ST}(U)$ are the spacetime residuals, equation (2.87). Each discrete location in the spacetime mesh corresponds to a given time $t_0$ and position $(x_0, y_0)$. The solution is obtained solving the following equation in pseudo-time

$$
\frac{dU}{dt} + R_{ST}(U, \nabla U) = 0 \tag{3.24}
$$

As outlined before, a fourth-order Runge-Kutta scheme is used. Similarly to the solution of time-accurate problems with an implicit solver, acceleration techniques may be used here because the intermediate solutions
Chapter 3: Solution method: time integration and equivalence to ALE

that drive to the spacetime solution $R_{ST} = 0$ can be neglected. In this work a local time-stepping technique has been utilised to increase the rate of convergence of simulations. This is very simple and easy to implement but not as effective as, for instance, multigrid methods [109, 110].

3.7 Comparison between an arbitrary Lagrangian-Eulerian formulation and the spacetime framework

The spacetime framework for the solution of unsteady aerodynamics problems is still a very young technique and, at the time of writing this thesis, is less efficient than the arbitrary Lagrangian-Eulerian counterpart, mainly due to the fact that the whole time domain is updated at every iteration in pseudo-time. However, its flexibility allows the simulation of arbitrary moving boundary problems with ease and avoids the introduction of complex mesh motion techniques (e.g. Chimera grids) and interpolation methods between the solution at consecutive time levels. A hybrid formulation where spacetime is used in areas of complex boundary motions or topological changes and an ALE framework is applied elsewhere would yield a very versatile yet fast and efficient solver. Unfortunately, with the current implementation (i.e. cell-centered finite volume) this is not straightforward since the spacetime solution differs from the solution of the arbitrary Lagrangian-Eulerian framework in general and there are only a few particular cases where both formulations are comparable, as shown in this section for an unsteady one-dimensional problem.

3.7.1 Arbitrary Lagrangian-Eulerian formulation

The integral formulation of the one-dimensional form of the Navier-Stokes equations in an arbitrary Lagrangian-Eulerian framework yields

$$\frac{d}{dt} \int_V U d\Omega + \oint_{\partial V} (F - U u_c) n_x dS = 0$$

where $U$ is the column vector of conserved variables, $F$ is the column vector of fluxes, $n_x$ is the one-dimensional normal vector and $u_c$ is the velocity of the boundaries of control volume $V$. Notice that term $-U u_c$ in the surface integral arises as a consequence of the application of the Reynolds transport theorem, equation (A.1), to the time derivative integral. Using a backward second-difference for the time derivative (see equation (F.9)
3.7. Comparison between an arbitrary Lagrangian-Eulerian formulation and the spacetime framework

\begin{equation}
U^n_j V^n_j \left( \frac{1}{t^n - t^{n-1}} + \frac{1}{t^n - t^{n-2}} \right) - U^{n-1}_j V^{n-1}_j \left( \frac{1}{t^{n-1} - t^n - 1} + \frac{1}{t^{n-1} - t^{n-2}} \right) \\
+ U^{n-2}_j V^{n-2}_j \left( \frac{1}{t^{n-2} - t^{n-1}} - \frac{1}{t^{n-2} - t^{n}} \right) + R(U^n)
\end{equation}

\begin{equation}
- \left( U^n_{j+\frac{1}{2}} (u_c)^n_{j+\frac{1}{2}} A^n_{j+\frac{1}{2}} - U^n_{j-\frac{1}{2}} (u_c)^n_{j-\frac{1}{2}} A^n_{j-\frac{1}{2}} \right) = 0
\end{equation}

where a variable time step size $\Delta t$ has been used for an easy comparison with the general spacetime formulation.

In reality, the most general spacetime formulation has no equivalent counterpart in conventional CFD techniques because the former allows for a variable time step size across the spatial domain which is impossible to do in the latter case. Here, a general formulation of the spacetime framework refers only to a variable time step size at different time levels.

\begin{equation}
V^n_{j \pm \frac{1}{2}} = (u_c)^n_{j \pm \frac{1}{2}} A^n_{j \pm \frac{1}{2}} \left( t^{n+\frac{1}{2}} - t^{-\frac{1}{2}} \right)
\end{equation}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure3.1.png}
\caption{One-dimensional moving mesh in arbitrary Lagrangian-Eulerian framework, equivalent to the problem formulated in spacetime given in Figure 3.2}
\end{figure}

in Appendix F and using the discretisation in Figure 3.1 equation (3.25) yields, for cell $j$ at time level $n$,
Therefore equation (3.26) can be cast to

\[ U^n_j V^n_j \left( \frac{1}{t^n - t^{n-2}} + \frac{1}{t^n - t^{n-1}} \right) - U^{n-1}_j V^{n-1}_j \left( \frac{1}{t^{n-1} - t^{n-2}} + \frac{1}{t^{n-1} - t^{n-2}} \right) \]

\[ + U^{n-2}_j V^{n-2}_j \left( \frac{1}{t^{n-1} - t^{n-2}} - \frac{1}{t^{n-1} - t^{n-2}} \right) - \frac{U^n_{j+\frac{1}{2}} V^*_{j+\frac{1}{2}} - U^n_{j-\frac{1}{2}} V^*_{j-\frac{1}{2}}}{t^{n+\frac{1}{2}} - t^{n-\frac{1}{2}}} + R(U^n) = 0 \]  

(3.28)

### 3.7.2 Cell-centered finite-volume spacetime framework

The spacetime formulation for the same one-dimensional problem in terms of spacetime surface integrals is

\[ \oint_{\partial V_{ST}} (U n_t + F n_x) dS = 0 \]  

(3.29)

where \( n_t \) denotes the normal vector component in the time direction \( t \). Looking at Figure 3.2 the discretization
3.7. Comparison between an arbitrary Lagrangian-Eulerian formulation and the spacetime framework

of equation (3.29) at spacetime cell \((j,n)\) yields

\[
\begin{align*}
U_{j}^{n+\frac{1}{2}}V_{j}^{n+\frac{1}{2}} - U_{j}^{n-\frac{1}{2}}V_{j}^{n-\frac{1}{2}} + U_{j+\frac{1}{2}} A_{j+\frac{1}{2}} - U_{j-\frac{1}{2}} A_{j-\frac{1}{2}} n_t \\
\quad + F_{j+\frac{1}{2}} A_{j+\frac{1}{2}} n_x - F_{j-\frac{1}{2}} A_{j-\frac{1}{2}} n_x = 0
\end{align*}
\]

(3.30)

where \(A_{j+\frac{1}{2}}\) are spacetime areas (i.e. of dimension \([L^2 T^2]\), where \(L\) is length and \(T\) is time) as depicted in Figure 3.2. The value of the conserved variables is calculated and stored at the spacetime cell centres, therefore the value of \(U_{j}^{n+\frac{1}{2}}\) at time level \(n+\frac{1}{2}\), and \(U_{j}^{n-\frac{1}{2}}\) at time level \(n-\frac{1}{2}\), via a second-order upwind discretisation may be worked out as follows

\[
\begin{align*}
U_{j}^{n+\frac{1}{2}} &= U_{j}^{n} + \left(\frac{\partial U}{\partial t}\right)_j^n \left(\frac{t^{n+\frac{1}{2}} - t^{n-\frac{1}{2}}}{2}\right) \\
U_{j}^{n-\frac{1}{2}} &= U_{j}^{n-1} + \left(\frac{\partial U}{\partial t}\right)_j^{n-1} \left(\frac{t^{n-\frac{1}{2}} - t^{n-\frac{3}{2}}}{2}\right)
\end{align*}
\]

(3.31) (3.32)

A first-order upwind approximation\(^2\) of the time derivatives at time levels \(n\) and \(n-1\) yields

\[
\begin{align*}
\left(\frac{\partial U}{\partial t}\right)_j^n &= \frac{U_{j}^{n} - U_{j}^{n-1}}{t^n - t^{n-1}} \\
\left(\frac{\partial U}{\partial t}\right)_j^{n-1} &= \frac{U_{j}^{n-1} - U_{j}^{n-2}}{t^{n-1} - t^{n-2}}
\end{align*}
\]

(3.33) (3.34)

Combining equations (3.31) (3.34) one can write

\[
\begin{align*}
U_{j}^{n+\frac{1}{2}} &= U_{j}^{n} \left(1 + \frac{t^{n+\frac{1}{2}} - t^n}{t^n - t^{n-1}}\right) - U_{j}^{n-1} \left(\frac{t^{n+\frac{1}{2}} - t^n}{t^n - t^{n-1}}\right) \\
U_{j}^{n-\frac{1}{2}} &= U_{j}^{n-1} \left(1 + \frac{t^{n-\frac{1}{2}} - t^{n-1}}{t^{n-1} - t^{n-2}}\right) - U_{j}^{n-2} \left(\frac{t^{n-\frac{1}{2}} - t^{n-1}}{t^{n-1} - t^{n-2}}\right)
\end{align*}
\]

(3.35) (3.36)

\(^2\)In the general case of a fully unstructured mesh in spacetime, the discretisation of time derivatives have some contributions from spatial terms.
which can be used in equation (3.30), as follows,

\[
U_j^n V_j^{n+\frac{1}{2}} \left( 1 + \frac{t^{n+\frac{1}{2}} - t^n}{t^n - t^{n-1}} \right) - U_j^{n-1} \left[ V_j^{n+\frac{1}{2}} \left( \frac{t^{n+\frac{1}{2}} - t^n}{t^n - t^{n-1}} \right) + V_j^{n-\frac{1}{2}} \left( 1 + \frac{t^{n-\frac{1}{2}} - t^{n-1}}{t^{n-1} - t^{n-2}} \right) \right] 
+ U_j^{n-2} V_j^{n-\frac{1}{2}} \left( \frac{t^{n-\frac{1}{2}} - t^{n-1}}{t^{n-1} - t^{n-2}} \right) + U_{j+\frac{1}{2}}^n V_{j+\frac{1}{2}}^* - U_{j-\frac{1}{2}}^n V_{j-\frac{1}{2}}^* 
+ \left( F_{j+\frac{1}{2}}^n A_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n A_{j-\frac{1}{2}}^n \right) \cdot \left( t^{n+\frac{1}{2}} - t^{n-\frac{1}{2}} \right) = 0
\]  

(3.37)

Finally, bearing in mind that

\[
t^{n+\frac{1}{2}} - t^{n-\frac{1}{2}} = 2 \left( t^{n+\frac{1}{2}} - t^n \right)
\]

(3.38)

and, rearranging, the spacetime discretisation can be cast to

\[
U_j^n V_j^{n+\frac{1}{2}} \left( \frac{1}{t^{n+\frac{1}{2}} - t^{n-\frac{1}{2}}} + \frac{1}{2} \frac{1}{t^n - t^{n-1}} \right) 
- U_j^{n-1} \left[ V_j^{n+\frac{1}{2}} \left( \frac{1}{2} \frac{1}{t^n - t^{n-1}} \right) + V_j^{n-\frac{1}{2}} \left( \frac{1}{2} \frac{1}{t^{n-\frac{1}{2}} - t^{n-2}} \right) \right] 
+ U_j^{n-2} V_j^{n-\frac{1}{2}} \left( \frac{1}{2} \frac{1}{t^{n-1} - t^{n-2}} \right) 
\]

(3.39)

Comparing equations (3.28) and (3.39), in arbitrary Lagrangian-Eulerian and spacetime formulations, respectively, it is clear that they are not directly equivalent with the current implementation of the spacetime framework. This is mainly due to the fact that the solution in spacetime is calculated at the centres of spacetime cells which, effectively, lie in between two time levels of the ALE formulation, hence the solution is shifted half a time step. Although equations (3.28) and (3.39) differ it is still possible to couple both formulations. For instance, if the solution \( U \) is calculated through an ALE formulation up to time level \( n \) across the whole spatial mesh (i.e. for all \( j \)), equation (3.39) may be used to initialize the solution of the spacetime domain between time planes \( t = t^{n-\frac{1}{2}} \) and \( t = t^{n+\frac{1}{2}} \). Notice that this coupling is only possible in this cell-centered case by the use of an interpolation to obtain the solution at time plane \( t = t^{n+\frac{1}{2}} \), as given by equation (3.31).

Moreover, a further simplification of equation (3.39) can be obtained in the following scenarios considered below: a constant time-step size throughout the whole time domain and a fixed spatial discretisation throughout time.
3.7. Comparison between an arbitrary Lagrangian-Eulerian formulation and the spacetime framework

Constant time-step size

In case the time-step size is constant one can write, at any time level \( k \),

\[
\Delta t = t^{k+\frac{1}{2}} - t^{k-\frac{1}{2}} = t^{k} - t^{k-1} = \frac{t^{k} - t^{k-2}}{2}
\]  

(3.40)

Therefore, equation (3.28) for the ALE formulation yields

\[
\frac{3U^n_jV^n_j - 4U^{n-1}_jV^{n-1}_j + U^{n-2}_jV^{n-2}_j}{2\Delta t} - \frac{U^n_{j+\frac{1}{2}}V^n_{j+\frac{1}{2}} - U^n_{j-\frac{1}{2}}V^n_{j-\frac{1}{2}}}{\Delta t} + R(U^n) = 0
\]  

(3.41)

whereas equation (3.39) for the spacetime framework yields

\[
\frac{3U^n_jV^n_{j+\frac{1}{2}} - U^{n-1}_jV^{n-1}_{j+\frac{1}{2}} + 3V^n_{j+\frac{1}{2}}}{2\Delta t} - \frac{U^n_{j+\frac{1}{2}}V^n_{j+\frac{1}{2}} - U^n_{j-\frac{1}{2}}V^n_{j-\frac{1}{2}}}{\Delta t} + R(U^n) = 0
\]  

(3.42)

Although very similar they are not directly equivalent, hence a coupling between an ALE and a spacetime formulation would not be straightforward. An interpolation method is necessary if a coupling between both frameworks is desired which may introduce numerical errors. For a direct coupling, the volume of cell \( j \) must also remain constant throughout time, i.e. \( V_{j+\frac{1}{2}} = V_{j-\frac{1}{2}} \), but this is only the case when no moving boundaries exist in the problem.

Constant time-step size with fixed grid throughout time

In the event the grid is fixed throughout time, on top of a constant time-step size, a further assumption can be made at time level \( k \), namely

\[
V_j = V_j^{k+\frac{1}{2}} = V_j^{k} = V_j^{k-\frac{1}{2}} = V_j^{k-1} = V_j^{k-2}
\]  

(3.43)

This can be seen in Figure 3.3. Both equations (3.41) and (3.42) can now be identically cast to

\[
\frac{3U^n_jV_j - 4U^{n-1}_jV_j + U^{n-2}_jV_j}{2\Delta t} + R(U^n) = 0
\]  

(3.44)
Although both the spacetime and ALE frameworks could, in theory, be coupled in this specific case, the flexibility that the spacetime formulation brings is unnecessary. The transient problem between constant and uniform fluid properties to the steady-state solution is one of such problems where both formulations are identical and, thus, could be used within the same simulation without further modifications or interpolations.

It is important to bear in mind that this analysis accounts only for the time derivatives and nothing has been said in regard to the spatial derivatives which determine the value of residuals $R(U^n)$. Hence, it applies to any spatial discretisation provided that the same is used in both formulations.

### 3.7.3 Vertex-centered finite-volume spacetime framework

Based on the above analysis, it may be hypothesised that a vertex-centered implementation of the spacetime framework would allow a direct coupling with an arbitrary Lagrangian-Eulerian formulation. In such a case (see Figure 3.4), the coupling could happen on a single $t = \text{constant}$ plane as opposed to a region between two planes $t^{n-\frac{1}{2}}$ and $t^{n+\frac{1}{2}}$ if a cell-centered implementation was used.
3.7. Comparison between an arbitrary Lagrangian-Eulerian formulation and the spacetime framework

The discretization of equation (3.29) at the spacetime cells outlined in Figure 3.4, between time levels \( n - 1 \) and \( n \), yields

\[
\left( \frac{U_{j+1}^n + U_j^n}{2} \right) \left( \frac{V_{j+1}^n + V_j^n}{2} \right) - \left( \frac{U_{j+1}^{n-1} + U_j^{n-1}}{2} \right) \left( \frac{V_{j+1}^{n-1} + V_j^{n-1}}{2} \right) + \left( \frac{U_{j+1}^n + U_j^{n-1}}{2} \right) A_{j+1}^n n_t
\]

\[= - \left( \frac{U_j^n + U_j^{n-1}}{2} \right) A_j^* n_t + \left( \frac{F_{j+1}^n + F_j^{n-1}}{2} \right) A_{j+1}^* n_x - \left( \frac{F_j^n + F_j^{n-1}}{2} \right) A_j^* n_x = 0 \tag{3.45} \]

and

\[
\left( \frac{U_j^n + U_{j-1}^n}{2} \right) \left( \frac{V_j^n + V_{j-1}^n}{2} \right) - \left( \frac{U_j^{n-1} + U_{j-1}^{n-1}}{2} \right) \left( \frac{V_j^{n-1} + V_{j-1}^{n-1}}{2} \right) + \left( \frac{U_j^n + U_{j-1}^{n-1}}{2} \right) A_{j-1}^n n_t
\]

\[= - \left( \frac{U_{j-1}^n + U_{j-1}^{n-1}}{2} \right) A_{j-1}^* n_t + \left( \frac{F_{j-1}^n + F_{j-1}^{n-1}}{2} \right) A_{j-1}^* n_x - \left( \frac{F_j^n + F_{j-1}^{n-1}}{2} \right) A_{j-1}^* n_x = 0 \tag{3.46} \]

where \( A_j^* \) and \( A_{j \pm 1}^* \) are spacetime areas (i.e. of dimension \([L^2 T] \), where \( L \) is length and \( T \) is time) given by (denote \( k \) an index which takes the values: \( j-1, j \) or \( j+1 \))

\[
A_k^* n_x = \left( \frac{A_k^n + A_k^{n-1}}{2} \right) (t^n - t^{n-1}) \tag{3.47} \]
\[ A_k^* n_t = V_k^* \]  

as depicted in Figure 3.4.

If the solution \( \mathbf{U} \) is calculated through an ALE formulation up to time level \( n \) across the whole spatial mesh (i.e. for all \( j \)), a coupling with the spacetime framework could be obtained applying equation (3.45) or (3.46) to all the spacetime cells at the initial boundary of the spacetime domain, i.e. at \( t = t^n \) plane. A direct comparison with the arbitrary Lagrangian-Eulerian formulation in equation (3.28) requires the combination of both equations (3.45) and (3.46). In contrast with the cell-centered spacetime formulation (section 3.7.2), it is interesting to realise that although a larger spatial stencil is necessary in this vertex-centered spacetime formulation (e.g. the solution of \( \mathbf{U}_j^n \) is coupled with the solution of \( \mathbf{U}_{j-1}^n \) and \( \mathbf{U}_{j+1}^n \)), there is no real need for interpolating the ALE solution \( \mathbf{U}^n \) to obtain the solution at \( t = t^{n+\frac{1}{2}} \) because both lie at the exact same time level.
Chapter 4

Stability analysis

4.1 Introduction

The stability of a numerical scheme can be assessed by looking at the errors of the solution throughout the time integration [105]. These errors need to be bounded and must decay from one step to the next when marching in time or pseudo-time. There exist many methods to assess the stability of a certain numerical scheme. In general, the consideration of real boundary conditions lead to very complicated analyses. A von Neumann stability analysis, which imposes periodic boundary conditions, yields a much simpler study and the condition obtained resembles closely that of the stability analysis including real boundary conditions.

Different combinations of space and time discretisations yield different behaviours. For a numerical method to be stable the eigenvalue spectrum of the semi-discrete system, i.e. the locus of all eigenvalues of the spatial discretisation throughout the time integration, must be contained within the region of stability of the numerical (time) integration method used. In particular, explicit schemes such as the fourth-order four-stages Runge-Kutta used in this work to march the solution in time or pseudo-time can be unstable if the values of $\Delta x$ and $\Delta t$ do not satisfy a CFL condition.

It was hypothesised that the CFL condition for the integration of the spacetime formulation of the Navier-Stokes equations [2,80] is different from that for a conventional finite difference method for the time integration of the semi-discrete system of equations. Therefore, a von Neumann stability analysis is performed in this chapter for the spacetime formulation. The results obtained in this chapter have been used in the simulations to accelerate the convergence of the solution through local time-stepping in pseudo-time $t^*$. 

First of all, the stability of a one-dimensional convective-diffusive equation is investigated in Section 4.2. This
shows that the eigenvalue spectrum of the semi-discrete system, $\bar{\lambda} \Delta t$, is represented by a region in the complex plane $\mathbb{C}$, where convective terms yield eigenvalues on the imaginary axis, $\Re (\bar{\lambda}_c \Delta t_c) = 0$, and diffusive terms on the real one, $\Im (\bar{\lambda}_d \Delta t_d) = 0$. The combination of both leads to eigenvalues in the complex plane, i.e. $\bar{\lambda} \Delta t = \Re (\bar{\lambda}_d \Delta t_d) + I \Im (\bar{\lambda}_c \Delta t_c) \in \mathbb{C}$ where $I = \sqrt{-1}$. Moreover, the necessary but not sufficient von Neumann stability condition, which states that the growth of the numerical solution must decrease for large numbers of iterations, requires that all eigenvalues lie on the left half of the complex plane, i.e. they all must have a negative real part, $\Re (\bar{\lambda} \Delta t) = \bar{\lambda}_d \Delta t_d \leq 0$.

Since the eigenvalue spectrum of the semi-discrete system must be contained within the region of stability of the time integration method and an explicit fourth-order Runge-Kutta method has been used in this work, looking at Figure 4.1 a constraint on the maximum time step size may be imposed as

$$\Delta t \leq \min \{ \Delta t_c, \Delta t_d \} \quad (4.1)$$

where $\Delta t_c$ and $\Delta t_d$ are the maximum allowed time step sizes if only convective or only diffusive terms were considered, i.e. such that $0 \leq \lambda_d \Delta t_d \leq 2.78$ and $2\sqrt{2} I \leq \lambda_c \Delta t_c \leq 2\sqrt{2} I$, respectively. Using condition (4.1) all eigenvalues are contained within the green rectangle in Figure 4.1. A more restrictive and safer condition is
used in the present work to ensure the stability of the numerical solution. It can be written as follows

\[
\frac{1}{\Delta t} \geq \frac{1}{\Delta t_c} + \frac{1}{\Delta t_d} \tag{4.2}
\]

This condition corresponds to all eigenvalues contained within the red rectangle in Figure 4.1. Notice that, although the red rectangle is always contained within the green one, its size depends on the value of some quantities specific to each problem.

Next, in Sections 4.3.1 and 4.3.2 a stability condition is obtained for the inviscid (or convective) terms of the spacetime formulation of the Navier-Stokes equations (2.80) via central-difference and upwind biased stencils, \((\Delta t^*_c)_{CD}\) and \((\Delta t^*_c)_{UW}\) respectively. In Section 4.3.3 the stability condition for viscous (or diffusive) terms of the spacetime formulation of the Navier-Stokes equations (2.80) is obtained for a central-difference discretisation, yielding the value of \((\Delta t^*_d)_{CD}\). From these values a CFL condition can be obtained using equation (4.2) for all the schemes used in this work, as summarised in Table 4.1. Notice that this condition in spacetime is imposed strictly on pseudo-time \(t^*\). Moreover, the eigenvalues \(\bar{\lambda}_c\) and \(\bar{\lambda}_d\) of the semi-discrete system include a contribution from the physical time discretisation.

### 4.2 One-dimensional convective-diffusive equation

Consider the one-dimensional convective-diffusive equation

\[u_t + au_x = \alpha u_{xx}\]

where \(a \in \mathbb{R}\) is the convective velocity and \(\alpha \in \mathbb{R}\) a diffusion coefficient. A semi-discretisation (i.e. spatial discretisation only) yields, at point \(i\) in the mesh,

\[
\frac{du_i}{dt} + \frac{a}{2\Delta x} (u_{i+1} - u_{i-1}) = \frac{\alpha}{\Delta x^2} (u_{i+1} - 2u_i + u_{i-1}) \tag{4.3}
\]

where a central-difference scheme has been used. The spatial and temporal contributions are decoupled by the assumption that the solution can be decomposed into a number of waves of time-dependent amplitude that span the spatial domain. Hence, a finite Fourier series is used to express the numerical solution as

\[
u (t, i\Delta x) = \sum_{m=-N/2}^{N/2} \hat{u}_m (t) e^{ik_m i\Delta x} = \sum_{m=-N/2}^{N/2} \hat{u}_m (t) e^{i\phi_m i} \tag{4.4}
\]

where \(N\) is the number of positive and negative harmonics, \(I = \sqrt{-1}\) and \(\hat{u}_m (t)\) is the amplitude of the \(m\)-th harmonic with wave number \(k_m\) and phase angle \(\phi_m = k_m \Delta x\). Substituting the contribution of the \(m\)-th
Chapter 4: Stability analysis

The harmonic in (4.3) yields

$$\frac{d\hat{u}_m}{dt} = \left[ \frac{2\alpha}{\Delta x^2} \left( e^{I\phi_m} + e^{-I\phi_m} \right) - Ia \left( e^{I\phi_m} - e^{-I\phi_m} \right) \right] \hat{u}_m(t) \tag{4.5}$$

Using the trigonometric identity

$$e^{I\theta} = \cos \theta + I \sin \theta \tag{4.6}$$

one can finally write, for the $m$-th harmonic,

$$\frac{d\hat{u}_m}{dt} = \bar{\lambda}_m \hat{u}_m(t) \quad \Rightarrow \quad \hat{u}_m(t) = C_m e^{\bar{\lambda}_m t} \tag{4.7}$$

where $C_m$ is a constant value and the associated eigenvalue $\bar{\lambda}_m$ is given by

$$\bar{\lambda}_m \Delta t = 2 \frac{\alpha\Delta t}{\Delta x^2} (\cos \phi_m - 1) - Ia \frac{\alpha\Delta t}{\Delta x} \sin \phi_m \tag{4.8}$$

Notice that equation (4.8) represents the eigenvalue spectrum of the $m$-th harmonic of the semi-discrete system (4.3) when the phase angle $\phi_m$ ranges from $-\pi$ to $\pi$. In other words, $\bar{\lambda}_m$ corresponds to the eigenvalue of the $m$-th harmonic of the spatial discretisation. Finally, a time discretisation of (4.7) yields the following amplification factor

$$G = \frac{\hat{u}_m^{n+1}}{\hat{u}_m^n} = e^{\lambda_m \Delta t} \tag{4.9}$$

where the subscript $m$ has been dropped for the sake of simplicity. For the numerical scheme to be stable the growth of the solution in time must decrease, hence $|G| \leq 1$, which yields the following condition

$$\Re (\bar{\lambda}_m \Delta t) \leq 0 \tag{4.10}$$

where $\Re(z)$ is the real part of complex number $z \in \mathbb{C}$. This is also known as the necessary but not sufficient von Neumann stability condition and $\bar{\lambda}_m \Delta t$ is the so-called Fourier symbol of the spatial discretisation. Writing
all degrees of freedom in the mesh, i.e. the time-dependent solution at each node \( \hat{u}_i(t) \), in a column vector 
\( \hat{U}(t) = [\hat{u}_1(t), \hat{u}_2(t), \ldots, \hat{u}_N(t)]^T \), the semi-discrete system \((4.3)\) can be written in matrix form as

\[
\frac{d\hat{U}}{dt} = S\hat{U} \tag{4.11}
\]

where matrix \( S \) represents the spatial discretisation. The eigenvalues of \( S \) are \( \lambda_m \) and their associated eigenvectors are the functions \( \hat{V}^{(m)} = e^{ik_m x} \). To prove this bear in mind that the right eigenvectors of \( S \) are the column vectors \( \hat{V}^{(j)} \) that satisfy the following equation

\[
S\hat{V}^{(j)} = \lambda_j \hat{V}^{(j)} \tag{4.12}
\]

where \( \lambda_j \) are their associated eigenvalues. Therefore, the solution \( \hat{U}(t) \) of \((4.11)\) can be written in terms of the eigenvectors \( \hat{V}^{(j)} \), which constitute a basis of the solution space, as

\[
\hat{U}(t) = \sum_{j=1}^{N} \hat{u}_j(t) \hat{V}^{(j)} \tag{4.13}
\]

This decomposition into spatial and temporal contributions is called modal decomposition \([105]\). Using \((4.13)\) in equation \((4.11)\) yields, for each mode \( j \),

\[
\hat{u}_j(t) = C_j e^{\lambda_j t} \tag{4.14}
\]

which is identical to the solution of the \( m \)-th harmonic in the von Neumann analysis, equation \((4.7)\).

Equation \((4.9)\) represents only the exact amplification factor of the semi-discrete system \((4.11)\) which is independent of the numerical (time) integration method used. The numerical amplification factor depends on the scheme used. Writing the update of the solution \( \hat{U} \) between time levels \( n \) and \( n + 1 \) as

\[
\hat{U}^{n+1} = C\hat{U}^n \tag{4.15}
\]

where matrix \( C \) represents the update operator, the stability condition requires, for all eigenvalues of \( C \), that

\[
|z_C| \leq 1 \tag{4.16}
\]
The maximum eigenvalue of the update operator $C$ defined in (4.15) for the fourth-order four-stages explicit Runge-Kutta scheme used throughout this work, given by equations (3.10)-(3.14), has been depicted in Figure 4.2 and can be written as [105]

$$z_C = 1 + \lambda \Delta t + \left(\frac{\lambda \Delta t}{2}\right)^2 + \left(\frac{\lambda \Delta t}{6}\right)^3 + \left(\frac{\lambda \Delta t}{24}\right)^4$$  \hspace{1cm} (4.17)

The eigenvalue spectrum of the spatial discretisation, equation (4.8), needs to be contained within the shaded area in Figure 4.2. Depending on the values of $a \Delta t / \Delta x$ and $\alpha \Delta t / \Delta x^2$, the scheme can be stable or unstable. For instance, if $\alpha = 0$ and $a \neq 0$ all eigenvalues lie on the imaginary axis and only the interval $[-2\sqrt{2}, 2\sqrt{2}] \in \mathbb{C}$ is contained within the stability region of the explicit fourth-order Runge-Kutta scheme. Hence the CFL condition in this case is

$$\left|\frac{a \Delta t_c}{\Delta x}\right| \leq 2\sqrt{2}$$  \hspace{1cm} (4.18)

Similarly, if $a = 0$ and $\alpha \neq 0$ all eigenvalues lie on the negative real axis and only the interval $[-2.78, 0] \in \mathbb{R}$ is contained within the stability region. Hence the CFL condition in this case is

$$0 \leq \frac{\alpha \Delta t_d}{\Delta x^2} \leq 0.695$$  \hspace{1cm} (4.19)

In general, when $a \neq 0$ and $\alpha \geq 0$, the eigenvalue spectrum of the semi-discrete system (4.3) spans the region inside the green dashed curve in Figure 4.2. In this case, the CFL condition can be estimated using the values of $\Delta t_c$ and $\Delta t_d$ from (4.18)-(4.19) in equation (4.2).

In the following sections a similar stability analysis is performed on the spacetime formulation of the Navier-Stokes equations. The Symbolic Math Toolbox available through MATLAB® has been used for the calculation of Jacobian matrices and the solution of eigenvalues and eigenvectors problems.
4.3. Spacetime formulation of the two-dimensional Navier-Stokes equations

The spacetime formulation of the two-dimensional Navier-Stokes equations (2.80) can be written in differential form as

\[
\frac{\partial U}{\partial t} + \frac{\partial U}{\partial x} (F_x - F^v_x) + \frac{\partial U}{\partial y} (F_y - F^v_y) = 0
\]

(4.20)

In order to simplify calculations, augmented kinematic viscosity \( \hat{\mu} \) and augmented Prandtl number \( \hat{Pr} \) are defined as

\[
\hat{\mu} = \mu + \mu_t
\]

(4.21)

\[
\hat{Pr} = \frac{\mu Pr_t \hat{Pr}_t}{\mu Pr_t + \mu_t \hat{Pr}_t}
\]

(4.22)
where notice that \( \hat{\rho}_r \) has been defined such that the ratio \( \hat{\mu} / \rho_r \) yields

\[
\hat{\rho}_r = \frac{\rho_r}{\mu} + \frac{\mu_t}{\rho_r}
\]  

(4.23)

Bearing in mind that inviscid fluxes are functions of the conserved variables, \( F_x(U) \) and \( F_y(U) \), and viscous fluxes are functions of their derivatives, \( F_{v,xx}(U_x, U_y) \) and \( F_{v,yy}(U_x, U_y) \), equation (4.20) can be linearised as

\[
\frac{\partial U}{\partial t} + A_x \frac{\partial U}{\partial x} + A_y \frac{\partial U}{\partial y} = A_{v,xx} \frac{\partial^2 U}{\partial x^2} + (A_{v,xy} + A_{v,yx}) \frac{\partial^2 U}{\partial x \partial y} + A_{v,yy} \frac{\partial^2 U}{\partial y^2}
\]

(4.24)

where the matrices \( A_x \) and \( A_y \) of linearised inviscid fluxes are given by

\[
A_x = \frac{\partial F_x}{\partial U} = \begin{bmatrix}
0 & 1 & 0 & 0 \\
-u^2 + (\gamma - 1) \frac{|v|^2}{2} & (3 - \gamma) u & -(\gamma - 1) v & \gamma - 1 \\
-uv & v & u & 0 \\
-u (\gamma E - (\gamma - 1) |v|^2) & \gamma E - (\gamma - 1) \left( u^2 + \frac{|v|^2}{2} \right) & -(\gamma - 1) uv & \gamma u
\end{bmatrix}
\]

(4.25)

\[
A_y = \frac{\partial F_y}{\partial U} = \begin{bmatrix}
0 & 0 & 1 & 0 \\
-uv & v & u & 0 \\
-v^2 + (\gamma - 1) \frac{|v|^2}{2} & -(\gamma - 1) u & (3 - \gamma) v & \gamma - 1 \\
-v (\gamma E - (\gamma - 1) |v|^2) & -(\gamma - 1) uv & \gamma E - (\gamma - 1) \left( v^2 + \frac{|v|^2}{2} \right) & \gamma v
\end{bmatrix}
\]

(4.26)

and matrices \( A_{v,xx}, A_{v,xy}, A_{v,yx}, \) and \( A_{v,yy} \) of linearised viscous fluxes are

\[
A_{v,xx} = \frac{\partial F_{v,xx}}{\partial U_x} = \frac{\hat{\mu}}{\rho Re_\infty} \begin{bmatrix}
0 & 0 & 0 & 0 \\
-4u & 4 & 0 & 0 \\
-\frac{3}{3} & 0 & 1 & 0 \\
-\frac{\gamma |v|^2}{2 \rho_r} - \left( \frac{4u^2}{3} + v^2 \right) - \frac{a^2}{\rho_r (\gamma - 1)} \left( \frac{4}{3} - \frac{\gamma}{\rho_r} \right) u \left( 1 - \frac{\gamma}{\rho_r} \right) v & \frac{\gamma}{\rho_r}
\end{bmatrix}
\]

(4.27)
4.3. Spacetime formulation of the two-dimensional Navier-Stokes equations

\[ A_{xy}^v = \frac{\partial F_x^v}{\partial U_y} = \frac{\hat{\mu}}{\rho Re \infty} \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{2v}{3} & 0 & -\frac{2}{3} & 0 \\ -u & 1 & 0 & 0 \\ \frac{-uv}{3} & v & -\frac{2u}{3} & 0 \end{bmatrix} \] (4.28)

\[ A_{yx}^v = \frac{\partial F_y^v}{\partial U_x} = \frac{\hat{\mu}}{\rho Re \infty} \begin{bmatrix} 0 & 0 & 0 & 0 \\ -v & 0 & 1 & 0 \\ \frac{2u}{3} & -\frac{2}{3} & 0 & 0 \\ \frac{-uv}{3} & -\frac{2v}{3} & u & 0 \end{bmatrix} \] (4.29)

\[ A_{yy}^v = \frac{\partial F_y^v}{\partial U_y} = \frac{\hat{\mu}}{\rho Re \infty} \begin{bmatrix} 0 & 0 & 0 & 0 \\ -u & 1 & 0 & 0 \\ \frac{-4v}{3} & 0 & 4 & 0 \\ \frac{\gamma |\mathbf{v}|^2}{2Pr} - \left( \frac{u^2 + 4v^2}{3} \right) - \frac{a^2}{Pr(\gamma - 1)} \left( 1 - \frac{\gamma}{Pr} \right) u \left( \frac{4}{3} - \frac{\gamma}{Pr} \right) v \frac{\gamma}{Pr} \end{bmatrix} \] (4.30)

where \(|\mathbf{v}|^2 = u^2 + v^2 + w^2\), \(U_x = \frac{\partial U}{\partial x}\) and \(U_y = \frac{\partial U}{\partial y}\). Notice that, according to Schwarz theorem [111], the following equality of the symmetry of partial derivatives of \(U\) has been used

\[ \frac{\partial^2 U}{\partial x \partial y} = \frac{\partial^2 U}{\partial y \partial x} \] (4.31)

Equation [4.24] may be written in terms of the primitive variables \(W = \{\rho, u, v, p\}^T\) as

\[ \frac{\partial W}{\partial t} + \frac{\partial W}{\partial t} + B_x \frac{\partial W}{\partial x} + B_y \frac{\partial W}{\partial y} = B_{xx}^v \frac{\partial^2 W}{\partial x^2} + (B_{xy}^v + B_{yx}^v) \frac{\partial^2 W}{\partial x \partial y} + B_{yy}^v \frac{\partial^2 W}{\partial y^2} \] (4.32)

where the matrices \(B_x\) and \(B_y\) of linearised inviscid fluxes in terms of the primitive variables \(W\) have been
computed as

$$B_x = T_{U \rightarrow W}^{-1} A_x T_{U \rightarrow W} = \begin{bmatrix}
    u & \rho & 0 & 0 \\
    0 & u & 0 & \frac{1}{\rho} \\
    0 & 0 & u & 0 \\
    0 & \rho a^2 & 0 & u
\end{bmatrix}$$  \hspace{1cm} (4.33)$$

$$B_y = T_{U \rightarrow W}^{-1} A_y T_{U \rightarrow W} = \begin{bmatrix}
v & 0 & \rho & 0 \\
0 & v & 0 & 0 \\
0 & 0 & v & \frac{1}{\rho} \\
0 & 0 & \rho a^2 & v
\end{bmatrix}$$  \hspace{1cm} (4.34)$$

and the matrices $B_{xx}^v$, $B_{xy}^v$, $B_{yx}^v$, and $B_{yy}^v$ of linearised viscous fluxes in terms of the primitive variables $W$ are given by

$$B_{xx}^v = T_{U \rightarrow W}^{-1} A_{xx} T_{U \rightarrow W} = \frac{\hat{\mu}}{\rho Re_{\infty}} \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 4/3 & 0 & 0 \\
0 & 0 & 1 & 0 \\
-\frac{a^2}{Pr} & 0 & 0 & \gamma/Pr
\end{bmatrix}$$  \hspace{1cm} (4.35)$$

$$B_{xy}^v = T_{U \rightarrow W}^{-1} A_{xy} T_{U \rightarrow W} = \frac{\hat{\mu}}{\rho Re_{\infty}} \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & -2/3 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}$$  \hspace{1cm} (4.36)$$

$$B_{yx}^v = T_{U \rightarrow W}^{-1} A_{yx} T_{U \rightarrow W} = \frac{\hat{\mu}}{\rho Re_{\infty}} \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & -2/3 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}$$  \hspace{1cm} (4.37)$$
4.3. Spacetime formulation of the two-dimensional Navier-Stokes equations

\[
B_{yy}^v = T_{U \to W}^{-1} A_{yy} T_{U \to W} = \frac{\hat{\mu}}{\rho Re_{\infty}} \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 4/3 & 0 \\
-\frac{a^2}{Pr} & 0 & 0 & \gamma/Pr
\end{bmatrix}
\] (4.38)

The transformation matrix \( T_{U \to W} \) between conserved and primitive variables, \( U \) and \( W \) respectively, is given by

\[
T_{U \to W} = \frac{\partial U}{\partial W} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
u & \rho & 0 & 0 \\
v & 0 & \rho & 0 \\
u^2 + v^2 - \frac{u}{\rho} & \rho \nu & \rho v & 1/(-\gamma + 1)
\end{bmatrix}
\] (4.39)

and the inverse transformation \( T_{U \to W}^{-1} \) by

\[
T_{U \to W}^{-1} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
\frac{-u}{\rho} & 1 & \frac{1}{\rho} & 0 & 0 \\
\frac{-v}{\rho} & 0 & \frac{1}{\rho} & 0 \\
(\gamma - 1) \frac{u^2 + v^2}{2} & -u(\gamma - 1) & -v(\gamma - 1) & (\gamma - 1)
\end{bmatrix}
\] (4.40)

Using the following generalised curvilinear spacetime coordinates

\[
\xi = \xi(t, x, y) \quad \eta = \eta(t, x, y) \quad \zeta = \zeta(t, x, y)
\] (4.41)

equation (4.32) can be cast into

\[
\frac{\partial W}{\partial t^*} + B_\xi \frac{\partial W}{\partial \xi} + B_\eta \frac{\partial W}{\partial \eta} + B_\zeta \frac{\partial W}{\partial \zeta} = B_{\xi \xi} \frac{\partial^2 W}{\partial \xi^2} + B_{\eta \eta} \frac{\partial^2 W}{\partial \eta^2} + B_{\zeta \zeta} \frac{\partial^2 W}{\partial \zeta^2} + 2B_{\xi \eta} \frac{\partial^2 W}{\partial \xi \partial \eta} + 2B_{\xi \zeta} \frac{\partial^2 W}{\partial \xi \partial \zeta} + 2B_{\eta \zeta} \frac{\partial^2 W}{\partial \eta \partial \zeta}
\] (4.42)
where the matrices \( \mathbf{B}_\xi \), \( \mathbf{B}_\eta \) and \( \mathbf{B}_\zeta \) of generalised inviscid fluxes are given by

\[
\mathbf{B}_\xi = \xi_t \mathbf{I} + \xi_x \mathbf{B}_x + \xi_y \mathbf{B}_y - \xi_{xx} \mathbf{B}_{xx}^v - \xi_{xy} \left( \mathbf{B}_{xy}^v + \mathbf{B}_{yx}^v \right) - \xi_{yy} \mathbf{B}_{yy}^v \tag{4.43}
\]

\[
\mathbf{B}_\eta = \eta_t \mathbf{I} + \eta_x \mathbf{B}_x + \eta_y \mathbf{B}_y - \eta_{xx} \mathbf{B}_{xx}^v - \eta_{xy} \left( \mathbf{B}_{xy}^v + \mathbf{B}_{yx}^v \right) - \eta_{yy} \mathbf{B}_{yy}^v \tag{4.44}
\]

\[
\mathbf{B}_\zeta = \zeta_t \mathbf{I} + \zeta_x \mathbf{B}_x + \zeta_y \mathbf{B}_y - \zeta_{xx} \mathbf{B}_{xx}^v - \zeta_{xy} \left( \mathbf{B}_{xy}^v + \mathbf{B}_{yx}^v \right) - \zeta_{yy} \mathbf{B}_{yy}^v \tag{4.45}
\]

and the matrices \( \mathbf{B}_{\xi\xi}^v \), \( \mathbf{B}_{\eta\eta}^v \), \( \mathbf{B}_{\zeta\zeta}^v \), \( \mathbf{B}_{\xi\eta}^v \), \( \mathbf{B}_{\xi\zeta}^v \) and \( \mathbf{B}_{\eta\zeta}^v \) of generalised viscous fluxes are defined as follows

\[
\mathbf{B}_{\xi\xi}^v = \xi_x \xi_x \mathbf{B}_{xx}^v + \xi_x \xi_y \left( \mathbf{B}_{xy}^v + \mathbf{B}_{yx}^v \right) + \xi_y \xi_y \mathbf{B}_{yy}^v \tag{4.46}
\]

\[
\mathbf{B}_{\eta\eta}^v = \eta_x \eta_x \mathbf{B}_{xx}^v + \eta_x \eta_y \left( \mathbf{B}_{xy}^v + \mathbf{B}_{yx}^v \right) + \eta_y \eta_y \mathbf{B}_{yy}^v \tag{4.47}
\]

\[
\mathbf{B}_{\zeta\zeta}^v = \zeta_x \zeta_x \mathbf{B}_{xx}^v + \zeta_x \zeta_y \left( \mathbf{B}_{xy}^v + \mathbf{B}_{yx}^v \right) + \zeta_y \zeta_y \mathbf{B}_{yy}^v \tag{4.48}
\]

\[
\mathbf{B}_{\xi\eta}^v = \xi_x \eta_x \mathbf{B}_{xx}^v + \left( \frac{\xi_x \eta_y + \xi_y \eta_x}{2} \right) \left( \mathbf{B}_{xy}^v + \mathbf{B}_{yx}^v \right) + \xi_y \eta_y \mathbf{B}_{yy}^v \tag{4.49}
\]

\[
\mathbf{B}_{\xi\zeta}^v = \xi_x \zeta_x \mathbf{B}_{xx}^v + \left( \frac{\xi_x \zeta_y + \xi_y \zeta_x}{2} \right) \left( \mathbf{B}_{xy}^v + \mathbf{B}_{yx}^v \right) + \xi_y \zeta_y \mathbf{B}_{yy}^v \tag{4.50}
\]

\[
\mathbf{B}_{\eta\zeta}^v = \eta_x \zeta_x \mathbf{B}_{xx}^v + \left( \frac{\eta_x \zeta_y + \eta_y \zeta_x}{2} \right) \left( \mathbf{B}_{xy}^v + \mathbf{B}_{yx}^v \right) + \eta_y \zeta_y \mathbf{B}_{yy}^v \tag{4.51}
\]

Again, there is a symmetry of the partial derivatives of \( \mathbf{W} \) according to Schwarz’s theorem, as follows

\[
\frac{\partial^2 \mathbf{W}}{\partial \xi \partial \eta} = \frac{\partial^2 \mathbf{W}}{\partial \eta \partial \xi} = \frac{\partial^2 \mathbf{W}}{\partial \xi \partial \zeta} = \frac{\partial^2 \mathbf{W}}{\partial \zeta \partial \xi} = \frac{\partial^2 \mathbf{W}}{\partial \eta \partial \zeta} = \frac{\partial^2 \mathbf{W}}{\partial \zeta \partial \eta} \tag{4.52}
\]

Moreover, due to the choice of a finite volume discretisation, the second derivatives of generalised coordinates vanish, i.e. \( \xi_{ij} = \eta_{ij} = \zeta_{ij} = 0 \) for \( i,j = x,y,z \). They only need to be taken into account when high-order discretisations (e.g. discontinuous-Galerkin) are used, where mesh elements are curved, i.e. the lines joining two nodes in the mesh are curved. Therefore, equations (4.43)-(4.45) in the current work can be simplified as

\[
\mathbf{B}_\xi = \xi_t \mathbf{I} + \xi_x \mathbf{B}_x + \xi_y \mathbf{B}_y \tag{4.53}
\]

\[
\mathbf{B}_\eta = \eta_t \mathbf{I} + \eta_x \mathbf{B}_x + \eta_y \mathbf{B}_y \tag{4.54}
\]

\[
\mathbf{B}_\zeta = \zeta_t \mathbf{I} + \zeta_x \mathbf{B}_x + \zeta_y \mathbf{B}_y \tag{4.55}
\]
Expanding matrices $B_\xi$, $B_\xi^\eta$ and $B_\eta^\eta$ yields

$$
B_\xi = \begin{bmatrix}
\xi_t + u\xi_x + v\xi_y & \rho\xi_x & \rho\xi_y & 0 \\
0 & \xi_t + u\xi_x + v\xi_y & 0 & \frac{\xi_x}{\rho} \\
0 & 0 & \xi_t + u\xi_x + v\xi_y & \frac{\xi_y}{\rho} \\
0 & \rho a^2\xi_x & \rho a^2\xi_y & \xi_t + u\xi_x + v\xi_y
\end{bmatrix}
$$

(4.56)

$$
B_\eta^\xi = \frac{\hat{\mu}}{\rho Re_\infty} \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & \frac{4}{3}\xi_x^2 + \xi_y^2 & \xi_x\xi_y & 0 \\
0 & \xi_y\xi_x & \xi_x^2 + \frac{4}{3}\xi_y^2 & 0 \\
-\frac{a^2}{Pr} (\xi_x^2 + \xi_y^2) & 0 & 0 & \frac{\gamma}{Pr} (\xi_x^2 + \xi_y^2)
\end{bmatrix}
$$

(4.57)

$$
B_\eta^\eta = \frac{\hat{\mu}}{\rho Re_\infty} \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & \frac{4}{3}\xi_x\eta_x + \xi_y\eta_y & \xi_x\eta_y + \xi_y\eta_x & 0 \\
0 & \xi_y\eta_x + \xi_x\eta_y & \xi_x\eta_x + \frac{4}{3}\xi_y\eta_y & 0 \\
-\frac{a^2}{Pr} (\xi_x\eta_x + \xi_y\eta_y) & 0 & 0 & \frac{\gamma}{Pr} (\xi_x\eta_x + \xi_y\eta_y)
\end{bmatrix}
$$

(4.58)

All other matrices can be found by replacing $\xi$ and $\eta$ in above equations by either $\xi$, $\eta$ or $\zeta$. In other words, the general form of matrices of inviscid fluxes $B_\xi$, $B_\eta$ and $B_\zeta$ can be cast into

$$
B_\alpha = \begin{bmatrix}
\alpha_t + u\alpha_x + v\alpha_y & \rho\alpha_x & \rho\alpha_y & 0 \\
0 & \alpha_t + u\alpha_x + v\alpha_y & 0 & \frac{\alpha_x}{\rho} \\
0 & 0 & \alpha_t + u\alpha_x + v\alpha_y & \frac{\alpha_y}{\rho} \\
0 & \rho a^2\alpha_x & \rho a^2\alpha_y & \alpha_t + u\alpha_x + v\alpha_y
\end{bmatrix}
$$

(4.59)
whereas the general form of matrices of viscous fluxes \( B^v_{\xi\xi}, B^v_{\eta\eta}, B^v_{\zeta\zeta}, B^v_{\xi\eta}, B^v_{\xi\zeta} \) and \( B^v_{\eta\zeta} \) is

\[
B^v_{\alpha\beta} = \frac{\hat{\mu}}{\rho Re_{\infty}} \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & \frac{4}{3} \alpha_x \beta_x + \alpha_y \beta_y & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{4}{3} \alpha_x \beta_x + \frac{4}{3} \alpha_y \beta_y & 0 \\
- \frac{a^2}{Pr} \left( \alpha_x \beta_x + \alpha_y \beta_y \right) & 0 & 0 & 0 & \frac{\gamma}{Pr} \left( \alpha_x \beta_x + \alpha_y \beta_y \right)
\end{bmatrix}
\]

(4.60)

where \( \alpha \) and \( \beta \) can be any of the generalized curvilinear coordinates \( \xi, \eta \) or \( \zeta \).

### 4.3.1 Inviscid terms with a second-order central-difference discretisation

Removing the viscous terms of the Navier-Stokes equations in (4.42) and adding some first order numerical dissipation terms as proposed by Jameson et al. [76], the inviscid Euler equations with general spacetime coordinates yield

\[
\frac{\partial W}{\partial t^*} + B_{\xi} \frac{\partial W}{\partial \xi} + B_{\eta} \frac{\partial W}{\partial \eta} + B_{\zeta} \frac{\partial W}{\partial \zeta} - \varepsilon \left( \Delta \xi \frac{\partial^2 W}{\partial \xi^2} + \Delta \eta \frac{\partial^2 W}{\partial \eta^2} + \Delta \zeta \frac{\partial^2 W}{\partial \zeta^2} \right) = 0
\]

(4.61)

where \( \varepsilon \in \mathbb{R} \) is a scaling factor for the dissipation and may depend on local pressure gradients. Decoupling this equation in the three different directions \( \xi, \eta \) and \( \zeta \) there exists a time-step size limitation for each of them. The maximum allowable value of \( (\Delta t^*_c)_{\text{CD}} \) for the three-dimensional spacetime system (4.61) is computed as

\[
(\Delta t^*_c)_{\text{CD}} = \min \left\{ (\Delta t^*_c)_{\xi}, (\Delta t^*_c)_{\eta}, (\Delta t^*_c)_{\zeta} \right\}
\]

(4.62)

Also, a more restrictive condition can be applied as follows

\[
\frac{1}{\Delta t^*_c} = \frac{1}{(\Delta t^*_c)_{\xi}} + \frac{1}{(\Delta t^*_c)_{\eta}} + \frac{1}{(\Delta t^*_c)_{\zeta}}
\]

(4.63)

Matrices \( B_{\xi}, B_{\eta} \) and \( B_{\zeta} \) in (4.61) have the same form, hence the same analysis applies in each of the three
4.3. Spacetime formulation of the two-dimensional Navier-Stokes equations

directions. For the general coordinate $\xi$ (the same applies to $\eta$ and $\zeta$) the following equation is considered

$$\frac{\partial W}{\partial t^*} + B_\xi \frac{\partial W}{\partial \xi} - \varepsilon \Delta \xi \frac{\partial^2 W}{\partial \xi^2} = 0 \quad (4.64)$$

The use of a central-difference scheme on the spacetime discretisation yields the semi-discrete system of equations

$$\frac{\partial W_{i,j,k}}{\partial t^*} + B_\xi \left( \frac{W_{i+1,j,k} - W_{i-1,j,k}}{2\Delta \xi} \right) - \varepsilon \left( \frac{W_{i+1,j,k} - 2W_{i,j,k} + W_{i-1,j,k}}{\Delta \xi} \right) = 0 \quad (4.65)$$

In order to calculate the Fourier symbol of the spacetime discretisation the solution is written as a finite Fourier series,

$$W(t, \xi, \eta, \zeta) = \frac{N}{2} \sum_{m=-N/2}^{N/2} \hat{W}_m(t) e^{i(k_m \Delta \xi + j \phi_m^\eta + k \phi_m^\zeta)} \quad (4.66)$$

where $\phi_m^\xi = k_m \Delta \xi$, $\phi_m^\eta = k_m \Delta \eta$ and $\phi_m^\zeta = k_m \Delta \zeta$ are the phase angles and $k_m$ is the wave number associated with the $m$-th harmonic. Introducing an arbitrary mode $m$ of solution (4.66) into equation (4.65) one can write

$$\frac{\partial \hat{W}_m}{\partial t^*} + \left( \frac{e^{I\phi_m^\xi} - e^{-I\phi_m^\xi}}{2\Delta \xi} \right) B_\xi - \varepsilon \left( \frac{e^{I\phi_m^\xi} - 2 + e^{-I\phi_m^\xi}}{\Delta \xi} \right) \hat{W}_m = 0 \quad (4.67)$$

Rearranging and using the trigonometric relationship (4.6) to write this equation in terms of sines and cosines yields

$$\frac{\partial \hat{W}_m}{\partial t^*} = \left[ \frac{2\varepsilon}{\Delta \xi} (\cos \phi_m^\xi - 1) I - \frac{I}{\Delta \xi} \sin \phi_m^\xi B_\xi \right] \hat{W}_m \quad (4.68)$$

Solving the eigenvalues and eigenvectors problem given by equation $\det (B_\xi - \lambda I) = 0$ matrix $B_\xi$ can be diagonalized as

$$\Lambda_\xi = Q_\xi^{-1} B_\xi Q_\xi = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{bmatrix} \quad (4.69)$$
Figure 4.3: Schematic representation of characteristic lines in spacetime. If $\xi = t$ then $\xi_t = 1$ and all eigenvalues are $dt/dt^* = \xi_t = 1$.

where the eigenvalues are (see diagram in Figure 4.3)

$$\lambda_{1,2} = \xi_t + u\xi_x + v\xi_y \pm a\sqrt{\xi_x^2 + \xi_y^2} \quad \lambda_{3,4} = \xi_t + u\xi_x + v\xi_y$$ (4.70)

and transformation matrix $Q_\xi$ and its inverse are

$$Q_\xi = \begin{bmatrix} \frac{1}{a^2} & \frac{1}{a^2} & 1 & 0 \\ \xi_x & -\frac{\xi_x}{\rho a\sqrt{\xi_x^2 + \xi_y^2}} & 0 & -\frac{\xi_y}{\xi_x} \\ \xi_y & -\frac{\xi_y}{\rho a\sqrt{\xi_x^2 + \xi_y^2}} & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$ (4.71)

$$Q_\xi^{-1} = \begin{bmatrix} 0 & \frac{\rho a\xi_x}{2\sqrt{\xi_x^2 + \xi_y^2}} & \frac{\rho a\xi_y}{2\sqrt{\xi_x^2 + \xi_y^2}} & 1 \\ 0 & -\frac{\rho a\xi_x}{2\sqrt{\xi_x^2 + \xi_y^2}} & -\frac{\rho a\xi_y}{2\sqrt{\xi_x^2 + \xi_y^2}} & 1 \\ 0 & \xi_x\xi_y & \xi_x^2 + \xi_y^2 & -\frac{1}{a^2} \\ 0 & -\xi_x\xi_y & \xi_x^2 + \xi_y^2 & 0 \end{bmatrix}$$ (4.72)

Defining new variables $X = Q_\xi^{-1}\dot{W}$ equation (4.68) can be cast in terms of diagonal matrix $\Lambda_\xi$ to

$$\frac{\partial \dot{X}_m}{\partial t^*} = \left[ \frac{2\varepsilon}{\Delta \xi} (\cos \phi^m_\xi - 1) I - \frac{I}{\Delta \xi} \sin \phi^m_\xi \Lambda_\xi \right] \dot{X}_m$$ (4.73)
This is a system of four independent scalar equations of the form

\[ \frac{\partial \hat{X}_m}{\partial t^*} = \left[ \frac{2 \varepsilon}{\Delta \xi} (\cos \phi^m_{\xi} - 1) - \frac{\lambda_i I}{\Delta \xi} \sin \phi^m_{\xi} \right] \hat{X}_m \quad (4.74) \]

where \( \lambda_i \) are each of the eigenvalues defined in (4.70). Therefore, the Fourier symbol of this discretisation is

\[ \hat{\lambda}_m \Delta t^* = \frac{2 \varepsilon \Delta t^*}{\Delta \xi} (\cos \phi^m_{\xi} - 1) - I \frac{\lambda_i \Delta t^*}{\Delta \xi} \sin \phi^m_{\xi} \quad (4.75) \]

In regions where the flow is smooth the value of \( \varepsilon = O(\Delta \xi^2) \) and the scheme is stable for CFL numbers

\[ \left| \frac{\lambda_i (\Delta t^*)_{CD}}{\Delta \xi} \right| \leq 2 \sqrt{2} \quad (4.76) \]

Notice that the above is true if \( \varepsilon \simeq 0 \) because the eigenvalue spectrum of the spacetime discretisation lies on the imaginary axis and the explicit Runge-Kutta scheme used is stable in the interval \([-2\sqrt{2}I, 2\sqrt{2}I] \in \mathbb{C} \) (see Figure 4.2). Also, bear in mind that \( \lambda_i \) are the spacetime eigenvalues defined in (4.70). In regions with large gradients \( \varepsilon \neq 0 \) and the eigenvalue spectrum lies within the region enclosed by the green dashed curve in Figure 4.2 with \( \varepsilon \simeq \alpha \).

### 4.3.2 Inviscid terms with a second-order upwind discretisation

A similar stability analysis is performed here with a second-order upwind spacetime discretisation which, effectively, adds some numerical dissipation to the central-difference scheme. Therefore, the equation to solve here is

\[ \frac{\partial \mathbf{W}}{\partial t^*} + B_\xi \frac{\partial \mathbf{W}}{\partial \xi} + B_\eta \frac{\partial \mathbf{W}}{\partial \eta} + B_\zeta \frac{\partial \mathbf{W}}{\partial \zeta} = 0 \quad (4.77) \]

It is important to realise that system (4.77) is obtained by linearisation of the fluxes. However, the fluxes defined by Van Leer in the flux-vector splitting approach used throughout this work are not differentiable [3], hence a linearisation of them is not possible. Therefore, the stability analysis presented in this section must be used carefully since it is not directly applicable to the flux-vector splitting approach.

Like in the central-difference case (Section 4.3.1) the stability analysis of equation (4.77) can be decoupled into three different directions \( \xi, \eta \) and \( \zeta \) and compute the limitation on the time-step size via (4.62) or (4.63). The
equation to be considered for each generalised coordinate (here $\xi$ is used) is

$$\frac{\partial W}{\partial t^*} + B_\xi \frac{\partial W}{\partial \xi} = 0 \quad (4.78)$$

Diagonalising matrix $B_\xi$ and defining $X = Q^{-1}_\xi \hat{W}$ yields

$$\frac{\partial X}{\partial t^*} + \Lambda_\xi \frac{\partial X}{\partial \xi} = 0 \quad (4.79)$$

where diagonal matrix $\Lambda_\xi$ is given in (4.69) and transformation matrix $Q_\xi$ and its inverse $Q^{-1}_\xi$ are defined in (4.71) and (4.72), respectively. Finally, each of the four independent equations in system (4.79) can be written as

$$\frac{\partial X}{\partial t^*} + \lambda \frac{\partial X}{\partial \xi} = 0 \quad (4.80)$$

where $\lambda$ are the eigenvalues of matrix $B_\xi$. The semi-discretisation of (4.80) yields (for $\lambda > 0$, i.e. the flow moves from left to right)

$$\frac{\partial X_i}{\partial t^*} + \lambda \left( \frac{X_{i+\frac{1}{2}} - X_{i-\frac{1}{2}}}{\Delta \xi} \right) = 0 \quad (4.81)$$

where the superscript “$-$” refers to the left side of the face. In the event of a negative eigenvalue, i.e. if the flow moves from right to left, the values at the right side of each face $X^{+}_{i+\frac{1}{2}}$ would be used instead. Notice that the subscripts $j$ and $k$ corresponding to the other two generalised coordinates, $\eta$ and $\zeta$, have been dropped for the sake of simplicity. The reconstruction of face values from cell-centre values for the second-order upwind scheme is given by equation (2.66) which can be cast, in the current one-dimensional case, to

$$X^-_{i+\frac{1}{2}} = X_i + \left( \frac{X_{i+1} - X_{i-1}}{2\Delta \xi} \right) \varphi(r_i) \frac{\Delta \xi}{2} \quad (4.82)$$

The slope limiter by Van Leer, equation (2.70), is the one used here and can be cast into

$$\varphi_{VL} (r) = \frac{r + |r|}{1 + |r|} = \begin{cases} 
\frac{2r}{1 + r} & \text{if } r > 0 \quad \text{(smooth flow)} \\
0 & \text{if } r \leq 0 \quad \text{(discontinuities)} 
\end{cases} \quad (4.83)$$
4.3. Spacetime formulation of the two-dimensional Navier-Stokes equations

For \( \lambda > 0 \) (an equivalent expression can be found for \( \lambda < 0 \)) and \( r > 0 \), \( \varphi \) can be computed at cell \( i \) as

\[
\frac{2r_i}{1 + r_i} = \frac{2\Delta X^+ + \Delta X^-}{\Delta X^+ - \Delta X^-} = \frac{2(X_{i+1} - X_i)}{(X_{i+1} - X_{i-1})} (4.84)
\]

where \( \Delta X^+ = X_{i+1} - X_i \), \( \Delta X^- = X_i - X_{i-1} \) and \( r_i = \frac{\Delta X^+}{\Delta X^-} \). Combining equations (4.82) and (4.84) yields

\[
X_{i+\frac{1}{2}}^- = X_i + \frac{1}{2} (X_{i+1} - X_i) (4.85)
\]

when \( \lambda > 0 \) and \( r > 0 \) (smooth flow). Similarly, one can write

\[
X_{i-\frac{1}{2}}^- = X_{i-1} + \frac{1}{2} (X_i - X_{i-1}) (4.86)
\]

and cast equation (4.81) into

\[
\frac{\partial X_i}{\partial t^*} + \lambda \left( \frac{X_i - X_{i-1}}{\Delta \xi} + \frac{X_{i+1} - 2X_i + X_{i-1}}{2\Delta \xi} \right) = 0 (4.87)
\]

Writing the solution in terms of Fourier series

\[
X(t, \xi) = \sum_{m=-N/2}^{N/2} \hat{X}_m(t) e^{I k_m \xi} = \sum_{m=-N/2}^{N/2} \hat{X}_m(t) e^{I \phi_m} (4.88)
\]

yields

\[
\frac{\partial \hat{X}_m}{\partial t^*} = -\frac{\lambda}{\Delta \xi} \left( 1 - e^{-I \phi_m} + \frac{e^{I \phi_m} - 2 + e^{-I \phi_m}}{2} \right) \hat{X}_m (4.89)
\]

and using trigonometric identity (4.6)

\[
\frac{\partial \hat{X}_m}{\partial t^*} = \left[ -I \frac{\lambda}{\Delta \xi} \sin \phi_m \right] \hat{X}_m (4.90)
\]
Therefore, the Fourier symbol of this spacetime discretisation is

$$\lambda_m \Delta t^* = -I \frac{\lambda \Delta t^*}{\Delta \xi} \sin \phi_m^\eta$$  \hspace{1cm} (4.91)$$

If a fourth-order four-stages explicit Runge-Kutta scheme is used to integrate the semi-discrete system in pseudo-time (see Figure 4.4) the scheme is stable for CFL numbers

$$\left| \frac{\lambda (\Delta t^*_{\text{cW}})}{\Delta \xi^2} \right| \leq 2 \sqrt{2} \hspace{1cm} (4.92)$$

This condition is only valid in regions of smooth flow. Near shocks $r \leq 0$ and $\varphi = 0$, hence the scheme reduces to

$$\frac{\partial X_i}{\partial t^*} + \frac{\lambda}{\Delta \xi^2} (X_i - X_{i-1}) = 0 \hspace{1cm} (4.93)$$

Therefore, the Fourier symbol is

$$\lambda_m \Delta t^* = \frac{\lambda \Delta t^*}{\Delta \xi} (\cos \phi_m^\eta - 1 - I \sin \phi_m^\eta)$$  \hspace{1cm} (4.94)$$

which corresponds to the circle of radius $\frac{\lambda \Delta t^*}{\Delta \xi}$ centered at $z_0 = -\frac{\lambda \Delta t^*}{\Delta \xi}$ in the complex plane. Looking at Figure 4.4 the limit of the region of stability of the explicit Runge-Kutta along the real axis is $-2.78$. Therefore, in this case, the scheme is stable for CFL numbers

$$\left| \frac{\lambda (\Delta t^*_{\text{cW}})}{\Delta \xi} \right| \leq 1.39 \hspace{1cm} (4.95)$$

### 4.3.3 Viscous terms with a second-order central-difference discretisation

Removing the convective terms of the Navier-Stokes equations in (4.42), i.e. leaving only viscous (diffusive) terms leads to

$$\frac{\partial W}{\partial t^*} = B_{\xi\xi} \frac{\partial^2 W}{\partial \xi^2} + B_{\eta\eta} \frac{\partial^2 W}{\partial \eta^2} + B_{\xi\eta} \frac{\partial^2 W}{\partial \xi \partial \eta} + 2B_{\xi\xi} \frac{\partial^2 W}{\partial \xi \partial \xi} + 2B_{\xi\eta} \frac{\partial^2 W}{\partial \xi \partial \eta} + 2B_{\eta\eta} \frac{\partial^2 W}{\partial \eta \partial \eta} \hspace{1cm} (4.96)$$
4.3. Spacetime formulation of the two-dimensional Navier-Stokes equations

\[ \left| 1 + \hat{\lambda} \Delta t + \frac{(\hat{\lambda} \Delta t)^2}{2} + \frac{(\hat{\lambda} \Delta t)^3}{6} + \frac{(\hat{\lambda} \Delta t)^4}{24} \right| \leq 1 \]

Figure 4.4: Region of stability of fourth-order four-stages explicit Runge-Kutta scheme (shaded area) and circle of radius \( R = 1.39 \) centered at \( z = -1.39 \) (green dashed line).

Because viscous terms are diffusive a central-difference scheme is used to discretise them. Second derivatives are discretised as derivatives of the first derivative. Assuming \( u \) is the function for which second derivatives are sought, the discretisation of its first derivative with respect to generalised coordinate \( \xi \) is

\[ \frac{\partial u_{i,j,k}}{\partial \xi} = \frac{u_{i+1,j,k} - u_{i-1,j,k}}{2\Delta \xi} \]  

\[ (4.97) \]

Therefore, the second derivative of \( u \) with respect to \( \xi \) twice yields

\[ \frac{\partial^2 u_{i,j,k}}{\partial \xi^2} = \frac{\partial}{\partial \xi}\left( \frac{\partial u_{i,j,k}}{\partial \xi} \right) = \frac{1}{2\Delta \xi}\left( \frac{\partial u_{i+1,j,k}}{\partial \xi} - \frac{\partial u_{i-1,j,k}}{\partial \xi} \right) = \frac{u_{i+2,j,k} - 2u_{i,j,k} + u_{i-2,j,k}}{4\Delta \xi^2} \]  

\[ (4.98) \]

and the second derivative derivative of \( u \) with respect to \( \xi \) and \( \eta \) yields

\[ \frac{\partial^2 u_{i,j,k}}{\partial \xi \partial \eta} = \frac{\partial}{\partial \eta}\left( \frac{\partial u_{i,j,k}}{\partial \xi} \right) = \frac{1}{2\Delta \eta}\left( \frac{\partial u_{i,j+1,k}}{\partial \xi} - \frac{\partial u_{i,j-1,k}}{\partial \xi} \right) = \frac{u_{i+1,j+1,k} - u_{i-1,j+1,k} - u_{i+1,j-1,k} + u_{i-1,j-1,k}}{4\Delta \xi \Delta \eta} \]  

\[ (4.99) \]
Therefore, the discretisation of equation (4.96) leads to

\[
\frac{\partial W_{i,j,k}}{\partial t^*} = B^v_{\xi \xi} \left( \frac{W_{i+2,j,k} - 2W_{i,j,k} + W_{i-2,j,k}}{4\Delta \xi^2} \right) + B^v_{\eta \eta} \left( \frac{W_{i,j+2,k} - 2W_{i,j,k} + W_{i,j-2,k}}{4\Delta \eta^2} \right) + B^v_{\zeta \zeta} \left( \frac{W_{i,j,k+2} - 2W_{i,j,k} + W_{i,j,k-2}}{4\Delta \zeta^2} \right) \\
+ 2B^v_{\xi \eta} \left( \frac{W_{i+1,j+1,k} - W_{i-1,j+1,k} - W_{i+1,j-1,k} + W_{i-1,j-1,k}}{4\Delta \xi \Delta \eta} \right) + 2B^v_{\xi \zeta} \left( \frac{W_{i+1,j,k+1} - W_{i-1,j,k+1} - W_{i+1,j,k-1} + W_{i-1,j,k-1}}{4\Delta \xi \Delta \zeta} \right) \\
+ 2B^v_{\eta \zeta} \left( \frac{W_{i,j+1,k+1} - W_{i,j+1,k-1} - W_{i,j-1,k+1} + W_{i,j-1,k-1}}{4\Delta \eta \Delta \zeta} \right) 
\] 
(4.100)

Assuming the solution can be written in terms of Fourier series, equation (4.66), and using an arbitrary mode \( m \) in (4.100) leads to

\[
\frac{\partial \hat{W}_m}{\partial t^*} = \left[ B^v_{\xi \xi} \left( \frac{e^{2i\phi} - 2 + e^{-2i\phi}}{4\Delta \xi^2} \right) + B^v_{\eta \eta} \left( \frac{e^{2i\phi} - 2 + e^{-2i\phi}}{4\Delta \eta^2} \right) + B^v_{\zeta \zeta} \left( \frac{e^{2i\phi} - 2 + e^{-2i\phi}}{4\Delta \zeta^2} \right) \\
+ 2B^v_{\xi \eta} \left( \frac{e^{i(\phi + \phi)} - e^{-i(\phi + \phi)} - e^{-i(\phi - \phi)} + e^{i(\phi - \phi)}}{4\Delta \xi \Delta \eta} \right) + 2B^v_{\xi \zeta} \left( \frac{e^{i(\phi + \phi)} - e^{-i(\phi + \phi)} - e^{-i(\phi - \phi)} + e^{i(\phi - \phi)}}{4\Delta \xi \Delta \zeta} \right) \\
+ 2B^v_{\eta \zeta} \left( \frac{e^{i(\phi + \phi)} - e^{-i(\phi + \phi)} - e^{-i(\phi - \phi)} + e^{i(\phi - \phi)}}{4\Delta \eta \Delta \zeta} \right) \right] \hat{W}_m 
\] 
(4.101)

Using trigonometric identity (4.6) and rearranging

\[
\frac{\partial \hat{W}_m}{\partial t^*} = \left[ B^v_{\xi \xi} \frac{\cos(2\phi) - 1}{2} + B^v_{\eta \eta} \frac{\cos(2\phi) - 1}{2} + B^v_{\zeta \zeta} \frac{\cos(2\phi) - 1}{2} \\
+ 2B^v_{\xi \eta} \frac{\cos(\phi + \phi) - \cos(\phi - \phi)}{2 \Delta \xi \Delta \eta} + 2B^v_{\xi \zeta} \frac{\cos(\phi + \phi) - \cos(\phi - \phi)}{2 \Delta \xi \Delta \zeta} \\
+ 2B^v_{\eta \zeta} \frac{\cos(\phi + \phi) - \cos(\phi - \phi)}{2 \Delta \eta \Delta \zeta} \right] \hat{W}_m 
\] 
(4.102)

Recall the following trigonometric identity

\[
\cos(\theta_1 + \theta_2) = \cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2 
\] 
(4.103)
Therefore

\[ \cos(2\theta) - 1 = -2 \sin^2 \theta \] (4.104)

\[ \cos(\theta_A + \theta_B) - \cos(\theta_A - \theta_B) = -2 \sin \theta_A \sin \theta_B \] (4.105)

Combining (4.102), (4.104) and (4.105) one can write

\[ \frac{\partial \hat{W}_m}{\partial t^*} = - \left[ \frac{B_v^{\xi \xi}}{\Delta \xi^2} \sin^2 \phi_\xi + \frac{B_v^{\eta \eta}}{\Delta \eta^2} \sin^2 \phi_\eta + \frac{B_v^{\zeta \zeta}}{\Delta \zeta^2} \sin^2 \phi_\zeta 
+ \frac{2B_v^{\xi \eta}}{\Delta \xi \Delta \eta} \sin \phi_\xi \sin \phi_\eta + \frac{2B_v^{\xi \zeta}}{\Delta \xi \Delta \zeta} \sin \phi_\xi \sin \phi_\zeta + \frac{2B_v^{\eta \zeta}}{\Delta \eta \Delta \zeta} \sin \phi_\eta \sin \phi_\zeta \right] \hat{W}_m \] (4.106)

From the general shape of matrix \( B_v^{\alpha \beta} \), given in equation (4.60), one can conclude that the first and fourth equations in system (4.106), which correspond to density \( \rho \) and pressure \( p \) conservation, respectively, are identically null. Therefore, writing the two-equation system (conservation of \( u \) and \( v \)) leads to

\[ \frac{\partial \hat{v}_m}{\partial t^*} = - \left[ \frac{E_v^{u \xi}}{\Delta \xi^2} \sin^2 \phi_\xi + \frac{E_v^{u \eta}}{\Delta \eta^2} \sin^2 \phi_\eta + \frac{E_v^{u \zeta}}{\Delta \zeta^2} \sin^2 \phi_\zeta 
+ \frac{2E_v^{u \xi \eta}}{\Delta \xi \Delta \eta} \sin \phi_\xi \sin \phi_\eta + \frac{2E_v^{u \xi \zeta}}{\Delta \xi \Delta \zeta} \sin \phi_\xi \sin \phi_\zeta + \frac{2E_v^{u \eta \zeta}}{\Delta \eta \Delta \zeta} \sin \phi_\eta \sin \phi_\zeta \right] \hat{v}_m \] (4.107)

where \( \hat{v}_m \) is the column vector

\[ \hat{v}_m = \begin{pmatrix} \hat{u}_m \\ \hat{v}_m \end{pmatrix} \] (4.108)

and \( E_v^{\alpha \beta} \) is the matrix defined as

\[ E_v^{\alpha \beta} = \frac{\tilde{\mu}}{\rho Re_c} \begin{bmatrix} \frac{4}{3} \alpha_x \beta_x + \alpha_y \beta_y & \frac{\alpha_x \beta_y + \alpha_y \beta_x}{6} \\ \frac{\alpha_x \beta_y + \alpha_y \beta_x}{6} & \alpha_x \beta_x + \frac{4}{3} \alpha_y \beta_y \end{bmatrix} \] (4.109)

where \( \alpha \) and \( \beta \) can be any of the generalized coordinates \( \xi, \eta, \zeta \). Defining functions \( f_x \) and \( f_y \) as follows

\[ f_x = \frac{\xi_x \sin \phi_\xi}{\Delta \xi} + \frac{\eta_x \sin \phi_\eta}{\Delta \eta} + \frac{\zeta_x \sin \phi_\zeta}{\Delta \zeta} \] (4.110)
\[ f_y = \frac{\zeta_y \sin \phi \xi}{\Delta \xi} + \frac{\eta_y \sin \phi \eta}{\Delta \eta} + \frac{\zeta_y \sin \phi \zeta}{\Delta \zeta} \] (4.111)

and matrix \( \hat{E}^v \) as

\[
\hat{E}^v = \frac{\hat{\mu}}{\rho Re_{\infty}} \begin{bmatrix}
\frac{4}{3} f_x^2 + f_y^2 & \frac{f_x f_y}{3} \\
\frac{f_x f_y}{3} & f_x^2 + \frac{4}{3} f_y^2
\end{bmatrix}
\] (4.112)

Equation (4.107) can be further simplified as

\[
\frac{\partial \hat{\mathbf{v}}_m}{\partial \hat{t}^*} = -\hat{E}^v \hat{\mathbf{v}}_m
\] (4.113)

Solving the eigenvalues and eigenvectors problem \( |\hat{E}^v - \lambda I| = 0 \), matrix \( \hat{E}^v \) can be diagonalized yielding

\[
\Lambda^v = T^{-1} \hat{E}^v T = \begin{bmatrix}
\lambda_1 & 0 \\
0 & \lambda_2
\end{bmatrix}
\] (4.114)

where the eigenvalues \( \lambda_1 \) and \( \lambda_2 \) are

\[
\lambda_1 = \frac{4}{3} \frac{\hat{\mu}}{\rho Re_{\infty}} (f_x^2 + f_y^2) \\
\lambda_2 = \frac{\hat{\mu}}{\rho Re_{\infty}} (f_x^2 + f_y^2)
\] (4.115)

and transformation matrix \( T \) and its inverse \( T^{-1} \) are

\[
T = \begin{bmatrix}
f_x & -f_y \\
\frac{f_x f_y}{f_x^2 + f_y^2} & \frac{f_x}{f_y}
\end{bmatrix} \\
1 & 1
\]

\[
T^{-1} = \begin{bmatrix}
1 & \frac{f_y}{f_x} \\
\frac{f_x}{f_y} & -1
\end{bmatrix}
\] (4.116)

Combining equations (4.113) and (4.114) and defining the column vector \( \hat{\mathbf{X}}_m = T^{-1} \hat{\mathbf{v}}_m \) yields

\[
\frac{\partial \hat{\mathbf{X}}_m}{\partial \hat{t}^*} = -\Lambda^v \hat{\mathbf{X}}_m
\] (4.117)
Since $\Lambda^v$ is diagonal, equation (4.117) can be decoupled into two independent scalar equations of the form

$$\frac{\partial \hat{X}_m}{\partial t^*} = -\lambda \hat{X}_m$$  \hspace{1cm} (4.118)

Therefore, the Fourier symbol of the spacetime discretisation of viscous terms yields $-\lambda \Delta t^*$ where $\lambda$ are each of the eigenvalues in (4.115). Since the eigenvalue spectrum lies on the negative part of the real axis, if a fourth-order four-stages explicit Runge-Kutta scheme is used to integrate the semi-discrete system in pseudo-time (see Figure 4.4) the scheme is stable for CFL numbers

$$\left| \hat{\mu} (\Delta t^*)_{CD} \rho \text{Re}_{\infty} \max_{\phi_{C,v,\xi}} \left( f_x^2 + f_y^2 \right) \right| \leq 2.78$$  \hspace{1cm} (4.119)

Following Pulliam [108, 107] the derivatives with respect to the curvilinear coordinates can be written in terms of the Cartesian coordinates using the chain rule, i.e.

$$\begin{Bmatrix} \partial_{\xi} \\ \partial_{\eta} \\ \partial_{\zeta} \end{Bmatrix} = \begin{bmatrix} t_{\xi} & x_{\xi} & y_{\xi} \\ t_{\eta} & x_{\eta} & y_{\eta} \\ t_{\zeta} & x_{\zeta} & y_{\zeta} \end{bmatrix} \begin{Bmatrix} \partial_t \\ \partial_x \\ \partial_y \end{Bmatrix}$$  \hspace{1cm} (4.120)

Similarly, the derivatives with respect to Cartesian coordinates can be written in terms of the curvilinear coordinates

$$\begin{Bmatrix} \partial_t \\ \partial_x \\ \partial_y \end{Bmatrix} = \begin{bmatrix} \xi_t & \eta_t & \zeta_t \\ \xi_x & \eta_x & \zeta_x \\ \xi_y & \eta_y & \zeta_y \end{bmatrix} \begin{Bmatrix} \partial_{\xi} \\ \partial_{\eta} \\ \partial_{\zeta} \end{Bmatrix}$$  \hspace{1cm} (4.121)

Inverting the transformation matrix in (4.120) yields

$$\begin{Bmatrix} \partial_t \\ \partial_x \\ \partial_y \end{Bmatrix} = \frac{1}{J} \begin{bmatrix} x_{\xi} y_{\eta} - x_{\eta} y_{\xi} & x_{\xi} y_{\zeta} - x_{\zeta} y_{\xi} & x_{\eta} y_{\xi} - x_{\xi} y_{\eta} \\ t_{\eta} y_{\zeta} - t_{\zeta} y_{\eta} & t_{\xi} y_{\eta} - t_{\eta} y_{\xi} & t_{\xi} y_{\zeta} - t_{\eta} y_{\xi} \\ t_{\zeta} x_{\eta} - t_{\eta} x_{\zeta} & t_{\xi} x_{\eta} - t_{\eta} x_{\xi} & t_{\eta} x_{\zeta} - t_{\xi} x_{\eta} \end{bmatrix} \begin{Bmatrix} \partial_{\xi} \\ \partial_{\eta} \\ \partial_{\zeta} \end{Bmatrix}$$  \hspace{1cm} (4.122)
where the Jacobian of the transformation is

$$J = t_\eta x_\xi y_\zeta - t_\xi x_\eta y_\zeta - t_\zeta x_\xi y_\eta + t_\xi x_\eta y_\zeta - t_\zeta x_\xi y_\eta$$  \hspace{1cm} (4.123)

Since (4.121) and (4.122) are equivalent, it is possible to write the metrics $\xi_t$, $\xi_x$, $\xi_y$, $\eta_t$, $\eta_x$, $\eta_y$, $\zeta_t$, $\zeta_x$ and $\zeta_y$ as functions of the metrics $t_\xi$, $t_\eta$, $t_\zeta$, $x_\xi$, $x_\eta$, $x_\zeta$, $y_\xi$, $y_\eta$ and $y_\zeta$

$$\xi_t = \frac{x_\zeta y_\eta - x_\eta y_\zeta}{J} \hspace{1cm} \xi_x = \frac{t_\eta y_\zeta - t_\zeta y_\eta}{J} \hspace{1cm} \xi_y = \frac{t_\xi x_\eta - t_\xi x_\zeta}{J}$$  \hspace{1cm} (4.124)

$$\eta_t = \frac{x_\zeta y_\eta - x_\eta y_\zeta}{J} \hspace{1cm} \eta_x = \frac{t_\zeta y_\xi - t_\xi y_\zeta}{J} \hspace{1cm} \eta_y = \frac{t_\eta x_\xi - t_\xi x_\eta}{J}$$  \hspace{1cm} (4.125)

$$\zeta_t = \frac{x_\eta y_\zeta - x_\zeta y_\eta}{J} \hspace{1cm} \zeta_x = \frac{t_\xi y_\eta - t_\eta y_\zeta}{J} \hspace{1cm} \zeta_y = \frac{t_\eta x_\xi - t_\xi x_\eta}{J}$$  \hspace{1cm} (4.126)

where the values of the metrics $t_\xi$, $t_\eta$, $t_\zeta$, $x_\xi$, $x_\eta$, $x_\zeta$, $y_\xi$, $y_\eta$ and $y_\zeta$ can be worked out at each cell in the mesh as follows

$$t_\xi = \frac{\Delta \xi}{\Delta \zeta} \hspace{1cm} t_\eta = \frac{\Delta \eta}{\Delta \zeta} \hspace{1cm} t_\zeta = \frac{\Delta \zeta}{\Delta \zeta}$$  \hspace{1cm} (4.127)

$$x_\xi = \frac{\Delta \xi}{\Delta \zeta} \hspace{1cm} x_\eta = \frac{\Delta \eta}{\Delta \zeta} \hspace{1cm} x_\zeta = \frac{\Delta \zeta}{\Delta \zeta}$$  \hspace{1cm} (4.128)

$$y_\xi = \frac{\Delta \xi}{\Delta \zeta} \hspace{1cm} y_\eta = \frac{\Delta \eta}{\Delta \zeta} \hspace{1cm} y_\zeta = \frac{\Delta \zeta}{\Delta \zeta}$$  \hspace{1cm} (4.129)

where $\Delta_i x_j$ represents the variation of coordinate $x_j$ along direction $i$. Substituting (4.127), (4.128) and (4.129) back into equations (4.110) and (4.111) leads to

$$f_x = \frac{(\Delta_\eta t \Delta_\zeta y - \Delta_\xi t \Delta_\eta y) \sin \phi^\zeta_n + (\Delta_\xi t \Delta_\zeta y - \Delta_\zeta t \Delta_\xi y) \sin \phi^\eta_n + (\Delta_\zeta t \Delta_\eta y - \Delta_\xi t \Delta_\eta y) \sin \phi^\zeta_n}{J \Delta_\xi \Delta_\eta \Delta_\zeta}$$  \hspace{1cm} (4.130)

$$f_y = \frac{(\Delta_\eta t \Delta_\zeta x - \Delta_\xi t \Delta_\eta x) \sin \phi^\zeta_n + (\Delta_\xi t \Delta_\zeta x - \Delta_\zeta t \Delta_\xi x) \sin \phi^\eta_n + (\Delta_\zeta t \Delta_\eta x - \Delta_\xi t \Delta_\eta x) \sin \phi^\zeta_n}{J \Delta_\xi \Delta_\eta \Delta_\zeta}$$  \hspace{1cm} (4.131)
4.3. Spacetime formulation of the two-dimensional Navier-Stokes equations

Defining $\Psi$ as follows

$$\Psi = \left( |\Delta t \Delta y - \Delta \xi \Delta y| + |\Delta t \Delta \xi y - \Delta \eta \Delta \xi y| + |\Delta \eta \Delta \xi y - \Delta \eta \Delta \xi| \right)^2$$

$$+ \left( |\Delta t \Delta x - \Delta \eta \Delta x| + |\Delta \xi \Delta x - \Delta \eta \Delta x| + |\Delta \xi \Delta x - \Delta \xi \Delta \eta x| \right)^2$$

(4.132)

and noting that $|\sin \theta| \leq 1$ it is possible to find an upper limit of $f_x^2 + f_y^2$ as

$$f_x^2 + f_y^2 \leq \frac{\Psi}{(J \Delta \xi \Delta \eta \Delta \zeta)^2}$$

(4.133)

The CFL condition (4.119) can now be cast into

$$\left\| \frac{\hat{\mu} (\Delta t^*_c)_{CD}}{\rho Re_{\infty}} \right\| \leq 2.78$$

(4.134)

The above can be written in terms of the cell volume for structured grids bearing in mind that $V_{\text{parallelepiped}} = J \Delta \xi \Delta \eta \Delta \zeta$ is the volume of a parallelepiped.

4.3.4 Estimation of the CFL condition

As outlined in the introduction of this chapter, a CFL condition can be estimated through equation (4.2) for the different numerical schemes used in this work. The value of the maximum pseudo-time step size needs to be calculated first for convective and diffusive terms separately, $\Delta t^*_c$ and $\Delta t^*_d$ respectively. Moreover, convective terms may be discretised via a second-order central-difference scheme or an upwind biased one leading to two different values for the largest convective time step. Diffusive terms are always discretised using the second-order central-difference stencil. Thereafter, all these values can be used to work out the final CFL condition taking convective and/or diffusive contributions into account as necessary. Table 4.1 describes the calculation of the maximum time-step size for each of the spacetime numerical schemes used in this work for the explicit fourth-order Runge-Kutta scheme used to march the solution in pseudo-time $t^*$. 

119
### CFL condition in spacetime

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Value of $\Delta t^<em>_c$ and $\Delta t^</em>_d$</th>
<th>Maximum $\Delta t^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euler equations with second-order central-difference. Equation (2.91) or (2.92) without viscous terms.</td>
<td>$(\Delta t^*<em>c)</em>{CD} \rightarrow$ Eq. (4.76)</td>
<td>$\Delta t^* \leq (\Delta t^*<em>c)</em>{CD}$</td>
</tr>
<tr>
<td>Euler equations with second-order upwind. Equation (2.131) or (2.132) without viscous terms.</td>
<td>$(\Delta t^<em><em>c)</em>{UW}$ (smooth) $\rightarrow$ Eq. (4.92), $(\Delta t^</em><em>c)</em>{UW}$ (gradients) $\rightarrow$ Eq. (4.95)</td>
<td>$\Delta t^* \leq (\Delta t^*<em>c)</em>{UW}$</td>
</tr>
<tr>
<td>Navier-Stokes equations with second-order central-difference convective terms and second-order central-difference diffusive terms. Equation (2.91) or (2.92).</td>
<td>$(\Delta t^<em><em>c)</em>{CD} \rightarrow$ Eq. (4.76), $(\Delta t^</em><em>d)</em>{CD} \rightarrow$ Eq. (4.134)</td>
<td>$\frac{1}{\Delta t^<em>} \geq \frac{1}{(\Delta t^</em><em>c)</em>{CD}} + \frac{1}{(\Delta t^*<em>d)</em>{CD}}$</td>
</tr>
<tr>
<td>Navier-Stokes equations with second-order upwind convective terms and second-order central-difference diffusive terms. Equation (2.131) or (2.132).</td>
<td>$(\Delta t^<em><em>c)</em>{UW}$ (smooth) $\rightarrow$ Eq. (4.92), $(\Delta t^</em><em>c)</em>{UW}$ (gradients) $\rightarrow$ Eq. (4.95), $(\Delta t^*<em>d)</em>{CD} \rightarrow$ Eq. (4.134)</td>
<td>$\frac{1}{\Delta t^<em>} \geq \frac{1}{(\Delta t^</em><em>c)</em>{UW}} + \frac{1}{(\Delta t^*<em>d)</em>{CD}}$</td>
</tr>
</tbody>
</table>

Table 4.1: Calculation of the CFL condition in spacetime based on maximum convective and diffusive pseudo-time step sizes, $\Delta t^*_c$ and $\Delta t^*_d$, for different numerical schemes.
Chapter 5

Results

5.1 Introduction

The main objective of this work is to demonstrate the ability to accurately solve unsteady aerodynamics problems with complex motions through a spacetime framework. As explained in previous chapters, the integration of the Navier-Stokes equations for fluids motion in space (of dimension $N$) and time is coupled and solved at once via the use of $N+1$ dimensional finite-volume or finite-element methods. Since this is a novel technique and little previous work has been done in this direction, a two-dimensional spacetime framework (2D+t) based on a finite-volume formulation is presented here. Bear in mind that finite-element formulations such as discontinuous-Galerkin are also possible as demonstrated by recent publications [72, 71]. A series of problems involving large deformations or topological changes in the geometry that conventional approaches find hard to solve are simulated here, namely a slat and flap deployment, a spoiler deflection, a rotor-stator configuration, aerofoils flying in opposite directions past each other and a full landing case with a combination of slat, flap and spoiler deployments along with ground effect.

Unlike traditional approaches, where a directionally biased finite-difference is used for the integration of time derivatives, the spacetime framework permits the use of bi-directional stencils such as central-difference schemes, not only in space but also in time. The implications of this are far-reaching as it introduces a violation of causality due to pressure waves effectively travelling backwards in time. This represents a great concern of the present work, hence alternative upwind stencils are introduced, at least in the time direction, tested and compared against the results of the central-difference counterpart and experimental data. Some of these test cases are: a one-dimensional piston problem, Sod’s shock-tube problem, AGARD R-702(3E3) cases 1, 3 and 5 for pitching NACA 0012 aerofoil and a two-dimensional isentropic Euler vortex.
Chapter 5: Results

The structure of this chapter is as follows. In the rest of this section simple one-dimensional problems are presented aiming to introduce the issues that arise from the use of a central-difference formulation in spacetime. Subsequent sections focus on the correlation of the results obtained through the spacetime method with empirical data, the comparison of the different stencils implemented and the showcase of its ability to produce reliable results for different types of unsteady aerodynamics problems with complex motions. Some of them include both inviscid and viscous solutions. These are the cases where, at the time of simulation, a functioning viscous solver was available.

5.1.1 Periodic semi-infinite piston

As a starting point in the validation of the spacetime formulation a periodic semi-infinite piston is simulated. This is a simple one-dimensional test case and an analytical solution exists, given by equation (5.3). The spacetime mesh used is depicted in Figure 5.1 where a coupled discretisation of space and time is performed through an unstructured two-dimensional mesh. Notice that the simplicity of this problem allows the use of a structured mesh also. A periodicity condition is applied in time, i.e. left and right vertical boundaries are connected. The top boundary is modelled as a moving solid wall and for the bottom one, non-reflecting boundary conditions are used. The motion is sinusoidal, defined by

\[ x(t) - x_0 = \frac{\Delta L}{2} \cos \left( \frac{2\pi}{T} t \right) \]  

and the reduced frequency

\[ k = \frac{\pi \Delta L}{T a_\infty} \]  

is equal to 0.016. In equations (5.1)-(5.2), \( \Delta L \) and \( T \) are the amplitude and the period of the piston’s motion,
respectively, and \( a_\infty \) is the speed of sound at initial conditions. Notice that the top boundary in Figure 5.2 is a sinusoid although due to the very small amplitude used this is almost imperceptible. This setup allows the results to be compared with piston theory at the moving wall \([112]\) given by equation

\[
\frac{p}{p_\infty} = \left(1 + \frac{\gamma - 1}{2} \frac{u_w}{a_\infty}\right)^{\frac{2}{\gamma - 1}}
\]  

(5.3)

Pressure contours for both the central-difference and upwind schemes (flux-splitting vector by Van Leer has been chosen in this case) are depicted in Figure 5.2. Also, the pressure at the moving wall is compared to theoretical results over one whole oscillation in Figure 5.3. These non-dimensional results correspond to a motion of amplitude \( \Delta L = 10.41 \) cm at 1000 rpm with sea-level ISA atmosphere conditions, i.e. \( \rho_\infty = 1.225 \) kg \( \cdot \) m\(^{-3}\) and \( p_\infty = 101325 \) Pa. The value of the heat capacity ratio used is \( \gamma = 1.403 \) and the maximum piston velocity at each cycle is \( V_{\text{max}} = 5.45 \) m \( \cdot \) s\(^{-1}\).

Figure 5.2: Pressure contours for one-dimensional periodic semi-infinite piston: central-difference (left) and upwind (right).

Figure 5.3: Comparison of central-difference and upwind results to piston theory
Results for both central-difference and upwind are in good agreement with piston theory. No noticeable differences appear between the central-difference and upwind (Van Leer) stencils which can be explained by the periodicity of the problem. Although intuitively information can only travel forwards in time, in periodic problems information may seem to go also backwards in time thus justifying the use of a central-difference time stencil. Bearing in mind that this is just an illusion, the explanation relies on the fact that, at each cycle, the problem is influenced by any previous temporal state, hence later stages of the previous cycle (ahead in physical time domain) determine the solution at the earliest stages of the current cycle (behind in physical time domain).

5.1.2 Motion of piston with a sharp change of direction

The suspicion that a non-periodic problem yields noticeable differences between a central-difference and an upwind stencils is confirmed by the following test case: a one-dimensional moving piston with a sharp change of direction, as given by the spacetime mesh in Figure 5.4. A structured mesh has been used to align mesh lines with \( t = \text{constant} \) lines and to allow a simpler assessment of whether information travels backwards in time. Initially, up to \( t = 0.4 \), the piston travels downwards at a constant speed, compressing the gas inside the chamber. In contrast with the periodic problem defined in Section 5.1.1, the bottom boundary has been defined as a solid wall, leading to some wave reflections as observed by the pressure contour plot in Figure 5.4. At time \( t = 0.4 \) the piston suddenly inverts its velocity and from this point onwards it moves upwards at a constant speed, expanding the gas inside the chamber. Unlike the periodic problem in Section 5.1.1 here the solution at any time level should only be influenced by the solution at previous time levels. In other words, the solution at any spacetime location \((t_0, x_0)\) such that \( t_0 < 0.4 \) should remain unaware of the rapid change in the direction of the piston motion.

![Figure 5.4](image)

Figure 5.4: Spacetime mesh (left) used on moving one-dimensional finite piston with a sharp change of direction, and pressure contours plot (right).

Therefore, the aim of this configuration is, firstly, to prove that a central-difference discretisation of the spacetime
domain yields non-physical solutions and, secondly, analyse whether an upwind stencil avoids this. Figure 5.5 depicts the pressure distribution throughout time along the $x = \text{constant}$ line represented by the dashed line in Figure 5.4. The upwind formulation improves considerably the prediction of sudden and fast movements when compared to a central-difference formulation. The typical oscillatory behaviour around shocks of JST solvers is observed here in the time direction when a sudden change in the movement of a boundary occurs. The upwind formulation damps out these oscillations successfully yielding a much smoother and accurate solution. The difference in the quality of the solutions is again explained by the fact that pressure waves always travel forwards in time.

![Comparison of pressure distribution throughout time at a location $x_0$ between central-difference and upwind for piston with sharp change of direction.](image)

Figure 5.5: Comparison of pressure distribution throughout time at a location $x_0$ between central-difference and upwind for piston with sharp change of direction.
5.2 Shock tube problem

This is the first problem computed with the spacetime solver in a series of correlations against experimental or analytical data. Sod’s shock tube problem [113] is a common test case to assess the validity and accuracy of CFD solvers, in particular those which solve a Riemann problem at every cell interface. Their ability to capture shock waves, rarefaction waves and contact discontinuities is investigated through this simple one dimensional problem. A closed tube of unit length between $x = 0$ and $x = 1$ is divided in two parts by a membrane initially located at $x = 0.5$. At each side of the tube there is gas at different conditions, as shown in Figure 5.6. At time $t = 0$ the membrane breaks leading to a shock-wave moving to the right and a rarefaction wave moving to the left. In between both waves a contact discontinuity also appears. In order to solve this problem in spacetime the one-dimensional tube is discretised with a uniform grid with 456 cells along the $x$-direction, one single cell in the $y$-direction and 400 cells of size $\Delta t = 0.0025$ in the $t$-direction up to time $t = 0.2$.

![Figure 5.6: Sod’s shock tube problem layout with initial conditions (t = 0) at both sides of the membrane, left L and right R.](image)

The solution at $t = 0$ is compared with the known analytical solution as depicted in Figure 5.7. As expected, the solution with the central-difference scheme yields oscillations around shocks whereas the upwind schemes tested (Van Leer and Roe) capture them much more accurately. No noticeable differences can be observed in regard to time-accuracy. All stencils seem to resolve the wave pattern of the solution correctly.
5.2. Shock tube problem

Figure 5.7: Sod’s shock tube problem: comparison of density (top), pressure (center) and Mach (bottom) distributions at time $t = 0.2$ between theory and CFD results via three different stencils (JST, Van Leer and Roe).
5.3 Two-dimensional isentropic Euler vortex

Taking advantage of one of the few available analytical solutions of the compressible Euler equations for fluid motion, a test case to assess the ability to capture and preserve vortical flow structures is computed next. A vortex submerged in a freestream flow is centered at \((x_0, y_0) = (5, 4)\) at time \(t = 0\) in a rectangular fluid domain which spans \(0 \leq x \leq 20\) and \(0 \leq y \leq 8\) (see Figure 5.8). The freestream conditions are \(\rho_\infty = 1\), \(u_\infty = 1\), \(v_\infty = 0\), \(p_\infty = 1\) and the problem is simulated until time \(t = 10\). Hence, the vortex is in theory convected up to location \((x_F, y_F) = (15, 4)\). The vortex is represented by an isentropic \((\delta S = 0)\) perturbation of the freestream conditions following Yee et al. [114]

\[
(\delta u, \delta v) = \frac{\beta_v}{2\pi} e^{\frac{1-x^2-y^2}{x}} (-y, x) \quad (5.4)
\]

\[
\delta T = -\frac{(\gamma - 1) \beta_v^2}{8\gamma\pi^2} e^{1-x^2-y^2} \quad (5.5)
\]

where \(\beta_v = 5\) is the strength of the vortex. Once \(u = u_\infty + \delta u\), \(v = v_\infty + \delta v\) and \(T = T_\infty + \delta T\) are calculated the density and pressure can be worked out from the isentropic relations

\[
\rho = T^{\frac{1}{\gamma-1}} \quad p = \rho^\gamma \quad (5.6)
\]

The spacetime domain has been discretised with a structured mesh of 160 by 64 cells along the \(x\) and \(y\) directions respectively, and 250 cells in the \(t\) direction. Therefore, each spacetime cell is of size \(\Delta x = \Delta y = 0.125\) and \(\Delta t = 0.04\). Pressure contour plots at \(t = 10\) have been depicted in Figure 5.8 for the JST, Van Leer and Roe stencils. All of them seem to resolve the vortex correctly although the central-difference scheme proves to be less dissipative with Van Leer flux-vector splitting method being the most dissipative of all three.

Looking at the comparison between the theoretical solution and each of the three solutions at \(t = 10\) separately, Figure 5.9 it can be observed that, although the central-difference scheme distorts the vortex less, it moves downwards slightly more than in the upwind biased schemes. Overall, the JST solution is the best out of the three simulated here and, although upwind formulations use a more realistic time stencil, this may imply that the upwind schemes have not really reached a high enough temporal accuracy.
Figure 5.8: Pressure contours plot for two-dimensional isentropic Euler (inviscid) vortex transport problem: initial solution (top), JST (second from top), Van Leer (second from bottom) and Roe (bottom).
Figure 5.9: Pressure contours plot comparison between theoretical solution and CFD results for two-dimensional isentropic Euler (inviscid) vortex transport problem at $t = 10$: JST (top), Van Leer (center) and Roe (bottom).
5.4 Oscillatory and transient pitching NACA 0012

In this section, several test cases for a pitching NACA 0012 aerofoil at different conditions are investigated. It can be concluded that the JST formulation is almost as good as the upwind counterparts for the solution of time-accurate periodic problems like the ones considered here. The spacetime mesh used is constructed from a two-dimensional structured mesh by stacking up grid planes in the time direction, as shown above in Figure 5.10. An O-grid of size $201 \times 60$ has been used to generate the spacetime meshes with 150 physical time-steps for the inviscid problems. Similarly, a C-grid of identical size, $201 \times 60$, has been used in the case of viscous problems with 100 physical time-steps in this case. In other words, the physical time-step size is $\Delta t = \frac{T}{150}$ in the former case and $\Delta t = \frac{T}{100}$ in the latter, where $T = \frac{\pi c}{U_\infty}$ is the period. In order to ensure a proper resolution of the boundary layer the first grid line normal to the wall is at a distance $\sim 10^{-5}$, where the chord of the aerofoil is $c = 1$. This ensures $y^+ \sim \mathcal{O}(1)$ in all the simulations in this section.

![Figure 5.10: Example of spacetime geometry for pitching NACA 0012. Meshes are constructed by stacking up two-dimensional grid planes in the t direction. An O-grid is used in the inviscid case (left) and a C-grid is used in the viscous case (right).](image)

The aerofoil follows a pitching motion about its quarter chord described by

$$\alpha = \alpha_0 + \Delta \alpha \sin \omega t$$  \hspace{1cm} (5.7)

where $\alpha_0$ and $\Delta \alpha$ are the initial angle of attack and the amplitude of the oscillation, respectively. The value of $\Delta \alpha$, which effectively describes the motion along with the value of the circular frequency $\omega$, is implicit in the definition of the spacetime geometry whereas the value of $\alpha_0$ is just the mean angle of incidence which can be (and has been) given as a parameter to the solver directly. The reduced frequency of the motion is the
non-dimensional parameter defined as

\[ k = \frac{\omega c}{2U_\infty} \]  

(5.8)

Radial basis functions (RBFs) are used to deform the two-dimensional mesh at each \( t = \) constant plane after a geometry transformation. All problems considered in this section are periodic, hence the first and last planes are connected to achieve the periodic boundary condition. Since the spacetime framework is conservative, both in space and time, periodic problems like the following are particularly well suited because the solution can be said to have converged to the final solution once the \( L^2 \)-norms of the residuals have dropped beyond a certain threshold (notice that the residuals in spacetime represent the change in the solution throughout the whole period), provided that the numerical scheme is stable and convergent.

5.4.1 AGARD R-702(3E3) Case 1

The first correlation is based on test case 1 of the AGARD R-702(3E3) [115]. The aerofoil is submerged in a freestream flow at Mach \( M_\infty = 0.6 \) and \( Re_\infty = 4.8 \times 10^6 \). The motion is defined by a mean angle of incidence \( \alpha_0 = 2.89 \) deg, an amplitude \( \Delta \alpha = 2.41 \) deg and a reduced frequency \( k = 0.0808 \). Both Euler and RANS solutions are computed. A central-difference (JST) scheme and an upwind method based on Van Leer fluxes are used in both cases. Moreover, a Roe-based solver is also used in the inviscid case.

A selection of \( C_p \) distributions at four different phase angles \( \omega t \) is depicted in Figure 5.11 and compared with experimental data extracted from the aforementioned AGARD report. The full set of solutions at all phase angles where empirical data is available can be found in Appendix G. The CFD results via the spacetime method correlate remarkably well in the inviscid cases. Surprisingly, the JST solution seems to be better than that of the upwind counterparts but this is probably explained by a slight phase lead over the upwind stencils. This is particularly noticeable for \( \omega t = 114.10 \) deg where the JST solution over-predicts the \( C_p \) in comparison with the empirical data. Moreover, the fact that upwind schemes are more dissipative can contribute to explain these results. Viscous spacetime solutions seem to under-predict the pressure coefficient, especially at high angles of attack. Perhaps this could be explained by the fact that a large physical time-step has been used, precisely one and a half times larger than the inviscid counterpart, hence a lower temporal-accuracy is obtained. In Figure 5.12 plots of the locus of the pitching moment coefficient, \( C_m \), and normal forces, \( C_N \), are also compared with experimental data. Again, inviscid solutions yield a better prediction of normal forces with no noticeable difference between them. However, if the moment coefficient \( C_m \) is sought, viscous solutions offer a better estimate in this case, the JST solution being the better fit once again. Although somewhat unexpected these results can probably be explained by the observed phase lead of the central-difference scheme due to information propagating backwards in time, as mentioned earlier. Also, as implied by the results of the inviscid isentropic vortex transport problem the upwind schemes may not be achieving as high a temporal accuracy as the JST.
5.4. Oscillatory and transient pitching NACA 0012

\[ \omega t = 53.61 \text{ deg} \]

\[ \omega t = 114.10 \text{ deg} \]

\[ \omega t = 195.65 \text{ deg} \]

\[ \omega t = 262.61 \text{ deg} \]

Figure 5.11: \( C_p \) distribution plots for pitching NACA 0012 with amplitude \( \Delta \alpha = 2.41 \text{ deg} \) at \( M_\infty = 0.6 \), \( \alpha_0 = 2.89 \text{ deg} \) and \( k = \frac{\omega}{2U_\infty} = 0.0808 \). Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 1.
Chapter 5: Results

5.4.2 AGARD R-702(3E3) Case 3

The second correlation is based on test case 3 of the AGARD R-702(3E3) [115]. The aerofoil is submerged in a freestream flow at Mach $M_\infty = 0.6$ and $Re_\infty = 4.8 \times 10^6$. The motion is defined by a mean angle of incidence $\alpha_0 = 2.44$ deg, an amplitude $\Delta \alpha = 4.89$ deg and a reduced frequency $k = \frac{\omega c}{U_\infty} = 0.0810$. Both Euler and RANS solutions are computed. A central-difference (JST) scheme and an upwind method based on Van Leer fluxes are used in both cases. Moreover, a Roe-based solver is also used in the inviscid case.

$C_p$ distributions at different phase angles $\omega t$ are depicted and compared with experimental data extracted from case 3 of the AGARD report R-702(3E3) in Figures 5.13 and 5.14. Like in case 1, CFD results via the spacetime method correlate well, especially in the inviscid cases. However, in this case the upwind solutions match the empirical data better most times with the exception of two phase angles, namely $\omega t = 59.85$ deg and $\omega t = 264.81$ deg. Everywhere else the JST inviscid solver seems to over-predict pressure slightly, particularly noticeable at the leading edge. Bear in mind that, out of the three AGARD test cases presented here, this is the one with the biggest amplitude of oscillation ($\Delta \alpha$) and, given that all three have similar reduced frequencies, it is therefore the one with the biggest angular velocity $\dot{\alpha}$. In other words, the angle of attack changes faster in this test case, hence the use of a non-realistic time stencil by the JST formulation is more noticeable than in the other two cases. Viscous spacetime solutions seem to under-predict the pressure coefficient in this case too, especially at high angles of attack where, perhaps, the turbulent boundary layer of the Spalart-Allmaras model delays, or even avoids, separation. Also, a larger physical time-step than in the Euler solutions has been used, hence a lower temporal-accuracy is obtained. It is interesting to notice the oscillatory solution of

![Figure 5.12: $C_N$ (left) and $C_m$ (right) coefficients for pitching NACA 0012 with amplitude $\Delta \alpha = 2.41$ deg at $M_\infty = 0.6$, $\alpha_0 = 2.89$ deg and $k = \frac{\omega c}{U_\infty} = 0.0808$. Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 1.](image)
the viscous JST solution at $\omega t = 135.51\,\text{deg}$, typical of central-difference solvers around shock waves. In this case, however, this is a transient effect coming from the integration in the time direction. This behaviour is similar to that observed in the non-periodic simple flap problem, presented and explained in Section 5.4.5 or in the highly unsteady pitching motion of Section 5.4.5. In Figure 5.15 plots of the locus of the pitching moment coefficient, $C_m$, and normal forces, $C_N$, are also compared with experimental data. Viscous solutions yield a better prediction of normal forces in this case. Moreover, the moment coefficient $C_m$ is predicted well by the RANS-SA JST solution whereas the viscous Van Leer and all three inviscid solutions are far from the empirical data.

$\omega t = 26.53\,\text{deg}$

$\omega t = 59.85\,\text{deg}$

$\omega t = 135.51\,\text{deg}$

$\omega t = 174.12\,\text{deg}$

Figure 5.13: $C_p$ distribution plots for pitching NACA 0012 with amplitude $\Delta \alpha = 2.44\,\text{deg}$ at $M_\infty = 0.6$, $\alpha_0 = 4.86\,\text{deg}$ and $k = \frac{\omega c}{U_\infty} = 0.0810$. Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 3. Continues in Figure 5.14.
Chapter 5: Results

ωt = 214.16 deg

ωt = 264.81 deg

ωt = 296.16 deg

ωt = 346.25 deg

Figure 5.14: Continuation from Figure 5.13. $C_p$ distribution plots for pitching NACA 0012 with amplitude $\Delta \alpha = 2.44$ deg at $M_\infty = 0.6$, $\alpha_0 = 4.86$ deg and $k = \frac{\omega}{2\omega_0} = 0.0810$. Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 3.
5.4. Oscillatory and transient pitching NACA 0012

Figure 5.15: $C_N$ (left) and $C_m$ (right) coefficients for pitching NACA 0012 with amplitude $\Delta \alpha = 2.44$ deg at $M_\infty = 0.6$, $\alpha_0 = 4.86$ deg and $k = \frac{\omega c}{U_\infty} = 0.0810$. Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 3.

5.4.3 AGARD R-702(3E3) Case 5

This correlation is based on test case 5 of the AGARD R-702(3E3) [115]. Now the aerofoil is submerged in a freestream flow at Mach $M_\infty = 0.755$ and $Re_\infty = 5.5 \times 10^6$. The motion is defined by a mean angle of incidence $\alpha_0 = 0.016$ deg, an amplitude $\Delta \alpha = 2.51$ deg and a reduced frequency $k = 0.0814$. Similarly, both Euler and RANS solutions are computed with JST and Van Leer solvers. Moreover, a Roe-based solver is also used in the inviscid case.

Similar results have been obtained in this case as depicted in Figures 5.16 and 5.17 for $C_p$ distributions and Figure 5.18 for $C_N$ and $C_m$ coefficients. Again, the CFD results are compared with experimental data extracted from the aforementioned AGARD report. In general, Euler solutions correlate better than RANS. Like before, this could be explained by the use of a larger physical time-step, hence obtaining a less time-accurate solution. In this case, although smaller, a slight phase lead of the JST solution compared to the upwind solvers (both Van Leer and Roe) can also be observed.
Figure 5.16: $C_p$ distribution plots for pitching NACA 0012 with amplitude $\Delta \alpha = 2.51$ deg at $M_\infty = 0.755$, $\alpha_0 = 0.016$ deg and $k = \frac{\omega t}{U_\infty} = 0.0814$. Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 5. Continues in Figure 5.17.
5.4. Oscillatory and transient pitching NACA 0012

\[ \omega t = 306.6 \text{ deg} \]
\[ \omega t = 347.2 \text{ deg} \]

Figure 5.17: Continuation from Figure 5.16. \( C_p \) distribution plots for pitching NACA 0012 with amplitude \( \Delta \alpha = 2.51 \) deg at \( M_\infty = 0.755 \), \( \alpha_0 = 0.016 \) deg and \( k = \frac{\omega^2}{2U_\infty} = 0.0814 \). Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 5.

Figure 5.18: \( C_N \) (left) and \( C_m \) (right) coefficients for pitching NACA 0012 with amplitude \( \Delta \alpha = 2.51 \) deg at \( M_\infty = 0.755 \), \( \alpha_0 = 0.016 \) deg and \( k = \frac{\omega^2}{2U_\infty} = 0.0814 \). Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 5.
5.4.4 A hybrid stencil (CSUT)

As explained in Section 2.4.4 it was hypothesised that taking advantage of a central-difference approach in space whilst still upwinding in time would allow the strength and robustness of the JST scheme to be retained and, at the same time, achieve more time accurate solutions. A first attempt to overcome the issues of the spacetime central-difference stencil via a hybrid scheme, referred to as CSUT (central-difference in space, upwind in time), is a transonic pitching NACA 0012. Only inviscid solutions are computed in this case and three different stencils are compared: JST, Van Leer and the aforementioned CSUT. The aerofoil is submerged in a freestream flow at Mach $M_\infty = 0.85$ and its motion is defined by a mean angle of incidence $\alpha_0 = 0.0$ deg, an amplitude $\Delta \alpha = 2.51$ deg and a reduced frequency $k = 0.0814$.

Pressure coefficient distributions for $\omega t = 30, 120, 210, 300$ deg have been depicted in Figure 5.19. The CSUT and JST formulations yield similar $C_p$ distributions for most phase angles, likely down to the periodicity of the problem (note that the only difference between CSUT and JST should come from the time stencil). However, the results with the upwind formulation, Van Leer in this case, differ significantly. It can be observed that the CSUT solution, although similar to JST, always lies in between that for JST and Van Leer. Nothing can be concluded yet in regard to the hybrid CSUT formulation since the periodicity of the problem in consideration attenuates any potential differences with respect to the JST formulation.
5.4. Oscillatory and transient pitching NACA 0012

\[ \omega t = 30.0 \text{ deg} \quad \omega t = 120.0 \text{ deg} \]
\[ \omega t = 210.0 \text{ deg} \quad \omega t = 300.0 \text{ deg} \]

Figure 5.19: \( C_p \) distributions for pitching NACA 0012 at \( M_\infty = 0.85 \) for \( \omega t = 30, 120, 210, 300 \text{ deg} \)
5.4.5 Negative Spalart-Allmaras model

As explained in Section C.2 of the Appendix C, although the Spalart-Allmaras turbulence model admits only positive solutions, it may become numerically unstable under certain circumstances. Numerical issues have been experienced in the spacetime version of this model when either large physical time-steps are used or the resolution of the grid in the space directions is not fine enough for the problem in consideration. In particular, this has been the case for the JST formulation in pitching aerofoils at high angles of attack or if the problem is highly unsteady. It is likely that these issues can be avoided by the use of a finer mesh in all three directions in spacetime, i.e. both $x$ and $y$ in space and time $t$. In most cases the use of the negative version of the Spalart-Allmaras model is sufficient. However, bear in mind that if a good spatial and temporal resolution is to be achieved, the discretisation of the spacetime domain needs to be fine enough.

High angle of attack

A periodic pitching NACA 0012 aerofoil in a freestream flow at $M_\infty = 0.5$, $Re_\infty = 10^6$ and $\alpha_0 = 9$ deg has been simulated with a central-difference stencil (JST) to demonstrate the use of the negative Spalart-Allmaras turbulence model. The amplitude of oscillation is $\Delta\alpha = 2.51$ deg and the reduced frequency is $k = 0.1229$. Velocity contours and $C_p$ distributions have been plotted in Figure 5.20 at several phase angles, namely $\omega t = 0, 60, 150, 180, 270$ deg. A similar mesh as in the AGARD test cases with 50 cells in the time direction has been used here. If the standard Spalart-Allmaras model is used, numerical issues arise due to big differences in the value of $\bar{\nu}$ at the interface between the boundary layer and the freestream flow. Decreasing the CFL number does not seem to improve stability. However, the use of the negative version of the Spalart-Allmaras solves the pitching motion at $\alpha = 9$ deg seamlessly. The flow on the upper side of the aerofoil separates between phase angles $\omega t \approx 40$ deg and $\omega t \approx 170$ deg approximately. The history of $L^2$-norms of density $\rho$ residuals is given in Figure 5.21. It is interesting to realise that residuals in spacetime account for changes across the spatial domain at all time levels on every iteration. In the event of a transient that leads to a steady state solution, residuals may decrease indefinitely until machine accuracy is reached. However, in the problem in consideration residuals never drop beyond $10^{-3}$ since the solution is intrinsically unstable.
5.4. Oscillatory and transient pitching NACA 0012

Figure 5.20: Periodic pitching NACA 0012 aerofoil ($\Delta \alpha = 2.51$ deg, $k = 0.1229$) in a freestream flow at $M_\infty = 0.5$, $Re_\infty = 10^6$ and $\alpha_0 = 9$ deg at phase angles $\omega t = 0, 60, 150, 180, 270$ deg: velocity contours (left) and $C_p$ distributions (right).
Figure 5.21: $L^2$-norms of density $\rho$ residuals for periodic pitching NACA 0012 aerofoil ($\Delta \alpha = 2.51$ deg, $k = 0.1229$) in a freestream flow at $M_{\infty} = 0.5$, $Re_{\infty} = 10^6$ and $\alpha_0 = 9$ deg.

Highly unsteady motion

A highly unsteady non-periodic pitching motion is simulated here with JST and Van Leer stencils. Initially a NACA 0012 aerofoil is submerged in a freestream flow at $M_{\infty} = 0.5$, $Re_{\infty} = 5.5 \times 10^6$ and $\alpha_0 = 5$ deg. At some point, the aerofoil moves in a very fast pitching motion at a reduced frequency $k = 0.25$ with an amplitude of $\Delta \alpha = 3$ deg and the transient is simulated. The negative version of the Spalart-Allmaras turbulence model in Section C.2 is used here to avoid numerical issues due to an under-resolved transient. A comparison between the JST and Van Leer formulations is given in Figure 5.22. Moreover, vorticity contours are depicted in Figure 5.23 at several time levels during the transient. As it has been the norm in the AGARD cases simulated in Sections 5.4.1, 5.4.2 and 5.4.3, the JST seems to over-predict the $C_p$, especially on the leading edge at high angles of attack. Notice the oscillations presented by the JST solution at phase angle $\omega t = 90$ deg while the upwind presents a much smoother and realistic solution. This is likely down to the non-physical time stencil used by the central-difference formulation.
5.4. Oscillatory and transient pitching NACA 0012

ωt = 30 deg

ωt = 60 deg

ωt = 90 deg

ωt = 120 deg

ωt = 180 deg

ωt = 270 deg

Figure 5.22: $C_p$ distributions for highly unsteady non-periodic pitching motion of a NACA 0012 aerofoil with amplitude $\Delta \alpha = 3$ deg at $M_\infty = 0.6$, $\alpha_0 = 5$ deg and $k = \frac{\omega_c}{2U_\infty} = 0.25$. Comparison of JST and Van Leer solutions.
ωt = 0 deg

ωt = 90 deg

ωt = 270

Figure 5.23: Vorticity contours at several time levels during the highly unsteady non-periodic pitching motion of a NACA 0012 aerofoil with amplitude Δα = 3 deg at $M_{\infty} = 0.6$, $\alpha_0 = 5$ deg and $k = \omega t / \omega_c = 0.25$. 
5.5 Simple flap

Since no clear conclusions regarding the suitability of the JST and CSUT spacetime formulations can be drawn from the analysis of periodic problems a simple unsteady non-periodic problem is computed here. Using radial basis functions to deform the two-dimensional NACA 0012 mesh used for Euler simulations in Section 5.4 and stacking up planes in the time direction a spacetime mesh is created to simulate the deflection of a simple flap, as depicted in Figure 5.24. Initially the aerofoil is flying with an angle of attack \( \alpha = 0 \) deg at Mach \( M_\infty = 0.7 \). After some time the flap deflects an angle \( \delta_F = 13.5 \) deg in a non-dimensional time \( \hat{t} = \frac{U_\infty}{c} \approx 0.295 \). The unsteady solution of the transient process is simulated with all four inviscid formulations, namely JST, Van Leer, Roe and CSUT.

![Spacetime mesh for simple flap deflection on a NACA 0012 aerofoil flying at \( M_\infty = 0.7 \).](image)

Pressure contour plots before, during and after (once the solution reaches the steady state) the flap deflection are depicted in Figures 5.25, 5.26 and 5.27, respectively, for all the different stencils considered here. Similarly, \( C_p \) distributions along the chord at various time levels are depicted in Figure 5.28 and the \( L^2 \)-norms of density residuals are given in Figure 5.29.
Figure 5.25: Pressure contour plots before the simple flap deflection ($\delta_F = 13.5$ deg, $\hat{t} = \frac{U_{\infty} t}{c} \approx 0.295$) on NACA 0012 flying at $M_\infty = 0.7$: JST (top left), CSUT (top right), Van Leer (bottom left) and Roe (bottom right).

Figure 5.26: Pressure contour plots during the simple flap deflection ($\delta_F = 13.5$ deg, $\hat{t} = \frac{U_{\infty} t}{c} \approx 0.295$) on NACA 0012 flying at $M_\infty = 0.7$: JST (top left), CSUT (top right), Van Leer (bottom left) and Roe (bottom right).
Although all formulations resolve the flap deflection successfully the JST solution presents a very erratic pattern, similar to the solution of the JST around shocks, particularly noticeable during the flap motion. Moreover, in the JST the flow anticipates to the flap motion as can be observed in Figure 5.25. This can be easily explained by the use of a non-physical stencil in time, hence allowing pressure waves to effectively travel backwards in time. On the other hand, the hybrid CSUT formulation exhibits a much smoother solution, even in the transient part, resembling the solution by upwind schemes closer. Since the only difference between the JST and the hybrid CSUT is the use of a more realistic time stencil it can be concluded that a central-difference scheme in spacetime is not appropriate when a non-periodic time-accurate solution is sought. It is interesting to realise that after the flap deflection, when the steady state solution has been reached, all four formulations yield similar results with slight differences probably down to the upwind schemes using a different spatial discretisation. Moreover, both the upwind and CSUT simulations converge much faster than the JST also explained by the use of a more realistic time stencil, as demonstrated by convergence residuals in Figure 5.29 and observed in several other simulations when comparing CSUT versus JST. This hybrid formulation represents a significant improvement with respect to JST and demonstrates the applicability of the method to transient problems. However, the CSUT scheme is not perfect and, although it overcomes the main issue of the central-difference formulation in spacetime, it is computationally almost as expensive as any of the upwind formulations considered here due to the need for second-order correction terms to extrapolate cell-centre values to face values if a second order accuracy is to be obtained in time.
Chapter 5: Results

Before the flap deflection

During the flap deflection

After the flap deflection

Figure 5.28: $C_p$ distributions before, during and after the flap deflection ($\delta_F = 13.5 \text{ deg, } \hat{t} = \frac{U_{\infty}}{t_c} \approx 0.295$) corresponding to those in Figures 5.25, 5.26 and 5.27, respectively, for NACA 0012 flying at $M_{\infty} = 0.7$.

Figure 5.29: $L^2$-norms of density residuals for simple flap deflection ($\delta_F = 13.5 \text{ deg, } \hat{t} = \frac{U_{\infty}}{t_c} \approx 0.295$) on a NACA 0012 aerofoil flying at $M_{\infty} = 0.7$. 

150
5.5. Simple flap

5.5.1 Investigation of the artificial dissipation in time

The implementation of the JST solver used in the simulation of the simple flap deflection in Section 5.5 only included the JST dissipation term in the spatial directions. Term $D_k$ at face $k$ in equations (2.91) and (2.92) was initially replaced by $D_k \sqrt{n_x^2 + n_y^2}$ in an attempt to avoid distorting the temporal accuracy of the solution. This produced very erratic results in transients like the simple flap deflection (see Figures 5.26 and 5.28). Hence, the addition of JST-like dissipation became necessary also in the time direction. Although this represents an improvement, unfortunately, the artificial dissipation term added in time did not seem to eliminate the oscillations in the transient flap. Figures 5.30 and 5.31 show a comparison of the pressure contours and $C_p$ distributions for the JST formulation with and without artificial dissipation in the time direction, at a certain time level during the simple flap deflection. Although the latter presents a smoother solution, more similar to upwind stencils, both cases fail to properly resolve the motion time-accurately. Nevertheless, this was implemented and used hereafter throughout all simulations in this work.

![Without artificial dissipation in time](image1.png)  ![With artificial dissipation in time](image2.png)

Figure 5.30: Pressure contour plots during the simple flap deflection ($\delta_F = 13.5$ deg, $\dot{t} = \frac{t_{ref}}{c} \approx 0.295$) on NACA 0012 flying at $M_\infty = 0.7$ via JST: without artificial dissipation in time (left) and with artificial dissipation in time (right).

![C_p distributions](image3.png)

Figure 5.31: $C_p$ distributions during the flap deflection ($\delta_F = 13.5$ deg, $\dot{t} = \frac{t_{ref}}{c} \approx 0.295$) corresponding to the time-level depicted in Figure 5.30 for NACA 0012 flying at $M_\infty = 0.7$. 
5.6 Slat and slotted flap

One of the main advantages of the spacetime method for aerodynamics relies on its versatility and the fact that it copes remarkably well with complex boundary motions. In particular, problems involving topological changes in the geometry, like the store separation, are difficult to solve via conventional means since the use of intricate, and not always obvious, interpolation methods is necessary where re-meshing is the only feasible option. In some other cases other techniques which do not require re-generating a completely new mesh, such as Chimera or overset grids, are a possibility. However, even in this case, an interpolation is necessary to connect the solution of both meshes. The case considered here represents one of such problems where the ALE formulation is not enough to cope with the separation and deflection of a slat and a slotted flap, initially indistinguishable from the main aerofoil. In the spacetime framework the difficulty resides only in the generation of a spacetime mesh and this is a trivial task in the present two-dimensional case since any commercial three-dimensional meshing tool can be used directly for this purpose.

![Figure 5.32: Spacetime mesh for a slat and flap deployment on a RAE 2815 aerofoil flying at $M_\infty = 0.23$.](image)

Initially a RAE 2815 aerofoil is immersed in a freestream flow at $M_\infty = 0.23$ with an angle of attack $\alpha = 5$ deg, and the slotted slat and flap are fully integrated within the aerofoil. At some point they move apart and deflect an angle $\delta_F = 30$ deg simultaneously, as can be understood looking at the spacetime geometry depicted in Figure 5.32. Notice that because the motion is implicitly defined by the spacetime mesh, the connectivity between different time-levels is automatically accomplished, hence simulating objects moving apart is easy when compared to the ALE formulation.

Three different stencils have been used in this simulation, namely JST, Van Leer and CSUT. Pressure contour plots with velocity streamlines are given in Figure 5.33 for all three methods and the history of convergence residuals is presented in Figure 5.34.
Figure 5.33: Pressure contours plots with velocity streamlines for slat and flap deployment on a RAE 2815 aerofoil flying at $M_\infty = 0.23$: JST (left), Van Leer (centre) and CSUT (right)
Chapter 5: Results

Figure 5.34: Convergence residuals for a slat and flap deployment on a RAE 2815 aerofoil flying at $M_{\infty} = 0.23$.

All three formulations resolve the transient satisfactorily, with the JST and CSUT almost identical given the only difference being the temporal stencil. It can be observed that the CSUT solution lies in between that for JST and Van Leer. Even the history of convergence is as expected, with the upwind solver not only the most realistic for non-periodic time-accurate problems like the present one but, at the same time, the one with the best convergence rate. Looking at these results it may be hypothesised that the main source of difference comes from the use of different schemes in space rather than a different stencil in time. Hence, it is not clear whether the extra computational effort required by the use of a hybrid stencil (central in space, upwind in time) makes up for the gains in time-accuracy when compared with the JST formulation.
5.7 Spoiler

Another typical problem where the versatility of the spacetime method is exploited is the spoiler case computed here. Like in the slat and slotted flap of Section 5.6 the simulation of large boundary motions is done seamlessly in spacetime. A three-dimensional mesh is required to describe the two-dimensional geometry and its motion or evolution throughout time. But this is a trivial task with available commercial meshing software. Figure 5.35 shows the unstructured spacetime mesh used in the solution of the current spoiler deployment case. Initially a NACA 0012 aerofoil at an angle of incidence of $\alpha = 0$ deg is immersed in a freestream flow at $M_{\infty} = 0.25$. At some point, a spoiler deflects up to an angle of $\delta_S = 45$ deg in a non-dimensional time $\frac{U_{\infty}}{c} t_c = 1.5$, and the whole transient process is calculated.

![Figure 5.35: Spacetime mesh for spoiler deployment case.](image)

As in Section 5.6 the problem is computed with three different stencils, namely JST, Van Leer and the hybrid CSUT. Pressure contour plots are depicted in Figure 5.36 for several time levels and the history of convergence residuals is given in Figure 5.37. Although all three formulations seem to resolve the transient correctly, there are differences between them. The JST and CSUT solutions use a different scheme in space and this is reflected in the prediction of a larger region of low pressure behind the spoiler. This is particularly important towards the end of the spoiler deployment, especially in the JST case. As in previous simulations this can be attributed to the fact that a non-physical time stencil is used in the JST formulation. Although a lot of effort is put into decoupling the central-difference and upwind stencils of the spatial and temporal fluxes in the CSUT formulation, this is impossible in practice in the case of unstructured meshes. It can be understood by considering the spatial fluxes at the interface between two spacetime cells whose centres are located at two different time levels, and bearing in mind that the cell average value of the conserved variables (i.e. $\rho$, $\rho u$, $\rho v$ and $\rho E$) depends not only on spatial coordinates but also on time. Therefore, temporal fluxes are inevitably affected slightly by spatial fluxes which, in the case of the CSUT formulation, prevent the time stencil from being purely upwind like in the case of Van Leer (or Roe). Finally, the history of convergence residuals is comparable for all three stencils considered here, with Van Leer the fastest convergence followed closely by CSUT, as expected.
Figure 5.36: Pressure contours for spoiler deployment on a NACA 0012 aerofoil flying at $M_\infty = 0.25$: JST (left), Van Leer (centre) and CSUT (right)
Figure 5.37: Convergence residuals for spoiler deployment on a NACA 0012 aerofoil flying at $M_{\infty} = 0.25$. 
5.8 Investigation of adverse lift

Due to the simplicity of spoiler deployment simulations via the spacetime method, the study of a rapidly deploying spoiler is done here. As explained by Yeung et al. [116] the deflection of a spoiler at a very high rotational speed produces a vortex shedding from the tip of the spoiler at the initial stages, which leads to an increase in lift. This phenomenon is usually referred to as adverse lift. Geisbauer et al. [117] state that a rapid deflection is one such that the non-dimensional time

\[ \hat{t} = \frac{U_\infty t}{c} \]  

(5.9)

associated with the spoiler deflection is \( \hat{t} \leq 5 \). Here, a spoiler of length 0.1c is located between \( x = 0.7c \) and \( x = 0.8c \) on a NACA 0012 aerofoil, as observed in Figure 5.38. Initially the aerofoil is submerged in a freestream flow at \( M_\infty = 0.25 \) and \( \alpha = 0 \) deg. At some point, the spoiler is deployed to an angle \( \delta_S = 90 \) deg in a non-dimensional time \( \hat{t} = 4.4 \). The transient is solved in spacetime via three different stencils: JST, Van Leer and Roe. Only inviscid solutions are calculated. The spacetime mesh used here is given in Figure 5.39.

Figure 5.38: Spoiler configuration (between 0.7c and 0.8c) on a NACA 0012 aerofoil flying at \( M_\infty = 0.25 \).

Figure 5.39: Spacetime mesh and geometry of the spoiler used in the investigation of the adverse lift.
Notice the importance of using unstructured meshes to define the spacetime geometry of problems involving topological changes (the spoiler is initially part of the aerofoil and there is no space between the spoiler and the main body) and large boundary motions like the current one. Moreover, as a consequence of the fully unstructured mesh in spacetime, the use of varying time-step sizes across the spatial domain requires no extra effort. The exploitation of this intrinsic ability of the spacetime method can improve the overall efficiency of the simulation since small time-steps can be used close to the geometry, where big gradients appear, whilst retaining very large time-steps far away, where the solution remains almost constant. For instance, approximately four time-steps have been used to discretise the time domain at the farfield boundaries while hundreds have been used in the neighbourhood of the aerofoil, as depicted in Figure 5.40.

![Figure 5.40: Spacetime volume mesh for the spoiler deployment used in the investigation of adverse lift: different time-step sizes at different locations.](image)

The lift coefficient $C_L$ is depicted throughout time in Figure 5.41 for each of the three stencils used. A representation of the motion of the spoiler is also given in Figure 5.41 to help understand better how and when adverse lift happens. Initially the lift is null since the aerofoil is flying at an angle of attack $\alpha = 0$ deg. The JST solution exhibits some oscillations, probably due to a non-realistic time stencil. At time $\hat{t} \simeq 1.47$ the spoiler starts to rotate at a high rate and straight after that a peak in the lift coefficient, $C_L$, can be observed for all three solutions. As explained by Yeung et al. [116] and observed in the pressure contours depicted in Figure 5.42, a strong vortex appears after the spoiler due to an "intensive shear flow near the tip". This, in turn, produces an adverse lift at the initial stages of the spoiler deployment. This vortex vanishes towards the end of the spoiler motion and the $C_L$ reaches the value of an equivalent spoiler in a static position. Here, the transient has been simulated only up to $\hat{t} = \frac{U_\infty t_c}{c} \simeq 10.27$, hence the steady state solution is not quite achieved. It is interesting to realise that the solutions of the JST and Roe formulations are very similar, both for pressure contours and integrated $C_L$. This is probably due to the JST solution achieving a higher temporal accuracy while, at the same time, using a non-realistic time stencil. It is possible that, in this particular case, the gains achieved by a better spatial and temporal accuracy cancel out with the losses due to a non-physical time stencil.
Figure 5.41: Adverse lift on rapidly deploying spoiler ($U_\infty/t = 4.4$ and $\delta_S = 90$ deg) on a NACA 0012 aerofoil flying at $M_\infty = 0.25$ and $\alpha = 0$ deg.
5.8. Investigation of adverse lift

Figure 5.42: Pressure contours for spoiler deployment in investigation of adverse lift at non-dimensional times $\hat{t} = \frac{U_\infty t}{c} = 0, 1.1, 2.2, 3.3, 4.4$ and 10: JST (left), Van Leer (centre) and Roe (right)

JST  Van Leer  Roe
Chapter 5: Results

5.9 Landing case

As outlined in previous sections, one of the main benefits of the spacetime formulation is its versatility and the fact that it is capable of handling very complex boundary motions with relative ease. Problems like the slat and slotted flap deployment in Section 5.6 can be solved without the need for further modifications to the solver nor the implementation of intricate interpolation methods that connect cells between different time levels. Moreover, the use of a finite-volume method, conservative by nature, in space and time simplifies the problem considerably. In contrast with this, defining flow properties at positions where there did not exist fluid at a former time level can be problematic with conventional arbitrary Lagrangian-Eulerian formulations. In general, this relies heavily on the accuracy of the interpolation method used.

A simplified version of all the motions that a wing undergoes during approach and landing has been modelled here, i.e. a slat and flap deployment on an aerofoil flying at $M_\infty = 0.15$ followed by an increase of its angle of incidence and a spoiler deployment which, in turn, decreases the incidence, all of which happens while approaching the ground. Figure 5.43 shows the spacetime mesh used to represent the geometry for this problem and define the motions involved in it.

Figure 5.43: Spacetime mesh for landing case, i.e. aerofoil with a slat and flap deployment followed by an increase in its angle of incidence and a spoiler deployment which, in turn, decreases the incidence, all of which happens while approaching to the ground.

As mentioned above, the solver can be left unchanged speeding up the overall simulation process, from meshing through to running the CFD code. A comparison of three different stencils has been done, namely JST, Van Leer and CSUT. Pressure contour plots with streamlines are shown in Figures 5.44 and 5.45 for all three formulations and the history of convergence residuals is plotted in Figure 5.46. In order to understand what the streamlines represent it is important to realize that the reference frame chosen for this simulation is not fixed to the aerofoil nor the ground; it moves with the aerofoil on the horizontal direction but remains fixed in the vertical direction, i.e. null vertical velocity.
Figure 5.44: Pressure contours and streamlines for landing case in a freestream flow at $M_\infty = 0.15$: (a) JST (left), (b) Van Leer (centre) and (c) CSUT (right). Continues in Figure 5.45.
As observed in the slat and flap case of Section 5.6 or the spoiler deployment problem of Section 5.7, the three stencils provide a similar solution with the main differences in the time-accuracy appearing as a consequence of a more or less realistic time stencil and the use of different spatial schemes. For instance, in the JST solution at the second time level depicted in Figure 5.44, where the aerofoil has just started to move downwards approaching the ground, the pressure at the lower side of the aerofoil seems to have raised more than in the Van Leer solution. The CSUT solution also seems to over-predict the pressure here but it lies in between JST and Van Leer. Moreover, in the last time level depicted in Figure 5.44 (the one at the bottom), where the aerofoil has just finished approaching the ground (i.e. null vertical velocity) and it has already started to pitch forwards, the JST solution seems to underestimate the pressure at the lower side of the aerofoil.
As it has been observed in many other cases, the rate of convergence of the hybrid CSUT solver is not much better than that of the central-difference one, as depicted in Figure 5.46. This can be explained by a less realistic time stencil than the Van Leer counterpart, since time fluxes are slightly affected by spatial fluxes when spacetime cell centres are not perfectly aligned at the same time level. Although this is a minor contribution it can become more noticeable in cases where the transient solution is unstable and changes rapidly.

Figure 5.46: History of convergence residuals for landing case in a freestream flow at $M_\infty = 0.15$
Chapter 5: Results

5.10 Problems with large relative motions

To contribute to the idea that the spacetime framework is very versatile and copes well with complex arbitrary boundary motions, a couple of problems showcase the potential for large relative motions in this section. Since the only objective of this section is to show the potential for very large (or, effectively, infinite) relative motions, only an upwind formulation based on Van Leer flux-splitting vector method is used.

5.10.1 Aerofoils flying in opposite directions

The spacetime method greatly simplifies problems like objects moving apart/closer that would otherwise need to be calculated via intricate mesh adaptation and interpolation techniques such as Chimera or overset grids. The problem of two parallel aerofoils flying towards another aerofoil that flies between them in the opposite direction is effectively solved with ease by the spacetime method. No further modifications to the solver are required and no mesh interpolations are needed as it would be the case with a conventional ALE formulation. The spacetime geometry used for this problem is depicted in Figure 5.47.

![Spacetime mesh for NACA 0012 aerofoils flying in opposite directions](image)

Figure 5.47: Spacetime mesh for NACA 0012 aerofoils flying in opposite directions

One of the key ideas of this of simulation is the fact that, as opposed to conventional methods, the solution is worked out in a fixed reference frame, i.e. the freestream condition is $M_\infty = 0$. In other words, instead of the fluid moving towards the aerofoils, the aerofoils themselves move at a certain speed with respect to the static air. Pressure contour plots at different time levels are given in Figure 5.48 for the aerofoils moving at $M = 0.8$ and $M = 1.6$ in opposite directions. The history of convergence residuals is provided in Figure 5.49.
5.10. Problems with large relative motions

Figure 5.48: Pressure contours for aerofoils flying in opposite directions at: (a) $M_\infty = 0.8$ (left) and (b) $M_\infty = 1.6$ (right)
It is interesting to realise that the same mesh can be (and has been) used here to simulate the transonic and supersonic cases by squeezing/stretching the mesh in time $t$ direction. There is just one thing that needs to be taken into account in this case. If the mesh used in the supersonic simulation is to be re-used for the transonic case, the size of the spacetime cells in the time direction once the mesh has been stretched (by a factor of 2 in this case) need not be too large for a good temporal accuracy to be attained.
5.10. Problems with large relative motions

5.10.2 Rotor-stator problem

Another relevant problem, and similar in complexity, is the case of a rotor-stator interaction. This problem is periodic both in space and time, hence it is particularly well-suited for the spacetime framework, which is conservative in space and time. The spacetime geometry for the rotor-stator problem is given in Figure 5.50. It consists of four NACA 0012 aerofoils in the rotor and another four in the stator. Throughout one whole period, each aerofoil in the rotor moves and occupies the next position. Hence, after four periods all aerofoils have rotated across the domain in the $y$ direction. Boundaries at $t =$ constant and $y =$ constant planes are connected to achieve temporal and spatial periodicity, respectively. Boundaries at $x =$ constant are modelled as farfield.

![Figure 5.50: Spacetime geometry for rotor-stator problem. The mesh at $t =$ constant boundaries is provided. Boundaries at $t =$ constant and $y =$ constant planes are connected to achieve temporal and spatial periodicity, respectively. Boundaries at $x =$ constant are modelled as farfield.](image)

This problem is computed to demonstrate the ability of the spacetime method to solve potentially infinite motions without the need for complex mesh deformation techniques or re-meshing. Therefore, it does not correspond to any particular geometry and none experimental data exists. However, this could represent a compressor. The flow is parallel to the $x$ direction and it is initialised at $M_{\infty} = 0.05$. The stator remains fixed and the rotor moves downwards at a speed $2.4 \cdot U_{\infty}$. Once the $L^2$-norm of the density residuals have dropped beyond $10^{-7}$, pressure contours are depicted at different time levels throughout one whole period in Figure 5.51. It can be observed that the rotor-stator compresses the air successfully when it flows through.
Figure 5.51: Pressure contours plots for rotor-stator problem at different time-levels.
Chapter 6

Conclusions and future work

6.1 Conclusions

A new methodology to discretise and solve the fluid equations of motion in spacetime for complex aerodynamics problems with large boundary motions and/or topological transformations has been presented. The applicability of the so-called spacetime framework is broad and a range of two-dimensional problems which are difficult to solve in an arbitrary Lagrangian-Eulerian formulation has been simulated here successfully, namely a slat, a slotted flap, a spoiler, a full landing case with ground effect, aerofoils flying in opposite directions past each other and a rotor-stator configuration. By means of a conventional method, any of these problems require remeshing or, in the best possible case scenario, the use of Chimera or overset grids on top of mesh deformation. In either case, there is the need for intricate, and not always obvious, interpolation methods to connect the solution between cells at different time-levels or, in the case of Chimera grids, to relate the solution at both sides of the interface between two different meshes. On the other hand, the integration of the Navier-Stokes equations in spacetime allows the solution of unsteady aerodynamics problems of dimension $N$ as steady state problems of dimension $N + 1$. Both the geometry and its motion are discretised through the use of spacetime finite-volumes (finite-elements are also possible \cite{71}) and integrated in space and time simultaneously. This, in turn, ensures the conservation of density, momentum and energy both in space and time. It is therefore very well-suited for time periodic problems, where initial and final states are connected geometrically, and convergence of the numerical solution to the exact one can be guaranteed once the $l^2$-norms of the spacetime residuals have dropped beyond a certain threshold. Moreover, a geometric conservation law is not necessary in spacetime since the mesh remains fixed throughout the whole simulation. Although any available commercial software may be used in the generation of spacetime geometries for the solution of two-dimensional unsteady problems, the main difficulty resides on the generation of spacetime meshes for three-dimensional problems. Currently, the only attempts to produce general four-dimensional meshes are limited to research groups. This
constitutes one of the main bottlenecks in the development of this young and novel technique.

If an unstructured mesh is used for the discretisation of the spacetime domain, different time-step sizes can be defined at different spatial locations. This can greatly improve the overall efficiency of the simulation if a clever exploitation of this intrinsic ability of the spacetime method is done. For instance, it is possible to use small time-steps in the neighbourhood of the geometry, where big gradients appear, whilst retaining very large time-steps far away in the farfield, where the solution remains almost constant throughout the whole simulation. This has been explained and demonstrated in Section 5.8 where roughly four time-steps are sufficient to cover the whole time domain at the farfield boundaries while hundreds have been used close to the aerofoil.

Industrial applications can benefit substantially from the use of the spacetime framework due to its potential for highly automated CFD simulations which could, in turn, speed up the design cycle. It has been shown that the solver can be left unchanged throughout the whole range of problems simulated in this work. Besides, shared- and/or distributed-memory parallelisation can be applied to the spacetime method just like any other arbitrary Lagrangian-Eulerian formulation. In particular, a shared-memory parallelisation based on OpenMP® has been implemented and used throughout the simulations in this work. One of the main objections for the use of this method is perhaps its low computational efficiency and the fact that it is, in general, slower than its dual-time counterpart since the whole time domain is updated on every iteration. However, there are two possible ways to circumvent this issue. On the one hand, taking advantage of information travelling always forwards in time at a pseudo-velocity \( u^* = \frac{dt}{dt} = 1 \), as implied from a characteristics analysis, only the solution behind this wavefront needs to be updated. This idea is analogous to a space-marching algorithm for supersonic flows. On the other hand, coupling spacetime and conventional solvers can lead to big savings in computational effort where simple or small motions happen and, at the same time, exploit the versatility of the spacetime method where large boundary motions or topological changes in the geometry need to be resolved.

Among the downsides of the spacetime method, one can outline the fact that it is hard to couple with structural solvers since motions need to be prescribed beforehand. In order to perform fluid-structure interaction simulations the spacetime domain can be subdivided into many smaller spacetime regions delimited by \( t = \) constant planes and solved one by one in order. This way, each spacetime subregion may be deformed to accommodate solid deformations. One may assume that the versatility offered by the spacetime method is lost once a subdivision like this is performed. However, topological changes in the geometry can still be simulated with the only condition of having \( t = \) constant mesh planes distributed throughout the spacetime domain. In fact, gains in efficiency can be expected since only small portions of the computational domain are updated on every iteration.

It has been demonstrated that a central-difference formulation in spacetime may introduce a violation of the principle of causality due to the use of a non-realistic time stencil. The effects of this are not noticeable in the case of periodic problems like the one-dimensional piston or the pitching aerofoils. However, pressure waves travelling backwards in time can be observed in non-periodic problems with rapid motions like the simple
flap presented in Section 5.5. The faster the motion of solid boundaries is, the more noticeable the effects consequence of a non-physical time stencil are. On the other hand, upwind formulations seem to solve this issue successfully with the use of directionally biased stencils. In general, they yield more representative solutions since there are no signs of information convected from future states. Second-order upwind solvers based on Van Leer and Roe fluxes have been implemented and used in this work. However, they do not seem to achieve as high a temporal accuracy as the JST counterpart in a number of cases simulated in this work. Besides, they are more dissipative than the central-difference and, thus, their time-accuracy is slightly compromised. The hybrid CSUT formulation (central in space, upwind in time) presented here has been found to produce solutions similar to the JST counterpart consistently, probably down to the same scheme used in space. Moreover, its ability to damp out the oscillations characteristic of the JST formulation in transients with rapidly moving boundaries, like the simple flap in Section 5.5 has also been demonstrated. Rates of convergence of the numerical solution have been found to have a connection with the physical-time stencil. The more realistic the time stencil is, the faster the $l^2$-norms of the residuals drop. In general, upwind formulations seem to converge faster than the central-difference counterpart. The history of convergence of the hybrid CSUT formulation usually lies in between the upwind and central-difference counterparts.

Following, a summary of the conclusions explained above is given as a set of bullet points:

- The spacetime method copes well with complex boundary motions and topological transformations in the geometry. It can simulate large relative motions without the use of an arbitrary Lagrangian-Eulerian (ALE) formulation in combination with a geometric conservation law (GCL).

- With the spacetime method there is no need for intricate, and not always obvious, interpolation methods to connect the solution between cells at different time-levels.

- The spacetime method ensures automatic conservation of mass, momentum and energy, not only in space but also in time. Therefore, it is particularly well-suited for the solution of time-periodic problems since initial and time boundaries are directly connected via cell interfaces.

- The spacetime method allows the solution of an unsteady problem of dimension $N$ as a steady problem of dimension $N + 1$. Hence, any of the methods (implicit or explicit) used for the solution of steady-state problems may be used directly for time-accurate problems.

- The spacetime method brings the potential for very efficient time-accurate simulations since it allows for varying time-step sizes across the space. It is possible to use fine time-steps close to the geometry, where big gradients appear, while retaining very large time-steps far away in the farfield.

- The spacetime method has a great potential for highly automated CFD simulations since no additional modifications are needed for different types of motions. The spacetime formulation presented here is general and can, in principle, cope with any kind of boundary motion.

- The spacetime method is suitable for shared- and/or distributed-memory parallelisation, just like any other conventional CFD method.
Chapter 6: Conclusions and future work

- A central-difference formulation in spacetime yields non-physical solutions as a result of pressure waves travelling backwards in physical-time. The use of upwind and hybrid stencils overcome this issue.

- Unfortunately, upwind stencils are more dissipative and they may not achieve as high a temporal accuracy as the central-difference counterpart.

- The spacetime method is, in general, slower than the ALE based counterpart since it updates the entire time domain at each pseudo-time iteration. However, taking advantage that information travels always forwards in time at a constant pseudo-velocity $u^* = \frac{dt}{dt^*} = 1$, as implied by a characteristics analysis, only the solution behind this wavefront needs to be updated in each iteration. This is similar to the idea underpinning space-marching algorithms for supersonic flows.

- Although it is hard to couple the spacetime solver with a structural solver due to the need to prescribe motions beforehand, it can still be done. Similarly, the coupling of a spacetime solver with an ALE based solver is possible.

- It is cumbersome to generate truly unstructured four-dimensional meshes to make use of the spacetime method in real life three-dimensional applications. This might constitute the major bottleneck for the development of the method.

6.2 Future work

The versatility and applicability of the spacetime method has been demonstrated in this work. However, since this is still a young and novel technique there are several topics that should be investigated further in future work.

A lot of effort has been put into multigrid techniques for faster convergence rates of conventional methods. The problem is simulated in several spatial meshes with different levels of refinement and the residuals are transferred from the coarsest to the finest mesh (other patterns are also possible) accelerating the convergence of the simulation considerably. The same idea should be applicable in spacetime, with the difference that the residuals in spacetime contain information regarding both spatial and temporal discretisations. This should improve the convergence of the simulation in spacetime considerably.

As outlined before, the implementation of the spacetime method done in this work is, in general, slower than its dual-time counterpart. This comes as a consequence of the whole time domain updated on every iteration. Moreover, as implied by a characteristics analysis, there exists a wavefront perpendicular to the $t$-direction that advances with a pseudo-velocity $u^* = \frac{dt}{dt^*} = 1$ across the spacetime domain. Big computational savings can be achieved if only the region behind this wavefront is updated on every iteration since, provided that a realistic
time-stencil is used, the region ahead of it is not influenced by past events. An implementation of this would be similar to the space-marching technique used for the solution of supersonic flows or detonations.

Given the versatility of the spacetime method and its ability to simulate with ease topological changes in the geometry, it is very well suited for kinematic optimisation problems. For instance, the investigation of high-lift devices, like a wing flap and a slat, could benefit substantially from the use of the spacetime method to solve the Navier-Stokes equations for fluids motion. The transient during the deployment of these devices can be simulated seamlessly with a spacetime formulation without the need for complex mesh motion methods such as Chimera or overset grids. Furthermore, no interpolation methods are required to connect the solution between cells at consecutive time-levels when a topological change (e.g. where the flap separates from the main wing) leads to cells appearing or disappearing.

Besides, the investigation of the solution of fluid-structure interaction problems with the spacetime method would be interesting. As explained before, this can be achieved by subdivision of the spacetime domain into smaller regions and solution of them independently in chronological order. The spacetime mesh in each of these sub-regions may be deformed to adapt to boundary motions. An increase in the efficiency of the simulation should also be expected since smaller regions of the spacetime domain are solved at a time.

Finally, and although not directly related to the work presented here, the generation of unstructured four-dimensional meshes should be investigated. None of the problems involving large boundary motions or topological changes can be simulated in spacetime with structured grids. Hence, if this work is to be extended to a more general three-dimensional space the ability to create fully unstructured four-dimensional meshes is of key importance. At the time of writing this thesis only a few attempts have been made to generate truly unstructured four-dimensional meshes [71, 72, 73]. Moreover, these have been restricted to simplex meshes based on a Delaunay triangulation in a four-dimensional space and subsequent refinement via flip edge operations. A more general algorithm to generate 4D unstructured meshes is necessary for an extension of the spacetime method to industrial applications.
Appendix A

Arbitrary Lagrangian-Eulerian formulation

Historically, the formulation of the Navier-Stokes equations for fluid motion has followed one of the two main approaches: Lagrangian or Eulerian. In the former, fluid variables are associated with particles and the solution is given for each of those particles at a given time $t$. In the latter, however, the fluid solution is given at spatial locations regardless of which individual particle occupies that position at that moment in time. In other words, if $\phi$ represents a certain fluid property (e.g. velocity, density, pressure, etc.) the Lagrangian formulation gives $\phi(x_0, t)$ at time $t$ for a fluid particle whose initial position is $x_0 = x(t_0)$, whereas the Eulerian formulation gives the solution $\phi(x, t)$ at a given location $x$ and time $t$, regardless of which particle is at that space-time location.

There are some straightforward implications derived from these definitions. In Lagrangian based computations mesh nodes follow the associated fluid particles. This description of the fluid makes the treatment of problems with free surfaces and/or multiple phases straightforward [118]. Such a formulation proves very difficult to implement in fluid dynamics problems where the necessary grid displacements may lead to very distorted and low quality meshes with inverted elements in some cases. Therefore, a pure Lagrangian formulation is much better suited to structural problems where deformations are small and the overall grid quality is preserved. In Eulerian formulations the grid may remain constant regardless of particle motions, allowing for a much simpler and more efficient representation of fluids. However, even though the Eulerian formulation can easily handle large displacements of fluid particles with respect to the underlying fixed grid, it is still necessary to transform the mesh (re-mesh or, in the best case scenario, deform the mesh) in situations where boundaries move or in case a better resolution is sought in order to capture shocks or other phenomena.

The arbitrary Lagrangian-Eulerian framework, commonly abbreviated as ALE, is a widely used solution for moving boundary problems in aerodynamics. It lies in between pure Lagrangian and pure Eulerian formulations.
The underlying grid does not follow material particles nor is it fixed in space. On the contrary, it deforms in order to accommodate boundary motions and changes in the geometry and/or interfaces. In the limit, an ALE framework can be seen as a purely Lagrangian description if the mesh moves with the fluid, or as a purely Eulerian one if the mesh remains fixed in space. However, the main strengths of both formulations can be retained under the ALE framework by achieving a more efficient computation than the Eulerian counterpart, with refined areas and a better quality grid, and being also capable of handling motions with larger displacements than Lagrangian methods. An extensive description of the ALE framework can be found in [119, 120, 121] and a representation of particles and mesh motions in the case of a one-dimensional grid is depicted in Figure A.1.

Formulating the Navier-Stokes equations for fluid motion in an ALE framework requires the introduction of the Reynolds transport theorem which establishes that the rate of change of a fluid property \( \phi(x,t) \) (e.g. mass) over a time-dependent control volume \( V_c(t) \) may be decomposed into two contributions, as follows

\[
\frac{d}{dt} \int_{V_c(t)} \phi(x,t) \, d\Omega = \int_{V_c(t)} \frac{\partial \phi(x,t)}{\partial t} \, d\Omega + \oint_{\partial V_c(t)} \phi(x,t) \mathbf{v}_c \cdot \mathbf{n} \, dS \tag{A.1}
\]

where \( \mathbf{v}_c \) is the velocity of the boundary of the control volume \( V_c(t) \). Let us assume that this control volume represents a volume element of an ALE formulation. Moreover, particularizing equation (A.1) to a fluid volume
V (t) which occupies the same portion of space at time t, i.e. $V (t) \equiv V_c (t)$, yields

$$\frac{d}{dt} \int_{V(t)} \phi (x,t) \, d\Omega = \int_{V(t)} \frac{\partial \phi (x,t)}{\partial t} \, d\Omega + \oint_{\partial V(t)} \phi (x,t) \mathbf{v} \cdot \mathbf{n} \, dS \tag{A.2}$$

where $\mathbf{v}$ is now the fluid velocity because, by definition, the fluid volume $V(t)$ moves with the fluid.

The conservation of mass can be written as $\frac{d}{dt} \int_{V(t)} \rho (x,t) \, d\Omega = 0$ or, subtracting equation (A.1) to (A.2) and substituting $\phi (x,t) = \rho (x,t)$,

$$\frac{d}{dt} \int_{V_c(t)} \rho (x,t) \, d\Omega + \oint_{\partial V_c(t)} \rho (x,t) (\mathbf{v} - \mathbf{v}_c) \cdot \mathbf{n} \, dS = 0 \tag{A.3}$$

where the first integral of the right hand side of equation (A.1) has been cancelled out with the first integral of the right hand side of equation (A.2) due to the aforementioned equivalence $V (t) \equiv V_c (t)$ at time $t$.

Equation (A.3) constitutes the conservation of mass for fluid motion in an arbitrary Lagrangian-Eulerian formulation applied to a volume element whose boundary moves with a certain velocity $\mathbf{v}_c$. Notice that, in general $\mathbf{v}_c \neq \mathbf{v}$ and $\mathbf{v}_c \neq 0$. However, there exist two special cases worth mentioning. In a purely Lagrangian case the grid (i.e. discretisation of the fluid domain into smaller volume elements) would follow the fluid, $\mathbf{v}_c = \mathbf{v}$, whereas in a purely Eulerian one the grid would remain unchanged through time, $\mathbf{v}_c = 0$.

An equivalent formulation to (A.3) can also be derived for the conservation of momentum and energy within an ALE framework, as given by equations (2.1)-(2.4).

### A.1 Geometric conservation law

As first introduced and explained by Thomas and Lombard [122], the arbitrary Lagrangian-Eulerian framework applied to the solution of unsteady aerodynamic problems via finite-volume or finite-element methods (e.g. discontinuous-Galerkin [121]) require a so-called geometric conservation law (GCL). An inexact integration of the volumes can lead to numerical errors which suppress the trivial uniform solution from the discretised equations. In other words, errors may be introduced if the coordinates of the grid nodes at a certain time level are used directly to work out cell volumes exactly because the numerical integration only solves mesh motion approximately. The GCL avoids this by working out the volumes by means of the area swept by each cell boundary (edge or face) between two consecutive time levels, using the same numerical integration scheme as the one used to solve the physical conservation laws.
Appendix A: Arbitrary Lagrangian-Eulerian formulation

With conventional solution methods, under the assumption that any numerical scheme should preserve a uniform solution, the GCL can be easily derived by applying the discretised conservation laws to a constant and uniform fluid flow in a moving grid. For instance, the application of the Euler equations of motion (see Appendix B) to a two-dimensional time-dependent volume $V(t)$ with uniform flow properties, i.e. where $\rho_0$, $u_0$, $v_0$, $p_0$ and $E_0$ are constant and uniform, gives

$$\frac{d}{dt} \int_{V(t)} U_0 d\Omega + \oint_{\partial V(t)} U_0 [v_0 - v_c]^T n \, dS = \oint_{\partial V(t)} P_0 n \, dS$$  \hspace{1cm} (A.4)$$

where $n = \{n_x, n_y\}$ is the normal vector at each face, $v_0 = \{u_0, v_0\}$ and $v_c(t) = \{u_c(t), v_c(t)\}$ are the column vectors of flow velocity and velocity of the control volume boundaries, respectively, $U_0$ is the column vector of conserved variables

$$U_0 = \begin{pmatrix} \rho_0 \\ \rho_0 u_0 \\ \rho_0 v_0 \\ \rho_0 E_0 \end{pmatrix}$$  \hspace{1cm} (A.5)$$

and $P_0$ is a matrix which represents the pressure terms

$$P_0 = \begin{bmatrix} 0 & 0 \\ p_0 & 0 \\ 0 & p_0 \\ p_0 u_0 & p_0 v_0 \end{bmatrix}$$  \hspace{1cm} (A.6)$$

Notice that $U_0 [v_0 - v_c]^T$ represents the outer product or tensor product between column vector $U_0$ and the transpose of column vector $[v_0 - v_c]$, and yields a matrix of size $4 \times 2$, i.e.

$$\begin{bmatrix} \rho_0 \\ \rho_0 u_0 \\ \rho_0 v_0 \\ \rho_0 E_0 \end{bmatrix} \begin{bmatrix} u_0 - u(t) \\ v_0 - v(t) \end{bmatrix} = \begin{bmatrix} \rho_0[u_0 - u(t)] & \rho_0 [v_0 - v(t)] \\ \rho_0 u_0[u_0 - u(t)] & \rho_0 u_0 [v_0 - v(t)] \\ \rho_0 v_0[u_0 - u(t)] & \rho_0 v_0 [v_0 - v(t)] \\ \rho_0 E_0[u_0 - u(t)] & \rho_0 E_0 [v_0 - v(t)] \end{bmatrix}$$  \hspace{1cm} (A.7)$$

Since $U_0$, $v_0$ and $P_0$ are constant for the case considered here, it is possible to move them out of the integrals
in equation (A.4) yielding

\[ U_0 \frac{d}{dt} \int_{V(t)} d\Omega + U_0 v_T^0 \oint_{\partial V(t)} \mathbf{n} dS - U_0 \oint_{\partial V(t)} v_T^c \mathbf{n} dS = P_0 \oint_{\partial V(t)} \mathbf{n} dS \tag{A.8} \]

Bearing in mind that \( \oint_{\partial V(t)} \mathbf{n} dS = 0 \) one can write

\[ U_0 \left( \frac{d}{dt} \int_{V(t)} d\Omega - \oint_{\partial V(t)} v_T^c \mathbf{n} dS \right) = 0 \tag{A.9} \]

Since a non-trivial uniform solution \( (U_0 \neq 0) \) must be preserved by the discretisation of the Euler equations, the second factor in the above equation must be identically zero. This can be integrated numerically via the same scheme used to solve the equations of fluid motion and obtain the sought geometric conservation law. For instance, using a finite volume approach and a second-order backward differentiation formula (BDF2) for the time derivative yields, at each cell in the mesh,

\[ V^{n+1} = \frac{4V^n - V^{n-1}}{3} + 2\Delta t \sum_{i=1}^{n_f} \left( v_{c,i}^{n+1} n_x^{n+1} + v_{c,i}^{n+1} n_y^{n+1} \right) \Delta S^{n+1} \tag{A.10} \]

where \( n \) is the current time level and \( n_f \) is the total number of cell faces.
Appendix B

Governing equations

In this appendix, a summary of the governing equations used in the current work is presented. The Navier-Stokes equations are derived from physical conservation laws and are the most general form of the partial differential equations that describe fluid behaviour. To study turbulent flows Reynolds and Favre average of the Navier-Stokes equations is required together with a turbulence model to close the problem. In this work a Spalart-Allmaras one equation model has been used and is presented in Appendix C. Finally, assuming the flow is inviscid and compressible, the Euler equations for fluid motion are derived from the Navier-Stokes equations.

B.1 Navier-Stokes equations

The Navier-Stokes equations for a compressible fluid can be written using tensor notation

\[
\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_j)}{\partial x_j} = 0 \tag{B.1}
\]

\[
\frac{\partial (\rho u_i)}{\partial t} + \frac{\partial (\rho u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho f_i \tag{B.2}
\]

\[
\frac{\partial (\rho E)}{\partial t} + \frac{\partial [(\rho E + p) u_j]}{\partial x_j} = \frac{\partial}{\partial x_j} \left( \kappa \frac{\partial T}{\partial x_j} \right) + \frac{\partial (\tau_{jk} u_k)}{\partial x_j} + \rho f_j^V u_j + Q_H \tag{B.3}
\]

where \(\rho\) is the density, \(u_i\) is the fluid velocity in the \(i\)-direction, \(p\) is the pressure, \(T\) is the temperature and \(E\)
Appendix B: Governing equations

is the total energy defined as

\[ E = e + \frac{1}{2} u_j u_j \]  

(B.4)

where \( e \) denotes the internal energy

\[ e = c_v T \]  

(B.5)

and \( c_v \) is the specific heat capacity at constant volume. The Cartesian coordinate aligned with the \( i \)-direction is \( x_i \) and the thermal conductivity \( \kappa \) is defined as

\[ \kappa = \frac{\mu c_p}{Pr} \]  

(B.6)

where \( \mu \) is the dynamic viscosity, \( c_p \) is the specific heat capacity at constant pressure and \( Pr \) is the Prandtl number. The kinematic viscosity \( \nu \) is defined as the ratio

\[ \nu = \frac{\mu}{\rho} \]  

(B.7)

Dynamic viscosity \( \mu \) is a function of the temperature \( T \) and can be obtained via Sutherland’s law \cite{124} as follows

\[ \frac{\mu}{\mu_\infty} = \left( \frac{T}{T_\infty} \right)^{\frac{3}{2}} \frac{T_\infty + 110}{T + 110} \]  

(B.8)

where the subscript \( \infty \) denotes reference or freestream values. Moreover, \( \tau_{ij} \) is the viscous shear stress tensor and in the case of a Newtonian fluid in local thermodynamic equilibrium the following relationship applies

\[ \tau_{ij} = \mu \left( 2 S_{ij} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) \]  

(B.9)

where \( S_{ij} \) is the strain rate tensor defined as

\[ S_{ij} = \frac{1}{2} \left( \frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right) \]  

(B.10)
and $\delta_{ij}$ is the Kronecker delta given by

$$\delta_{ij} = \begin{cases} 
1 & \text{if } i = j \\
0 & \text{if } i \neq j 
\end{cases}$$  \hspace{1cm} (B.11)

Finally, $f_i^V$ is the sum of all external volume forces per unit volume and $Q_H$ accounts for heat fluxes due to chemical reactions. For ideal gases the equation of state, which provides a relationship between the pressure, the density and the temperature, can be written as

$$p = \rho R_g T$$  \hspace{1cm} (B.12)

where $R_g$ is the specific gas constant given by

$$R_g = c_v (\gamma - 1)$$  \hspace{1cm} (B.13)

and $\gamma$ is the heat capacity ratio

$$\gamma = \frac{c_p}{c_v}$$  \hspace{1cm} (B.14)

Solving these equations, i.e. implementing a direct numerical simulation (DNS) solver is, in principle, possible to predict any flow structure or behaviour provided that the mesh is fine enough to resolve the smallest scales in the problem. However, the cost of performing such calculations is so big that, in practice, this is limited to specific problems at low Reynolds numbers. Turbulence models provide decent solutions of high Reynolds flows at a fraction of the cost. Depending on whether turbulence is actually calculated or modelled or a combination of calculation and modelling is used, one can distinguish the following mainstream techniques: (Favre-)Reynolds-averaged Navier-Stokes (RANS), where only turbulence models are used; large-eddy simulation (LES), where larger eddies are calculated while models are used for the smaller scales; detached-eddy simulation (DES), where regions near solid walls are treated in a RANS manner whereas an LES formulation is used elsewhere, hence being cheaper than the purely LES approach; unsteady Reynolds-averaged Navier-Stokes (URANS), where transient terms are retained in the RANS equations; and any of the numerous hybrid LES/URANS techniques.

In the present thesis a RANS approach has been used and is explained in the following sections. Firstly, an introduction to Reynolds (or time) averaging and Favre (or density weighted) averaging is given in section B.1.1. Next, the derivation of the (Favre-)Reynolds-averaged Navier-Stokes equations is presented in section

185
Appendix B: Governing equations

B.1.1 Reynolds and Favre averaging

Turbulent flows are characterised by rapid variations of the flow properties, as depicted in Figure B.1.

![Sample fluid property in a turbulent flow. The total value can be decomposed into a mean value and a fluctuating part.](image)

In order to study turbulent flows an average value of a variable in the time domain, called a Reynolds average \[ \overline{\phi} \] may be introduced as follows

\[
\overline{\phi}(x_j, t) = \frac{1}{\delta T} \int_{t-\delta T}^{t} \phi(x_j, t+\tau) \, d\tau \quad (B.15)
\]

where the over-bar \( \overline{\phi} \) means it is a time-averaged quantity. To retain the time dependency of the Navier-Stokes equations the value of \( \delta T \), over which fluid variables are averaged, needs to be sufficiently small compared to all other unsteady phenomena but big enough in comparison with turbulence effects. Should the time scales corresponding to turbulence and other unsteady phenomena be of the same order of magnitude, the Reynolds averaging would fail to capture unsteadiness in the flow. Any quantity in the flow may now be split into an average \( \overline{\phi} \) and a fluctuating \( \phi' \) part, as follows

\[
\phi(x_j, t) = \overline{\phi}(x_j, t) + \phi'(x_j, t) \quad (B.16)
\]

where, by definition, the time-average of the fluctuating part is identically zero, \( \overline{\phi'} = 0 \). Applying this time-averaging to density and velocity in the continuity equation for a one-dimensional compressible flow yields

\[
\frac{\partial}{\partial t} (\overline{\rho} + \rho') + \frac{\partial}{\partial x_j} (\overline{\rho u_j} + \rho' u_j + \rho' u_j') = 0 \quad (B.17)
\]
Averaging equation \([B.17]\) leads to the Reynolds averaged form of the continuity equation

\[
\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j} \left( \rho u_j + \rho' u'_j \right) = 0 \tag{B.18}
\]

where the correlation terms between density and velocities \(\rho' u'_j\) are unknown and an estimation is needed to close the problem. If the Reynolds averaging was to be applied to the momentum equations other extra correlation terms involving not only density and velocities but also density and internal energy would need to be evaluated. Therefore, in the case of compressible flows, a density weighted averaging, also known as Favre averaging, has been historically used, as follows. Define

\[
\tilde{\phi} = \frac{\rho \phi}{\rho} \tag{B.19}
\]

where the tilde \(\tilde{\phi}\) represents a Favre or density weighted average quantity. Fluid variables may be written in terms of the Favre average as

\[
\phi(x_i, t) = \tilde{\phi}(x_i, t) + \phi''(x_i, t) \tag{B.20}
\]

It can be shown that the Favre average of the fluctuating part is identically zero

\[
\tilde{\phi''} = \frac{\rho \phi''}{\rho} = 0 \tag{B.21}
\]

### B.1.2 Reynolds-averaged Navier-Stokes equations for compressible flow

Averaged Navier-Stokes \([126, 127]\) can be derived from equations \([B.1], [B.2]\) and \([B.3]\) to yield

\[
\frac{\partial \rho}{\partial t} + \frac{\partial (\rho \tilde{u}_j)}{\partial x_j} = 0 \tag{B.22}
\]

\[
\frac{\partial (\rho \tilde{u}_i)}{\partial t} + \frac{\partial}{\partial x_j} \left( \rho \tilde{u}_i \tilde{u}_j + \rho' u'_i u'_j \right) = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} \tag{B.23}
\]
Appendix B: Governing equations

\[
\frac{\partial}{\partial t} \left( \bar{p} \bar{E} \right) + \frac{\partial}{\partial x_j} \left[ (\bar{p} \bar{E} + \bar{p}) \bar{u}_j \right] = \frac{\partial}{\partial x_j} \left( \tau_{jk} \bar{u}_k + \tau_{jk} u''_k \right) + \frac{\partial}{\partial x_j} \left[ \mu_p \frac{\partial T}{\partial x_j} - c_p \bar{p} u''_j T'' - \rho u''_j u''_k \bar{u}_k - \rho u''_j \frac{1}{2} u''_j u''_j \right] \tag{B.24}
\]

where a Reynolds decomposition has been applied to density \( \rho \) and pressure \( p \)

\[
\rho = \bar{p} + \rho' \quad \quad p = \bar{p} + p'
\tag{B.25}
\]

and a Favre decomposition to velocities \( u_i \), total energy \( E \), kinematic viscosity \( \mu \) and temperature \( T \)

\[
u_i = \bar{u}_i + u''_i \quad \quad E = \bar{E} + E'' \quad \quad \mu = \bar{\mu} + \mu'' \quad \quad T = \bar{T} + T''
\tag{B.26}
\]

Total energy \( \bar{E} \) includes now the kinetic energy per unit volume \( k_t \) of the turbulent fluctuations, i.e.

\[
\bar{E} = \bar{\varepsilon} + \frac{1}{2} \bar{u}_j \bar{u}_j + k_t
\tag{B.27}
\]

with

\[
k_t = \frac{1}{2} \bar{u}_j'' \bar{u}_j''
\tag{B.28}
\]

Bear in mind that external volume forces \( (\rho f_j V = 0) \) and heat fluxes due to chemical reactions \( (Q_H = 0) \) have been neglected. Using the Favre averaging the equation of state for ideal gases \( \text{[B.12]} \) can be written as

\[
p = \bar{p} R_g \bar{T}
\tag{B.29}
\]

The molecular viscosity term, \( \tau_{jk} u''_k \), and the turbulent transport one, \( \rho u''_j u''_j' \), in equation \( \text{[B.24]} \) may be approximated by \( \text{[123]} \)

\[
\tau_{jk} u''_k - \rho u''_j' u''_j' \simeq \left( \bar{\mu} + \frac{\bar{\mu}}{\sigma_k} \right) \frac{\partial k_t}{\partial x_j}
\tag{B.30}
\]

188
where \( \sigma_k \) is a closure constant. In the case of non-hypersonic problems these terms can be neglected because the turbulent kinetic energy is negligible compared to pressure terms, i.e. \( \bar{pk}_t \ll \bar{p} \). In this case, the turbulent kinetic energy term can be also removed from the total energy equation in (B.27). Moreover, the turbulent heat flux term \(-c_p \bar{mu}_j^T \partial \tilde{T} / \partial x_j\) can be approximated using the following analogy between momentum and heat transfer

\[
-c_p \bar{mu}_j^T \partial \tilde{T} / \partial x_j = \tilde{\mu}_t c_p \Pr_t \partial \tilde{T} / \partial x_j
\]  

(B.31)

where \( \tilde{\mu}_t \) is the so-called eddy viscosity which accounts for turbulent effects and \( Pr_t \) is the turbulent Prandtl number defined as the ratio between momentum and heat transfer. The value of \( Pr_t \) can be approximated by \( Pr_t \simeq 0.89 \) in the case of shock-free flows with non-hypersonic speeds, provided that they are not heat dominated problems [123].

Using (B.30) and (B.31), assuming non-hypersonic flows, hence using \( \bar{pk}_t \ll \bar{p} \), one can re-write the energy equation (B.24) as

\[
\frac{\partial}{\partial t} \left( \bar{E} \right) + \frac{\partial}{\partial x_j} \left[ (\bar{pE} + \bar{p}) \bar{u}_j \right] = \frac{\partial}{\partial x_j} \left( \bar{\sigma}_{jk} \bar{u}_k - \tilde{\mu}_t \left( \tilde{S}_{ij} - \frac{1}{3} \frac{\partial \bar{u}_k}{\partial x_k} \delta_{ij} \right) \right) + \frac{\partial}{\partial x_j} \left[ c_p \left( \frac{\tilde{\mu}_t}{Pr_t} + \frac{\bar{\mu}_t}{Pr_t} \right) \frac{\partial \tilde{T}}{\partial x_j} \right]
\]  

(B.32)

where the total energy can now be written as

\[
\tilde{E} = \bar{e} + \frac{1}{2} \bar{u}_j \bar{u}_j
\]  

(B.33)

### B.1.3 Boussinesq approximation

An estimation of the Favre-averaged Reynolds-stress tensor, term \(-\bar{mu}_j^T \), can be worked out using the Boussinesq approximation [123] as

\[
-\bar{mu}_j^T = 2\tilde{\mu}_t \left( \tilde{S}_{ij} - \frac{1}{3} \frac{\partial \bar{u}_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \tilde{p} k_i \delta_{ij}
\]  

(B.34)

where, again, the term \(-\frac{2}{3} \tilde{p} k_i \delta_{ij}\) can be neglected for non-supersonic flows. Using the definition of the viscous
shear stress tensor in [B.9] one can write equation (B.34) as

\[-\rho \mu_i \mu_j = \tilde{\mu}_t \tau_{ij} \]  

(B.35)

or, multiplying by \( \tilde{u}_j \) and adding \( \tau_{ij} \tilde{u}_j \),

\[ \tau_{ij} \tilde{u}_j - \rho \mu_i \mu_j \tilde{u}_j = \left( 1 + \frac{\tilde{\mu}_t}{\tilde{\mu}} \right) \tau_{ij} \tilde{u}_j \]  

(B.36)

Combining (B.23) and (B.35) the momentum equation yields

\[ \frac{\partial (\tilde{\rho} \tilde{u}_i)}{\partial t} + \frac{\partial (\tilde{\rho} \tilde{u}_i \tilde{u}_j)}{\partial x_j} = -\frac{\partial \tilde{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left[ \left( 1 + \frac{\tilde{\mu}_t}{\tilde{\mu}} \right) \tau_{ij} \right] \]  

(B.37)

Likewise, combining (B.32) and (B.36) the energy equation finally yields

\[ \frac{\partial (\tilde{\rho} \tilde{E})}{\partial t} + \frac{\partial (\tilde{\rho} \tilde{E} + \tilde{p}) \tilde{u}_j}{\partial x_j} = \frac{\partial}{\partial x_j} \left[ \left( 1 + \frac{\tilde{\mu}_t}{\tilde{\mu}} \right) \tau_{jk} \tilde{u}_k \right] + \frac{\partial}{\partial x_j} \left[ c_p \left( \frac{\tilde{\mu}}{\tilde{P}_r} + \frac{\tilde{\mu}_t}{\tilde{P}_r} \right) \frac{\partial \tilde{T}}{\partial x_j} \right] \]  

(B.38)

B.1.4 Non-dimensionalisation

Let \( L_0 \) be some characteristic length in the problem, and \( \rho_\infty, U_\infty, \mu_\infty \) and \( T_\infty \) be the reference or freestream values of density, velocity, kinematic viscosity and temperature, respectively. All the variables may now be written in terms of these reference values as summarised in Table B.1 where the corresponding non-dimensional variables are denoted by the superscript *. Also, notice that the Reynolds (\( \tilde{\phi}, \phi' \)) and Favre (\( \tilde{\phi}, \phi'' \)) averaging notation has been dropped for the sake of simplicity.

<table>
<thead>
<tr>
<th>( x_i^* )</th>
<th>( t^* )</th>
<th>( \rho^* )</th>
<th>( u_i^* )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_i )</td>
<td>( t )</td>
<td>( \rho )</td>
<td>( u_i )</td>
</tr>
<tr>
<td>( L_0 )</td>
<td>( L_0/\rho_\infty )</td>
<td>( \rho_\infty )</td>
<td>( U_\infty )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \mu^* )</th>
<th>( p^* )</th>
<th>( T^* )</th>
<th>( E^* )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mu/\mu_\infty )</td>
<td>( p/\rho_\infty U_\infty^2 )</td>
<td>( T/T_\infty )</td>
<td>( E/U_\infty^2 )</td>
</tr>
</tbody>
</table>

Table B.1: Non-dimensionalisation used in RANS equations

The non-dimensional RANS equations for compressible flow may now be written as
B.1. Navier-Stokes equations

\[
\frac{\partial \rho^*}{\partial t^*} + \frac{\partial (\rho^* u_j^*)}{\partial x_j^*} = 0 \quad (B.39)
\]

\[
\frac{\partial (\rho^* u_i^*)}{\partial t^*} + \frac{\partial (\rho^* u_i^* u_j^* + p^* \delta_{ij})}{\partial x_j^*} = \frac{\partial \sigma_{ij}^*}{\partial x_j^*} \quad (B.40)
\]

\[
\frac{\partial (\rho^* E^*)}{\partial t^*} + \frac{\partial [(\rho^* E^* + p^*) u_j^*]}{\partial x_j^*} = \frac{\partial}{\partial x_j^*} \left( \sigma_{jk}^* u_k^* + q_j^* \right) \quad (B.41)
\]

where the non-dimensional Favre-averaged shear stress tensor has been re-defined to include the effects of turbulence as

\[
\sigma_{ij}^* = \left( 1 + \frac{\mu_t^*}{\mu^*} \right) \tau_{ij}^* = \mu^* + \mu_t^* \frac{2 S_{ij}^* - \frac{2}{3} \partial u_i^*}{3 \partial x_j^*^*} \quad (B.42)
\]

with

\[
S_{ij}^* = \frac{1}{2} \left( \frac{\partial u_i^*}{\partial x_j^*} + \frac{\partial u_j^*}{\partial x_i^*} \right) \quad (B.43)
\]

and the non-dimensional heat fluxes \( q_j^* \) are

\[
q_j^* = \frac{1}{Re_\infty M_\infty^2 (\gamma - 1)} \left( \frac{\mu^*}{Pr_t^*} + \frac{\mu_t^*}{Pr_t^*} \right) \frac{\partial T^*}{\partial x_j^*} \quad (B.44)
\]

The Reynolds number at the freestream is defined as the non-dimensional quantity

\[
Re_\infty = \frac{\rho_\infty U_\infty L_0}{\mu_\infty} \quad (B.45)
\]

and the freestream Mach number \( M_\infty \) as

\[
M_\infty = \frac{U_\infty}{a_\infty} \quad (B.46)
\]
where the speed of sound at the freestream is

\[ a_\infty = \sqrt{\left( \frac{\partial p}{\partial \rho} \right)_s} \bigg|_{\infty} = \sqrt{\gamma R_s T_\infty} \]  

(B.47)

The subscript \( s \) in the partial derivative represents that the derivative is done at constant entropy.

### Derivation of non-dimensional RANS equations

The derivation of the non-dimensional RANS equations (B.39), (B.40) and (B.41) is straightforward. Using \( L_0 \), \( t_0 \), \( \rho_\infty \), \( U_\infty \), \( p_\infty \), \( E_\infty \), \( \mu_\infty \) and \( T_\infty \) as the reference values of length, time, density, velocity, pressure, energy, dynamic viscosity and temperature, respectively, dimensional variables can be expressed as

\[
\begin{align*}
    x_i &= L_0 x_i^* \\
    t &= t_0 t^* \\
    \rho &= \rho_\infty \rho^* \\
    u_i &= U_\infty u_i^* \\
    \mu &= \mu_\infty \mu^* \\
    p &= p_\infty p^* \\
    T &= T_\infty T^* \\
    E &= E_\infty E^*
\end{align*}
\]  

(B.48)

Dropping the Reynolds and Favre averaging notation for the sake of simplicity, equations (B.22), (B.37) and (B.38) can be written using (B.48) as

\[
\begin{align*}
\frac{\rho_\infty}{t_0} \frac{\partial \rho^*}{\partial t^*} + \frac{\rho_\infty}{L_0} \frac{U_\infty}{L_0} \frac{\partial \left( \rho^* u_i^* \right)}{\partial x_j^*} &= 0 \\
\frac{\rho_\infty}{t_0} U_\infty \frac{\partial \left( \rho^* u_i^* \right)}{\partial t^*} + \frac{\rho_\infty}{L_0} \frac{U_\infty^2}{L_0} \frac{\partial \left( \rho^* u_i^* u_j^* \right)}{\partial x_j^*} &= \frac{p_\infty}{L_0} \frac{\partial \rho^*}{\partial x_i^*} + \frac{\mu_\infty}{L_0} \frac{U_\infty}{L_0} \frac{\partial}{\partial x_i^*} \left[ \left( 1 + \frac{\mu_i^*}{\mu^*} \right) \tau_{ij}^* \right] \\
\frac{\rho_\infty}{t_0} E_\infty \frac{\partial \left( \rho^* E^* \right)}{\partial t^*} + \frac{\rho_\infty}{L_0} \frac{U_\infty}{L_0} \frac{\partial \left( \rho^* E^* u_j^* \right)}{\partial x_j^*} + \frac{p_\infty}{L_0} U_\infty \frac{\partial \left( \rho^* u_j^* \right)}{\partial x_j^*} &= \frac{\mu_\infty U_\infty^2}{L_0} \frac{\partial}{\partial x_j^*} \left[ \left( 1 + \frac{\mu_i^*}{\mu^*} \right) \tau_{kj}^* u_k \right] + \frac{\mu_\infty T_\infty}{L_0^2} \frac{\partial}{\partial x_j^*} \left[ c_p \left( \frac{\rho^*}{\rho^*} + \frac{\mu_i^*}{\mu^*} \right) \frac{\partial T^*}{\partial x_j^*} \right]
\end{align*}
\]  

(B.50)

(B.51)

Multiplying the continuity equation by \( \frac{L_0}{\rho_\infty U_\infty} \), the momentum equations by \( \frac{L_0}{\rho_\infty U_\infty^2} \) and the energy equation by \( \frac{L_0}{\rho_\infty U_\infty^3} \) yields

\[
\frac{L_0}{U_\infty t_0} \frac{\partial \rho^*}{\partial t^*} + \frac{\partial \left( \rho^* u_i^* \right)}{\partial x_j^*} = 0
\]  

(B.52)

192
B.1. Navier-Stokes equations

\[
\frac{L_0}{U_\infty l_0} \frac{\partial (\rho^* u_i^*)}{\partial t^*} + \frac{\partial (\rho^* u_i^* u_j^*)}{\partial x_j^*} = -\frac{p_\infty}{\rho_\infty U_\infty^2} \frac{\partial p^*}{\partial x_i^*} + \frac{1}{Re_\infty} \frac{\partial}{\partial x_j^*} \left[ \left( 1 + \frac{\mu^*_T}{\mu^*} \right) \tau_{ij}^* \right] \tag{B.53}
\]

\[
\frac{L_0 E_\infty}{U_\infty l_0} \frac{\partial (\rho^* E^*)}{\partial t^*} + \frac{E_\infty}{U_\infty^2} \frac{\partial (\rho^* E^* u_j^*)}{\partial x_j^*} + \frac{p_\infty}{\rho_\infty U_\infty^2} \frac{\partial (\rho^* u_j^*)}{\partial x_j^*} = \frac{1}{Re_\infty} \frac{\partial}{\partial x_j^*} \left[ \left( 1 + \frac{\mu^*_T}{\mu^*} \right) \tau_{jk}^* u_k^* \right] + \frac{T_\infty}{U_\infty^2} \frac{1}{Re_\infty} \frac{\partial}{\partial x_j^*} \left[ c_p \left( \frac{\mu^*_T}{Pr} + \frac{\mu^*_T}{Pr_T} \right) \frac{\partial T^*}{\partial x_j^*} \right] \tag{B.54}
\]

In the above equations there are four independent dimensionless quantities, namely

\[
\frac{L_0}{U_\infty l_0}, \quad \frac{p_\infty}{\rho_\infty U_\infty^2}, \quad \frac{E_\infty}{U_\infty^2} \quad \text{and} \quad \frac{c_p T_\infty}{U_\infty^2}
\]

To simplify the problem further, it is possible to equate the first three dimensionless parameters to one. In other words, the characteristic time \( t_0 \), pressure \( p_\infty \) and total energy \( E_\infty \) may be chosen such that

\[
t_0 = \frac{L_0}{U_\infty} \tag{B.55}
\]

\[
p_\infty = \rho_\infty U_\infty^2 \tag{B.56}
\]

\[
E_\infty = U_\infty^2 \tag{B.57}
\]

The fourth non-dimensional quantity may be written in terms of the heat capacity ratio \( \gamma \) and the freestream Mach number \( M_\infty \) using equations (B.13), (B.14), (B.46) and (B.47) as follows

\[
\frac{c_p T_\infty}{U_\infty^2} = \frac{1}{\gamma - 1} \cdot \frac{1}{M_\infty^2} \tag{B.58}
\]

Finally, introducing (B.45), (B.55), (B.56), (B.57) and (B.58) into equations (B.52), (B.53) and (B.54) yields the non-dimensional RANS equations for compressible flow given by (B.39), (B.40) and (B.41).
Appendix B: Governing equations

B.2 Euler equations

For inviscid and compressible flows the Navier-Stokes equations simplify to the so-called Euler equations. These equations do not include viscous forces but give accurate solutions for attached flows at high Reynolds numbers, where viscous effects can be neglected without incurring big accuracy losses. Many applications of the Euler equations can be found throughout the history of CFD \cite{106, 7, 128, 76, 1}. Removing all viscous terms in the non-dimensional RANS equations (B.39), (B.40) and (B.41), and dropping the superscript * notation corresponding to non-dimensional variables for the sake of simplicity, yields

\[
\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_j)}{\partial x_j} = 0 \quad (B.59)
\]

\[
\frac{\partial (\rho u_i)}{\partial t} + \frac{\partial (\rho u_i u_j + p \delta_{ij})}{\partial x_j} = 0 \quad (B.60)
\]

\[
\frac{\partial (\rho E)}{\partial t} + \frac{\partial [(\rho E + p) u_j]}{\partial x_j} = 0 \quad (B.61)
\]

where all the variables are non-dimensional according to Table B.1.
Appendix C

Turbulence model

In this appendix, the one equation Spalart-Allmaras turbulence model used along with the governing RANS equations described in Appendix B is presented. The standard model is described first. However, the model can experience numerical issues that arise due to under-resolved grids or non-physical transients and so the negative version of the Spalart-Allmaras, which can overcome these issues, is introduced.

C.1 Standard Spalart-Allmaras one-equation model

In order to solve equations (B.39), (B.40) and (B.41) an evaluation of the eddy viscosity $\mu_t$ (or $\nu_t$) is required. The one-equation Spalart-Allmaras turbulence model aims to solve for this new variable through the definition of the conserved quantity $\overline{\nu}$ (or $\overline{\tau}$). The original equation derived by Spalart-Allmaras [129] for the eddy viscosity $\overline{\nu}$ is

$$\frac{\partial \overline{\nu}}{\partial t} + u_j \frac{\partial \overline{\nu}}{\partial x_j} = c_{b1} (1 - f_{t2}) \left( \left| \Omega \right| + \frac{\overline{\nu}}{\kappa^2 d^2} f_{t2} \right) \overline{\nu} = \left( c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2} \right) \left( \frac{\overline{\nu}}{d} \right)^2$$

$$+ \frac{1}{\sigma_{SA}} \left\{ \frac{\partial}{\partial x_j} \left[ (\nu + \overline{\nu}) \frac{\partial \overline{\nu}}{\partial x_j} \right] + c_{b2} \left( \frac{\partial \overline{\nu}}{\partial x_j} \right)^2 \right\} + f_{t1} \left( \frac{\partial^2 u_j}{\partial x_j^2} \right)^2$$

The ultimate goal here is to solve for the eddy viscosity $\mu_t$ (or $\nu_t$), used in equations (B.40) and (B.41), which
can be obtained from $\mu$ (or $\nu$) as follows

\[ \mu_t = \mu f_{v1} \]
\[ \nu_t = \nu f_{v1} \] (C.2)

Function $f_{v1}$ can be expressed as

\[ f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}} \] (C.3)

where $\chi$ is the ratio

\[ \chi = \frac{\mu}{\nu} = \frac{\nu}{\mu} \] (C.4)

and $c_{v1} = 7.1$ is a constant parameter. The production term contains the modified vorticity $\Omega$ given by

\[ \Omega = |\Omega| + \hat{\Omega} \] (C.5)

where the magnitude of vorticity $|\Omega|$ and function $\hat{\Omega}$ are

\[ |\Omega| = \sqrt{2\Omega_{ij}\Omega_{ij}} \] (C.6)

\[ \hat{\Omega} = \frac{\nu}{\kappa^2 d^4} f_{v2} \] (C.7)

The von Karman constant is $\kappa = 0.41$ and $d$ is the distance to the wall or slip line (in the wake). The vorticity may be worked out as

\[ \Omega_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right) \] (C.8)
and function $f_{v2}$ is defined as

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}$$  \hspace{1cm} (C.9)$$

Moreover, function $f_w$ in the destruction term is defined as

$$f_w = \varsigma(\bar{r}) \left[ 1 + \frac{c_{w3}^6}{\varsigma(\bar{r})^b + c_{w3}^6} \right]^{\frac{1}{6}}$$  \hspace{1cm} (C.10)$$

where $c_{w3} = 2$ is a constant and function $\varsigma(\bar{r})$ is given by

$$\varsigma(\bar{r}) = \bar{r} + c_{w2} (\bar{r}^6 - \bar{r})$$  \hspace{1cm} (C.11)$$

$$\bar{r} = \frac{\nu}{\Omega^2 \kappa d^2}$$  \hspace{1cm} (C.12)$$

with the constant value $c_{w2} = 0.3$. The trip term $f_{t1} \left( \frac{\partial^2 u_j}{\partial x_j^2} \right)^2$ is used to control the transition from laminar to turbulent flow. It is therefore common to neglect this term if the model is used for fully turbulent simulations. This is the case in the present work where all calculations are assumed to have a fully developed turbulent flow. In the event that the trip term cannot be neglected, function $f_{t1}$ is defined as

$$f_{t1} = c_{t1} g_t \exp \left( -c_{t2} \frac{\omega_t^2}{\Delta U^2} \left[ d^2 + g_t^2 \Delta x^2 \right] \right)$$  \hspace{1cm} (C.13)$$

where $d_t$ is the distance to the trip position, $\Delta U = U - U_t$ is the difference in velocities between the field point $(U)$ and the trip $(U_t)$, $\omega_t$ is the wall vorticity at the trip and $c_{t1} = 1$ and $c_{t2} = 2$ are constants. The value of $g_t$ is

$$g_t = \min \left( 0.1, \frac{\Delta U}{\omega_t \Delta x} \right)$$  \hspace{1cm} (C.14)$$

where $\Delta x$ is the grid spacing along the wall at the trip. Likewise, the trip function $f_{t2}$ used to provide stability to the solution $\nu = 0$ is given by

$$f_{t2} = c_{t3} \exp \left( -c_{t4} \chi^2 \right)$$  \hspace{1cm} (C.15)$$
where \( c_{t3} = 1.1 \) and \( c_{t4} = 2 \) are constant values. Unlike the trip term which can be neglected for fully turbulent applications, it is recommended to keep terms involving \( f_{t2} \). In order to ensure stability of the solution the following conditions are enforced \[ \Omega > 0 \quad \Omega \geq 0.3|\Omega| \quad \chi > 0 \quad \hat{r} \leq 10 \] (C.16)

The following values have been used for the constants throughout the Spalart-Allmaras model

| \( c_{b1} = 0.1355 \) | \( c_{b2} = 0.622 \) | \( \sigma_{SA} = \frac{2}{3} \) | \( c_{w1} = \frac{c_{b1}}{\hat{\kappa}^2} + \frac{1 + c_{b2}}{\sigma_{SA}} \) |
| \( c_{w2} = 0.3 \) | \( c_{w3} = 2 \) | \( \hat{\kappa} = 0.41 \) | \( c_{v1} = 7.1 \) |
| \( c_{v2} = 0.7 \) | \( c_{v3} = 0.9 \) | \( c_{t1} = 1 \) | \( c_{t2} = 2 \) |
| \( c_{t3} = 1.1 \) | \( c_{t4} = 2 \) | \( c_{n1} = 16 \) |

Table C.1: Constant values used in Spalart-Allmaras one equation turbulence model

C.1.1 An alternative definition of the modified vorticity \( \bar{\Omega} \)

In a physically meaningful solution, i.e. for \( \nu \geq 0 \), the modified vorticity \( \bar{\Omega} \) should always be greater than zero, as understood by looking at its definition in equation (C.5). A more realistic condition would set the limit on \( \bar{\Omega} \geq 0.3|\Omega| \) as suggested in \[99\]. However in the numerical solution of the discrete Spalart-Allmaras model a negative value can arise for certain values of \( \chi \). Allmaras \[99\] presents a modified \( \bar{\Omega} \) which preserves the original definition for values \( \bar{\Omega} \geq 0.3|\Omega| \) and remains non-negative otherwise

\[
\bar{\Omega} = \begin{cases} 
|\Omega| + \hat{\Omega} & \text{if } \hat{\Omega} \geq -c_{v2}|\Omega| \\
|\Omega| + \frac{|\Omega| (c_{v2}^2|\Omega| + c_{v3}\hat{\Omega})}{(c_{v3} - 2c_{v2})|\Omega| - \hat{\Omega}} & \text{if } \hat{\Omega} < -c_{v2}|\Omega|
\end{cases}
\] (C.17)

where the constants are \( c_{v2} = 0.7 \) and \( c_{v3} = 0.9 \). Notice that this alternative definition of \( \bar{\Omega} \) is also \( C^1 \) continuous, i.e. it is continuous and differentiable at least once. A comparison between the standard and alternative definitions of the modified vorticity \( \bar{\Omega} \), as found in reference \[99\], has been plotted in Figure C.1.
C.1. Standard Spalart-Allmaras one-equation model

\[ \hat{\Omega} |_{\Omega} \]

Figure C.1: Comparison between the standard and alternative definitions of the modified vorticity \( \Omega \), equations (C.5) and (C.17). As proposed by Allmaras et al. [99]

Table C.2: Non-dimensionalisation used for turbulent eddy-viscosity

\[ \mu^* = \frac{\mu}{\mu_\infty} \quad \nu^* = \frac{\nu}{\mu_\infty / \rho_\infty} \]

becomes clear that the alternative definition limits the values to \( \Omega \geq 0.1 |\Omega| \) as given by the asymptote

\[ \frac{\Omega}{|\Omega|} \to 0.1 \quad \text{when} \quad \frac{\hat{\Omega}}{|\Omega|} \to -\infty \quad \text{(C.18)} \]

C.1.2 Non-dimensionalisation of Spalart-Allmaras model without trip term

The first assumption made here is that the flow is fully turbulent, i.e. no trip term need to be considered

\[ f_{11} \left( \frac{\partial^2 u_j}{\partial x_j} \right)^2 = 0 \quad \text{(C.19)} \]

The terms involving \( f_{12} \) can be left unchanged as they ensure stability of the solution \( \bar{\mu} = \bar{\nu} = 0 \) which may arise during the transient that drives to the final solution. Moreover, the discretisation of term \( \left( \frac{\partial \bar{\nu}}{\partial x_j} \right)^2 \) does not easily ensure the achievement of positive turbulence fields for all transient solution states [129]. Using the
Following identity

$$\frac{∂}{∂x_j} [(ν + \nu) \frac{∂\nu}{∂x_j}] = \frac{∂ν}{∂x_j} \frac{∂\nu}{∂x_j} + (\nu + \nu) \frac{∂}{∂x_j} \left( \frac{∂\nu}{∂x_j} \right)$$ \hspace{1cm} (C.20)

and taking some liberties with differentiation of molecular viscosity $ν$, i.e. neglecting term $\frac{∂ν}{∂x_j} \frac{∂\nu}{∂x_j}$ as recommended by Spalart-Allmaras \[129\], diffusion in (C.1) can be written as

$$\frac{1}{σSA} \left\{ \frac{∂}{∂x_j} [(ν + \nu) \frac{∂\nu}{∂x_j}] + c_{b2} \left( \frac{∂\nu}{∂x_j} \right)^2 \right\} = \frac{1 + c_{b2}}{σSA} \frac{∂}{∂x_j} [(ν + \nu) \frac{∂\nu}{∂x_j}] - c_{b2} (ν + \nu) \frac{∂}{∂x_j} \left( \frac{∂\nu}{∂x_j} \right)$$ \hspace{1cm} (C.21)

Multiplying equation \( \text{(C.1)} \) by the density $ρ$, using equations \( \text{(C.19)} \) and \( \text{(C.21)} \), using non-dimensionalisation in Tables \[ B.1 \] and \[ C.2 \] and re-arranging terms, yields

$$\frac{\partial \hat{\Omega}}{\partial t} + \frac{∂ (\hat{\Omega}u_j)}{∂x_j} = \hat{\Omega} \left( 1 - f_{t2} \right) \left( |Ω| + \frac{1}{Re_\infty} \frac{ν}{κ^2 d^2 f_{v2}} \right) \hat{Ω} - \frac{1}{Re_\infty} \left( c_{w1} f_w - \frac{c_{b1}}{κ^2} f_{t2} \right) \left( \frac{∂\nu}{∂x_j} \right)$$

$$+ \frac{1}{σSA Re_\infty} \left\{ (1 + c_{b2}) \frac{∂}{∂x_j} [(μ + μ) \frac{∂\nu}{∂x_j}] - c_{b2} (μ + μ) \frac{∂}{∂x_j} \left( \frac{∂\nu}{∂x_j} \right) - (1 + c_{b2}) (ν + ν) \frac{∂}{∂x_j} \frac{∂ρ}{∂x_j} \right\}$$ \hspace{1cm} (C.22)

Notice that the superscript $^*$ denoting non-dimensional variables has been dropped for the sake of simplicity.

The non-dimensional form of function $\hat{Ω}$ and variable $\hat{r}$ are now as follows

$$\hat{Ω} = \frac{1}{Re_\infty} \frac{ν}{κ^2 d^2 f_{v2}}$$ \hspace{1cm} (C.23)

$$\hat{r} = \frac{ν}{Re_\infty Ωκ^2 d^2}$$ \hspace{1cm} (C.24)

**C.2 Negative Spalart-Allmaras one-equation model without trip term**

Although the original (standard) Spalart-Allmaras one-equation model \[129\] admits only positive solutions the discrete form can sometimes yield negative solutions, particularly in the case of non-physical transient states.
or under-resolved grids. In these situations, the values of turbulence inside boundary layers and wakes can be dramatically higher than those at the outer freestream yielding sharp changes and undershoots [99]. Attempts to overcome these numerical issues as a consequence of the discretisation include the numerical method proposed by Lorin et al. [130], based on a mixed finite element/finite volume solver; the introduction of an artificial viscosity by Nguyen et al. [131], active only in areas where the grid is too coarse such as the edge of boundary layers; and the modification of the Spalart-Allmaras model itself formulated by Moro et al. [132], which replaces the working variable \( \nu \) in regions where the eddy viscosity is smaller than the molecular viscosity, i.e. \( \nu \leq \nu \).

Finally, a pure continuation of the Spalart-Allmaras model for negative values of \( \nu \) is given in [99]. While preserving the original model for \( \nu \geq 0 \) the negative Spalart-Allmaras deals successfully with values of \( \nu < 0 \) and is energy stable. The form of the negative Spalart-Allmaras model [99] without trip term is

\[
\frac{\partial \nu}{\partial t} + u_j \frac{\partial \nu}{\partial x_j} = P_n - D_n + \frac{1}{\sigma_{SA}} \left( \frac{\partial}{\partial x_j} \left( (\nu + \nu) \frac{\partial \nu}{\partial x_j} \right) + c_b \left( \frac{\partial \nu}{\partial x_j} \right)^2 \right)
\]  

(C.25)

where \( P_n \) is the production term, \( D_n \) is the destruction and the function \( f_n (\chi) \) modifies the diffusion. Notice here that equation (C.25) is identical to the standard model, equation (C.1), in case

\[
f_n = \begin{cases} 
1 & \forall \chi \geq 0 \\
\end{cases}
\]  

(C.26)

\[
P_n = c_b (1 - f_{t2}) \left( |\omega| + \frac{\nu \nabla^2 f_{t2}}{\nu^2} \right) \nu
\]  

(C.27)

\[
D_n = \left( c_w f_w - \frac{c_b}{\nu^2} f_{t2} \right) \left( \frac{\nu}{\eta} \right)^2
\]  

(C.28)

Taking some special requirements to ensure \( C^4 \) continuity of this negative Spalart-Allmaras model at \( \nu = 0 \) and other requirements to guarantee energy stability, by forcing the integrated energy to always decrease in time, yields the definition of function \( f_n (\chi) \) as

\[
f_n = \frac{c_{n1} + \chi^3}{c_{n1} - \chi^3}
\]  

(C.29)

where \( c_{n1} = 16 \) has been chosen to maximize the region over which the diffusion coefficient turns from \( \nu + \nu \) for positive to \( \nu + |\nu| \) for negative [99]. Moreover, in order to accomplish the constraints above, the production and destruction terms in equation (C.25) have been defined as
$P_n = c_{b1} \left(1 - c_{ct} \right) |\Omega| \overline{\nu}$  \hspace{1cm} (C.30)

$D_n = -c_{w1} \left( \overline{\nu} \right)^2$  \hspace{1cm} (C.31)

where $|\Omega|$ is now the actual vorticity as defined by equation (C.8).

C.2.1 Non-dimensionalisation of negative Spalart-Allmaras model without trip term

The non-dimensionalisation of the negative Spalart-Allmaras model is obtained using a similar approach as in Section C.1.2. As before, the discretisation of term $\left( \frac{\partial \overline{\nu}}{\partial x_j} \right)^2$ does not easily ensure the achievement of positive turbulence fields for all transient solution states [129]. An identity equivalent to (C.20) for the negative model can be written as

$$\frac{\partial}{\partial x_j} \left[ (\nu + \nu f_n) \frac{\partial \overline{\nu}}{\partial x_j} \right] = \frac{\partial \nu}{\partial x_j} \frac{\partial \overline{\nu}}{\partial x_j} + f_n \left( \frac{\partial \overline{\nu}}{\partial x_j} \right)^2 + \overline{\nu} \frac{\partial \nu}{\partial x_j} \frac{\partial f_n}{\partial x_j} + (\nu + \nu f_n) \frac{\partial \overline{\nu}}{\partial x_j} \frac{\partial \overline{\nu}}{\partial x_j} \hspace{1cm} (C.32)$$

and, similarly, taking some liberties with differentiation of molecular viscosity $\nu$, i.e. neglecting term $\frac{\partial \nu}{\partial x_j} \frac{\partial \overline{\nu}}{\partial x_j}$ as recommended by Spalart-Allmaras [129], diffusion in (C.25) can be written as

$$\frac{1}{\sigma_{SA}} \left\{ \frac{\partial}{\partial x_j} \left[ (\nu + \nu f_n) \frac{\partial \overline{\nu}}{\partial x_j} \right] \right\} = f_n + c_{b2} \frac{\sigma_{SA}}{f_n} \frac{\partial}{\partial x_j} \left[ (\nu + \nu f_n) \frac{\partial \overline{\nu}}{\partial x_j} \right]$$

$$= \frac{c_{b2}}{\sigma_{SA} f_n} \left[ (\nu + \nu f_n) \frac{\partial \overline{\nu}}{\partial x_j} \right] + \overline{\nu} \frac{\partial \nu}{\partial x_j} \frac{\partial f_n}{\partial x_j} \hspace{1cm} (C.33)$$

Finally, multiplying by the density $\rho$, using non-dimensionalisation in Tables B.1 and C.2 and re-arranging
terms, yields

\[
\frac{\partial \mu}{\partial t} + \frac{\partial (\mu u_j)}{\partial x_j} = c_{b_1} (1 - c_{t_3}) |\Omega| \mu + \frac{c_{d_1}}{R \infty} \left( \frac{\partial p}{\partial t} \right) \\
+ \frac{1}{\sigma_{SA} R \infty} \left\{ \frac{\partial}{\partial x_j} \left[ \left( 1 + \frac{c_{b_2}}{f_n} \right) (\mu + \overline{p}) \frac{\partial \overline{\nu}}{\partial x_j} \right] - \frac{c_{b_2}}{f_n} (\mu + \overline{p}) \frac{\partial \overline{\nu}}{\partial x_j} \left( \frac{\partial \overline{\nu}}{\partial x_j} \right) \right\} \\
diffusion \]

\[
- c_{b_2} \left[ \mu + \left( f_n + \frac{1}{f_n} \right) \overline{p} \right] \frac{\partial \overline{\nu}}{\partial x_j} \frac{\partial f_n}{\partial x_j} - \left( 1 + \frac{c_{b_2}}{f_n} \right) (\nu + p f_n) \frac{\partial \overline{\nu}}{\partial x_j} \frac{\partial \rho}{\partial x_j} \right\}
\]

\text{(C.34)}

Notice the use of the magnitude of vorticity $|\Omega|$ instead of the modified vorticity $\overline{\Omega}$ and the ‘+’ sign in the destruction term.
Appendix D

Dissipation in upwind schemes

Unlike the central-difference scheme, there is no need for an artificial dissipation term in upwind biased schemes, as explained below. Consider for instance the one-dimensional case of the Euler equations (B.59), (B.60) and (B.61) written in matrix form

\[
\frac{\partial U}{\partial t} + A(U) \frac{\partial U}{\partial x} = 0
\]  \hspace{1cm} \text{(D.1)}

where matrix \( A = \frac{\partial F}{\partial U} \) comes from the linearisation of the fluxes \( F(U) \) given by

\[
F = \begin{bmatrix}
\rho u \\
\rho u^2 + p \\
(\rho E + p) u
\end{bmatrix}
\]  \hspace{1cm} \text{(D.2)}

where the column vector of conserved variables \( U \) is

\[
U = \begin{bmatrix}
\rho \\
\rho u \\
\rho E
\end{bmatrix}
\]  \hspace{1cm} \text{(D.3)}

Solving equation

\[
\det (A - \lambda I_3)
\]  \hspace{1cm} \text{(D.4)}
where $I_3$ is the identity matrix of size $3 \times 3$, yields the eigenvalues $\lambda_i$ and right eigenvectors $V_i$ of matrix $A$. Therefore, the diagonalisation of matrix $A$ is given by

$$D = T^{-1}AT$$  \hspace{1cm} (D.5)$$

where diagonal matrix $D$ is

$$D = \begin{bmatrix} 
\lambda_1 & 0 & 0 \\ 
0 & \lambda_2 & 0 \\ 
0 & 0 & \lambda_3 
\end{bmatrix}$$  \hspace{1cm} (D.6)$$

and the transformation matrix $T$ is defined as

$$T = \frac{\partial U}{\partial V} = \begin{bmatrix} 
\{V_1\} \\ 
\{V_2\} \\ 
\{V_3\} 
\end{bmatrix}$$  \hspace{1cm} (D.7)$$

where the column vectors $V_i$ are the right eigenvectors of $A$. Using equation (D.7) the linearised Euler equations (D.1) may be cast to a system of three non-coupled equations

$$\frac{\partial V}{\partial t} + D(V) \frac{\partial V}{\partial x} = 0$$  \hspace{1cm} (D.8)$$

or, solving one equation at a time,

$$\frac{\partial v}{\partial t} + \lambda(v) \frac{\partial v}{\partial x} = 0$$  \hspace{1cm} (D.9)$$

This is equivalent to solving the one-dimensional scalar wave equation three times, once for continuity, once for momentum and another one for energy. Discretising each of them via a first-order upwind scheme yields

$$v_{j+1}^{n+1} = v_j^n - \lambda_{j+\frac{1}{2}}^n \frac{\Delta t}{\Delta x} \begin{cases} v_{j+1}^n & \text{if } \lambda_{j+\frac{1}{2}}^n < 0 \\ v_j^n & \text{if } \lambda_{j+\frac{1}{2}}^n > 0 \end{cases} + \lambda_{j-\frac{1}{2}}^n \frac{\Delta t}{\Delta x} \begin{cases} v_j^n & \text{if } \lambda_{j-\frac{1}{2}}^n < 0 \\ v_{j-1}^n & \text{if } \lambda_{j-\frac{1}{2}}^n > 0 \end{cases}$$  \hspace{1cm} (D.10)$$
which can be cast to contain both first and second-order central differences (see [77]) as

\[
v^n_{j+1} = v^n_j - \frac{\Delta t}{\Delta x} \left[ \lambda^n_{j+\frac{1}{2}} \left( \frac{v^n_{j+1} + v^n_{j+1}}{2} \right) - \lambda^n_{j-\frac{1}{2}} \left( \frac{v^n_j + v^n_{j-1}}{2} \right) \right] + \frac{\Delta t}{\Delta x} \left[ \lambda^n_{j+\frac{1}{2}} \left( \frac{v^n_{j+1} - v^n_j}{2} \right) - \lambda^n_{j-\frac{1}{2}} \left( \frac{v^n_j - v^n_{j-1}}{2} \right) \right]
\]

(D.11)

If there are \( N \) cells in the mesh, the discrete problem consists of a system of \( N \) equations like (D.11), one for each cell in the mesh, which may be written in matrix form as follows

\[
V^{n+1}_0 = V^n_0 - \frac{\Delta t}{\Delta x} \left[ A_+ \left( \frac{V^n_0 + V^n_0}{2} \right) - A_- \left( \frac{V^n_0 + V^n_0}{2} \right) \right] + \frac{\Delta t}{\Delta x} \left[ A_+ \left( \frac{V^n_0 - V^n_0}{2} \right) - A_- \left( \frac{V^n_0 - V^n_0}{2} \right) \right]
\]

(D.12)

where matrices \( \begin{bmatrix} A_+ \end{bmatrix} \) and \( \begin{bmatrix} A_- \end{bmatrix} \), of size \( N \times N \) (do not confuse with the determinant of \( A_+ \) and \( A_- \), which are scalars), are defined as

\[
\begin{bmatrix} A_+ \end{bmatrix} = \begin{bmatrix} \lambda_{1+\frac{1}{2}} & 0 & \cdots & 0 & 0 \\ 0 & \lambda_{2+\frac{1}{2}} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda_{N+\frac{1}{2}} & 0 \\ \end{bmatrix}
\]

\[
\begin{bmatrix} A_- \end{bmatrix} = \begin{bmatrix} \lambda_{1-\frac{1}{2}} & 0 & \cdots & 0 & 0 \\ 0 & \lambda_{2-\frac{1}{2}} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda_{N-\frac{1}{2}} & 0 \\ \end{bmatrix}
\]

(D.13)

and column vectors \( V^n_0 \), \( V^n \) and \( V^n_+ \) are

\[
V^n_0 = \begin{bmatrix} v^n_1 \\ v^n_2 \\ \vdots \\ v^n_{N-1} \\ v^n_N \end{bmatrix}, \quad V^n = \begin{bmatrix} a^n \\ v^n_1 \\ \vdots \\ v^n_{N-2} \\ v^n_{N-1} \end{bmatrix}, \quad V^n_+ = \begin{bmatrix} v^n_2 \\ v^n_3 \\ \vdots \\ v^n_N \\ b^n \end{bmatrix}
\]

(D.14)

where the value of \( a^n \) and \( b^n \) depends on the time-dependent boundary conditions at both sides of the one-dimensional domain. Looking at equation (D.12) it becomes clear that the upwind formulation introduces a second-difference numerical dissipation term on top of the central-difference term, hence there is no need for the addition of an extra artificial dissipation term.

This formulation can be further simplified in the case of a constant coefficients matrix \( A \) in the conservation
law in equation (D.1). Diagonalising via (D.5), the system of equations can be written as a system of three independent equations of the form

\[
\frac{\partial v}{\partial t} + \lambda \frac{\partial v}{\partial x} = 0 \quad (D.15)
\]

where the values of \( \lambda \) are now constant across the spatial domain. Discretising again via a first-order upwind scheme yields

\[
v^{n+1}_j = v^n_j - \lambda \frac{\Delta t}{\Delta x} \begin{cases} 
  v^n_{j+1} - v^n_j, & \lambda < 0 \\
  v^n_j - v^n_{j-1}, & \lambda > 0 
\end{cases}
\]

or

\[
v^{n+1}_j = v^n_j - \lambda \frac{\Delta t}{2\Delta x} (v^n_{j+1} - v^n_j) + |\lambda| \frac{\Delta t}{2\Delta x} (v^n_{j+1} - 2v^n_j + v^n_{j-1}) \quad (D.17)
\]

Writing the system of \( N \) equations, one for each cell in the mesh, yields, in matrix form,

\[
V^{n+1}_0 = V^n_0 - \lambda \frac{\Delta t}{2\Delta x} (V^n_+ - V^n_-) + |\lambda| \frac{\Delta t}{2\Delta x} (V^n_+ - 2V^n_0 + V^n_-) \quad (D.18)
\]

where column vectors \( V^n_0, V^n_+ \) and \( V^n_- \) are defined in (D.14) and the value of \( a^n \) and \( b^n \) depends on the time-dependent boundary conditions at both sides of the one-dimensional domain.
Appendix E

Gradients calculation

E.1 Spatial gradients

The calculation of gradients in a finite-volume discretisation may be done via the Green-Gauss theorem, also known as the theorem of divergence, which states that

$$\int_V \nabla \phi(x, t) \, d\Omega = \oint_{\partial V} \phi(x, t) \cdot n \, dS \quad (E.1)$$

where \(V\) is the volume of the cell where the gradient \(\nabla \phi\) is to be calculated, \(\partial V\) is the closed boundary of volume \(V\) and \(n = [n_x, n_y]^T\) is the normal vector at the boundary. Notice that in the case of a finite-volume discretisation, where there is only one integration point per cell, \(x_0\), the gradient \(\nabla \phi\) may be taken out of the integral yielding

$$\nabla \phi(x_0, t) = \frac{1}{V} \oint_{\partial V} \phi(x, t) \cdot n \, dS \quad (E.2)$$

Discretising equation (E.2) yields

$$\nabla \phi(x_0, t) = \frac{1}{V} \sum_{i=1}^{n_f} \phi(x_i, t) \cdot n_i \, \Delta S_i \quad (E.3)$$

where \(n_f\) is the number of faces at the cell and \(x_i, n_i, \Delta S_i\) are the centre, the normal vector and the area of face \(i\), respectively. Notice that time \(t\) is just a parameter in the above equations.
E.2 Spacetime gradients

An expression similar to (E.3) for the calculation of spacetime gradients in the spacetime framework can be derived applying the Green-Gauss theorem in spacetime. For a finite-volume discretisation yields

$$\nabla_{ST} \phi(x_0, t_0) = \frac{1}{V_{ST}} \sum_{i=1}^{n_f} \phi(x_i, t_i) \cdot n_{ST,i} \Delta S_{ST,i}$$  \hspace{1cm} (E.4)

where $V_{ST}$ is the spacetime volume of the cell where the spacetime gradient $\nabla_{ST} \phi$ is to be calculated, $(x_0, t_0)$ is the spacetime location of the cell centre, $(x_i, t_i)$ is the spacetime location of the centre of face $i$, $n_{ST,i} = [n_t, n_x, n_y]^T$ is the spacetime normal vector at face $i$ and $\Delta S_{ST,i}$ is the face area. Here, gradient $\nabla_{ST} \phi(x_0, t_0)$ is a spacetime gradient and it has components in space and time.

E.3 Spatial gradients in spacetime

Spatial gradients are required in the calculation of viscous terms and in the Spalart-Allmaras turbulence model even in the case of the spacetime formulation. This is a straightforward process and it is done substituting the spacetime normal vector $n_{ST}$ with the spatial normal vector $n = n_{ST} - (n_{ST} \cdot e_t) e_t$ in equation (E.4). Notice that this approach is equivalent to working out the spacetime gradients through (E.4) and then neglecting their component in the time $t$ direction.
Appendix F

Backward second-difference

The derivation of the backward second-difference of a function $f(t)$ for a variable step size $\Delta t^n$ is given below. Assume $f$ is $C^n$, i.e. the function and its first $n$ derivatives are continuous and differentiable. Then, consider its Taylor polynomial

$$f(t) = f(t_0) + \frac{f'(t_0)}{1!} (t-t_0) + \frac{f''(t_0)}{2!} (t-t_0)^2 + \frac{f'''(t_0)}{3!} (t-t_0)^3 + \cdots + \frac{f^{(n)}(t_0)}{n!} (t-t_0)^n + O([t-t_0]^n) \quad (F.1)$$

Centering (F.1) at $t_0 = t^n$ corresponding to the $n$-th timestep, the approximation of $f(t)$ at $t = t^{n-1}$ and $t = t^{n-2}$ yields

$$f(t^{n-1}) = f(t^n) - f'(t^n) (t^n - t^{n-1}) + \frac{f''(t^n)}{2} (t^n - t^{n-1})^2 + O \left( [t^n - t^{n-1}]^2 \right) \quad (F.2)$$

$$f(t^{n-2}) = f(t^n) - f'(t^n) (t^n - t^{n-2}) + \frac{f''(t^n)}{2} (t^n - t^{n-2})^2 + O \left( [t^n - t^{n-2}]^2 \right) \quad (F.3)$$

Subtracting $\frac{1}{(t^n - t^{n-2})^2} \times [F.3]$ from $\frac{1}{(t^n - t^{n-1})^2} \times [F.2]$ and neglecting higher-order terms, one can write

$$\frac{f(t^{n-1})}{(t^n - t^{n-1})^2} - \frac{f(t^{n-2})}{(t^n - t^{n-2})^2} = f(t^n) \left[ \frac{1}{(t^n - t^{n-1})^2} - \frac{1}{(t^n - t^{n-2})^2} \right] - f'(t^n) \left( \frac{1}{t^n - t^{n-1}} - \frac{1}{t^n - t^{n-2}} \right) \quad (F.4)$$
Appendix F: Backward second-difference

Notice that

\[
\left( \frac{1}{t^n - t^{n-1}} - \frac{1}{t^n - t^{n-2}} \right) = \frac{t^{n-1} - t^{n-2}}{(t^n - t^{n-1})(t^n - t^{n-2})}
\]  
(F.5)

Multiplying equation (F.4) by \( \frac{(t^n - t^{n-1})(t^n - t^{n-2})}{t^{n-1} - t^{n-2}} \) and re-arranging gives

\[
f'(t^n) = f(t^n) \left[ \frac{t^n - t^{n-2}}{(t^n - t^{n-1})(t^n - t^{n-2})} - \frac{t^n - t^{n-1}}{(t^n - t^{n-2})(t^{n-1} - t^{n-2})} \right] \\
- f(t^{n-1}) \frac{t^n - t^{n-2}}{(t^n - t^{n-1})(t^{n-1} - t^{n-2})} + f(t^{n-2}) \frac{t^n - t^{n-1}}{(t^n - t^{n-2})(t^{n-1} - t^{n-2})}
\]  
(F.6)

Finally, using

\[
t^n - t^{n-1} = (t^n - t^{n-2}) - (t^{n-1} - t^{n-2})
\]  
(F.7)

\[
t^n - t^{n-2} = (t^n - t^{n-1}) - (t^{n-2} - t^{n-1})
\]  
(F.8)

yields the backward second-difference of \( f(t) \) for variable step size \( \Delta t \) as follows

\[
f'(t^n) = f(t^n) \left( \frac{1}{t^n - t^{n-1}} + \frac{1}{t^n - t^{n-2}} \right) \\
- f(t^{n-1}) \left( \frac{1}{t^{n-1} - t^{n-2}} + \frac{1}{t^n - t^{n-1}} \right) + f(t^{n-2}) \left( \frac{1}{t^{n-1} - t^{n-2}} - \frac{1}{t^n - t^{n-2}} \right)
\]  
(F.9)

Bear in mind that this equation reduces to the well known formula for the backward second-difference with constant step size \( \Delta t \) in equation (F.11). In other words, imposing that

\[
\Delta t = t^n - t^{n-1} = t^{n-1} - t^{n-2} = \frac{t^n - t^{n-2}}{2}
\]  
(F.10)

it is possible to cast equation (F.9) as

\[
f'(t^n) = \frac{3f(t^n) - 4f(t^{n-1}) + f(t^{n-2})}{2\Delta t}
\]  
(F.11)
Appendix G

Full results for AGARD R-702(3E3) Case 1

The full set of $C_p$ distributions for the AGARD R-702(3E3) Case 1 are given below in Figures G.1 to G.6.

Figure G.1: $C_p$ distribution plots for pitching NACA 0012 with amplitude $\Delta \alpha = 2.41$ deg at $M_\infty = 0.6$, $\alpha_0 = 2.89$ deg and $k = \frac{\omega}{2M_\infty} = 0.0808$. Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 1. Continues in Figure G.2.
Appendix G: Full results for AGARD R-702(3E3) Case 1

\[ \omega t = 23.22 \text{ deg} \]

\[ \omega t = 33.78 \text{ deg} \]

\[ \omega t = 43.86 \text{ deg} \]

\[ \omega t = 53.61 \text{ deg} \]

\[ \omega t = 60.14 \text{ deg} \]

\[ \omega t = 67.10 \text{ deg} \]

Figure G.2: Continuation from Figure G.1. \( C_p \) distribution plots for pitching NACA 0012 with amplitude \( \Delta \alpha = 2.41 \) deg at \( M_\infty = 0.6 \), \( \alpha_0 = 2.89 \) deg and \( k = \frac{\omega}{2 \pi F} = 0.0808 \). Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 1. Continues in Figure G.3.
Figure G.3: Continuation from Figure G.2. $C_p$ distribution plots for pitching NACA 0012 with amplitude $\Delta \alpha = 2.41$ deg at $M_{\infty} = 0.6$, $\alpha_0 = 2.89$ deg and $k = \frac{\omega_c}{U_{\infty}} = 0.0808$. Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 1. Continues in Figure G.4.
Appendix G: Full results for AGARD R-702(3E3) Case 1

Figure G.4: Continuation from Figure G.3. $C_p$ distribution plots for pitching NACA 0012 with amplitude $\Delta \alpha = 2.41$ deg at $M_\infty = 0.6$, $\alpha_0 = 2.89$ deg and $k = \frac{\omega c}{U_\infty} = 0.0808$. Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 1. Continues in Figure G.5.
\[ \omega t = 225.88 \text{ deg} \]

\[ \omega t = 237.83 \text{ deg} \]

\[ \omega t = 250.37 \text{ deg} \]

\[ \omega t = 262.61 \text{ deg} \]

\[ \omega t = 270.0 \text{ deg} \]

\[ \omega t = 281.69 \text{ deg} \]

Figure G.5: Continuation from Figure G.4. \( C_p \) distribution plots for pitching NACA 0012 with amplitude \( \Delta \alpha = 2.41 \) deg at \( M_\infty = 0.6, \alpha_0 = 2.89 \) deg and \( k = \frac{\omega c}{U_\infty} = 0.0808 \). Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 1. Continues in Figure G.6.
Appendix G: Full results for AGARD R-702(3E3) Case 1

\[ \omega t = 294.67 \text{ deg} \]

\[ \omega t = 306.79 \text{ deg} \]

\[ \omega t = 318.40 \text{ deg} \]

\[ \omega t = 329.86 \text{ deg} \]

\[ \omega t = 341.37 \text{ deg} \]

\[ \omega t = 352.37 \text{ deg} \]

Figure G.6: Continuation from Figure G.5. \( C_p \) distribution plots for pitching NACA 0012 with amplitude \( \Delta \alpha = 2.41 \) deg at \( M_{\infty} = 0.6 \), \( \alpha_0 = 2.89 \) deg and \( \kappa = \frac{\omega t}{M_{\infty}} = 0.0808 \). Comparison of CFD results via spacetime solver against experimental data from AGARD R-702(3E3) Case 1.

---

218
Appendix H

Implementation details

The CFD solver developed in this work uses a finite-volume discretization of the two-dimensional Euler and Navier-Stokes equations for fluids motion in spacetime. There are four available stencils: a central-difference, two upwind (based on Van Leer and Roe) and a hybrid one (central in space, upwind in time). A spacetime face based mesh (only faces and nodal information are provided) is required as input along with the simulation parameters, which are specified as command line arguments. A flowchart of the simulation process for the spacetime framework developed in this work is given in Figure H.1.

Figure H.1: Workflow for the simulation of aerodynamics problems in the spacetime presented here.
Appendix H: Implementation details

The generation of a $2D+t$ spacetime mesh is done by means of a three-dimensional unstructured mesh generator where, in this case, the third dimension represents time $t$. The mesh is given by a text file in ASCII format and the information in it is arranged as follows:

\[
\begin{array}{ccc}
\text{numPoints} & \text{numFaces} \\
\text{x} & \text{y} & \text{t} \\
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
N & \text{p1} & \text{p2} & \ldots & \text{pN} & \text{leftCell} & \text{rightCell} & \text{boundMarker} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{array}
\]

where \text{numPoints} and \text{numFaces} are the total number of points and faces in the spacetime mesh; \text{x}, \text{y} and \text{t} are the spacetime coordinates for each point in the mesh; \text{N} is the number of vertices at each face and \text{p1}, \text{p2}, \ldots \text{pN} are the vertices; \text{leftCell} and \text{rightCell} are the left and right cell indices at each face and \text{boundMarker} is the boundary marker at each face. In the case of a viscous simulation the velocity of solid grid nodes needs to be provided as well. The velocities are given in a separate text file in ASCII format as follows:

\[
\begin{array}{cc}
\text{node} & \text{Vx} & \text{Vy} \\
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
\end{array}
\]

where \text{node} is the solid node index and \text{Vx} and \text{Vy} are components of the node’s velocity in the $x$ and $y$ directions.

The mesh and the simulation parameters (i.e. freestream flow conditions and solver settings) are given as inputs to the solver which, unless specified, applies the non-dimensionalisation defined by tables B.1 and C.2 in Appendices B and C. In a spacetime simulation an initial condition must be supplied on top of the boundary conditions unless the problem is periodic in time. The initial condition is given by a $2D$ solution which can be passed directly to the solver or calculated based on some initial conditions. In reality, the spacetime solver can be launched with a uniform flow as the initial solution but this is, in general, not a physically meaningful flow (unless in the absence of a solid geometry). If the initial solution needs to be calculated, a spacetime mesh is generated by extrusion of the initial $2D$ plane. The initial and final boundaries are connected by setting a time-periodic boundary condition on them. The initial solution is then calculated by the spacetime solver as a simple time-periodic problem where no boundary motions exist. Once the initial solution is obtained, it is saved to a file and the spacetime simulation starts.
The spacetime solution is obtained by solution of equation (2.86) by means of an explicit or implicit solver, just like any steady state simulation. Here, a four-stages fourth-order explicit Runge-Kutta integration scheme has been used to march the solution in pseudo-time until convergence is reached. A diagram of the updating mechanism at each pseudo-time step is given in Figure H.2. Bear in mind that the RANS solver uses the same code as the inviscid Euler solver with the addition of viscous terms. Moreover, an additional equation needs to be solved in the viscous case for the Spalart-Allmaras turbulence model. Diffusive terms are only calculated once and re-used throughout all stages in the Runge-Kutta iteration whereas convective terms are calculated at every stage. Also the time-step size is only calculated once at the beginning of the Runge-Kutta iteration. Notice that the implementation of the different stencils differs only at the calculation of fluxes. Therefore, most of the code has been re-used for all the different available combinations. At the end of the spacetime simulation, the solution is saved into a Tecplot® file. Also, a seeding file is created to be able to re-start the simulation later.

The post-processing of the results has been done using Tecplot® by slicing the spacetime solution into several time-steps, as desired.

A copy of the source code described here can be found at:

https://bitbucket.org/flamarique/spacetime/

Please be aware that this code is used only for research purposes and it may contain bugs or inconsistencies. Therefore, use only at your own risk.
Appendix H: Implementation details

solution at current pseudo-time step: \( U^n, \bar{\mu}^n \)

calculate gradients: \( \nabla \rho, \nabla u, \nabla v, \nabla p, \nabla T, \nabla \bar{\nu} \)

reconstruct face values: \( W^f = \frac{W^L + W^R}{2} \) or \( W^f = W^C + \nabla W^C \cdot \Phi \cdot (x^f - x^C) \)

augment face values \( \rightarrow \) calculate \( E, T, \mu, \mu_t \)

calculate inviscid face fluxes (central-difference, Van Leer or Roe): \( F_x, F_y, F_t \)

if stage = 1

calculate viscous face fluxes: \( F^v_x, F^v_y, F^v_t \)

calculate residuals for RANS equations: \( R(U, \nabla U) \)

if stage = 1

calculate pseudo-time step size: \( \Delta t^* \)

calculate volume contribution of the Spalart-Allmaras equation: \( Q \)

calculate surface contributions to diffusive term of the Spalart-Allmaras equation: \( H^d \)

calculate surface contributions to convective term of the Spalart-Allmaras equation: \( H^c \)

calculate residuals for SA equation: \( R_{SA} \)

calculate solution at stage \( s \): \( U^{(s)}, \bar{\mu}^{(s)} \)

calculate solution at halo cells through the boundary conditions

augment cell values \( \rightarrow \) calculate \( E, T, \mu, \mu_t \)

calculate solution at next pseudo-time step: \( U^{n+1}, \bar{\mu}^{n+1} \)

Figure H.2: Flowchart of the updating mechanism at each pseudo-time step.
List of References


[63] Ray Hixon. Method and system for the efficient calculation of unsteady processes on arbitrary space-time domains.


231
List of References


