<table>
<thead>
<tr>
<th>Màster</th>
<th>Master’s degree in Numerical Methods in Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Títol</td>
<td>Simulation of atmospheric effects with the DLR code TAU</td>
</tr>
<tr>
<td>Autor</td>
<td>Laia Alcaraz Capsada</td>
</tr>
<tr>
<td>Tutor</td>
<td>Dr. Ralf Heinrich i Dr. Roberto Flores</td>
</tr>
<tr>
<td>Intensificació</td>
<td>Aeronàutica, Dinàmica de Fluids Computational</td>
</tr>
<tr>
<td>Data</td>
<td>Juliol 2015</td>
</tr>
</tbody>
</table>
Simulation of atmospheric effects with the DLR code TAU

Author: Laia Alcaraz Capsada
Supervisors: Dr. Ralf Heinrich
Dr. Roberto Flores

A thesis submitted in fulfilment of the requirements for the Master’s degree in Numerical Methods in Engineering

July 2015
Declaration of Authorship

I, Laia Alcaraz Capsada, declare that this thesis titled, "Simulation of atmospheric effects with the DLR code TAU" and the work presented in it are my own. I confirm that:

- This work was done wholly while in candidature for a research degree at the Universitat Politècnica de Catalunya.

- Where I have consulted the published work of others, this is always clearly attributed.

- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.

- I have acknowledged all main sources of help.

- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Laia Alcaraz Capsada.
Barcelona, July 2015.
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Abstract

Escola Tècnica Superior d’Enginyers de Camins, Canals i Ports de Barcelona

Simulation of atmospheric effects with the DLR code TAU

by Laia Alcaraz Capsada

The Computational Fluid Dynamics code TAU, developed at the Deutsches Zentrum für Luft- und Raumfahrt (German Aerospace Center) is used to model encounters between atmospheric effects (wind gusts and aircraft wake vortices) and aircraft.

In the case of wind gusts, two different numerical methods have been used. The first one is the so-called Disturbance Velocity Approach, a simple method that captures the influence of a gust on an aircraft, but the influence of the aerodynamics of the aircraft on the gust properties is not captured. The second method is the Resolved Atmosphere Approach, which feeds a gust into the flow field by using an unsteady boundary condition. Compared to the Disturbance Velocity Approach, this method requires more computational effort, but it is more accurate: it captures the mutual interaction of the gust and the aircraft. In order to find the validity range for the Disturbance Velocity Approach, both methods are compared for two-dimensional as well as three-dimensional geometries. In the two-dimensional cases, there is an excellent agreement between the simple method and the highly accurate one for gust wavelengths greater than or equal to two reference chord lengths and an on-flow Mach number less than or equal to 0.78. In the three-dimensional cases and a vertical gust, an unexpected agreement between both methods is found for all the gusts tested. In the three-dimensional cases and a lateral gust, no validity range for the Disturbance Velocity Approach can be given, because the resolution of the grid should be improved.

Finally, the Disturbance Velocity Approach has been implemented into the TAU code to enable the simulation of wake vortices encounter problems. The correctness of the implementation has been verified through different test cases.

Key words: Computational Fluid Dynamics, wind gust, wake vortex, Disturbance Velocity Approach, Resolved Atmosphere Approach.
Acknowledgements

Above all, I would like to acknowledge Dr. Ralf Heinrich, my supervisor at the Deutsches Zentrum für Luft- und Raumfahrt (DLR), for giving me the opportunity to do my Master’s thesis there and for the economic support. I thank him for his continuous patience and dedication, for teaching me so many things and solving all my doubts. I would also like to acknowledge Dr. Roberto Flores, my supervisor at the Universitat Politècnica de Catalunya (UPC), for revising the text and giving me very useful advice.

I am also grateful to many people at the DLR. First of all, to Britta Ernst, who assisted me during my first days there and taught me how to use TAU. Since then, she has always been willing to help me and answer my questions, and in fact, she provided one of the unstructured grids of this thesis. In addition, I acknowledge Ana Manso Jaume, who solved me many doubts about the grid generation software. The help of Alexander Kuzolap and Markus Widhalm was very useful for some of the plots of the thesis. I also thank the former one for many advices related to Linux and LaTeX, and for the nice working atmosphere. Moreover, I acknowledge my friend David Azócar Paredes, for answering many doubts, giving me advice, and for all the funny moments shared together.

My stance at the DLR and in Braunschweig has been very pleasant thanks to a lot of people. Among others, I acknowledge Kağan Atç, Christoph Bosshard, Andreas Michler, Eduard Frick, Britta Roll, Ana Pal, Verena Kuhlemann and Andrei Merle. It has been very nice to have lunch with them, to learn about the German culture and to improve my German communication skills.

I am also grateful to the Barcelona staff from the Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE). In particular, to Dr. Jordi Pons-Prats and Dr. Antonia Larese de Tetto, who found the opportunity at the DLR and encouraged me to do my Master’s thesis there. In addition, I thank Lelia Zielonka, from the CIMNE secretariat, who has been really helpful with many administration details not only before moving to Braunschweig, but also during my whole stance there.

And last, but by no means least, I would like to thank my family, who has always supported me since the beginning of my studies, and who encouraged me to do my thesis abroad. Finally, I really acknowledge my boyfriend for his patience and his unconditional support.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Declaration of Authorship</td>
<td>i</td>
</tr>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>iii</td>
</tr>
<tr>
<td>Contents</td>
<td>iv</td>
</tr>
<tr>
<td>List of Figures</td>
<td>vi</td>
</tr>
<tr>
<td>List of Tables</td>
<td>ix</td>
</tr>
<tr>
<td>List of Symbols</td>
<td>xii</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation and objectives of the thesis</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Structure of the thesis</td>
<td>4</td>
</tr>
<tr>
<td>I Mathematical models</td>
<td>5</td>
</tr>
<tr>
<td>2 The DLR code TAU</td>
<td>6</td>
</tr>
<tr>
<td>2.1 The mathematical description of the flow</td>
<td>6</td>
</tr>
<tr>
<td>2.1.1 Reynolds Averaging and Favre (Mass) Averaging</td>
<td>9</td>
</tr>
<tr>
<td>2.1.2 The Favre- and Reynolds-Averaged Navier-Stokes equations</td>
<td>9</td>
</tr>
<tr>
<td>2.2 Spatial discretization</td>
<td>12</td>
</tr>
<tr>
<td>2.2.1 Primary grid and dual grid</td>
<td>12</td>
</tr>
<tr>
<td>2.2.2 Central scheme</td>
<td>14</td>
</tr>
<tr>
<td>2.2.3 Upwind scheme</td>
<td>15</td>
</tr>
<tr>
<td>2.3 Temporal discretization</td>
<td>15</td>
</tr>
<tr>
<td>2.3.1 Global time-stepping</td>
<td>16</td>
</tr>
<tr>
<td>2.3.2 Dual time-stepping (DTS)</td>
<td>17</td>
</tr>
<tr>
<td>2.4 Initial and Boundary conditions</td>
<td>20</td>
</tr>
<tr>
<td>2.4.1 Boundary conditions</td>
<td>20</td>
</tr>
<tr>
<td>2.4.2 Initial conditions</td>
<td>22</td>
</tr>
<tr>
<td>2.5 Acceleration techniques: Multigrid</td>
<td>22</td>
</tr>
<tr>
<td>2.5.1 Agglomeration multigrid</td>
<td>23</td>
</tr>
<tr>
<td>2.5.2 Full multigrid cycle</td>
<td>23</td>
</tr>
</tbody>
</table>
# Contents

3 The Chimera technique 25
  3.1 Introduction and motivation ........................................... 25
  3.2 The Chimera technique in the DLR-TAU code ....................... 27
    3.2.1 Node identification ............................................. 27
    3.2.2 Interpolation at the Chimera boundaries .................... 29

4 CFD modelling of atmospheric effects 33
  4.1 Characterization of atmospheric effects .......................... 33
    4.1.1 Atmospheric wind gusts ....................................... 33
    4.1.2 Aircraft wake vortex ......................................... 36
  4.2 Disturbance Velocity Approach (DVA) ............................. 40
    4.2.1 Description of the method .................................... 40
    4.2.2 Advantages and disadvantages ................................. 41
  4.3 Resolved Atmosphere Approach (RAA) ............................... 41
    4.3.1 Description of the method .................................... 41
    4.3.2 Advantages and disadvantages ................................. 42

II Numerical applications 44

5 Grid set-up 45
  5.1 Grid density study .................................................. 45
    5.1.1 Uniform grid .................................................. 46
    5.1.2 Overset grid .................................................. 47
    5.1.3 Conclusions .................................................... 50
  5.2 2D grids ............................................................ 51
    5.2.1 Grids for gusts encounters .................................... 51
    5.2.2 Grid for wake vortex encounters ............................... 53
  5.3 3D grids ............................................................ 54

6 Results and discussion 58
  6.1 Atmospheric wind gusts ............................................. 58
    6.1.1 2D computations .............................................. 58
    6.1.2 3D computations .............................................. 64
  6.2 Interaction with wake vortices .................................... 76

7 Conclusions 84
  7.1 Summary of the results ........................................... 84
  7.2 Future work ....................................................... 86

A Conservation laws for moving grids 88

B Conservation laws averaging 91

Bibliography 93
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Primary grid (black lines) and dual grid (red lines) around a NACA0012 airfoil.</td>
<td>13</td>
</tr>
<tr>
<td>2.2</td>
<td>Unsteady computation with dual time-stepping and Runge-Kutta.</td>
<td>19</td>
</tr>
<tr>
<td>2.3</td>
<td>Example of multigrid cycle.</td>
<td>23</td>
</tr>
<tr>
<td>2.4</td>
<td>Generation of coarse grids by the agglomeration multigrid method in 2D.</td>
<td>24</td>
</tr>
<tr>
<td>3.1</td>
<td>Grids for the Chimera technique.</td>
<td>28</td>
</tr>
<tr>
<td>3.2</td>
<td>Translation of $\Omega_{cg}$ and $\Omega_b$ with respect to $\Omega_{bg}$.</td>
<td>31</td>
</tr>
<tr>
<td>3.3</td>
<td>NACA0012 sample configuration with movable flap.</td>
<td>32</td>
</tr>
<tr>
<td>4.1</td>
<td>Wind data in flight and height directions. Taken from [18], modified.</td>
<td>34</td>
</tr>
<tr>
<td>4.2</td>
<td>&quot;1 − cos&quot; gust velocity profile (FAR Part 25.341).</td>
<td>35</td>
</tr>
<tr>
<td>4.3</td>
<td>Encounter between a SDM configuration and a &quot;1 − cos&quot; gust.</td>
<td>35</td>
</tr>
<tr>
<td>4.4</td>
<td>Possible vortex encounters and their consequences. Taken from [20].</td>
<td>36</td>
</tr>
<tr>
<td>4.5</td>
<td>Scheme of the tangential velocity and definition of vortex parameters.</td>
<td>37</td>
</tr>
<tr>
<td>4.6</td>
<td>Comparison of the tangential velocity $v_t$ predicted by the Lamb-Oseen and the Burnham-Hallock models for different values of the core radius $r_c$.</td>
<td>38</td>
</tr>
<tr>
<td>4.7</td>
<td>Up: $v$ contour field. Down: $w$ contour field.</td>
<td>39</td>
</tr>
<tr>
<td>4.8</td>
<td>Gust travelling relative to a NACA0012 airfoil.</td>
<td>40</td>
</tr>
<tr>
<td>4.9</td>
<td>Resolution of a lateral gust with the RAA. Boundaries where unsteady boundary conditions are applied (left and bottom) are dashed.</td>
<td>42</td>
</tr>
<tr>
<td>5.1</td>
<td>Geodesic ($x_g, y_g, z_g$), TAU ($x, y, z$) and aircraft fixed ($x_b, y_b, z_b$) coordinate systems for $t = 0$ and $t &gt; 0$. Taken from [13].</td>
<td>46</td>
</tr>
<tr>
<td>5.2</td>
<td>Comparison between the analytical gust profile and the numerical one, using a uniform Cartesian grid with 100 cells per $\lambda$ in the direction of movement.</td>
<td>47</td>
</tr>
<tr>
<td>5.3</td>
<td>Overset grid for the grid density study. $\Omega_{bg}$ is shown in blue, while $\Omega_{tg}$, which is moving, is plotted in red. A vertical &quot;1 − cos&quot; gust (in yellow) moves together with $\Omega_{tg}$.</td>
<td>48</td>
</tr>
<tr>
<td>5.4</td>
<td>Comparison between the analytical gust profile and the numerical one, using three overset grids with different resolutions in the $x_g$−direction.</td>
<td>50</td>
</tr>
<tr>
<td>5.5</td>
<td>Unstructured grid for a NACA0012 airfoil and a HTP (in red) and rectilinear background grid (in blue). Due to the effect of the holes, the nodes of the background grid around the airfoil and the HTP have been removed.</td>
<td>53</td>
</tr>
<tr>
<td>5.6</td>
<td>Grid around a NACA0012 airfoil.</td>
<td>54</td>
</tr>
</tbody>
</table>
5.7 Near field grid (in blue) and surface grid (in red) for one half of the LamAiR aircraft. .......................... 56
5.8 Surface grid for the complete LamAiR aircraft. .......................... 56
5.9 Assembled grid and hole geometry for the complete aircraft. The grid for the aircraft, $\Omega_{bg}$ and $\Omega_{tg}$ are shown in red, blue and green, respectively. The hole geometry is plotted in black. .......................... 57

6.1 Lift coefficient history as a function of the dimensionless time predicted by the DVA and the RAA, with $Ma_\infty = 0.28$. The reference Reynolds number is $Re_\infty = 6.53 \cdot 10^6$. .......................... 59
6.2 Lift coefficient history as a function of the dimensionless time predicted by the DVA and the RAA, with $Ma_\infty = 0.78$. The reference Reynolds number is $Re_\infty = 18.20 \cdot 10^6$. .......................... 59
6.3 Interaction of a gust with $\lambda/C_{ref} = 1$ and the NACA0012 airfoil with both the DVA and the RAA. .......................... 60
6.4 Interaction of a gust with $\lambda/C_{ref} = 1$ and the HTP with both the DVA and the RAA. .......................... 60
6.5 Convergence history of the residual and the lift coefficient for the steady simulation with $Ma_\infty = 0.28$ and $\lambda/C_{ref} = 1$. .......................... 62
6.6 Convergence history of the residual and the lift coefficient for the unsteady simulation with $Ma_\infty = 0.28$ and $\lambda/C_{ref} = 1$ and the DVA. .......................... 62
6.7 Zoom-in of figure 6.6. .......................... 63
6.8 Lift coefficient history as a function of the dimensionless time predicted by the DVA and the RAA, with $Ma_\infty = 0.78$. The reference Reynolds number is $Re_\infty = 24.98 \cdot 10^6$. .......................... 65
6.9 Details of the interaction of vertical gusts and an aircraft with both the DVA and the RAA for $\lambda/C_{ref} = 1$ and $\lambda/C_{ref} = 2$. .......................... 66
6.10 Convergence history of the residual and the lift coefficient for the steady simulation with $Ma_\infty = 0.78$ and $\lambda/C_{ref} = 1$. .......................... 67
6.11 Convergence history of the residual and the lift coefficient for the unsteady simulation with $Ma_\infty = 0.78$, $\lambda/C_{ref} = 1$ and the RAA. .......................... 68
6.12 Zoom-in of figure 6.11. .......................... 68
6.13 Lateral force coefficient history as a function of the dimensionless time predicted by the DVA and the RAA, with $\lambda/C_{ref} = 1$. .......................... 69
6.14 Lateral force coefficient history as a function of the dimensionless time predicted by the DVA and the RAA, with $\lambda/C_{ref} = 2$. .......................... 70
6.15 Lateral force coefficient history as a function of the dimensionless time predicted by the DVA and the RAA, with $\lambda/C_{ref} = 4$. .......................... 70
6.16 Slice at $x_g = -29$ m of the $y-$ component of the velocity field at time step 1000, for $\lambda/C_{ref} = 4$ and the RAA. .......................... 71
6.17 Slice at $z_g = 0$ m of the $y-$ component of the velocity field at time step 1900, for $\lambda/C_{ref} = 1$ and the RAA. .......................... 72
6.18 Slice at $z_g = 0$ m of the $y-$ component of the velocity field at time step 1900, for $\lambda/C_{ref} = 2$ and the RAA. .......................... 72
6.19 Slice at $z_g = 0$ m of the $y-$ component of the velocity field at time step 1900, for $\lambda/C_{ref} = 4$ and the RAA. .......................... 73
6.20 Slice at $z_g = 0$ m of the $y-$ component of the velocity field at time step 1900, for $\lambda/C_{ref} = 4$ and the RAA. .......................... 74
6.21 Slice at $z_g = 0$ m of the $y$– component of the velocity field at time step 1500, for $\lambda/C_{ref} = 4$ and the RAA. .................................................. 74
6.22 Convergence history of the residual and the lateral force coefficient for the steady simulation with $Ma_{\infty} = 0.78$ and $\lambda/C_{ref} = 1$. ............................. 75
6.23 Convergence history of the residual and the lateral force coefficient for the unsteady simulation with $Ma_{\infty} = 0.78$, $\lambda/C_{ref} = 1$ and the RAA. . . 75
6.24 Zoom-in of figure 6.23. ................................................................. 76
6.25 First possible encounter between a generic aircraft and the wake vortices of another aircraft. Not drawn at scale. ............................. 77
6.26 Second possible encounter between a generic aircraft and the wake vortices of another aircraft. Not drawn at scale. ............................. 78
6.27 First test case for the verification of the DVA implementation. Not drawn at scale. ................................................................. 78
6.28 Second test case for the verification of the DVA implementation. Not drawn at scale. ................................................................. 78
6.29 Time history of the lift coefficient for test cases 1 and 2. .................. 79
6.30 Time history of the lift coefficient for test cases 1 and 3. .................. 80
6.31 Time history of the lift coefficient for test cases 4 and 5. ................. 80
6.32 Local $z$–component of the velocity field (in blue) of the right vortex. The situation corresponds to the first test case, and it is not drawn at scale. . 81
6.33 Convergence history of the residual and the lift coefficient for the steady simulation of test case 1. .................................................. 82
6.34 Convergence history of the residual and the lift coefficient for the unsteady simulation of test case 1. .................................................. 82
6.35 Zoom-in of figure 6.34. ................................................................. 83
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Coefficients for the BDF integration schemes.</td>
<td>17</td>
</tr>
<tr>
<td>4.1</td>
<td>Maximum value of the tangential velocity profiles of figure 4.6.</td>
<td>38</td>
</tr>
<tr>
<td>5.1</td>
<td>Overset grids for the density study.</td>
<td>49</td>
</tr>
<tr>
<td>5.2</td>
<td>Error in the $L_2$ norm for the overset grid.</td>
<td>50</td>
</tr>
<tr>
<td>5.3</td>
<td>Features of $\Omega_y$ depending on $\lambda/C_{ref}$.</td>
<td>52</td>
</tr>
<tr>
<td>5.4</td>
<td>Number of nodes of the 3D assembled grids.</td>
<td>56</td>
</tr>
<tr>
<td>6.1</td>
<td>$e_{C_L,max}$ errors for the 2D computations.</td>
<td>61</td>
</tr>
<tr>
<td>6.2</td>
<td>$e_{C_L,max}$ values for the 3D computations involving a vertical gust.</td>
<td>65</td>
</tr>
<tr>
<td>6.3</td>
<td>$e_{C_{Fy,max}}$ values for the 3D computations involving a lateral gust.</td>
<td>71</td>
</tr>
<tr>
<td>6.4</td>
<td>Parameters for the different pairs of equivalent test cases.</td>
<td>77</td>
</tr>
</tbody>
</table>
Abbreviations

1D one-dimensional.

2D two-dimensional.

3D three-dimensional.

ALE Arbitrary-Lagrange-Eulerian.

AoA Angle of Attack.

AUSMDV Advection Upstream Splitting Method combining flux Difference and Vector splitting.

BDF Backward Difference Formulas.

C²A²S²E Center for Computer Applications in AeroSpace Science and Engineering.

CFD Computational Fluid Dynamics.

CFL Courant-Friedrichs-Lewy.

CIMNE Centre Internacional de Mètodes Numèrics en Enginyeria.

DD Domain Decomposition.

DLR Deutsches Zentrum für Luft- und Raumfahrt.

DTS Dual Time Stepping.

DVA Disturbance Velocity Approach.

FAR Federal Aviation Regulations.

GCL Geometric Conservation Law.

HTP Horizontal Tail Plane.
Abbreviations

JST  Jameson-Schmidt-Turkel.

LamAiR  Laminar Aircraft Research.

LUSGS  Lower-Upper Symmetric Gauss-Seidel.

MegaCads  Multiblock elliptic grid generation and Computer aided system.

MG-cycles  Multigrid-cycles.

MPI  Message Passing Interface.

NACA  The National Advisory Committee for Aeronautics, USA.

RAA  Resolved Atmosphere Approach.

RANS  Reynolds Averaged Navier-Stokes.

SA(O)  Original Spalart-Allmaras one-equation.

SA-neg  Negative Spalart-Allmaras one-equation.

SDM  Standard Dynamic Model.

UPC  Universitat Politècnica de Catalunya.

URANS  Unsteady Reynolds Averaged Navier-Stokes.

VTP  Vertical Tail Plane.
List of Symbols

\( a \)  
\( speed \ of \ sound \)
\( A \)  
\( planform \ area \)
\( b \)  
\( wake \ vortex \ separation \)
\( c \)  
\( closed \ curve \)
\( c_p \)  
\( specific \ heat \ coefficient \ at \ constant \ pressure \)
\( c_v \)  
\( specific \ heat \ coefficient \ at \ constant \ volume \)
\( C_{F_y} \)  
\( Lateral \ force \ coefficient \)
\( C_L \)  
\( lift \ coefficient \)
\( C_{rej} \)  
\( reference \ chord \ length \)
\( d \)  
\( wall \ distance; \ dimension \)
\( e \)  
\( internal \ energy \ per \ unit \ mass \)
\( e_{C_L,\text{max}} \)  
\( error \ of \ the \ maximum \ lift \ coefficient \)
\( e_{F_y,\text{max}} \)  
\( error \ of \ the \ maximum \ lateral \ force \ coefficient \)
\( e_{L_2} \)  
\( error \ in \ the \ L_2 \ norm \)
\( E \)  
\( total \ energy \ per \ unit \ mass \)
\( f_i \)  
\( weight \ of \ the \ finite \ element \ interpolation \)
\( F_y \)  
\( lateral \ force \ (in \ span \ direction) \)
\( F \)  
\( flux \ vector \)
\( h \)  
\( enthalpy; \ grid \ spacing \)
\( H \)  
\( total \ (stagnation) \ enthalpy \)
\( IBLANK [\Omega_f] \)  
\( blanking \ array \)
\( k \)  
\( thermal \ conductivity \ coefficient \)
\( \tilde{K} \)  
\( Favre-averaged \ turbulent \ kinetic \ energy \)
\( L \)  
\( lift \ force \)
\( m \)  
\( number \ of \ stages; \ meters \)
\( Ma \)  
\( Mach \ number \)
\( n_x, n_y, n_z \)  
\( components \ of \ the \ unit \ normal \ vector \ in \ the \ x-, \ y-, \ z- \ direction \)
\( N \)  
\( number \ of \ individuals \ in \ an \ ensemble; \ number \ of \ time \ steps \)
\( N_F \)  
\( number \ of \ faces \ surrounding \ a \ control \ volume \)
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n )</td>
<td>unit normal vector (outward pointing) of control volume face; geodesic normal vector</td>
</tr>
<tr>
<td>( p )</td>
<td>static pressure</td>
</tr>
<tr>
<td>( P )</td>
<td>number of processors</td>
</tr>
<tr>
<td>( \dot{q}_h )</td>
<td>heat flux due to radiation, chemical reactions, etc.</td>
</tr>
<tr>
<td>( r )</td>
<td>distance from the vortex centre (radius)</td>
</tr>
<tr>
<td>( r_c )</td>
<td>core radius</td>
</tr>
<tr>
<td>( R )</td>
<td>specific gas constant</td>
</tr>
<tr>
<td>( Re )</td>
<td>Reynolds number</td>
</tr>
<tr>
<td>( R )</td>
<td>residual</td>
</tr>
<tr>
<td>( \mathbb{R} )</td>
<td>set of real numbers</td>
</tr>
<tr>
<td>( s )</td>
<td>seconds</td>
</tr>
<tr>
<td>( ds )</td>
<td>line element</td>
</tr>
<tr>
<td>( S )</td>
<td>mean rotation rate</td>
</tr>
<tr>
<td>( S_{ij} )</td>
<td>components of strain-rate tensor</td>
</tr>
<tr>
<td>( dS )</td>
<td>surface element</td>
</tr>
<tr>
<td>( \Delta S )</td>
<td>length / area of a face of a control volume</td>
</tr>
<tr>
<td>( S )</td>
<td>strain-rate tensor</td>
</tr>
<tr>
<td>( t )</td>
<td>time</td>
</tr>
<tr>
<td>( t_{ref} )</td>
<td>time to travel a distance ( C_{ref} ) at a velocity ( u_{\infty} )</td>
</tr>
<tr>
<td>( \Delta t )</td>
<td>time step</td>
</tr>
<tr>
<td>( T )</td>
<td>static temperature; time interval (in Reynolds averaging)</td>
</tr>
<tr>
<td>( u, v, w )</td>
<td>Cartesian velocity components</td>
</tr>
<tr>
<td>( v_i )</td>
<td>velocity components (in general coordinates)</td>
</tr>
<tr>
<td>( v_t )</td>
<td>tangential velocity</td>
</tr>
<tr>
<td>( \mathbf{v} )</td>
<td>velocity vector with Cartesian components ( u, v ) and ( w )</td>
</tr>
<tr>
<td>( \mathbf{v}_b )</td>
<td>velocity of the surface of a control volume</td>
</tr>
<tr>
<td>( \mathbf{v}_{flex} )</td>
<td>velocity due to a deformation of a control volume</td>
</tr>
<tr>
<td>( \mathbf{v}_i )</td>
<td>induced velocity by an atmospheric disturbance</td>
</tr>
<tr>
<td>( \mathbf{v}_{rot} )</td>
<td>rotational velocity of a control volume</td>
</tr>
<tr>
<td>( \mathbf{v}_{trans} )</td>
<td>translational velocity of a control volume</td>
</tr>
<tr>
<td>( V )</td>
<td>contravariant velocity of the flow</td>
</tr>
<tr>
<td>( V_b )</td>
<td>contravariant velocity of a control volume</td>
</tr>
<tr>
<td>( V_r )</td>
<td>contravariant velocity relative to the motion of a control volume</td>
</tr>
<tr>
<td>( w_0 )</td>
<td>amplitude of a vertical wind gust</td>
</tr>
<tr>
<td>( \mathbf{W} )</td>
<td>vector of the conservative variables</td>
</tr>
<tr>
<td>( x, y, z )</td>
<td>Cartesian coordinate system; TAU coordinate system</td>
</tr>
<tr>
<td>( \Delta x )</td>
<td>grid spacing in ( x )-direction</td>
</tr>
<tr>
<td>( x_i )</td>
<td>corner of a planar element</td>
</tr>
</tbody>
</table>
List of Symbols

\( x_e \) fringe point
\( \Delta y \) grid spacing in \( y \)-direction
\( \Delta z \) grid spacing in \( z \)-direction

\( \alpha \) angle of attack
\( \alpha_k \) coefficient of the Runge-Kutta scheme (in stage \( k \))
\( \beta_k \) coefficients for time integration scheme (\( k = -1, 0, 1, 2 \))
\( \gamma \) ratio of specific heat coefficients at constant pressure and volume
\( \gamma_k \) coefficients for time integration scheme (\( k = 0, 1 \))

\( \Gamma \) circulation
\( \delta_{ij} \) Kronecker delta symbol
\( \zeta \) type of artificial dissipation
\( \eta \) coordinate of a reference finite element
\( \lambda \) wavelenght
\( \mu \) dynamic viscosity coefficient; coordinate of a reference finite element

\( \mu_T \) eddy viscosity
\( \nu \) kinematic viscosity coefficient (\( = \mu/\rho \))
\( \xi \) coordinate of a reference finite element; yaw angle
\( \rho \) density

\( \tau_{ij} \) components of viscous stress tensor
\( \tau_{ij}^F \) components of Favre-averaged Reynolds stress tensor

\( \tau \) viscous stress tensor (normal and shear stresses)
\( \tau^F \) Favre-averaged Reynolds-stress tensor
\( \phi \) roll angle
\( \Omega \) control volume; domain; grid; geometry
\( \Omega_i \) subdomain
\( \Omega_{ij} \) components of rotation-rate tensor
\( d\Omega \) volume element
\( \partial\Omega \) boundary of a control volume; boundary of a domain; boundary of a grid; boundary of a geometry

\( \nabla U \) gradient of scalar \( U \) \( \left( = \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}, \frac{\partial U}{\partial z} \right) \)

Subscripts

\( b \) aircraft fixed coordinate system
\( bg \) background grid
\( c \) related to convection
\( cg \) Chimera grid
\( g \) geodesic coordinate system
### List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ip$</td>
<td>interpolated</td>
</tr>
<tr>
<td>$I$</td>
<td>index of a control volume</td>
</tr>
<tr>
<td>$h$</td>
<td>grid spacing; hole-cutting geometry</td>
</tr>
<tr>
<td>$L$</td>
<td>left</td>
</tr>
<tr>
<td>$m$</td>
<td>index of control volume face; species</td>
</tr>
<tr>
<td>$max$</td>
<td>maximum</td>
</tr>
<tr>
<td>$min$</td>
<td>minimum</td>
</tr>
<tr>
<td>$ref$</td>
<td>reference</td>
</tr>
<tr>
<td>$R$</td>
<td>right</td>
</tr>
<tr>
<td>$tg$</td>
<td>gust-transport grid</td>
</tr>
<tr>
<td>$v$</td>
<td>viscous part</td>
</tr>
<tr>
<td>$x, y, z$</td>
<td>components in the $x-, y-, z-$ direction</td>
</tr>
<tr>
<td>$\infty$</td>
<td>at infinity (farfield)</td>
</tr>
</tbody>
</table>

### Superscripts

<table>
<thead>
<tr>
<th>Superscript</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$bg$</td>
<td>background grid</td>
</tr>
<tr>
<td>$cg$</td>
<td>Chimera grid</td>
</tr>
<tr>
<td>$h$</td>
<td>numerical solution</td>
</tr>
<tr>
<td>$m$</td>
<td>current inner iteration</td>
</tr>
<tr>
<td>$n-2$</td>
<td>two previous time levels</td>
</tr>
<tr>
<td>$n-1$</td>
<td>previous time level</td>
</tr>
<tr>
<td>$n$</td>
<td>current time level</td>
</tr>
<tr>
<td>$n+1$</td>
<td>next time level</td>
</tr>
<tr>
<td>$\tilde{}$</td>
<td>Favre averaged mean value; virtual</td>
</tr>
<tr>
<td>$''$</td>
<td>fluctuating part of Favre decomposition</td>
</tr>
<tr>
<td>$'$</td>
<td>Reynolds averaged mean value; complement (in set theory)</td>
</tr>
<tr>
<td>$'$</td>
<td>fluctuating part of Reynolds decomposition</td>
</tr>
<tr>
<td>$*$</td>
<td>dual</td>
</tr>
<tr>
<td>$+$</td>
<td>improved solution (in multigrid)</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Motivation and objectives of the thesis

The importance of the air-transportation industry has grown considerably during the last decades, and at the same time, the necessity of making it safer and more efficient. Although the security of aircraft has been increased a lot, there are situations that can still be very dangerous and can have very serious consequences.

One example is the encounter of an aircraft with an atmospheric wind gust, which is a very strong discrete wind pulse, usually with a random and sudden character. Wind gusts are responsible for the appearance of additional unsteady air-loads on the aircraft, and can lead to wind shear, which is an abrupt change in wind speed and/or direction, causing turbulence or a rapid increase or decrease in velocity. Wind shear is especially delicate if it is produced during approach and landing, and for instance, it was the reason for the crash of the Delta Airlines flight 191 in 1985. Therefore, it is really of interest to predict the additional loads that arise during an encounter with a gust. This is important for the layout of the flight control system and the control surfaces, which are used to control the aircraft’s direction in flight and the aircraft flight’s attitude, respectively. In addition, gust loading has to be taken into account for the design of the structure, since for transport aircraft, gusts are the largest source of fatigue loading for the major part of the structure.

The encounter between an aircraft and vortices in the atmosphere is another potentially hazardous situation, yielding for instance to an imposed roll or yaw, to additional structural loads or to a loss of altitude. Obviously, this is especially dangerous during take-off and landing: when the aircraft is close to the ground, the pilot has only a limited altitude left to recover. One of the sources for the creation of vortices lies in
the planes themselves: the wake behind an aircraft generally consists of two coherent counter-rotating swirling flows of equal strength, proportional to the lift created by the aircraft. These flows, which are like horizontal tornadoes, are called wake vortex, and are a hazard to oncoming aircraft. In fact, some accidents have been reported due to wake vortex encounters, for example the crash of the American Airlines flight 587 in 2001. To avoid such encounters, a safety distance must be maintained between two aircraft. That is, there must be a waiting time of several minutes (depending on the size of the aircraft involved and other factors) before allowing another aircraft to take-off or land on the same runway. Nevertheless, the standard separations already limit the capacity of many airports, something really problematic in view of the strong growth of worldwide air traffic. Thus, modelling and understanding an encounter between an aircraft and the wake vortices can contribute to optimize the aircraft separation distances, hence increasing airport capacities while maintaining safety levels.

In the early 1970’s, and parallel to the improvements of modern aircraft, started the history of the Computational Fluid Dynamics (CFD), which can be defined as a combination of physics, numerical mathematics and computer sciences employed to simulate fluid flows. The importance of this discipline has grown very fast, and thanks to the rapidly increasing speed of supercomputers and the improvement of numerical methodologies, CFD is nowadays employed in many fields: from aircraft, turbomachinery, or car and ship design to meteorology, astrophysics or oceanography. In the field of aeronautics, CFD is seen as a complementary tool to wind tunnel and flight tests, and it has become very important for aircraft design in industry and research.

To date, several powerful CFD codes have been developed. One of them is the TAU code, developed at the Deutsches Zentrum für Luft- und Raumfahrt (DLR), the national aeronautics and space research centre of the Federal Republic of Germany. TAU is a finite volume software which can simulate viscous and inviscid flows about complex geometries from the low subsonic up to the hypersonic flow regime, employing hybrid unstructured grids [22]. The development of the TAU code, which started more than a decade ago, is still being carried out at the Institut für Aerodynamik und Strömungstechnik (Institute of Aerodynamics and Flow Technology), in particular at its numerical methods department, called Center for Computer Applications in AeroSpace Science and Engineering (C2A2S2E). Nowadays, the code is routinely used not only at the DLR, but also at many universities and in the European aeronautical industry. In particular, the aircraft manufacturer Airbus uses the TAU code for the aerodynamic design and the prediction of aerodynamic performance of aircraft.

This thesis aims to use the TAU code (release 2014.2.0) to simulate encounters between aircraft and atmospheric effects, in particular wind gusts and wake vortex.
Chapter 1. Introduction

On the one hand, regarding gust encounters, two different approaches had been previously implemented in TAU, whose basic two-dimensional (2D) verification was done in [12]. This work has been taken as a starting point, and one of the objectives of the thesis is to improve it for 2D geometries and extend the verification to three-dimensional (3D) cases. Such simulations are very costly and take a lot of time, so this is why another of the intentions of the thesis is to obtain accurate results with the smallest possible computational cost. For doing so, attention must be paid during the grid generation process and also when using the solver module of TAU.

On the other hand, the two approaches implemented to model gust-aircraft encounters can also be adapted to model encounters with wake vortex. Thus, the last purpose of the present work is to implement one of these methods in TAU, and to verify it through some equivalent test cases.

To sum up, the main objectives of the thesis are:

- To analyse the two gust modelling methods implemented in TAU for 2D and for 3D geometries.
- To implement a numerical method to simulate wake vortex encounters in TAU.
- To work with the smallest possible computational cost, but having sufficient accuracy.

As it has been already said, the numerical simulations of the present thesis are very time consuming. In this context, it is essential to perform parallel simulations on a High-Performance-Computing system. In particular, the computations of the present thesis have been run on the C$^2$A$^2$S$^2$E cluster, which has 13440 cores and uses the Son of Grid Engine resource management system. The TAU code, which is parallelized based on Domain Decomposition (DD) and on the Message Passing Interface (MPI), can be efficiently run in parallel. In fact, this an important feature of the software.

Before starting a CFD computation, it is important to generate suitable meshes. TAU does not have a module for grid generation, so the grids for this thesis have been created with two external grid generation software: CENTAUR [1] and the Multiblock elliptic grid generation and Computer aided system (MegaCads). The former one is a hybrid grid generation package from the company CentaurSoft, and the second one, which is property of the DLR, generates structured multiblock grids.

Another very important point is the postprocessing (i.e., the visualization) of data and results. For doing so, the professional software Tecplot [2], the IsoPlot99 software (property of the DLR) and the LaTeX package PGFPplots have been used.
1.2 Structure of the thesis

This subsection provides an overview of the structure of the present thesis, which is divided into two different parts and has two appendices. Chapters 2, 3 and 4 belong to the first part, called 'Mathematical models', which contains the theory of the models and the numerical methods used in the thesis. The second part (chapters 5, 6 and 7) is related to the numerical simulations of the thesis.

Chapter 2 is devoted to the TAU code, and explains the most relevant theory behind the software. First of all, the governing equations of the flow are defined, and some information about the turbulence models implemented is given. After that, the governing equations are discretized both in space and time, something necessary to perform CFD simulations. The section devoted to the temporal discretization is very important, since the simulations of this thesis are mainly unsteady. In the next section, the initial and boundary conditions that must be prescribed are introduced. For the sake of simplicity, details on the procedures for solving the systems of equations have been avoided. Nevertheless, the last section of the chapter is devoted to a technique used in the solution procedure, which is very important for the reduction of the computational time.

Chapter 3 is related to the Chimera technique, a DD method which has been crucial for the purposes of this thesis. After motivating the usage of this technique, its implementation in the TAU code is discussed. Finally, a description about how it allows to do simulations with moving grids is given.

Chapter 4 describes the two atmospheric effects considered in this work: wind gusts and wake vortex. Within the first section, both disturbances are defined and characterized in detail. After that, the first method for modelling them, called Disturbance Velocity Approach (DVA), is described. Then, the same is done with the second method, which is the so-called Resolved Atmosphere Approach (RAA). In both cases, the advantages and the disadvantages of the methods are presented.

The second part of the thesis begins with chapter 5, where the grids for all the computations are described. The first section aims to find a sufficient spatial resolution to achieve good results in the simulations. Then, and taking the results of the first section into account, the grids for the 2D computations are described, both for wind gusts and for wake vortex. Finally, the 3D grids are presented.

Chapter 6 describes the numerical simulations done and presents their results, which are also discussed in detail.

Finally, the last chapter of the thesis presents the conclusions of the main results achieved, and suggests some work to be done in the future.
Part I

Mathematical models
Chapter 2

The DLR code TAU

This chapter is devoted to the numerical methods implemented in the DLR code TAU [22], which is a second order unstructured finite volume solver for the compressible 3D Navier-Stokes equations. The DLR code TAU (from now on simply called TAU) is not only a code, but a software system which can be used for the prediction of viscous and inviscid flows about complex geometries (mainly complex aircraft-type configurations), from the low subsonic up to the hypersonic flow regime.

TAU solves the so-called Unsteady Reynolds Averaged Navier-Stokes equations, which are introduced at the beginning of this chapter. After that, these equations are discretized both in space and time, and boundary and initial conditions are introduced. Finally, some information about the multigrid technique for convergence acceleration is also given.

2.1 The mathematical description of the flow

The dynamical behaviour of a fluid is determined by the conservation of mass, momentum and energy. In the absence of body forces, the conservation of these quantities is considered in the following equations

\[
\begin{align*}
\frac{\partial}{\partial t} & \int_{\Omega} \rho d\Omega + \oint_{\partial \Omega} \rho (v \cdot n) dS = 0 \\
\frac{\partial}{\partial t} & \int_{\Omega} \rho v d\Omega + \oint_{\partial \Omega} \rho v (v \cdot n) dS = -\oint_{\partial \Omega} p n dS + \oint_{\partial \Omega} (\tau \cdot n) dS \\
\frac{\partial}{\partial t} & \int_{\Omega} \rho E d\Omega + \oint_{\partial \Omega} \rho E (v \cdot n) dS = \oint_{\partial \Omega} k (\nabla T \cdot n) dS \\
& + \int_{\Omega} \dot{q}_h d\Omega - \oint_{\partial \Omega} p (v \cdot n) dS + \oint_{\partial \Omega} (\tau \cdot v) \cdot n dS
\end{align*}
\] (2.1)
Chapter 2. The DLR code TAU

which are called the continuity equation, the momentum equation and the energy equation, respectively. In (2.1), $\Omega$ is an arbitrary control volume and $\partial \Omega$ its closed boundary, $\rho$ is the density, $\mathbf{v}$ the velocity vector, $\mathbf{n}$ a normal vector to $\partial \Omega$, $p$ the static pressure, $\mathbf{\tau}$ the so-called viscous stress tensor, $E$ the total energy per unit mass, $k$ the thermal conductivity coefficient, $T$ the temperature and $\dot{q}_h$ the heat flux. To close equations (2.1), additional relations (i.e., constitutive equations) have to be considered.

One of these relations involves the viscous stress tensor $\mathbf{\tau}$. If its components are

$$\tau_{ij} = 2\mu S_{ij} - \frac{2}{3}\mu \frac{\partial v_k}{\partial x_k} \delta_{ij}$$

where $v_k$ is the k-th component of $\mathbf{v}$, $\mu$ is the dynamic viscosity coefficient, $\delta_{ij}$ is the Kronecker delta symbol and the components of the strain-rate tensor $\mathbf{S}$ are

$$S_{ij} = \frac{1}{2} \left( \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) - \frac{1}{3} \frac{\partial v_k}{\partial x_k}$$

the fluid is defined as Newtonian.

The following equation for the thermodynamic variables must also be considered

$$E = e + \frac{|\mathbf{v}|^2}{2}$$

with

$$e = c_v T = h - \frac{p}{\rho}, \quad h = c_p T$$

where $e$ is the internal energy per unit mass, $h$ is the enthalpy, and $c_v$ and $c_p$ are the specific-heat coefficients for constant volume and pressure processes, respectively. They are constant because the fluid is assumed to be calorically perfect.

Finally, the last relation needed to close equations (2.1) is the equation of state, which in this case, is the perfect gas law

$$p = \rho RT$$

where $R = c_p - c_v$ is the specific gas constant.

Equations (2.1), together with the constitutive equations that have been just introduced are the so-called Navier-Stokes equations in the absence of body forces. Note, however, that the Navier-Stokes equations have been introduced in a fixed frame, that is, assuming that $\Omega$ is static. For the purposes of this thesis, it is of interest to consider that the boundary of the control volume is not fixed but moves with a certain velocity $\mathbf{v}_b$. As stated in [13], the most general expression for $\mathbf{v}_b$ is

$$\mathbf{v}_b = \mathbf{v}_{trans} + \mathbf{v}_{rot} + \mathbf{v}_{flex}$$
to take into account the velocity induced by a rigid translation, a rigid rotation, or a
deformation of the control volume, respectively. In the case of this thesis, $\mathbf{v}_{\text{flex}} = \mathbf{v}_{\text{rot}} = \mathbf{0}$. Equations (2.1) can be rewritten to take into account $\mathbf{v}_b$

$$\frac{\partial}{\partial t} \int_{\Omega} \rho d\Omega + \oint_{\partial \Omega} \rho \left[ (\mathbf{v} - \mathbf{v}_b) \cdot \mathbf{n} \right] dS = 0$$

$$\frac{\partial}{\partial t} \int_{\Omega} \rho \mathbf{v} d\Omega + \oint_{\partial \Omega} \rho \mathbf{v} \left[ (\mathbf{v} - \mathbf{v}_b) \cdot \mathbf{n} \right] dS = -\oint_{\partial \Omega} \rho \mathbf{n} dS + \oint_{\partial \Omega} (\mathbf{\tau} \cdot \mathbf{n}) dS$$

$$\frac{\partial}{\partial t} \int_{\Omega} \rho E d\Omega + \oint_{\partial \Omega} \rho E \left[ (\mathbf{v} - \mathbf{v}_b) \cdot \mathbf{n} \right] dS = \oint_{\partial \Omega} k \left( \nabla T \cdot \mathbf{n} \right) dS$$

$$+ \int_{\Omega} \mathbf{q}_h d\Omega - \oint_{\partial \Omega} \mathbf{p} (\mathbf{v} \cdot \mathbf{n}) dS + \oint_{\partial \Omega} (\mathbf{\tau} \cdot \mathbf{v}) \cdot \mathbf{n} dS$$

and are defined as the Navier-Stokes equations for an arbitrary control volume (from
now on, simply called Navier-Stokes equations). As said in [5], it is possible to write
these equations as

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{W} d\Omega + \oint_{\partial \Omega} (\mathbf{F}_c - \mathbf{F}_v) dS = 0$$

(2.9)

where $\mathbf{W}$ is the vector of the conservative variables (the unknowns), $\mathbf{F}_c$ is the vector of
convective fluxes and $\mathbf{F}_v$ is the vector of viscous fluxes. The definition of each of these
terms can be found in the appendix A.

It must be highlighted that, in a moving frame, not only (2.9) but also the so-called
Geometric Conservation Law (GCL) [26] must be satisfied everywhere in the domain
during the whole computation. The integral form of the GCL is

$$\frac{\partial}{\partial t} \int_{\Omega} d\Omega - \oint_{\partial \Omega} \mathbf{v}_b \cdot \mathbf{n} dS = 0$$

(2.10)

If this equation did not hold, the solution would be perturbed by numerical errors
induced by the presence of $\mathbf{v}_b$. If $\mathbf{v}_{\text{flex}} \neq \mathbf{0}$ (the grid is deformed), the GCL should be
discretized and solved using the same scheme applied to the Navier-Stokes equations,
see [5].

For high Reynolds numbers $Re$, the flow usually becomes turbulent: the solution is 3D,
time-dependent and also includes a large range of time- and length-scales. Since the
current computer power resources are not sufficient to take all scales into account, a
suitable approach is to average the Navier-Stokes equations and decompose the flow
variable in mean (or averaged) values and turbulent fluctuations. It is advisable to
perform the so-called Reynolds averaging for the density and pressure, while the mean
values of the rest of the flow variables (for instance, the velocity $\mathbf{v}$) shall be computed
through Favre (mass) averaging. Otherwise, as stated in [5], additional correlations involving density turbulent fluctuations should be considered, i.e., the set of equations to solve would become more complicated.

2.1.1 Reynolds Averaging and Favre (Mass) Averaging

In 1895, Reynolds proposed a decomposition of the flow variables in a mean and a fluctuating part in order to treat a turbulent flow. This means, for instance, that the pressure $p$ can be written as

$$p = \bar{p} + p'$$  \hspace{1cm} (2.11)

where the overbar denotes the mean value and the prime denotes the turbulent fluctuations. The following two ways of averaging are of interest:

- **Time averaging**, appropriate for statistically steady turbulence

$$\bar{p} = \lim_{T \to \infty} \frac{1}{T} \int_{t}^{t+T} p dt$$  \hspace{1cm} (2.12)

where the mean pressure is not a function of time but space if the time interval $T$ is large compared to the typical time-scale of the turbulent fluctuations.

- **Ensemble averaging**, appropriate for general turbulence

$$\bar{p} = \lim_{N \to \infty} \frac{1}{N} \sum_{m=1}^{N} p_m$$  \hspace{1cm} (2.13)

where $N$ is the number of individuals in the ensemble.

On the other hand, the Favre (Mass) Averaging consists in splitting, for example, the velocity components $v_i$ as

$$v_i = \tilde{v}_i + v''_i$$  \hspace{1cm} (2.14)

where $\tilde{v}_i$ is the mean value of $v_i$, and $v''_i$ is its fluctuating part. In this case, the mean value is computed through

$$\tilde{v}_i = \frac{1}{\bar{\rho}} \lim_{T \to \infty} \frac{1}{T} \int_{t}^{t+T} \rho v_i dt$$  \hspace{1cm} (2.15)

2.1.2 The Favre- and Reynolds-Averaged Navier-Stokes equations

Applying the Reynolds averaging (2.12) or (2.13) to $\rho$ and $p$, the Favre averaging (2.15) to the rest of the flow variables, and considering $\rho' \ll \bar{\rho}$ (Morkovin’s hypothesis), the
Chapter 2. The DLR code TAU

Favre- and Reynolds-Averaged Navier-Stokes equations in integral form read

$$\frac{\partial}{\partial t} \int_{\Omega} \bar{W} d\Omega + \oint_{\partial \Omega} (\bar{F}_c - \bar{F}_v) dS = 0$$  \hspace{1cm} (2.16)$$

where the bar over each of the terms means that they have been averaged. These equations are usually abbreviated as Unsteady Reynolds Averaged Navier-Stokes (URANS) equations. In the case of an steady problem, they are simply called Reynolds Averaged Navier-Stokes (RANS) equations.

All the details related to the definition of (2.16) can be found in the Appendix B. As it can be seen there, these equations have an additional contribution to the viscous stress tensor, which is called Favre-averaged Reynolds-stress tensor $\tau^F$. Thanks to the Boussinesq eddy viscosity hypothesis, its components can be written as

$$\tau^F_{ij} = -\bar{\rho} \dddot{v}_j' v_i'' = 2\mu_T \frac{\partial \bar{v}_k'}{\partial x_k} \delta_{ij} - \frac{2}{3} \bar{\rho} \bar{K} \delta_{ij}$$ \hspace{1cm} (2.17)$$

where $\mu_T$ is defined as the eddy viscosity, and $\bar{K} = \frac{1}{2} v_i'' v_i''$ is the Favre-averaged turbulent kinetic energy. These two variables have to be computed through one of the many turbulence models that have been implemented in TAU, see [22]. In the case of this thesis, the so-called Negative Spalart-Allmaras one-equation (SA-neg) model [3] has been used in the numerical simulations. This first-order model (thus, one of the easiest ways to compute $\tau^F_{ij}$) is a modification of the Original Spalart-Allmaras one-equation (SA(O)) model [25], where "O" stands for "original". For the sake of completeness, an introduction to the SA(O) model is given before describing the SA-neg model.

2.1.2.1 The Spalart-Allmaras one-equation (SA(0)) model

This turbulence model was developed based on empiricism, dimensional analysis and Galilean invariance. It consists in neglecting the turbulent kinetic energy $\bar{K}$ and defining $\mu_T$ through

$$\mu_T = f_{v1} \bar{\rho} \bar{v}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}, \quad \chi = \frac{\bar{\nu}}{\nu}$$ \hspace{1cm} (2.18)$$

being $\nu = \frac{\mu}{\rho}$ the kinematic viscosity coefficient. To compute $\bar{\nu}$, the following transport equation has to be solved (with appropriate initial and boundary conditions)

$$\frac{\partial \bar{\nu}}{\partial t} + \frac{\partial}{\partial x_j} (\bar{\nu} v_j) = C_{b1} (1 - f_{t2}) \bar{S} \bar{v} + \frac{1}{\sigma} \left\{ \frac{\partial}{\partial x_j} \left[ (\nu + \bar{\nu}) \frac{\partial \bar{\nu}}{\partial x_j} \right] + C_{b2} \frac{\partial \bar{\nu}}{\partial x_i} \frac{\partial \bar{\nu}}{\partial x_i} \right\} - \left[ C_{w1} f_w - \frac{C_{b1}}{\kappa^2} f_{t2} \right] \left( \frac{\bar{\nu}}{\bar{\rho}} \right)^2$$ \hspace{1cm} (2.19)$$
where $d$ is the distance from the field point to the nearest wall. Among others, the right-hand side term of this equation includes eddy viscosity production and destruction, in which the following parameter is of importance

$$\tilde{S} = S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2} \quad (2.20)$$

being $S = \sqrt{2\Omega_{ij}\Omega_{ij}}$ the magnitude of the mean rotation rate.

Finally, the model is closed by introducing

$$\Omega_{ij} = \frac{1}{2} \left( \frac{\partial v_i}{\partial x_j} - \frac{\partial v_j}{\partial x_i} \right)$$

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \quad f_w = g \left[ 1 + \frac{c_{w3}^6}{g^6 + c_{w3}^6} \right]^{1/6}$$

$$g = r + c_{w2}(r^6 - r), \quad r = \min \left[ \frac{\tilde{\nu}}{S\kappa^2 d^2}, 10 \right], \quad f_{t2} = c_{t3} \exp \left( -c_{t4} \chi^2 \right) \quad (2.21)$$

and also the following constants

$$c_{b1} = 0.1355, \quad \sigma = 2/3, \quad c_{b2} = 0.622, \quad \kappa = 0.41$$

$$c_{w2} = 0.3, \quad c_{w3} = 2, \quad c_{v1} = 7.1, \quad c_{t3} = 1.2, \quad c_{t4} = 0.5 \quad (2.22)$$

The SA(O) model is popular because of its simplicity and satisfactory results in a wide range of applications. For instance, it provides reasonably accurate predictions of turbulent flows with adverse pressure gradients, and it is also capable of simulating smooth transitions from laminar to turbulent flows at user specified locations. The good numerical properties of the SA(O) model are another advantage. The transport equation to be solved for $\tilde{\nu}$ is local, and this simplifies the implementation in an unstructured code. Furthermore, the SA(O) model is usually more robust compared to other models, for example the $k-\omega$ type ones.

### 2.1.2.2 The negative Spalart-Allmaras one-equation (SA-neg) model

Although the solution of (2.19) must be always positive (that is, $\tilde{\nu} > 0$), its numerical resolution can lead to $\tilde{\nu} < 0$, resulting in a reduction of the numerical robustness of the model. In order to improve that, a modification of the original model [25] was done in [3], resulting in the so-called Negative Spalart-Allmaras one-equation.
Chapter 2. The DLR code TAU

The SA-neg model can cope with negative values of \( \tilde{\nu} \) without degrading the numerical properties of the turbulence model. The model consists in solving (2.19), and if its solution \( \tilde{\nu} \) is negative, obtain better numerical properties by solving

\[
\begin{aligned}
\frac{\partial \tilde{\nu}}{\partial t} + v_j \frac{\partial \tilde{\nu}}{\partial x_j} &= c_{b1}(1 - c_{d3})S\tilde{\nu} + c_{w1} \left( \frac{\tilde{\nu}}{d} \right)^2 \\
&\quad + \frac{1}{\sigma} \left[ \frac{\partial}{\partial x_j} \left( (\nu + \tilde{\nu} f_n) \frac{\partial \tilde{\nu}}{\partial x_j} \right) + c_{b2} \frac{\partial \tilde{\nu}}{\partial x_i} \frac{\partial \tilde{\nu}}{\partial x_i} \right]
\end{aligned}
\] (2.23)

with

\[
f_n = \frac{c_{n1} + \chi^3}{c_{n1} - \chi^3}, \quad c_{n1} = 16.
\] (2.24)

2.2 Spatial discretization

The URANS equations have been introduced in the previous section. In order to solve them with the flow solver module of TAU, they need to be discretized both in space and time. For doing so, a separate discretization for space and time (i.e., the method of lines) is followed.

In this section, spatial discretization is carried out through a second-order finite volumes scheme.

2.2.1 Primary grid and dual grid

The spatial discretization of the computational domain is done through the so-called primary grid, which consists of polyedral elements with triangular and quadrilateral faces. The quadrilateral elements are not always planar, and in fact, TAU allows the usage of hexahedral, prismatic, pyramidal and tetrahedral elements as fundamental control volumes. Prismatic or hexahedral elements are usually used in the region near the wall for a proper resolution of strong gradients in the boundary layer. Tetrahedra and pyramids are used to resolve the space between the boundary layer and the farfield.

As stated in [22], TAU does not have a module for the generation of unstructured primary grids, so an external software for grid generation is used. The grid generation is done such that:

- The computational domain is entirely covered by the grid.
- All the grid cells boundaries match, i.e., there is no free space left between them.
- There is no overlapping between the cells.
TAU solves the URANS equations on the so-called dual grid, which is computed in a preprocessing step based on the primary grid, before the solution process is started. The dual grid is composed of several control volumes (arbitrary polyhedra), stored in a so-called edge based data structure, which makes the solver independent of the type of element of the primary grid. All metrics are given by normal vectors, representing the size and orientation of the faces, the geometric coordinates of the grid nodes and the volumes of the computational or dual cells.

Each dual cell is constructed around the nodes of the primary mesh, by connecting the midpoints of the (primary) cells with the midpoints of the neighbouring edges, as it is shown in figure 2.1.

Once the dual grid has been created, the URANS equations are discretized and evaluated in each of its control volumes. Since $v_b \neq 0$ (grids may move and control volumes may move in time), the time derivative of $\bar{W}$ in (2.16) is written as

$$\frac{\partial}{\partial t} \int_{\Omega} \bar{W} d\Omega = \frac{\partial (\Omega \bar{W})}{\partial t}$$

(2.25)

Hence, (2.16) becomes

$$\frac{\partial (\Omega \bar{W})}{\partial t} = - \oint_{\partial \Omega} (\bar{F}_c - \bar{F}_e) dS$$

(2.26)

Now, (2.26) has to be discretized in space by evaluating it for a control volume $\Omega$. Since TAU is of second order in space [15], it is sufficient to approximate the surface integral
by a sum of the fluxes crossing the faces of $\Omega$. The fluxes are considered constant across
a face, and are evaluated at the midpoint of the face.

The evaluation of (2.26) at a particular control volume leads to

$$
\frac{d(\Omega \bar{W})_I}{dt} = - \left[ \sum_{m=1}^{N_F} (\bar{F}_c - \bar{F}_v)_m \Delta S_m \right]
$$

(2.27)

where the subscript $I$ references the control volume in the computational domain, $N_F$
is the number of faces surrounding $\Omega_I$ and $\Delta S_m$ is the area of the face $m$.

Defining the term in square brackets on the right-hand side of (2.27) as the residual
$\bar{R}_I = \bar{R}(\bar{W}_I)$, the equation becomes

$$
\frac{d(\Omega \bar{W})_I}{dt} = -\bar{R}_I \quad \leftrightarrow \quad \frac{d(\Omega \bar{W})_I}{dt} + \bar{R}_I = 0
$$

(2.28)

The expression above has to be be evaluated in all the control volumes of the dual grid,
which leads to a system of first order hyperbolic ordinary differential equations. The
hyperbolic character of the system must be taken into account when applying boundary
conditions, as it is discussed in subsection 2.4.1.

The main issue regarding the spatial discretization is related to the evaluation of the
fluxes $\bar{F}_c$ and $\bar{F}_v$. As briefly described in the following (see [15] for a detailed explana-
tion), this can be done with either a central or an upwind scheme.

### 2.2.2 Central scheme

Given a face of a control volume, a central scheme is based on a finite-differences average
of variables associated with the centres of the grid cells adjacent to the left ($L$) and to
the right ($R$) of the face. For instance, $\bar{F}_c$ is computed through

$$
\bar{F}_c \approx \frac{1}{2} (\bar{F}_{c,L} + \bar{F}_{c,R}) - \frac{1}{2} \zeta \frac{1}{2} (\bar{W}_L - \bar{W}_R)
$$

(2.29)

Artificial dissipation is added in (2.29) to compensate the lack of stability of a central
scheme. The parameter $\zeta$ describes the type of dissipation that can be used:

- Scalar dissipation: Also known as Jameson-Schmidt-Turkel (JST) scheme, it is
  based on a blend of second and fourth-order differences using the so-called pseudo-
  Laplacian (a simplification of the Laplacian that offers less computational cost).
• Matrix dissipation: Despite its simplicity, scalar dissipation may result in a lack of accuracy of the results. However, by considering a matrix (the Jacobian of the convective flux), the dissipation terms can be weighted in a proper way. As a result, the accuracy of the central scheme is improved.

2.2.3 Upwind scheme

Several upwind schemes are implemented in TAU. A common feature of them is the consideration of the transport of information across the cell faces in the formulation of the convective fluxes. Compared to a central scheme with scalar artificial dissipation, these methods show an improved resolution of shocks, so they are of importance specially at the supersonic or hypersonic flow regime. Nevertheless, the numerical effort is increased, in particular in the case of unstructured grids.

Several options are available in TAU, for instance the Advection Upstream Splitting Method combining flux Difference and Vector splitting (AUSMDV), the Roe scheme or the Van Leer’s scheme, either of first or second order spacial accuracy. The choice of the scheme is a decision of the user and influences the results: for example, the Van Leer’s scheme usually leads to results of lower accuracy because of its diffusive character, but if one aims to improve the convergence of the simulation, it might be a good option to perform the first iterations of the computation with this scheme [5, 15].

2.3 Temporal discretization

As said previously, the approximation of the governing equations follows the method of lines, which decouples the discretization of space and time. Thus, after dealing with spatial discretization in the previous section, (2.28) shall now be discretized in time.

Before defining the time discretization techniques, two important concepts are introduced:

• Global time-stepping: Consists in using the same value of the time step $\Delta t$ for the computations in all the cells of the dual grid. On the one hand, global time-stepping is easy to implement, but on the other hand, the value of $\Delta t$ is restricted by the grid geometry and the characteristics of the governing equations. In practice, this means that the scheme is only stable if the so-called Courant-Friedrichs-Lewy (CFL) condition is fulfilled, and this is a strong limitation for the computation. More information about schemes using global time-stepping is given in subsection 2.3.1.
• Local time-stepping: Consists in prescribing a different value of $\Delta t$ at each of the dual cells. The value of $\Delta t$ is chosen in order to enforce a stable computation at each cell. Local time-stepping can be used with explicit schemes for steady computations, where global time-stepping is extremely inefficient because of the restrictions of the CFL condition. Therefore, local time-stepping is also regarded as a convergence acceleration technique, see section 2.5.

After introducing the two ways of defining $\Delta t$, the temporal discretization of (2.28) can be done through the two schemes available in TAU: global time-stepping and dual time-stepping.

### 2.3.1 Global time-stepping

In TAU, a Runge-Kutta multistage explicit scheme has been implemented using global time-stepping. In this case, the vector of averaged flow variables $\bar{W}_I$ is computed through

\[
\begin{align*}
\bar{W}_I^{(0)} &= \bar{W}_I^n \\
\bar{W}_I^{(1)} &= \bar{W}_I^{(0)} - \alpha_1 \frac{\Delta t_I}{\Omega_I} \bar{R}_I^{(0)} \\
\bar{W}_I^{(2)} &= \bar{W}_I^{(0)} - \alpha_2 \frac{\Delta t_I}{\Omega_I} \bar{R}_I^{(1)} \\
& \vdots \\
\bar{W}_I^{n+1} &= \bar{W}_I^{(m)} = \bar{W}_I^{(0)} - \alpha_m \frac{\Delta t_I}{\Omega_I} \bar{R}_I^{(m-1)}
\end{align*}
\]

where $\alpha_k$ are the stage coefficients, and $\bar{R}_I^{(k)}$ is the residual evaluated with the solution $\bar{W}_I^{(k)}$ of the k-th stage. Superscripts $n$ and $n+1$ denote data at time $t$ and at $t+\Delta t$, respectively, while $m$ is the number of steps (stages) in which the solution advances in time.

As it is shown in (2.30), global time-stepping explicit schemes use known data at time level $n$ to compute the solution at the next time level, $n + 1$. This makes such methods easy and fast to implement. However, as said before, its main disadvantage is that the CFL condition has to be fulfilled, which in an explicit time scheme, implies that $\Delta t$ should be equal or smaller than the time required to transport information across the stencil of the spatial discretization scheme. Otherwise, the method is not stable. The fulfilment of the CFL condition for $\Delta t$ depends on the governing equations to be solved (Euler, Navier-Stokes...), on the dimension of the problem (one-dimensional (1D), 2D or 3D) and on the grid used. A detailed estimation of $\Delta t$ for several cases can be found in [5].
2.3.2 Dual time-stepping (DTS)

This method was defined with the aim of avoiding the limitations that methods with global time-stepping have due to the CFL condition. In fact, it combines local and global time-stepping, resulting in a scheme that is suitable for time-accurate computations. Hence, it has been used in the unsteady simulations of the present work.

As stated in [13], the temporal discretization of (2.28) can be carried out using Backward Difference Formulas (BDF). This yields to the following implicit equation

\[
\frac{\beta_1 (\bar{W}\Omega)^{n+1}_j + \beta_0 (\bar{W}\Omega)^n_j + \beta_{-1} (\bar{W}\Omega)^{n-1}_j + \beta_{-2} (\bar{W}\Omega)^{n-2}_j}{\Delta t} + \gamma_1 \bar{R}_j^{n+1} + \gamma_0 \bar{R}_j^n = 0
\] (2.31)

where the superscript \( n - 1 \) denotes the previous time level \((t - \Delta t)\), and \( n - 2 \) indicates two previous time levels \((t - 2\Delta t)\). The value of the coefficients \( \beta \) and \( \gamma \) influences the accuracy of the BDF, which can be of first, second or third order. Table 2.1 presents the coefficients for each one of the schemes:

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Order</th>
<th>( \beta_1 )</th>
<th>( \beta_0 )</th>
<th>( \beta_{-1} )</th>
<th>( \beta_{-2} )</th>
<th>( \gamma_1 )</th>
<th>( \gamma_0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDF1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>BDF2</td>
<td>2</td>
<td>3/2</td>
<td>-2</td>
<td>1/2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>BDF3</td>
<td>3</td>
<td>11/6</td>
<td>-3</td>
<td>3/2</td>
<td>-1/3</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2.1: Coefficients for the BDF integration schemes.

Due to the presence of the residual \( \bar{R}_j^{n+1} = \bar{R} (\bar{W}_j^{n+1}) \), (2.31) is a non-linear equation in the unknowns \( \bar{W}_j^{n+1} \). Hence, it has to be solved iteratively. In TAU, this is done through the so-called Dual Time Stepping (DTS) method [16], which is very popular for unsteady flows.

The DTS scheme suggests solving (2.31) by defining, at each physical time step, a pseudo steady state-problem with a fictitious dual time \( t^* \). In this state, the following equation holds

\[
\frac{d(\Omega \bar{W}^*_j)}{dt^*} + \bar{R}^*_j = 0
\] (2.32)

where \( \bar{W}^*_j \) is an approximation to \( \bar{W}_j^{n+1} \), the unsteady residual \( \bar{R}^*_j \) is defined as

\[
\bar{R}^*_j = \frac{\beta_1 (\bar{W}\Omega)^{n+1}_j + \beta_0 (\bar{W}\Omega)^n_j + \beta_{-1} (\bar{W}\Omega)^{n-1}_j + \beta_{-2} (\bar{W}\Omega)^{n-2}_j}{\Delta t} + \gamma_1 \bar{R} (\bar{W}_j^*) + \gamma_0 \bar{R}_j^n
\] (2.33)
and the dual time \( t^* \) satisfies

\[
\lim_{t^* \to \infty} \begin{cases} 
\frac{d(\Omega \bar{\mathbf{W}}^*)}{dt^*} &= 0 \\
\bar{\mathbf{R}}^* &= 0 \\
\bar{\mathbf{W}}^* &= \bar{\mathbf{W}}^{n+1}_f
\end{cases}
\] (2.34)

Due to the definition of \( t^* \), (2.31) is fulfilled when (2.32) is evaluated at the so-called pseudo-steady state \( (t^* \to \infty) \). Therefore, (2.32) is solved (that is, \( \bar{\mathbf{W}}^*_f \) is computed) by advancing in dual time until the pseudo-steady state is reached and (2.34) holds. In that case, \( \bar{\mathbf{W}}^*_f = \bar{\mathbf{W}}^{n+1}_f \), i.e., the solution of (2.31) is found. The advantage of the DTS scheme is that the global time step \( \Delta t \) can be chosen based solely on flow physics and is not restricted by the CFL condition.

Equation (2.32) can be solved with the multistage Runge-Kutta method. The idea is to advance in \( t^* \), and at each dual time step \( \Delta t^* \), perform Runge-Kutta iterations (see subsection 2.3.1 for the numerical scheme) until the pseudo-steady state is reached and \( \bar{\mathbf{W}}^*_f \) approximates \( \bar{\mathbf{W}}^{n+1}_f \) with sufficient accuracy (in practice, this is achieved when \( \bar{\mathbf{R}}^*_f = \bar{\mathbf{R}}_{f,min}^* \), where \( \bar{\mathbf{R}}_{f,min}^* \) is the residual reduced by at least two or three orders of magnitude). Once the pseudo-steady state has been reached, the next physical time step \( \Delta t \) is conducted. The computation ends when \( N_{max} \), the maximum number of physical time steps, has been reached.

This procedure is shown in the flow chart of figure 2.2, where \( m \) and \( n \) control the iterations in dual and physical time, respectively. The time-marching process starts with \( (\bar{\mathbf{W}}^*_f)^m \) as an input, which can be \( \bar{\mathbf{W}}^*_f \) (the vector of averaged flow variables at the previous physical time level) or can be extrapolated from previous time steps, as said in [5, 13].

Equation (2.32) can also be solved using a Lower-Upper Symmetric Gauss-Seidel (LUSGS) scheme, which is implicit and therefore usually more robust. However, the explicit Runge-Kutta multistage method combined with acceleration techniques has a great popularity. These techniques are presented in section 2.5.
Figure 2.2: Unsteady computation with dual time-stepping and Runge-Kutta.
2.4 Initial and Boundary conditions

The aim of this section is to deal with the initial and boundary conditions that must be specified when solving (2.28) and computing $\bar{W}$.

2.4.1 Boundary conditions

The correct implementation of boundary conditions is one of the crucial points of the solver, since they influence the accuracy of the solution, and also the robustness and convergence speed of the numerical scheme. Two different type of boundaries can be distinguished:

- Artificial boundaries. Since the computational domain is only a certain part of the (real) physical domain, the last one has to be truncated, and this results in the creation of artificial boundaries like the symmetry plane or the farfield.

- Surface boundaries. These boundaries, which have a real physical meaning, have to be considered when the surface of a body is exposed to the flow.

For the simulations of this thesis, some of the artificial boundaries that can be considered in TAU have been used. In the following, they are generally described (details can be found in [15]) with the exception of the so-called Chimera boundaries, to which a complete chapter is devoted (see chapter 3 for the details of the Chimera technique). Finally, the surface boundaries used are also introduced.

Farfield

This type of boundary condition is of importance when simulating external flows past aerodynamic configurations. The farfield boundaries have to be implemented such that there are not notable effects on the flow solution as compared to the infinite domain. In addition, any outgoing information must not be reflected back into the flow field.

As said before, the hyperbolic character of the system of equations (2.28) influences how boundary conditions are applied, both in the inflow and the outflow boundaries. In the governing equations, information travels along the characteristics, and the sign of the eigenvalues of the Jacobian of the convective flux determines if it travels out or into the computational domain. According to [5], the number of variables to be imposed from outside the boundary should be equal to the number of incoming characteristics, and the remaining ones should be determined from the solution inside the domain.
Since all the simulations in this thesis have been performed in a subsonic regime (on-flow Mach number lower than 0.8), the treatment of farfield boundary conditions will be restricted to this particular case. It turns out that, in a 3D domain, a subsonic inflow boundary has four incoming characteristics and one outgoing, while the situation for a subsonic outflow boundary is just the opposite. Therefore, four variables have to be specified for the inflow boundary, which in TAU are the three components of the velocity and the temperature [15, 28]

$$\mathbf{v} = \mathbf{v}_\infty, \quad T = T_\infty$$  \hspace{1cm} (2.35)

where the subscript $\infty$ references data at the farfield. The remaining variable, $\rho$, gets a value extrapolated from the first inner grid point of the domain. On the other hand, the following is prescribed at an outflow boundary

$$\rho = \rho_\infty$$  \hspace{1cm} (2.36)

while the values for $\mathbf{v}$ and $T$ are extrapolated from the first inner grid point of the domain. After prescribing the flow conditions at the farfield boundaries, the fluxes crossing the boundary faces have to be determined by solving a Riemann problem.

**Symmetry plane**

This type of boundary condition has to be considered when the flow is symmetrical with respect to a plane. In that case, the velocity perpendicular to the symmetric boundary must be zero, which is equivalent to enforcing zero flux across this boundary. In particular, the following shall be imposed in a symmetry plane

$$\mathbf{v} \cdot \mathbf{n} = 0, \quad (\nabla \rho) \cdot \mathbf{n} = 0, \quad (\nabla T) \cdot \mathbf{n} = 0$$  \hspace{1cm} (2.37)

**Viscous wall**

This type of surface boundary, which has a physical meaning, has to be considered when a viscous fluid passes a solid wall. In this case, a no-slip boundary condition has to be implemented, that is,

$$\mathbf{v} = 0 \text{ at the surface}$$  \hspace{1cm} (2.38)

Moreover, the following is prescribed for the density and the temperature

$$(\nabla \rho) \cdot \mathbf{n} = 0, \quad (\nabla T) \cdot \mathbf{n} = 0$$  \hspace{1cm} (2.39)

When creating the primary grid, it is advisable to discretize the viscous region near the surface of the geometry (the viscous wall) with prismatic elements, which are well-suited
to capture boundary layers.

**Inviscid/Euler wall**

This type of surface boundary, which has a physical meaning, neglects all the viscous effects of the flow, and considers

\[ \mathbf{v} \cdot \mathbf{n} = 0 \text{ at the surface, } \quad (\nabla \rho) \cdot \mathbf{n} = 0, \quad (\nabla T) \cdot \mathbf{n} = 0 \]  

(2.40)

### 2.4.2 Initial conditions

Before the solution process starts, the initial flow conditions have to be prescribed. They determine the state of the flow at the first iteration \((t = 0)\) in a steady (unsteady) simulation. It is important to choose appropriate initial conditions, since they can influence in how fast the final solution is found, and also in the behaviour of the numerical scheme.

In the present work, the values defined at the farfield are taken as initial values in the whole flow field. In the case of an unsteady simulation, these values are initialized from a steady solution that has to be computed before.

### 2.5 Acceleration techniques: Multigrid

As stated in [15], different techniques have been implemented in TAU to accelerate the convergence of the solution, and therefore reduce the computational time. Among them, the so-called local time-stepping (see section 2.3 for more details) and multigrid methods have been really useful for the numerical simulations. Given the dual grid of the problem, the multigrid technique consists in solving the governing equations on a series of successively coarser grids. By doing so, the following holds:

- Since the control volumes are larger in a coarser grid, the value of the local time step can also be increased, resulting in an increase of the convergence speed.

- The high-frequency component of the error is reduced during the computation (due to the properties of the time-stepping and numerical schemes), but this does not happen with the low-frequency components. Nevertheless, the low-frequency components of the error in the finer grid can be seen as high-frequency ones in the successively coarser grids. Therefore, they can be reduced. If this scheme is followed in a sequence of grids, the error is successively damped, and as a consequence, the convergence speed is increased.
2.5.1 Agglomeration multigrid

Figure 2.4 shows the coarser grid levels generated from the dual grid of figure 2.1. These grids are generated through the so-called agglomeration multigrid method: given a fine dual grid, the next coarser level is generated by fusing the control volumes with their neighbours, so the dual cells of the successively coarser grids are larger and irregularly shaped. More information about the agglomeration multigrid can be found in [5].

2.5.2 Full multigrid cycle

The aim of this section is to describe how a computation with the multigrid technique is done, taking the grids of figures 2.1 and 2.4 as an example. For consistency, data in the finest grid has the subscript $h$ (in reference to its grid spacing), while data from the coarser grids from figure 2.4 have the subscripts $2h$, $4h$ and $8h$, respectively.

The so-called full multigrid cycle has been implemented in TAU to accelerate the convergence speed. It begins by computing the solution of (2.28) in one of the grid levels ($h$, $2h$, $4h$ or $8h$). If the computation has started in a level different from $h$, it is interpolated to the consecutive finer levels, until the finest one (level $h$) is reached. By doing that, a first solution on the finest grid is obtained efficiently. Once this has been done, the solution and the residual $\bar{W}_h$, $\bar{R}_h$ are transferred to the next coarser level by means of interpolation, and $\bar{W}_{2h}$, $\bar{R}_{2h}$ are computed. This procedure is repeated until the coarsest grid (level $8h$) is reached.

After computing $\bar{W}_{8h}$, $\bar{R}_{8h}$, the so-called coarse grid correction is calculated [5]. Then, this correction is interpolated to the next finer level ($4h$), and thanks to it, data in that level is improved, that is, a new solution $\bar{W}_{4h}^+$ and a new residual $\bar{R}_{4h}^+$ are computed. This procedure is repeated successively until the finest grid level is reached. As a consequence, low-frequency error components of the solution and the residual are smoothed.

\[
\begin{array}{c}
\bar{W}_h, \bar{R}_h \\
\bar{W}_{2h}, \bar{R}_{2h} \\
\bar{W}_{4h}, \bar{R}_{4h} \\
\bar{W}_{8h}, \bar{R}_{8h} \\
\bar{W}_{2h}^+, \bar{R}_{2h}^+ \\
\bar{W}_{4h}^+, \bar{R}_{4h}^+ \\
\bar{W}_{8h}^+, \bar{R}_{8h}^+ \\
\end{array}
\]

Figure 2.3: Example of multigrid cycle.
Chapter 2. The DLR code TAU

Figure 2.4: Generation of coarse grids by the agglomeration multigrid method in 2D.

(a) Second multigrid level, $2h$.

(b) Third multigrid level, $4h$.

(c) Fourth multigrid level, $8h$. 
Chapter 3

The Chimera technique

The aim of the present chapter is to introduce the so-called Chimera technique, a DD method which has been crucial for the numerical simulations done in this thesis. As said in chapter 2, this technique is related to one of the artificial boundary conditions that TAU has.

First of all, the Chimera technique is introduced, and a motivation for using it is also given. Then, the implementation of this technique in TAU is presented. Finally, the Chimera grid with moving subdomains (moving grids) is introduced.

3.1 Introduction and motivation

DD methods consist in splitting a boundary value problem defined in a computational domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$ into smaller boundary value problems defined on subdomains $\Omega_i \subset \Omega$, $i = 1, \ldots, n$, with $\bigcup_i \Omega_i = \Omega$ and adequate boundary conditions prescribed at the interfaces $\partial \Omega_i$. The main advantage of these methods is that they are capable of solving the problem in each $\Omega_i$ (local problem) separately from each other.

Different DD methods used in CFD can be found in the literature (see [14, 28] for further details). Among others, these methods have the following features in common:

- They introduce an easy way to parallelize a numerical code. In the particular case of TAU, the parallelization of the code is based on DD and MPI [22]. Thus, given a number $P$ of processors, the grid for $\Omega$ is partitioned into $P$ subgrids with similar size with respect to the number of points. Then, each of the $P$ processors solves the URANS equations on each of the subgrids.
• The meshing of complex geometries can be simplified by meshing each subdomain independently. This simplifies the meshing process, specially for block-structured meshes. In addition, it is really useful when working with complex 3D configurations, where the effort to generate an acceptable mesh is high.

• They allow the simulation of bodies or flow entities in relative motion.

All these features have been of importance for the present work. For instance, different subgrids have been combined to generate the 2D and the 3D grids. The features of the flow have also been transported during the simulations (more details about that are given in chapter 5). For doing so, several techniques are available [14, 28], in particular:

• The Arbitrary-Lagrange-Eulerian (ALE) technique, combined with an automatic remeshing technique that adapts and deforms the mesh to track a specified entity. This technique is not a DD method, but can capture the movement of bodies or flow entities through the domain in the case of small displacements.

• The Chimera technique, also known as overset grid technique. It is an overlapping DD method which can deal with an assembled primary grid composed of several component grids that have been generated around (simpler) geometrical entities. This technique also allows a relative motion between grids.

The ALE technique combined with an automatic remeshing that deforms the grid is not useful for the purposes of this thesis, because it is only efficient if the flow entity is transported a small distance (such that nodal displacement may be sufficient, and the nodal connectivity of the mesh remains unchanged). However, displacements of tens of meters are expected in the present case, so if (2.28) was to be solved combined with an automatic remeshing technique in an unsteady computation, the connectivity of the grid should be recomputed at each time step. The dual grid would change, and it would be necessary to compute it several times, resulting in extremely costly simulations.

The Chimera technique turns out to be the most general, flexible and efficient technique to treat the flows with moving components that are considered in the present work. This technique was first developed as a tool for simplifying the grid generation process [4], but it can now be used to solve the URANS equations with relative movement between grid blocks. For example, it is useful to simulate a maneuvering aircraft (moving flaps).
3.2 The Chimera technique in the DLR-TAU code

This section presents the implementation of the Chimera technique in TAU [15, 21, 23]. For the sake of simplicity, the 2D cartesian grid from figure 3.1 will be considered. The grid is assembled from two component grids: a background grid $\Omega_{bg}$ that covers the whole computational domain and a Chimera grid $\Omega_{cg} \mid \tilde{\Omega}_{cg} \subset \Omega_{bg}$. The extension to 3D, the use of more than two grids, unstructured grids and grids that contain solid bodies can be done in an equivalent way.

3.2.1 Node identification

The Chimera technique is an overlapping DD method: after setting up the assembled grid, $\Omega_{bg}$ and $\Omega_{cg}$ overlap in a certain part of the computational domain. It is necessary to identify where both grids overlap, since communication between them is carried out at the boundaries of the overlapping region, see subsection 3.2.2.

As figure 3.1(a) shows, a so-called hole-cutting geometry (or simply hole) $\Omega_h \mid \tilde{\Omega}_h \subset \Omega_{cg}$, which has a boundary $\partial \Omega_h$, must be used in a Chimera computation. The creation of a hole-cutting geometry is not an straightforward task: although it usually consists in one or very few cells, a more elaborated hole is necessary when cutting around a more complex geometry (see [11] for the mesh set-up of a LANN wing with aileron) or in component grids which move relatively to each other.

During the solution process, the hole removes some nodes of $\Omega_{bg}$ and defines an apparent interface, as it is shown in figure 3.1(b). This is a crucial point for the Chimera technique, since, as stated in [11], it generates the so-called Chimera boundaries, which belong to the group of artificial boundaries introduced in subsection 2.4.1. Chimera boundaries are either the outer boundary of a component grid (for instance, the boundary marked by "outer Chimera boundary" in figure 3.1(b), which corresponds to $\partial \Omega_{cg}$, the boundary of $\Omega_{cg}$) or the result of cutting a component grid with a hole (for example, the boundary marked by "inner Chimera boundary", which corresponds to $\partial \Omega_h$).

All the nodes of the primary grid that are placed at the Chimera boundaries are called fringe points, and communication between $\Omega_{bg}$ and $\Omega_{cg}$ is made in these nodes by means of interpolation. The fringe points can be classified into:

- The ones that are placed at the inner Chimera boundary (adjacent to $\partial \Omega_h$). These points, marked in figure 3.1(c), define the interpolation region from $\Omega_{cg}$ onto $\Omega_{bg}$, which following the notation of [21], shall be referred as $\Omega_{bg\leftarrow cg,ip} \subset \Omega_{bg}$.
• The ones placed at the outer Chimera boundary (adjacent to $\partial \Omega_{cg}$), marked in figure 3.1(d). This set of points defines $\Omega_{cg\leftarrow bg,ip} \subset \Omega_{cg}$, the interpolation region from $\Omega_{bg}$ onto $\Omega_{cg}$.

Figure 3.1: Grids for the Chimera technique.

Data at the fringe points are interpolated from the corner nodes of the so-called donor cells. As an example, figures 3.1(c) and 3.1(d) show a coloured donor cell in each of the component grids. Searching donor cells is a crucial point of the Chimera technique, since they can only be found if sufficient overlap exists between the component grids. Otherwise, the Chimera interpolation cannot be performed. Unfortunately, the necessity of a sufficient overlap yields to the major disadvantage of the Chimera technique: many
grid cells have to be spent in the overlapping region of two grids, and the governing
equations have to be solved twice there, which increases the computational cost [28].

3.2.2 Interpolation at the Chimera boundaries

This subsection is devoted to the details of the interpolation at the fringe points, where
the unknowns ($\bar{W}$) are interpolated between component grids. After setting up the
assembled grid and $\Omega_h$, (2.28) has to be solved. Thus, the following holds

$$\frac{d}{dt}(\Omega\bar{W}_{bg}) = \bar{R}(\bar{W}_{bg}) = 0 \quad \forall \Omega_I \subset \Omega_{bg} \setminus \Omega_h$$

(3.1)

where $\bar{W}_{bg}$ is data in $\Omega_{bg} \setminus \Omega_h$, and the original boundary conditions of the problem are
applied on $\partial\Omega_{bg} \setminus \Omega_h$. After solving (3.1), the conservative variables are interpolated in
$\Omega_{bg} \leftarrow \Omega_{ip}$, so $\bar{W}_{bg} = \bar{W}^{cg}$ (being $\bar{W}^{cg}$ the solution in $\Omega_{cg}$). In addition, in the Chimera
grid,

$$\frac{d}{dt}(\Omega\bar{W}^{cg}) = \bar{R}((\bar{W}^{cg}) = 0 \quad \forall \Omega_I \subset \Omega_{cg}$$

(3.2)

with $\bar{W}^{cg} = \bar{W}_{bg}$ in $\Omega_{cg} \leftarrow \Omega_{ip}$ by interpolation.

At each iterative step, the conservative variables $\bar{W}$ are updated at the Chimera boundaries through (3.1) and (3.2). Nevertheless, one must ensure that in the blanked points
(i.e., the points which are removed by the hole) no update of the solution is computed.
This is done by defining an array $IBLANK[\Omega_I]$ for the dual cells [21, 23]

$$IBLANK[\Omega_I] = \begin{cases} 0 & \text{if } \Omega_I \in \Omega_{bg} \text{ and } \Omega_I \subset \Omega_h \\ 1 & \text{else} \end{cases}$$

(3.3)

and modifying the Runge-Kutta scheme of (2.30)

$$\bar{W}_I^{(0)} = \bar{W}_I^{n}$$
$$\bar{W}_I^{(1)} = \bar{W}_I^{(0)} - IBLANK[\Omega_I] \alpha_1 \frac{\Delta t}{\Omega_I} \bar{R}_I^{(0)}$$
$$\bar{W}_I^{(2)} = \bar{W}_I^{(0)} - IBLANK[\Omega_I] \alpha_2 \frac{\Delta t}{\Omega_I} \bar{R}_I^{(1)}$$
$$\vdots$$
$$\bar{W}_I^{n+1} = \bar{W}_I^{(m)} = \bar{W}_I^{(0)} - IBLANK[\Omega_I] \alpha_m \frac{\Delta t}{\Omega_I} \bar{R}_I^{(m-1)}$$

(3.4)

It turns out that a key point in the above scheme is the transfer of data (interpolation)
between grids at the Chimera boundaries. This is not an straight-forward process, and
special care must be taken when doing it, as described in the following.
In TAU, the values of the flow variables at the fringe points are interpolated from the corner nodes of the donor cell by means of iso-parametric mapping, i.e. classical finite-element interpolation. In particular, trilinear interpolation is sufficient to maintain the second-order accuracy of the spatial discretization [21]. Besides, it is advisable to take the following into account for the sake of good interpolation [11]:

- In the overlap region between two grids, cells should be of similar size.
- To get rather clear-cut Chimera boundaries along hole-cutting geometries, small cell sizes are preferred in the regions where there is hole-cutting.
- Although one of the advantages of using the Chimera technique is the possibility of creating different independent grids, one has to take into account, when generating them, that a sufficient overlap must exist between the grids. The size of the overlapping region is related to the degree of interpolation, so an insufficient overlapping decreases the degree of interpolation to zero. As a consequence, the flow solution is locally degraded, and usually, the computation process stops. A sufficient overlap is specially of importance when there is relative movement between the grids (moving subdomains), since the Chimera boundaries will move.
- In the case of moving subdomains, the size of the donor cells through which another mesh is moving should be as uniform as possible.

As it has been already said, the Chimera technique has been useful not only when generating the grids but also to enable relative movement between different flow entities through the movement of a grid (see chapter 5 for more details). Consider the situation of figure 3.1, but let $\Omega_{cg}$ be in motion relative to $\Omega_{bg}$ (translation, rotation, or a composed trajectory described as a time-dependent Fourier or polynomial series) in an unsteady computation. In this case, the hole-cutting geometry $\Omega_h$ remains, by definition, attached to $\Omega_{cg}$. Hence, $\Omega_h$ moves together with $\Omega_{cg}$, and $\Omega_{bg\rightarrow cg,ip}$ and $\Omega_{cg\rightarrow bg,ip}$ change at each physical time step. Assuming a sufficient overlapping between $\Omega_{bg}$ and $\Omega_{cg}$, the Chimera boundaries (related to the search of new donor cells) are updated at each time step by doing the following [21]:

1. Search the possible donor cells through an alternating digital tree [6]. This yields to a cartesian box which contains the possible donor cells.
2. For each one of the possible donor cells of the cartesian box, construct an enveloping element with planar faces.
3. Given one of the cells of the box, check if the fringe point \( x_\varepsilon \) where the flow values will be interpolated is located inside the planar element to which the donor cell is related. If so, solve the following system of equations using a Newton method

\[
\sum_i f_i (\xi, \eta, \mu) x_i = x_\varepsilon
\]  

(3.5)

being \( x_i \) a corner of the planar element, and \( \xi, \eta, \mu \) the coordinates of a reference finite element. The weight of the interpolation \( f_i \) is 0 if \( x_\varepsilon \) is outside the planar element, and has a value between 0 and 1 if \( x_\varepsilon \) is inside the planar element.

When performing an unsteady computation with the DTS scheme of (2.31), the solutions \((\overline{W\Omega})_{n-1}^T\) and \((\overline{W\Omega})_{n-2}^T\) of the two previous time steps are required. Nevertheless, the hole moves together with the grid to which is associated (in the present case, \( \Omega_{cg} \)), so a point in \( \Omega_{bg} \setminus \Omega_h \) can become active. Against this backdrop, data at old time steps are reconstructed by interpolation of the respective values on \( \Omega_{cg} \) [21].

As an example, the following figures show two types of relative motion between grids. In the first one, there is a translation of one of the grids, while in the second one, one of the grids is rotated such that the flap of a NACA0012 airfoil is deflected. Note that, in both cases, the hole associated to the moving grid moves together with it.

![Figure 3.2: Translation of \( \Omega_{cg} \) and \( \Omega_h \) with respect to \( \Omega_{bg} \).](image-url)
(a) Initial configuration, $t = 0$.  
(b) Final configuration ($t > 0$), flap deflected $20^\circ$.

**Figure 3.3:** NACA0012 sample configuration with movable flap.
Chapter 4

CFD modelling of atmospheric effects

This chapter is devoted to the numerical methods implemented in TAU to enable encounters between aircraft and atmospheric effects (also called disturbances). In the first section, the particular atmospheric turbulences considered in this thesis are defined and characterized. After that, the second and the third sections present the two numerical methods used in the simulations of chapter 6: the Disturbance Velocity Approach and the Resolved Atmosphere Approach.

4.1 Characterization of atmospheric effects

The two examples of atmospheric effects considered in the present work are atmospheric wind gusts and aircraft wake vortex.

4.1.1 Atmospheric wind gusts

Among others, pressure differences in the air-layer around the Earth, together with solar radiation, the Coriolis effect or the presence of oceans and mountains, lead to a global air circulation in the atmosphere which changes in time and in space (see figure 4.1). Atmospheric motion can be classified into different scales, since it comprises phenomena of different magnitudes. If the smallest time-scales are considered (minutes and seconds), the variations in wind speed and direction are caused by wind gusts and turbulence [27].

A wind gust (from now on, simply called gust) is a strong, discrete wind pulse. Caused by a combination of thermal effects and friction with the Earth’s surface, the suddenness
and the random character of a gust can lead to very dangerous situations when interacting with an aircraft: gusts are responsible for the appearance of additional unsteady air-loads, which can have serious consequences for the aircraft structure, stability and control.

TAU can simulate encounters with vertical and lateral gusts, whose shape can correspond to a sharp edge gust (modelled by a Heaviside step function), to the one considered in [18], or to the ”1 − cos” law of the Federal Aviation Regulations (FAR) Part 25.341 [8]. For example, a ”1 − cos” vertical gust is defined as

\[ w(x) = \frac{1}{2} w_0 \left[ 1 - \cos \left( \frac{2\pi x}{\lambda} \right) \right] \]  

Figure 4.1: Wind data in flight and height directions. Taken from [18], modified.

where \( w \) is the \( z \)-velocity component (in Cartesian coordinates), \( w_0 \) is the amplitude of the gust and \( \lambda \) is its wavelength.
The "1 − cos" gust shape is mandatory for the aircraft certification process, and has been used in aircraft design processes and loads calculations in the last 60 years [18]. Thus, it is the one considered in this thesis, where gusts move relative to the aircraft at a translational velocity in the \( x \)-direction

\[
 u_\infty = a_\infty M a_\infty = \sqrt{\frac{\gamma p_\infty}{\rho_\infty}} M a_\infty
\]  

being \( a \) the speed of sound, \( M a \) the Mach number and \( \gamma = 1.4 \).

Figure 4.3 shows two examples of gust encounters with the Standard Dynamic Model (SDM) configuration [29], which is a generic fighter configuration similar to an F16.
4.1.2 Aircraft wake vortex

Aircraft wake vortex are the second type of atmospheric effect considered in this thesis. As stated in [10], the wake flow behind an aircraft is a natural consequence of the existence of aerodynamic lift. On the one hand, when an aircraft is flying, an upward motion, called "upwash", prevails in the region beyond both wing tips. On the other hand, there is a stronger downward motion just behind the trailing edge of each of the wings, which is known as "downwash". These two motions are shown in the following figure:

![Figure 4.4: Possible vortex encounters and their consequences. Taken from [20].](image)

In the near field of the wing, small vortices emerge from the so-called vortex sheet at the wing tips and at the edges of the landing flaps [10], in a process that is governed by several physical phenomena: boundary-layer separation, initiation of vortex stabilities, etc. The vortex sheet experiences a roll-up and the co-rotating vortices merge, resulting in a wake behind an aircraft which consists of two coherent counter-rotating swirling flows: the aircraft wake vortex. Each of these flows, which are like horizontal tornadoes of about equal strength, has its own strong circulation, which is proportional to the weight of the aircraft.

Once the aircraft has passed, the two wake vortex dissipate slowly in the atmosphere. Nevertheless, the wingtip vortices cause the so-called wake turbulence, which is hazardous to other aircraft. The situation is specially delicate during take-off and landing, where strong wake vortex are created because aircraft fly at high Angle of Attack (AoA). In this case, an encounter between a light aircraft and the wake turbulence of a heavier one can be really dangerous, see figure 4.4 for some of the consequences.
Consider a system of coordinates with origin in the centre of gravity of an aircraft, and $x$, $y$ and $z$ defining the axes of flight, span and height, respectively. In this context, several parameters can be defined to describe the wake vortex behind an aircraft [10]:

- Tangential velocity $v_t = \sqrt{v^2 + w^2}$, which is the most obvious feature of a swirling flow.
- Circulation $\Gamma$, related to the amount of vortex strength
  \[
  \Gamma = \Gamma (r) = \oint_{c} v_{t} \, ds \tag{4.3}
  \]
  where $c$ is a closed curve, and $r = \sqrt{y^2 + z^2}$ is the distance from the vortex centre.
- Core radius $r_c$, which, once the roll-up is completed, defines the value of $r$ corresponding to the maximum tangential velocity $v_{t,\text{max}}$.
- Distance between both vortices $b$, corresponding to a 70% or 75% of the wingspan of the preceding aircraft.

![Diagram of vortex parameters](image)

**Figure 4.5:** Scheme of the tangential velocity and definition of vortex parameters.

The tangential velocity $v_t$ is modelled independently for each of the vortices through an analytical model. In the present work, the following ones have been considered:

- Lamb-Oseen model [19]:
  \[
  v_t = \frac{\Gamma}{2\pi r} \left[ 1 - \exp \left( -1.2544 \left( r/r_c \right)^2 \right) \right] \tag{4.4}
  \]
• Burnham-Hallock model [7]:

\[ v_t = \frac{\Gamma}{2\pi r_c^2 + r^2} \]  \hspace{1cm} (4.5)

Figure 4.6 shows a comparison between both models, considering only one vortex. The plots have been done with \( \Gamma = 158 \text{ m}^2/\text{s} \) and \( b = 21.5 \text{ m} \), corresponding to a VFW-Fokker 614 jetliner, see [9].

![Figure 4.6: Comparison of the tangential velocity \( v_t \) predicted by the Lamb-Oseen and the Burnham-Hallock models for different values of the core radius \( r_c \).](image)

It is clear that, for the same core radius \( r_c \), the value of the maximum tangential velocity \( v_{t,max} \) is different. The opposite holds for its position, \( r_{max}/b \). This is detailed in the table below:

<table>
<thead>
<tr>
<th>Model</th>
<th>( r_c ) [m]</th>
<th>( v_{t,max} ) [m/s]</th>
<th>( r_{max}/b ) [-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lamb-Oseen</td>
<td>1.075</td>
<td>16.72</td>
<td>0.050</td>
</tr>
<tr>
<td>Burnham-Hallock</td>
<td>1.075</td>
<td>11.70</td>
<td>0.050</td>
</tr>
<tr>
<td>Burnham-Hallock</td>
<td>0.75</td>
<td>16.76</td>
<td>0.035</td>
</tr>
</tbody>
</table>

**Table 4.1:** Maximum value of the tangential velocity profiles of figure 4.6.

For the same core radius \( r_c \), the value of \( v_{t,max} \) predicted by the Lamb-Oseen model is around 40% greater than the value of \( v_{t,max} \) predicted by the Burnham-Hallock model. However, the position of the maximum is the same, \( r_{max}/b = 0.050 \). The maximum value of \( v_{t,max} \) predicted by the Burnham-Hallock model with \( r_c = 0.75 \text{ m} \) is really similar to the Lamb-Oseen one, but \( r_{max}/b \) diminishes.
The wake vortex are modelled by a superposition of two ideal vortices with an opposite circulation (the left vortex rotates clockwise, while the right one rotates counterclockwise), see figures 4.4 and 4.5.

Defining \((x_{vortex,1}, y_{vortex,1}, z_{vortex,1})\) and \((x_{vortex,2}, y_{vortex,2}, z_{vortex,2})\) as the Cartesian coordinates from the left vortex and the right vortex, respectively, the following radii can be computed [9]

\[
r_1 = \sqrt{(y - y_{vortex,1})^2 + (z - z_{vortex,1})^2}
\]

\[
r_2 = \sqrt{(y - y_{vortex,2})^2 + (z - z_{vortex,2})^2}
\]

Then, the superposition of the left and the right vortices yields to a velocity field that, in Cartesian coordinates, has the following components in span and height direction

\[
v = v_t \left( \frac{z - z_{vortex,1}}{r_1} \right) - v_t \left( \frac{z - z_{vortex,2}}{r_2} \right)
\]

\[
w = -v_t \left( \frac{y - y_{vortex,1}}{r_1} \right) + v_t \left( \frac{y - y_{vortex,2}}{r_2} \right)
\]

where, as it has been said before, the tangential velocity \(v_t\) is given by (4.4) or (4.5). Plots of \(v\) and \(w\), using \(\Gamma = 300 \text{ m}^2/\text{s}, b = 21.5 \text{ m}\) and \(r_c = 0.75 \text{ m}\) are shown in figure 4.7 for the Burhnam-Hallock model.
4.2 Disturbance Velocity Approach (DVA)

The Disturbance Velocity Approach (DVA) is the first method used to model encounters between aircraft and atmospheric effects (gusts and wake vortex).

4.2.1 Description of the method

The approach followed to implement the DVA is very similar to the one done in section 2.1 when introducing the velocity \( v_b \) of a control volume \( \Omega \). In (2.8), \( v_b \) slightly alters the flux balance, yielding, for instance, to the following continuity equation

\[
\frac{\partial}{\partial t} \int_{\Omega} \rho d\Omega + \oint_{\partial \Omega} \rho [(v - v_b) \cdot n] dS = 0 \tag{4.10}
\]

where the convection across the cell interface of the control volume is altered from \( v \) (in the original continuity equation (2.1)) to \( v - v_b \). This balance also changes if the velocity induced by an atmospheric effect \( v_i \) is taken into account. Then, (4.10) becomes [12]

\[
\frac{\partial}{\partial t} \int_{\Omega} \rho d\Omega + \oint_{\partial \Omega} \rho [(v - v_b - v_i) \cdot n] dS = 0 \tag{4.11}
\]

The same can be done for the momentum and the energy equations of (2.8). The induced velocity \( v_i \) is a function of space and time, and in TAU it can be prescribed either for vertical and lateral gusts or wake vortex. In this last case, the normal vector to the plane containing the vortices can have any orientation in space.

Figure 4.8 shows a vertical gust of wavelength \( \lambda \) that moves relative to a NACA0012 airfoil with a translational speed \( u_\infty \) given by (4.2). If the gust is still far from the leading edge, the boundary of the control volume (in gray) has \( v_i = 0 \). However, when the gust is just beneath the airfoil, the induced velocity in the \( z \)-direction is equal to the amplitude of the gust. As said in [13], the local effect of the gust is the same, as if the airfoil is moving with a velocity \( (0, 0, -w_0) \) downward.

![Figure 4.8: Gust travelling relative to a NACA0012 airfoil.](a) Initial situation \( (t = 0) \). (b) Gust beneath the airfoil \( (t > 0) \).
4.2.2 Advantages and disadvantages

On the one hand, the DVA is a simple method which is straight-forward to implement in a CFD code like TAU. Furthermore, it can be used in combination with standard CFD meshes, usually characterized by a reduced mesh resolution with growing distance from the aircraft. Hence, no spatial high resolution is required to transport the atmospheric effect (either gusts or vortices) through the computational domain.

On the other hand, the main disadvantage of this method is that it is not capable of solving the whole coupled problem between the gust and the aircraft: although it can predict the effect of the atmospheric disturbances on the aircraft, it cannot predict the feedback of the aerodynamics of the aircraft on the properties of the disturbance (either gusts or wake vortex). Hence, a prediction error for the DVA can be expected, specially for gusts of small wavelength.

4.3 Resolved Atmosphere Approach (RAA)

The Resolved Atmosphere Approach (RAA) is the second method used for simulating encounters between aircrafts and atmospheric disturbances. This method aims to eliminate the disadvantage of the DVA by resolving the atmospheric disturbance in the flow field. Doing so, the mutual interaction of aircraft and atmospheric effects can be captured. Therefore, the RAA can be used to quantify the prediction errors of the DVA.

4.3.1 Description of the method

As said in [12], to enable the simulation of a mutual interaction between an aircraft and an atmospheric effect, the later one has to be resolved in the flow field. In TAU, this is done by feeding the disturbance into the flow field at certain farfield boundaries. For doing so, the non-reflecting boundary condition has to be modified. Moreover, the constant farfield state $v_\infty = (u_\infty, v_\infty, w_\infty)$ that is usually prescribed at a farfield boundary must be adapted.

In the case of a gust, $v_\infty$ is specified as a function of space and time at the suitable farfield boundaries in dependency of the actual position of the gust. However, for wake vortex, not only the velocity components but also the farfield pressure $p_\infty$ and density $\rho_\infty$ have to be specified.

When doing a RAA computation with TAU, the user can select at which farfield boundaries the atmospheric disturbance is fed into the farfield. At these boundaries, boundary
conditions become unsteady, since the disturbance has to be fed in the domain and transported through it.

An example of such boundaries is shown in figure 4.9 for a "1 − cos" lateral gust and a 2D rectangular computational domain. The boundaries used to feed in the gust are dashed, and it is clear that their boundary conditions must be a function of time. At first, in figure 4.9(a), the $y-$ component of the velocity at the left boundary is $v = 22$ m/s, because the gust is being fed into the domain. However, in figure 4.9(b), the gust is far away from the left boundary, and the prescribed value at the left boundary is $v = 0$ m/s. A similar thing happens for the bottom boundary.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure.png}
\caption{Resolution of a lateral gust with the RAA. Boundaries where unsteady boundary conditions are applied (left and bottom) are dashed.}
\end{figure}

4.3.2 Advantages and disadvantages

The main advantage of the RAA with respect to the DVA is that it is capable of predicting the effects of the mutual interaction between an atmospheric disturbance and an aircraft, providing more accurate and realistic results.

Nevertheless, the RAA requires a high resolution in space to transport the gust through the domain without too much numerical losses, since TAU has a second-order space accuracy [12]. Standard CFD meshes have usually not enough resolution, and this yields to expensive computations because special meshes are required. Furthermore, at the beginning of an unsteady computation, more inner iterations in dual time are necessary to feed the gust into the domain (situation of figure 4.9(a)) compared to the DVA. This also contributes to the high computational cost of a RAA computation. Therefore, for the sake of cheaper numerical simulations, the RAA should only be used
in the critical cases where the DVA yields to results which are not satisfactory. The gust simulations in chapter 6 aim to set the range of validity of the DVA by comparing its results with the RAA ones.
Part II

Numerical applications
Chapter 5

Grid set-up

The aim of this chapter is to describe the grids used for the computations of chapter 6. The first section presents a grid density study, which aims to find a sufficient grid resolution to perform accurate simulations. After that, and based on the results of this study, grids for the 2D and the 3D simulations are presented in the second and third sections, respectively.

5.1 Grid density study

One of the disadvantages of the RAA is the requirement of a high spatial resolution to transport a disturbance without too much numerical diffusion. Hence, following the idea of [12], a grid density study has to be performed to find a reasonable compromise of computational costs and accuracy.

The grid density study consists in transporting an atmospheric disturbance (for instance, a ”1 – cos” gust with $\lambda = 1$ m and $w_0 = 0.1u_\infty$) through a computational domain without aerodynamic configurations. This means that the gust moves through the domain without interacting with neither an airfoil or an aircraft. Therefore, if the spatial resolution in the direction of movement is sufficient, the properties of the gust are expected to be conserved during the simulation.

For the sake of simplicity, a 2D square domain $\Omega = x_g \times z_g \in [-20, 20] \times [-20, 20]$ m$^2$ is taken into account, being $x_g$ and $z_g$ the so-called geodesic coordinates. They belong to one of the three right-hand coordinate systems considered in TAU, which are [13]:

- The geodesic coordinate system $(x_g, y_g, z_g)$, used as inertial system for encounters between aircraft and atmospheric disturbances.
• The body fixed coordinate system \((x, y, z)\), also called TAU coordinate system.

• The aircraft fixed coordinate system \((x_b, y_b, z_b)\) with origin on the center of gravity.

These three coordinate systems are presented in figure 5.1, which also shows that for time \(t = 0\), the geodesic and the aircraft fixed coordinate systems are the same, while they usually differ for \(t > 0\).

![Figure 5.1](image)

(a) Initial situation, \(t = 0\).

(b) Situation for \(t > 0\).

**Figure 5.1**: Geodesic \((x_g, y_g, z_g)\), TAU \((x, y, z)\) and aircraft fixed \((x_b, y_b, z_b)\) coordinate systems for \(t = 0\) and \(t > 0\). Taken from [13].

### 5.1.1 Uniform grid

For the first density study, the domain is meshed with a single uniform Cartesian primary grid. The grid spacing in the \(x_g\) direction is \(\Delta x_g = 0.01\) m, i.e., a resolution of 100 cells per wavelength in the direction of movement of the gust is used.

With the RAA, the gust is fed into the domain through unsteady farfield boundary conditions in the left and the bottom farfield boundaries (see subsection 4.3.1 for more details). Then, the gust moves to the right of the domain with a translational velocity defined in (4.2). When it has travelled 20 m, the profile of the gust is compared to the analytical one given by (4.1), as it is shown in figure 5.2.

The comparison between both gusts clearly shows that a uniform grid with a resolution of 100 cells per \(\lambda\) is not sufficient for a proper transportation of the gust. With this resolution, numerical oscillations appear upstream of the gust, and the maximum value of \(w/u_\infty\) is shifted to the left compared to the analytical velocity profile.
The error observed is measured using an error defined in the $L_2$ norm

$$e_{L_2} = \sqrt{\int_{\Omega} (w^h - w)^2 d\Omega / \int_{\Omega} w^2 d\Omega}$$  \hspace{1cm} (5.1)$$

where $\Omega$ is the computational domain, $w^h$ is the numerical data, and $w$ is the analytical gust profile, given by (4.1). In the case of figure 5.2, the error is $e_{L_2} = 15.06\%$. Therefore, an uniform Cartesian grid with $\Delta x_g = 0.01$ m (100 cells per $\lambda$) yields to an unacceptably large error.

The value of $e_{L_2}$ can be reduced by refining the grid in the direction of the gust’s movement. However, this implies much more expensive simulations, which is especially critical for the 3D cases. The alternative approach suggested in [12] provides much better results within an acceptable computational time, as it is explained in the following section.

### 5.1.2 Overset grid

Motivated by the results of figure 5.2, a new density study is made. In this case, the primary grid is assembled from the following two component grids:

- A background grid, $\Omega_{bg}$. This grid is rectilinear, i.e., its spacings in both $x_g$– and $z_g$– directions are not constant. This is done in order to reduce the number of nodes.
• A Chimera grid, which, following the nomenclature of [12], is called gust-transport grid, $\Omega_{tg}$. This Cartesian grid, whose length in $x_g$ direction is $2\lambda$ and overlaps with $\Omega_{bg}$, is used to transport the gust through the domain. At the beginning of the computation ($t = 0$), the gust is just in front of the computational domain. For $t > 0$, the gust is fed into the flow field through unsteady farfield boundary conditions at the left and bottom boundaries. After that, the gust advances in time, moving to the right of the domain. Once the gust reaches the center of $\Omega_{tg}$, the gust-transport grid starts to move, together with the gust, with the convection velocity of the flow. In the moving grid, the velocity of the boundary of a control volume is $v_b = (-u_\infty, 0, 0)$ (in geodesic coordinates). Therefore, equations (2.8), presented in chapter 2, have to be taken into account.

To create such an overset grid and enable the movement of $\Omega_{tg}$, it is necessary to use the Chimera technique, which is described in detail in chapter 3.

Figure 5.3: Overset grid for the grid density study. $\Omega_{bg}$ is shown in blue, while $\Omega_{tg}$, which is moving, is plotted in red. A vertical "1 − cos" gust (in yellow) moves together with $\Omega_{tg}$.

Once the gust is centred in $\Omega_{tg}$, it is placed in the same nodes (the ones belonging to $\Omega_{tg}$) until the end of the computation. This implies that $\frac{\partial w}{\partial t} = 0$ at these nodes, and the properties of the gust are expected to be conserved. Note that this is contrary to what happens for the single grid of subsection 5.1.1. In that case, $\frac{\partial w}{\partial t} \neq 0$ at the nodes of the
grid, because the gust moves through the domain, so the properties of the gust are not conserved (see figure 5.2).

As it has been said in chapter 3, the Chimera technique needs a hole-cutting geometry, \( \Omega_h \), to define the Chimera boundaries where communication between the grids is done. Furthermore, the governing equations are solved twice in the region where both grids overlap, so it is usually more expensive compared to standard computations. In the grid density study of [12], three Cartesian component grids are used: \( \Omega_{bg} \), \( \Omega_{tg} \), and a fine grid at the left part of the domain, used to feed the gust into the domain with sufficient resolution. With the aim of reducing the computational cost of using three component grids, only \( \Omega_{bg} \) and \( \Omega_{tg} \) have been used in the present work, but \( \Omega_{bg} \) is rectilinear instead of Cartesian (see figure 5.3). Hence, \( \Omega_{bg} \) has enough spatial resolution to feed in the gust successfully, and once the gust is centred in \( \Omega_{tg} \), it becomes coarser in order to reduce the number of nodes (because the gust is already placed in \( \Omega_{tg} \), not in \( \Omega_{bg} \)). This type of grid has been created with the Multiblock elliptic grid generation and Computer aided system (MegaCads), which is a grid generation software property of the DLR suitable for creating block-structured meshes.

The grid density study has been done with the overset grid just described for three different gust resolutions: 25 cells per \( \lambda \), 50 cells per \( \lambda \) and 100 cells per \( \lambda \), where \( \lambda = 1 \) m. Hence, several grids \( \Omega_{bg} \) and \( \Omega_{tg} \) have to be generated for each one of the resolutions. Some details of the grids are given in the table below:

<table>
<thead>
<tr>
<th>Resolution</th>
<th>( \Delta x_g ) for ( \Omega_{tg} ) [m]</th>
<th>Nodes (assembled grid)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25 cells per ( \lambda )</td>
<td>0.04</td>
<td>147204</td>
</tr>
<tr>
<td>50 cells per ( \lambda )</td>
<td>0.02</td>
<td>161304</td>
</tr>
<tr>
<td>100 cells per ( \lambda )</td>
<td>0.01</td>
<td>189504</td>
</tr>
</tbody>
</table>

**Table 5.1: Overset grids for the density study.**

The simulation described in subsection 5.1.1 has been done with the three overset grids, and their results are compared to the analytical gust profile in figure 5.4. It can be seen, for a resolution of 25 cells per \( \lambda \), that the maximum value of \( w/u_\infty \) is lower than the analytical one. Besides, deviations from the analytical solution are specially visible upstream of the gust. As expected, higher resolutions offer better results, as it is shown by the values of \( e_{L_2} \) in table 5.2. Although a resolution of 50 cells per \( \lambda \) yields very good results, using 100 cells per \( \lambda \) is the best option: the amplitude of the numerical gust is conserved during the transportation, and only minor deviations are visible upstream and downstream of the gust, so the value of \( e_{L_2} \) is acceptable.
5.1.3 Conclusions

With the aim of finding a sufficient grid resolution in the $x_g$-direction for both 2D and 3D simulations, two different grid density studies have been performed. The first one considers a Cartesian uniform grid with a resolution of 100 cells per $\lambda$, while for the second one, an overset grid has been used. The overset grid consists of a background grid and a gust-transport grid, which are assembled using the Chimera technique.

Figures 5.2 and 5.4 compare the numerical gust and the analytical one for the two types of grids described. It is clear that the overset grid offers much better results than the uniform grid, where the error in the $L_2$ norm is high ($e_{L_2} = 15.06 \%$). This is because:

- $\frac{\partial w}{\partial t} = 0$ in the case of the overset grid. Once the gust is centred in $\Omega_{tg}$ and its movement starts, the gust is always placed in the same nodes (the ones belonging to $\Omega_{tg}$), so the properties of the gust are conserved.

- $\frac{\partial w}{\partial t} \neq 0$ in the case of an uniform grid. Thus, the properties of the gust are not conserved as the gust moves through the domain.
In the case of an uniform grid, the properties of the gust can be conserved if $\Delta x_g$ is sufficiently small. Unfortunately, this implies very expensive computations, so using an overset grid is more advisable. In this case, the gust is transported with higher fidelity, and even with a resolution of 25 cells per $\lambda$, the value of $e_{L_2}$ is clearly better than the one from an uniform grid, for a significant lower computational cost.

Motivated by the results obtained in the grid density studies, the grids used for gust simulations, which are described in the following sections, contain a moving grid with a resolution of 100 cells per $\lambda$ in the direction of the gust movement.

5.2 2D grids

In this section, all details related to the 2D grids are given. In the first subsection, the grids for the simulations involving atmospheric gusts are described. The same is done in the last subsection for the grid to simulate wake vortex encounters.

5.2.1 Grids for gusts encounters

The gust simulations of chapter 6 aim to set the range of validity of the DVA by comparing its results with the RAA ones. One of the critical points of the RAA is the requirement of a high spatial resolution to transport the gust without too much numerical diffusion. According to the grid density study of the previous section, an overset grid with a resolution of 100 cells per $\lambda$ is sufficient. A lower resolution might be sufficient for the DVA, but for the sake of consistency, the same grid has to be used to compare it to the RAA.

The DVA and the RAA are compared using a configuration consisting in a symmetrical NACA0012 airfoil and a Horizontal Tail Plane (HTP), also called horizontal stabilizer. Both airfoils are at $\alpha = 0^\circ$, where $\alpha$ is the AoA. The reason for choosing such a configuration is the expectation, that the aerodynamics of the airfoil will influence the properties of the gust, which afterwards interacts with the HTP [12]. As said in chapter 4, the DVA cannot capture the effect of the aerodynamics of the airfoil on the gust, but this can be done with the RAA. Hence, prediction errors for the DVA are expected, especially for gusts of short wavelength. In order to check this, an encounter between a "$1 - \cos$" vertical gust and the NACA0012-HTP configuration has to be computed for different wavelengths.
The primary grid is created with the Chimera technique, and it is assembled from the following component grids:

- A rectilinear background grid $\Omega_{bg} = x_g \times z_g \in [-20, 20] \times [-20, 20]$ m², which has a growing value of $\Delta x_g$. The finest zone ($\Delta x_g = \lambda/100$) has a length of $\lambda/2$ in the $x_g$– direction, and, as explained in subsection 5.1.2, is used to feed the gust into the domain with sufficient spatial resolution. After that, and within a region of length $2\lambda$, the grid spacing grows, according to a Poisson distribution, until $\Delta x_g = 0.1$ m. With the aim of saving nodes and reducing the computational cost, the grid is also rectilinear in the direction of height, with a higher resolution for $z_g \in [-3, 3]$ m (close to the airfoils). This grid has been created with MegaCads.

- A Cartesian gust-transport grid $\Omega_{tg}$, which has a size of $2\lambda \times 40$ and a constant spacing $\Delta x_g = \lambda/100$. As said in subsection 5.1.2, this grid, which has been created with MegaCads, moves during the computation thanks to the Chimera technique.

- An unstructured mesh containing the near field of the NACA0012 airfoil and the HTP, with a size of $[-5.5, 2] \times [-2, 2]$ m². The distance between the inflow farfield boundary and the airfoil is $20 \ C_{ref}$, being $C_{ref} = 1$ m the reference chord length of the airfoil. This grid has been created with the grid generation software CENTAUR [1], and has prismatic elements close to the viscous wall to properly capture the boundary layer of the flow.

In order to check different gust wavelengths, simulations with $\lambda/C_{ref} = 1$, $\lambda/C_{ref} = 2$ and $\lambda/C_{ref} = 4$ are made. Since many features of $\Omega_{bg}$ and $\Omega_{tg}$ depend on $\lambda$, different grids have to be created for each value of $\lambda$. Table 5.3 presents some details of the different $\Omega_{tg}$ grids.

<table>
<thead>
<tr>
<th>$\Delta x_g$ [m]</th>
<th>$\lambda/C_{ref} = 1$ [-]</th>
<th>$\lambda/C_{ref} = 2$ [-]</th>
<th>$\lambda/C_{ref} = 4$ [-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size [m²]</td>
<td>2 × 40</td>
<td>4 × 40</td>
<td>8 × 40</td>
</tr>
</tbody>
</table>

Table 5.3: Features of $\Omega_{tg}$ depending on $\lambda/C_{ref}$.

When using the Chimera technique, the hole-cutting geometries remove some nodes of the grids and create the Chimera boundaries, see chapter 3. In the present case, there are three 2D holes:

- The first hole is associated to $\Omega_{tg}$, the gust-transport grid (with movement). During the simulation, this hole removes some nodes of $\Omega_{bg}$. Therefore, before reaching
the airfoil, the gust is placed within the nodes of $\Omega_{tg}$, which has a sufficient spatial resolution. Since this hole does not cut the near field grid, its nodes are not removed, but this does not affect the accuracy of the results (the near field grid has sufficient spatial resolution).

- The second hole is associated to the NACA0012 airfoil, and removes nodes of $\Omega_{bg}$ and $\Omega_{tg}$. Thus, the near field of the airfoil is always discretized with the unstructured mesh that has also elements for resolving the boundary layer.

- The third hole is associated to the HTP, and removes nodes of $\Omega_{bg}$ and $\Omega_{tg}$. It works like the hole for the NACA0012 airfoil.

In TAU, the user can select in which grids nodes should be removed, and also the grid, to which a hole is associated. This is specially of importance in unsteady computations with moving grids, because the hole can move together with the grid to which it is associated. In the present case, this is done for the hole associated to $\Omega_{tg}$.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure5.5.png}
\caption{Unstructured grid for a NACA0012 airfoil and a HTP (in red) and rectilinear background grid (in blue). Due to the effect of the holes, the nodes of the background grid around the airfoil and the HTP have been removed.}
\end{figure}

### 5.2.2 Grid for wake vortex encounters

One of the objectives of this thesis is to implement the DVA to enable the simulation of encounters with wake vortices. The implementation has been done in the most general
way, allowing the normal vector to the plane containing the vortices to have any orientation in space. The code has to be verified through some equivalent test cases (see section 6.2 for more details), which, for the sake of simplicity, are inviscid and consider a NACA0012 airfoil with $\alpha = 0^\circ$.

The primary grid for the airfoil has been created with CENTAUR and it is shown in figure 5.6. Since the simulations are inviscid, the inviscid wall boundary condition has to be prescribed (see subsection 2.4.1). Thus, the grid does not have any prismatic elements near the wall.

![Figure 5.6: Grid around a NACA0012 airfoil.](image)

### 5.3 3D grids

In subsection 5.2.1, a configuration involving a NACA0012 airfoil and an HTP is suggested for comparing the DVA and RAA. Nevertheless, more realistic results are expected with a 3D simulation involving a complete aircraft, where before reaching the tail of the aircraft, the gust interacts with the fuselage, and also with its wings. As a consequence, more discrepancies between the DVA and RAA may arise. In this case, the following simulations are considered:

- An encounter between an aircraft ($\alpha = 0^\circ$) and a "1 - cos" vertical gust with $\lambda/C_{ref} = 1$, $\lambda/C_{ref} = 2$ and $\lambda/C_{ref} = 4$, where $C_{ref} = 4$ m. Such simulations have a clear advantage: there is a symmetry with respect to the $y_g = 0$ m plane.
(where \(y_g\) is the span axis in geodesic coordinates), so only one half of the aircraft has to be considered. This reduces the number of mesh nodes and hence the computational cost of the simulation by a factor of 2.

- An encounter between an aircraft \((\alpha = 0^\circ)\) and a "1 – cos" lateral gust with \(\lambda/C_{ref} = 1\), \(\lambda/C_{ref} = 2\) and \(\lambda/C_{ref} = 4\). In this case, the situation is not symmetric with respect to the \(y_g = 0\) plane, so the complete aircraft must be considered.

When creating the 3D primary grids, the conclusions of the 2D grid density study done in section 5.1 can be also considered to be valid. Thus, the approach followed for the 2D grids in subsection 5.2.1 (that is, overset grid and \(\Omega_{tg}\) with a resolution of 100 cells per \(\lambda\)) is also used for the 3D applications. The primary grid is assembled from:

- A rectilinear background grid \(\Omega_{bg} = x_g \times y_g \times z_g \in [-40, 70] \times [-40, 40] \times [-40, 45] \text{ m}^3\) for the half configuration. In the case of the complete configuration, \(y_g \in [-80, 80] \text{ m}\). The resolution in span and height direction is increased close to the near field of the aircraft, according to a Poisson distribution, and the finest spacings in span and height direction are \(\Delta y_g = \Delta z_g = 0.25 \text{ m}\). As in the 2D case, the spacing also changes in the direction of flight in order to feed in the gust properly: the finest spacing is equal to \(\lambda/100\), while the coarsest one is equal to 0.25 m.

- A Cartesian gust-transport grid \(\Omega_{tg} = x_g \times y_g \times z_g \in [0, 2\lambda] \times [-40, 40] \times [-40, 45] \text{ m}^3\) for the half configuration. In the case of the complete configuration, \(y_g \in [-80, 80] \text{ m}\). This grid has a spacing \(\Delta x_g = \lambda/100\), and moves with the convection velocity \(u_\infty\) of the gust to transport it through the domain.

- An unstructured grid for the surface and the near field of the aircraft. The aircraft chosen is the one from the DLR project Laminar Aircraft Research (LamAiR), see [24] for an overview of the work. The grid has been created with CENTAUR, and has prismatic elements close to the viscous wall to capture the boundary layer of the flow. Figure 5.7 shows both the surface and the near field grids for one half of the aircraft, while figure 5.8 shows the complete aircraft with its surface grid.

As in the 2D case, the features of the grids depend on \(\lambda\), so several \(\Omega_{bg}\) and \(\Omega_{tg}\) grids have to be created. The number of nodes for the final assembled grid is shown in table 5.4 for each one of the cases. As it can be seen, the number of nodes is much higher compared to the 2D cases, in particular for the complete aircraft. Thus, these simulations are the most expensive ones that have been considered in this thesis.
Grid set-up

\[ \lambda/C_{ref} = 1 [-] \quad \lambda/C_{ref} = 2 [-] \quad \lambda/C_{ref} = 4 [-] \]

<table>
<thead>
<tr>
<th></th>
<th>( \lambda/C_{ref} = 1 )</th>
<th>( \lambda/C_{ref} = 2 )</th>
<th>( \lambda/C_{ref} = 4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Half aircraft</td>
<td>24.68 ( \cdot 10^6 )</td>
<td>24.45 ( \cdot 10^6 )</td>
<td>23.49 ( \cdot 10^6 )</td>
</tr>
<tr>
<td>Complete aircraft</td>
<td>49.16 ( \cdot 10^6 )</td>
<td>51.72 ( \cdot 10^6 )</td>
<td>50.21 ( \cdot 10^6 )</td>
</tr>
</tbody>
</table>

**Table 5.4:** Number of nodes of the 3D assembled grids.

**Figure 5.7:** Near field grid (in blue) and surface grid (in red) for one half of the LamAiR aircraft.

**Figure 5.8:** Surface grid for the complete LamAiR aircraft.
In the simulation involving a lateral gust, there are two 3D hole-cutting geometries:

- A first hole associated to $\Omega_{tg}$, the grid with movement. It works exactly as the one described in subsection 5.2.1.

- A second hole, associated to the grid for the aircraft, that removes some of the nodes of the background grid. This hole is shown in figure 5.9 for the simulation with the complete aircraft (the one involving a lateral gust).

![Figure 5.9: Assembled grid and hole geometry for the complete aircraft. The grid for the aircraft, $\Omega_{bg}$ and $\Omega_{tg}$ are shown in red, blue and green, respectively. The hole geometry is plotted in black.](image)

There is also another hole for the vertical gust simulations, which is associated to $\Omega_{tg}$. It removes some nodes of the near field grid for the aircraft, and it is introduced to ensure that the computations are done within the nodes of the moving grid (which has a very high resolution). Due to geometric reasons, this hole can not be used for the simulations with a lateral gust.
Chapter 6

Results and discussion

In this chapter, the numerical computations done and their results are described and analysed. The SA(O) turbulence model is used, and to improve the convergence properties, the Multigrid technique as well as local time-stepping have been considered.

6.1 Atmospheric wind gusts

To find the range of validity for the DVA, several computations have been done using the DVA and the RAA, with a ”1 − cos” gust and $\lambda/C_{\text{ref}} = 1$, $\lambda/C_{\text{ref}} = 2$ and $\lambda/C_{\text{ref}} = 4$. The gust moves through the domain at $\mathbf{v} = (-u_\infty, 0, 0)$ (in geodesic coordinates), and has an amplitude of $0.1u_\infty$, where $u_\infty$ is defined in (4.2). In order to study the influence of compressibility, two different on-flow Mach numbers have been selected. For $Ma_\infty = 0.28$, there is a nearly incompressible flow, while compressibility effects are expected for $Ma_\infty = 0.78$.

6.1.1 2D computations

These simulations involve a NACA0012-HTP configuration with $C_{\text{ref}} = 1$ m that interacts with a vertical gust. In this case, variations in the lift force are expected. The lift is analysed through the dimensionless lift coefficient $C_L$

$$C_L = \frac{2L}{\rho_\infty u_\infty^2 A} \quad (6.1)$$

where $L$ is the lift force and $A$ is the planform area. Figures 6.1 and 6.2 present the $C_L$ history as a function of the dimensionless time $t/t_{\text{ref}}$, being $t_{\text{ref}} = C_{\text{ref}}/u_\infty$ the time needed by a disturbance (in this case, a gust) to travel a distance $C_{\text{ref}}$ with velocity $u_\infty$. 

58
Chapter 6. Numerical results

Figure 6.1: Lift coefficient history as a function of the dimensionless time predicted by the DVA and the RAA, with $Ma_{\infty} = 0.28$. The reference Reynolds number is $Re_{\infty} = 6.53 \cdot 10^6$.

Figure 6.2: Lift coefficient history as a function of the dimensionless time predicted by the DVA and the RAA, with $Ma_{\infty} = 0.78$. The reference Reynolds number is $Re_{\infty} = 18.20 \cdot 10^6$. 
As figures 6.1 and 6.2 show, the major differences between the DVA and the RAA correspond to $\lambda/C_{\text{ref}} = 1$. In particular, discrepancies are found for $t/t_{\text{ref}} \in [20, 22]$ and $t/t_{\text{ref}} \in [23, 25]$, which correspond to the interaction with the NACA0012 airfoil and the HTP, respectively. This is shown in the zoom-in of the lift history of figures 6.3 and 6.4. Since the airfoil and the HTP are symmetrical and are at zero AoA, the lift variations are purely created by gust loading.

**Figure 6.3**: Interaction of a gust with $\lambda/C_{\text{ref}} = 1$ and the NACA0012 airfoil with both the DVA and the RAA.

**Figure 6.4**: Interaction of a gust with $\lambda/C_{\text{ref}} = 1$ and the HTP with both the DVA and the RAA.

It can be seen, regarding the interaction with the NACA0012 airfoil, that the maximum lift computed with the DVA is under-predicted compared to the RAA one. The maximum lift is related to the maximum load, and can be used to compute the prediction
error of the DVA relative to the RAA, defined as
\[ e_{C_{L,\text{max}}} = \frac{|C_{L,\text{max,RAA}} - C_{L,\text{max,DVA}}|}{|C_{L,\text{max,RAA}}|} \]  
(6.2)

being \( C_{L,\text{max,DVA}} \) and \( C_{L,\text{max,RAA}} \) the maximum lift coefficient computed with the DVA and the RAA, respectively. Table 6.1 presents the value of this error for the three wavelengths and the two different Mach numbers:

<table>
<thead>
<tr>
<th>( \lambda/C_{\text{ref}} ) [-]</th>
<th>( M_{\infty} ) = 0.28 [-]</th>
<th>( M_{\infty} ) = 0.78 [-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.07 %</td>
<td>13.19 %</td>
</tr>
<tr>
<td>2</td>
<td>1.22 %</td>
<td>3.38 %</td>
</tr>
<tr>
<td>4</td>
<td>0.45 %</td>
<td>0.78 %</td>
</tr>
</tbody>
</table>

Table 6.1: \( e_{C_{L,\text{max}}} \) errors for the 2D computations.

The errors corresponding to \( \lambda/C_{\text{ref}} = 4 \) are very low for both Mach numbers. This means that there is nearly no difference between the DVA and the RAA, i.e., there is a good agreement between both approaches. Both errors are also quite small for \( \lambda/C_{\text{ref}} = 2 \). In the case of the smallest wavelength \( \lambda/C_{\text{ref}} = 1 \) and an incompressible flow regime, the agreement between the DVA and the RAA is also good. Nevertheless, this does not happen for \( M_{\infty} = 0.78 \), where \( e_{C_{L,\text{max}}} = 13.19 \% \). In this case, the prediction error of the DVA is not negligible, so its results cannot be considered valid. Therefore, a safe range of validity for the DVA is \( \lambda/C_{\text{ref}} \geq 2 \) for \( M_{\infty} \leq 0.78 \).

As said before, there are also differences between the DVA and the RAA during the interaction with the HTP, in particular for \( \lambda/C_{\text{ref}} = 1 \), (shown in detail in figure 6.4). When using the RAA, the gust that interacts with the HTP is ”weaker” compared to the DVA, because it has previously interacted with the airfoil and its properties have been modified. This does not happen for the DVA, because this method is not capable of capturing the mutual interaction between the aerodynamics of the airfoil and the gust, i.e., the shape of the gust is not modified. Hence, the DVA predicts lift overpeaks for both Mach numbers, specially in the case of the smallest wavelength. In the case of \( \lambda/C_{\text{ref}} = 2 \) and \( \lambda/C_{\text{ref}} = 4 \), there is an agreement between both methods in the interaction with the NACA0012 airfoil (see table 6.1), so the differences during the interaction with the HTP are minimal.

The convergence of the solutions has been checked in terms of the residual and \( C_{L} \) for all the computations. As an example, figures 6.5, 6.6 and 6.7 show the convergence history for the DVA computation with \( M_{\infty} = 0.28 \) and \( \lambda/C_{\text{ref}} = 1 \), but similar results are also obtained for the rest of the computations. These figures present the convergence of both the residual (in red, scale shown at the left ordinate axis) and \( C_{L} \) (in blue, scale shown
at the right ordinate axis) as a function of the Multigrid-cycles (MG-cycles) performed during the solution procedure.

Figure 6.5: Convergence history of the residual and the lift coefficient for the steady simulation with $Ma_\infty = 0.28$ and $\lambda/C_{ref} = 1$.

Figure 6.6: Convergence history of the residual and the lift coefficient for the unsteady simulation with $Ma_\infty = 0.28$, $\lambda/C_{ref} = 1$ and the DVA.
As it has been already said, figure 6.5 shows the convergence history of the steady simulation, used to find the initial flow conditions for the following unsteady computation. A central scheme with matrix dissipation and local time-stepping with a Runge-Kutta scheme has been used. Notice that, after 1500 MG-cycles, the residual has decreased some orders of magnitude (it is of order $10^{-7}$), and $C_L$ has a constant value (the lift is stable). This is the criterion considered to ensure that the computation has converged.

The basic numerical parameters of the unsteady simulation are the same as in the steady one. Nevertheless, the DTS scheme has been used to allow a time accurate flow solution. The physical time step size $\Delta t$ is computed as

$$\Delta t = \frac{t_{\text{ref}}}{100} = \frac{C_{\text{ref}}}{100u_{\infty}}$$  \hspace{1cm} (6.3)

In particular, 4000 time steps are necessary for the simulation of figure 6.6. As the flow chart of figure 2.2 shows, several iterations in dual time (now MG-cycles, because the Multigrid technique is used) are performed within each physical time step. As described in chapter 2, the solution in dual time has to converge, within each physical time step, to the so called pseudo-steady state, which is reached, in practice, if the residual of the computation decreases some orders of magnitude and $C_L$ has an horizontal tangent.

A zoom-in of figure 6.6 is presented in figure 6.7. The plot clearly shows the behaviour of the residual and $C_L$ as the simulation advances in physical time. It can also be seen that about 50 MG-cycles in dual time are necessary to reach the pseudo-steady state,
where the residual has decreased some orders of magnitude and $C_L$ has a constant value (horizontal tangent).

In order to reduce the computational effort, the number of MG-cycles that are performed within each time step changes depending on the requirements of the computation. In particular, it changes from a minimum of 5 to a maximum of 50 (shown in figure 6.7). For example, 50 MG-cycles are necessary to reach the pseudo-steady state when the gust is interacting with the airfoil, but when it is still far away from its leading edge, 5 MG-cycles are sufficient. The number of MG-cycles also depends on the numerical method chosen to model the gust: compared to the DVA, the RAA needs more MG-cycles at the beginning of the computation, to feed the gust into the domain with sufficient resolution. Therefore, the RAA has greater computational cost compared to the DVA.

### 6.1.2 3D computations

If a gust interacts with a realistic 3D configuration including the fuselage and the tail of an aircraft, more discrepancies between the DVA and the RAA may arise compared to the 2D case: the gust interacts three times with the aerodynamics of the aircraft instead of two. As a consequence, the range of validity of the DVA set in 2D may change.

In the following, encounters between a vertical and a lateral gust with $\lambda/C_{ref} = 1$, $\lambda/C_{ref} = 2$ and $\lambda/C_{ref} = 4$ with the LamAiR aircraft (with $C_{ref} = 4$ m, presented in section 5.3) are simulated with both the DVA and the RAA. The vertical gust influences the aerodynamics of the fuselage, the wings and horizontal tail, whereas the lateral one mainly influences the fuselage and the vertical tail. The computations done have a large computational effort, specially in the case of a lateral gust, since a grid for the complete aircraft has to be used. Hence, to avoid many expensive calculus, only simulations with $Ma_\infty = 0.78$ have been done, because it corresponds to the biggest discrepancies between the DVA and the RAA in 2D (see table 6.1).

#### 6.1.2.1 Interaction with a vertical gust

Figure 6.8 shows the history of the lift coefficient $C_L$ as a function of the dimensionless time $t/t_{ref}$. At the beginning of the simulation, the value of $C_L$ is constant, because the gust is still far away from the aircraft. Then, the gust interacts with the fuselage of the aircraft, as the first growth in $C_L$ shows, and after that, there is an interaction with the wing, yielding to the major growth of the lift. Finally, the gust interacts with the tail of the aircraft, which corresponds to the last growth of the lift.
Chapter 6. Numerical results

Figure 6.8: Lift coefficient history as a function of the dimensionless time predicted by the DVA and the RAA, with $Ma_\infty = 0.78$. The reference Reynolds number is $Re_\infty = 24.98 \cdot 10^6$.

As in the 2D case, the maximum $C_L$ can be related to the maximum load, and it is used to evaluate the discrepancies between the DVA and the RAA through $\epsilon_{C_{L,\text{max}}}$, defined in (6.2). Table 6.2 shows the value of $\epsilon_{C_{L,\text{max}}}$ corresponding to data of figure 6.8:

<table>
<thead>
<tr>
<th>$\lambda/C_{\text{ref}}$ [-]</th>
<th>$\epsilon_{C_{L,\text{max}}}$ [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.90</td>
</tr>
<tr>
<td>2</td>
<td>1.02</td>
</tr>
<tr>
<td>4</td>
<td>0.41</td>
</tr>
</tbody>
</table>

Table 6.2: $\epsilon_{C_{L,\text{max}}}$ values for the 3D computations involving a vertical gust.

Compared to the errors of table 6.1, the ones from table 6.2 are surprisingly small. There is a very good agreement between the DVA and the RAA in terms of the maximum load, independently of the wavelength of the wind gust. Presumably, there are lift cancellations due to the global character of this magnitude, so the values of $\epsilon_{C_{L,\text{max}}}$ are smaller than expected. Lift cancellations are also assumed to be the reason why $\epsilon_{C_{L,\text{max}}}$ for $\lambda/C_{\text{ref}} = 1$ is even smaller than $\epsilon_{C_{L,\text{max}}}$ for $\lambda/C_{\text{ref}} = 2$, which is unexpected and contrary to the results of the 2D simulations.

Figure 6.9 shows zoom-ins of figure 6.8 for both $\lambda/C_{\text{ref}} = 1$ and $\lambda/C_{\text{ref}} = 2$, the two wavelengths that present more differences between the DVA and the RAA.
**Figure 6.9:** Details of the interaction of vertical a gust and an aircraft with both the DVA and the RAA for $\lambda/C_{ref} = 1$ and $\lambda/C_{ref} = 2$.

Note that the agreement between both methods is surprisingly well for the interaction with the wing of the plane, and also for the interaction with the tail. This is something unexpected, because before reaching the tail, the gust experiences two interactions with
the aircraft’s aerodynamics (with the fuselage and with the wing), so in principle, there
should be larger discrepancies between the DVA and RAA. As said before, lift cancella-
tions are supposed to be the reason for the good agreement of both methods.

Convergence of the solutions has been checked in terms of the residual and $C_L$. The
convergence history of the steady simulation for $\lambda/C_{ref} = 1$ is shown in figure 6.10,
while figure 6.11 presents the convergence of the unsteady computation for $\lambda/C_{ref} = 1$
and the RAA. A zoom-in of it is shown in figure 6.12. Similar plots have been obtained
for the rest of the computations.

In order to improve the convergence properties of the steady simulation, 500 MG-cycles
using an upwind scheme with AUSMDV have been first performed, followed by 9000
MG-cycles using a central scheme with scalar dissipation. In both cases, local time-
stepping with the Runge-Kutta scheme has been used. It can be seen that the residual
decreases some orders of magnitude, and that after 6000 MG-cycles, it is of order $10^{-6}$.
The lift coefficient has also a constant value, so the convergence criterion is fulfilled.

The unsteady simulation has been performed using a central scheme with scalar dissi-
pation and the DTS method with 2750 physical time steps, with $\Delta t$ defined in (6.3).
Convergence within each time step is achieved if the residual has decreased some orders
of magnitude and the $C_L$ has an horizontal tangent, which can be seen in figure 6.12.
As in the 2D unsteady simulations, the number of MG-cycles in dual time that are
performed within each time step changes depending on the necessities of the simulation.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure6.10.png}
\caption{Convergence history of the residual and the lift coefficient for the steady
simulation with $Ma_{\infty} = 0.78$ and $\lambda/C_{ref} = 1$.}
\end{figure}
Chapter 6. Numerical results

Figure 6.11: Convergence history of the residual and the lift coefficient for the unsteady simulation with $Ma_{\infty} = 0.78$, $\lambda/C_{ref} = 1$ and the RAA.

Figure 6.12: Zoom-in of figure 6.11.
6.1.2.2 Interaction with a lateral gust

For the interaction with a lateral gust, the DVA and the RAA are compared for $\lambda/C_{ref} = 1$, $\lambda/C_{ref} = 2$ and $\lambda/C_{ref} = 4$ in terms of the lateral force coefficient $C_{Fy}$. This coefficient is defined as

$$C_{Fy} = \frac{2F_y}{\rho_\infty u_\infty^2 A}$$

where $F_y$ is the lateral force (in span direction) that acts on the aircraft.

Figures 6.13, 6.14 and 6.15 show the time history of $C_{Fy}$ as a function of the dimensionless time $t/t_{ref}$ for an encounter with a gust of $\lambda/C_{ref} = 1$, $\lambda/C_{ref} = 2$ and $\lambda/C_{ref} = 4$, respectively. For all the cases, the on-flow Mach number is $Ma_\infty = 0.78$ and the on-flow Reynolds number is $Re_\infty = 24.98 \cdot 10^6$.

The value of $C_{Fy}$ is equal to zero at the beginning of the simulation, because the lateral gust is still far away from the aircraft and it does not influence the flow around it. The first decrease of $C_{Fy}$ corresponds to the interaction with the fuselage of the aircraft. After that, the gust reaches the wings. It can be seen that the change of $C_{Fy}$ during the interaction between the wings and the gust really depends on the value of $\lambda$: for example, for $\lambda/C_{ref} = 4$, there is a small change on the value of $C_{Fy}$. Finally, the gust interacts with the Vertical Tail Plane (VTP), yielding to the major decrease of $C_{Fy}$.

![Figure 6.13: Lateral force coefficient history as a function of the dimensionless time predicted by the DVA and the RAA, with $\lambda/C_{ref} = 1$.](image-url)
Chapter 6. Numerical results

Figure 6.14: Lateral force coefficient history as a function of the dimensionless time predicted by the DVA and the RAA, with $\lambda/C_{ref} = 2$.

Figure 6.15: Lateral force coefficient history as a function of the dimensionless time predicted by the DVA and the RAA, with $\lambda/C_{ref} = 4$. 
The prediction error of the DVA is defined as

\[ e_{C_{Fy,\text{max}}} = \frac{|C_{Fy,\text{max},\text{RAA}} - C_{Fy,\text{max},\text{DVA}}|}{|C_{Fy,\text{max},\text{RAA}}|} \] (6.5)

being \( C_{Fy,\text{max},\text{DVA}} \) and \( C_{Fy,\text{max},\text{RAA}} \) the maximum lateral force coefficient computed with the DVA and the RAA, respectively, which correspond to the interaction with the VTP. Table 6.3 presents the value of this error for the three wavelengths considered:

<table>
<thead>
<tr>
<th>( \lambda/C_{\text{ref}} ) [-]</th>
<th>( e_{C_{Fy,\text{max}}} ) [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.47</td>
</tr>
<tr>
<td>2</td>
<td>6.40</td>
</tr>
<tr>
<td>4</td>
<td>9.73</td>
</tr>
</tbody>
</table>

Table 6.3: \( e_{C_{Fy,\text{max}}} \) values for the 3D computations involving a lateral gust.

As it can be seen, the larger \( \lambda \) is, the greater is the value of \( e_{C_{Fy,\text{max}}} \). Note, however, that this is totally opposite to what is predicted for the interaction with the fuselage: figures 6.13, 6.14 and 6.15 show that the greatest differences between the DVA and the RAA correspond to \( \lambda/C_{\text{ref}} = 1 \). This change of behaviour is assumed to be due to the two vortices that emerge from the tips of the wings. Hence, when the gust interacts with the wings, it is perturbed due to the presence of the vortices, and this lasts until the end of the computation. The presence of the vortices is shown in the following figure:

![Slice at \( x_g = -29 \) m of the \( y \)- component of the velocity field at time step 1000, for \( \lambda/C_{\text{ref}} = 4 \) and the RAA.](image.png)
The effect of the vortices on the gust can be clearly seen in the following figures, which show the $y-$ component of the velocity field at time step 1900, for all the values of $\lambda$ and the RAA:

**Figure 6.17:** Slice at $z_g = 0$ m of the $y-$ component of the velocity field at time step 1900, for $\lambda/C_{ref} = 1$ and the RAA.

**Figure 6.18:** Slice at $z_g = 0$ m of the $y-$ component of the velocity field at time step 1900, for $\lambda/C_{ref} = 2$ and the RAA.
The wake vortices influence the flow behind the wings of the aircraft, as figures 6.17, 6.18 and 6.19 show. Therefore, they also perturb the gust, from the interaction with the wings until the end of the computation. As it can be seen in figures 6.20 and 6.21, the gust is symmetric in flight and span direction before and after the interaction with the fuselage. Thus, until the interaction with the wings, the behaviour of the DVA agrees with [12] (better agreement with the RAA for large wavelengths).

Once the gust reaches the wings, it becomes influenced by the wake vortices. The vortex that rotates clockwise (the one on the left wing, as shown by figures 4.4, 4.5 and 6.16) opposites the effect of the gust, increasing the value of $v$. On the contrary, the right vortex, which rotates counter-clockwise, decreases the value of $v$. As a consequence, the gust is no longer symmetric. The larger the value of $\lambda$ is, the stronger is this effect, and the larger are the discrepancies between the DVA and the RAA. Nevertheless, it is not fully understood, that only the wake vortices influence the results. For instance, an insufficient resolution of the near field grid may also affect the results. Therefore, no recommendation for the range of validity of the DVA can be given at the moment.

Finally, convergence has been checked for all the computations done, which use the same numerical parameters as in the 3D encounters with a vertical gust. Figures 6.22, 6.23 and 6.24 present plots of the steady and the unsteady simulations for $\lambda/C_{ref} = 1$ and the RAA (similar plots have been obtained for the rest of the computations). As it can
be seen, the convergence criteria are the same as in the previous sections, but in terms of $C_{F_y}$ instead of $C_L$.

Figure 6.20: Slice at $z_a = 0$ m of the $y-$ component of the velocity field at time step 1000, for $\lambda/C_{ref} = 4$ and the RAA.

Figure 6.21: Slice at $z_a = 0$ m of the $y-$ component of the velocity field at time step 1500, for $\lambda/C_{ref} = 4$ and the RAA.
Figure 6.22: Convergence history of the residual and the lateral force coefficient for the steady simulation with $M_{a_\infty} = 0.78$ and $\lambda/C_{ref} = 1$.

Figure 6.23: Convergence history of the residual and the lateral force coefficient for the unsteady simulation with $M_{a_\infty} = 0.78$, $\lambda/C_{ref} = 1$ and the RAA.
6.2 Interaction with wake vortices

The aim of the simulations of this subsection is to verify the implementation of the DVA for wake vortex encounters in TAU. The implementation has been done such that \( n \), which is the normal vector to the plane that contains the vortices (in geodesic coordinates), can have any orientation in space. Thus, the user can simulate encounters between aircraft flying in different directions. In addition, both the Lamb-Oseen and the Burnham-Hallock models, presented in chapter 4, have been implemented.

Figures 6.25 and 6.26 present schemes of two possible encounters which, for the sake of simplicity, are not drawn at scale. For example, figure 6.25 shows a lateral interaction between a generic aircraft and the wake vortices of another aircraft, which are contained within a plane. As in the case of the gust, the vortices can advance in time during the simulation, with a velocity \( \mathbf{v} = (-u_\infty, 0, 0) \). The translational velocity \( u_\infty \) is defined in (4.2) and depends on the flow properties at the farfield.

In order to verify the correctness of the DVA implementation, five test cases are defined. These test cases, which involve a NACA0012 airfoil at zero AoA and are totally equivalent, are presented in table 6.4. In particular, equal results are expected for test cases 1 and 2, 1 and 3, and 4 and 5.
## Chapter 6. Numerical results

<table>
<thead>
<tr>
<th>Test case</th>
<th>$\phi$ [$^\circ$]</th>
<th>$\xi$ [$^\circ$]</th>
<th>$\mathbf{n}$</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>(0, 1, 0)</td>
<td>Burnham-Hallock</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>90</td>
<td>(−1, 0, 0)</td>
<td>Burnham-Hallock</td>
</tr>
<tr>
<td>3</td>
<td>60</td>
<td>0</td>
<td>(0, $1/2$, $\sqrt{3}/2$)</td>
<td>Burnham-Hallock</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>(0, 1, 0)</td>
<td>Lamb-Oseen</td>
</tr>
<tr>
<td>5</td>
<td>−45</td>
<td>0</td>
<td>(0, $1/\sqrt{2}$, $−1/\sqrt{2}$)</td>
<td>Lamb-Oseen</td>
</tr>
</tbody>
</table>

### Table 6.4: Parameters for the different pairs of equivalent test cases.

Figures 6.27 and 6.28 show a scheme of the test cases 1 and 2, respectively (for the sake of clarity, they are not drawn at scale). In the first test case, the plane of the vortices (dashed line) is restricted to $y_g = 0$ m, and the normal vector $\mathbf{n}$ and the velocity vector $\mathbf{v}$ of the vortices are perpendicular. The position of the airfoil is fixed, and its roll angle $\phi$ and yaw angle $\xi$ are zero.

In the second test case, the geodesic normal vector is set to $\mathbf{n} = (−1, 0, 0)$ and the airfoil is rotated $90^\circ$ around the $z_g$ axis. In addition, the position of the vortices is fixed and the airfoil moves at $\mathbf{v} = (0, 0, −u_\infty)$.

![Diagram](image.png)

**Figure 6.25:** First possible encounter between a generic aircraft and the wake vortices of another aircraft. Not drawn at scale.
Figure 6.26: Second possible encounter between a generic aircraft and the wake vortices of another aircraft. Not drawn at scale.

\[ \phi = \xi = 0^\circ, \quad n = (0, 1, 0) \]
\[ \mathbf{v} = (-u_\infty, 0, 0) \]

\( \phi = \xi = 0^\circ \), \( x_g, z_g \)

Figure 6.27: First test case for the verification of the DVA implementation. Not drawn at scale.

\[ \phi = \xi = 0^\circ, \quad n = (-1, 0, 0) \]
\[ \mathbf{v} = (0, 0, -u_\infty) \]

\( \phi = 0^\circ, \xi = 90^\circ \), \( y_g, z_g \)

Figure 6.28: Second test case for the verification of the DVA implementation. Not drawn at scale.
Both, a steady and an unsteady simulation, have been done for each of the test cases just described. These computations have the following features in common:

- The initial distance between the leading edge of the airfoil and the right vortex is 33 m, i.e., there is no influence on the flow around the airfoil.
- The cores of the two vortices are separated by $b = 21.5$ m, corresponding to a VFW-Fokker 614 jetliner.
- The circulation is $\Gamma = 300$ m$^2$/s.
- The core radius is $r_c = 0.75$ m.
- The reference Mach and Reynolds numbers are $Ma_{\infty} = 0.5$ and $Re_{\infty} = 11.06 \cdot 10^6$, respectively.

Figures 6.29, 6.30 and 6.31 show the history of the $C_L$ as a function of $t/t_{\text{ref}}$ for the different pairs of equivalent test cases. As expected, each equivalent pair leads to the same results. For the sake of simplicity, the $C_L$ time history corresponding to the first test case (figure 6.29) is discussed in what follows. Nevertheless, equivalent analysis can be done for the rest of the test cases.

![Figure 6.29: Time history of the lift coefficient for test cases 1 and 2.](image-url)
Figure 6.30: Time history of the lift coefficient for test cases 1 and 3.

Figure 6.31: Time history of the lift coefficient for test cases 4 and 5.

Since the airfoil is symmetric and has a zero AoA, the variations of the lift coefficient are purely created due to the interaction with the vortices. Thus, $C_L = 0$ for $t/t_{ref} \leq 15$ and $t/t_{ref} \geq 75$: the flow around the airfoil is not perturbed by the wake vortices, which are far away from it. For $t/t_{ref} > 15$, the vortices advance in the domain. When the right one starts to disturb the flow, there is a lift growth, because the vortex rotates counter-clockwise. The closer is the airfoil to $r_c$, the stronger is the velocity field of the vortex, as it is shown in figure 6.32(a) for the local $z-$component of the velocity (plotted in blue). As the right vortex advances in time, stronger loads act on the airfoil, and
there is a progressive growth of the lift. The maximum $C_L$ corresponds to the maximum value of the tangential velocity at $r_c$ (see figure 4.5).

![Diagram](image.png)

(a) Counter-clockwise flow, before the interaction with the core of the right vortex. (b) Clockwise flow, after the interaction with the core of the right vortex.

**Figure 6.32:** Local $z$–component of the velocity field (in blue) of the right vortex.

The situation corresponds to the first test case, and it is not drawn at scale.

Just after interacting with the core of the right vortex, the direction of the flow changes, as it is shown for the tangential velocity in figure 4.5. The situation resembles the one from figure 6.32(b), which results in a strong lift decrease. Once the right vortex is away from the airfoil, its influence on the flow decreases. Hence, the loads applied on the airfoil also decrease, and the value of $C_L$ grows for $t/t_{ref} \in [35, 45]$. From $t/t_{ref} = 46$, the flow starts to be influenced by the left vortex, which rotates clockwise (see figure 4.5). Thus, the lift decreases, and after the airfoil interacts with the core of the vortex, the direction of the flow changes and the lift grows. From $t/t_{ref} = 56$, the influence of the left vortex on the airfoil decreases, and the initial lift is recovered, when both wake vortices are far away from the airfoil.

Notice the difference between figures 6.29 and 6.31: the values of $C_L$ predicted with the Lamb-Oseen model are greater than the ones predicted with the Burnham-Hallock model. This agrees with figure 4.6, where for the same value of $r_c$, the maximum tangential velocity prescribed by the Lamb-Oseen model is greater than the one prescribed by the Burnham-Hallock model. Hence, the greater the maximum tangential velocity, the greater is the maximum lift.

The convergence of the solution is analysed for the first test case in figures 6.33, 6.34 and 6.35. Similar results have been obtained for the rest of test cases. In the case of the steady simulation (figure 6.33), the residual is of order $10^{-14}$ after 225 MG-cycles, while the value of $C_L$ is constant. A central scheme with matrix dissipation is used to achieve convergence without much computational effort both for the steady and the unsteady simulations.
In the unsteady simulation, 20 MG-cycles in pseudo-time have been performed within each physical time step $\Delta t$, where $\Delta t = 0.0001$ s. Figure 6.34 shows that the residual decreases some orders of magnitude for each physical time step. This is sufficient to reach the pseudo-steady state, as indicated by the behaviour of $C_L$ in figure 6.35.

**Figure 6.33:** Convergence history of the residual and the lift coefficient for the steady simulation of test case 1.

**Figure 6.34:** Convergence history of the residual and the lift coefficient for the unsteady simulation of test case 1.
Figure 6.35: Zoom-in of figure 6.34.
Chapter 7

Conclusions

7.1 Summary of the results

This section presents an overview of the main results achieved in the numerical computations of this thesis, where the accuracy of the DVA has been compared to the RAA for wind gusts, and the implementation of the DVA to simulate wake vortex encounters has been verified. As explained in chapter 6, this has been done through encounters between airfoils or aircraft and atmospheric disturbances. In particular, the following simulations have been performed:

- Vertical gust encounters with a 2D NACA0012-HTP configuration.
- Vertical gust encounters with the 3D LamAiR aircraft.
- Lateral gust encounters with the 3D LamAiR aircraft.
- Wake vortex encounters with a 2D NACA0012 airfoil.

As stated in chapter 1, one of the objectives of this thesis is to reduce the computational cost of the simulations. This has been mainly done by varying the number of MG-cycles depending on the requirements of the computation and by using the Chimera technique to set up the grids and enable the movement of a high-resolution grid for a gust.

Vertical gust encounters with a 2D NACA0012-HTP configuration

The work done in [12] has been taken as a starting point for the 2D comparison of the DVA and RAA. However, the grid density study and the grid set-up have been improved, which has resulted in a reduction of the computational cost. In order to check the influence of compressibility, two different on-flow Mach numbers have been
selected: for $Ma_\infty = 0.28$, there is a nearly incompressible flow, while compressibility effects are expected for $Ma_\infty = 0.78$.

The two gust approaches have been compared in terms of the maximum lift coefficient for both Mach numbers and three different gust wavelengths, through the prediction error of the DVA. On the one hand, the effect of compressibility results in greater discrepancies between the DVA and the RAA for $Ma_\infty = 0.78$, independently of the wavelength of the gust. On the other hand, the greatest differences between both methods correspond to the smallest wavelength, independent of the Mach number. Therefore, a safe range of validity of the DVA corresponds to gust wavelengths greater than or equal to two reference chord lengths and an on-flow Mach number less than or equal to 0.78. These results agree with the ones from [12].

**Vertical gust encounters with the 3D LamAiR aircraft**

In order to study the influence of the fuselage in the comparison of the DVA and the RAA, 3D computations have been made with $Ma_\infty = 0.78$ and a vertical gust. The number of nodes of the simulation is reduced by only considering one half of the aircraft, so the computation is less costly.

The two gust modelling methods are compared in terms of the maximum lift coefficient, which is related to the interaction with the wing. The results obtained do not agree with the ones from the 2D simulations, which is totally unexpected. In particular, the DVA results are acceptable for all the wavelengths tested, since the DVA prediction errors are surprisingly small. In addition, the agreement between the DVA and the RAA is better for a gust wavelength equal to two reference chord lengths than to one reference chord length. Presumably, there are lift cancellations due to the global character of this magnitude that cause these unexpected results.

**Lateral gust encounters with the 3D LamAiR aircraft**

Simulations involving the complete aircraft and a lateral gust have been performed with $Ma_\infty = 0.78$. In this case, the DVA and the RAA have been compared in terms of the maximum lateral force coefficient, which is related to the interaction with the VTP.

The wake vortices that emerge from the tips of the wings influence the flow, so the gust is perturbed by the presence of the vortices (from the interaction with the wings until the end of the computation). On the one hand, when the gust has not reached the wings, the discrepancies between the DVA and the RAA are greater for small wavelengths. On the other hand, during the interaction with the VTP, the most important discrepancies between both methods are found for the largest wavelength (four reference chord lengths). This change of behaviour is presumably associated to the presence of the
vortices, whose effect on the gust is more important for large wavelengths. Nevertheless, the computations can also be influenced by the resolution of the near field grid, which is maybe insufficient. Therefore, no range of validity for the DVA and a lateral gust can be given at the moment, because the resolution of the grid should be improved.

Wake vortex encounters with a 2D NACA0012 airfoil

The DVA has been implemented in TAU to enable encounters between an aircraft and the wake vortices from another aircraft. The implementation has been done in the most general way, such that encounters between aircraft flying in different directions can be simulated.

To verify the correctness of the implementation, some equivalent test cases have been defined. To reduce the computational cost, the simulations consider an inviscid flow and a NACA0012 airfoil. The time history of the lift coefficient that has been obtained for each pair of equivalent test cases is the same, so the implementation can be considered successful.

7.2 Future work

The work that has been done in this thesis can be extended and improved in the future. For doing so, it is worth to take the following suggestions into account:

- Study the global distribution of the lift force in the span direction. This is necessary to ensure that the small values of $e_{C_L,\text{max}}$ obtained for the interaction between the LamAiR aircraft and a vertical gust are really caused by lift cancellations. In addition, it might be recommended to compute the prediction error of the DVA in terms of another magnitude, like for instance the wing root bending moment.

- Improve the resolution of the near field grid for the LamAiR aircraft. As said in chapter 5, the lateral gust simulations consider only two hole geometries, so during the interaction with the aircraft, the gust is not contained in the nodes of the moving grid but in the nodes of the near field grid. Thus, it could happen that not only the wake vortices but also the lower resolution of the near field grid have an influence on the numerical results. In order to discard this, the resolution of the near field grid should be improved.

- Do more realistic simulations by considering not only the influence of the gust and the aircraft’s aerodynamics, but also the coupling to other relevant disciplines. In particular, the next logical step is to consider the coupling to flight mechanics and also the fluid-structure interaction.
• Implement the so-called Source Gust Model, which is a third method for gust modelling developed in [17]. Like the RAA, this method allows a mutual interaction of gust and aircraft, but at a lower computational cost. Therefore, the main disadvantage of the RAA (very costly simulations) could be avoided.

• Implement the RAA to allow encounters with wake vortex turbulence, by prescribing unsteady boundary conditions for the velocity components, the pressure and the density at the farfield. As in the case of gust encounters, the RAA has to be compared to the DVA, to set the range of validity of the last approach.
Appendix A

Conservation laws for moving grids

If a control volume $\Omega$ moves with a certain velocity $v_b$, the conservation laws of mass, momentum and energy can be written in integral form as

$$\frac{\partial}{\partial t} \int_{\Omega} \rho d\Omega + \oint_{\partial \Omega} \rho [(v - v_b) \cdot \mathbf{n}] dS = 0$$

$$\frac{\partial}{\partial t} \int_{\Omega} \rho v d\Omega + \oint_{\partial \Omega} \rho v [(v - v_b) \cdot \mathbf{n}] dS = - \oint_{\partial \Omega} p n dS + \oint_{\partial \Omega} (\tau \cdot \mathbf{n}) dS$$

$$\frac{\partial}{\partial t} \int_{\Omega} \rho E d\Omega + \oint_{\partial \Omega} \rho E [(v - v_b) \cdot \mathbf{n}] dS = \oint_{\partial \Omega} k (\nabla T \cdot \mathbf{n}) dS$$

$$+ \int_{\Omega} \dot{q}_h d\Omega - \oint_{\partial \Omega} p (v \cdot \mathbf{n}) dS + \oint_{\partial \Omega} (\tau \cdot v) \cdot \mathbf{n} dS$$

Equations (A.1) can be rewritten in a more compact way

$$\frac{\partial}{\partial t} \int_{\Omega} W d\Omega + \oint_{\partial \Omega} (F_c - F_v) dS = 0$$

(A.2)

where $W$ is the vector of the conservative variables, which has the following components

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

(A.3)
For the sake of convenience, let $V$ and $V_b$ be the contravariant velocities of the flow and $\Omega$, respectively

\[ V = v \cdot n = n_x u + n_y v + n_z w \tag{A.4} \]
\[ V_b = v_b \cdot n = n_x \frac{\partial x}{\partial t} \bigg|_\Omega + n_y \frac{\partial y}{\partial t} \bigg|_\Omega + n_z \frac{\partial z}{\partial t} \bigg|_\Omega \tag{A.5} \]

where $n$ is the unit normal vector of a control volume face. Hence, note that the contravariant velocity is the velocity normal to the surface element $dS$. From $V$ and $V_b$, the contravariant velocity relative to the motion of $\Omega$ (in the case of this thesis, to the motion of the grid) can also be defined

\[ V_r = n_x u + n_y v + n_z w - V_b = V - V_b \tag{A.6} \]

Then, the term $F_c$, which is the so-called vector of the convective fluxes, is

\[ F_c = \begin{bmatrix} \rho V_r \\ \rho u V_r + n_x p \\ \rho v V_r + n_y p \\ \rho w V_r + n_z p \\ \rho H V_r + V_b p \end{bmatrix} \tag{A.7} \]

where $p$ stands for the static pressure, and $H$ stands for the total enthalpy

\[ H = h + \frac{|v|^2}{2} \tag{A.8} \]

being $h$ the enthalpy defined in (2.5).

Finally, the term $F_v$ is the vector of viscous fluxes

\[ F_v = \begin{bmatrix} 0 \\ n_x \tau_{xx} + n_y \tau_{xy} + n_z \tau_{xz} \\ n_x \tau_{yx} + n_y \tau_{yy} + n_z \tau_{yz} \\ n_x \tau_{zx} + n_y \tau_{zy} + n_z \tau_{zz} \\ n_x \Theta_x + n_y \Theta_y + n_z \Theta_z \end{bmatrix} \tag{A.9} \]

where

\[ \Theta_x = u \tau_{xx} + v \tau_{xy} + w \tau_{xz} + k \frac{\partial T}{\partial x} \]
\[ \Theta_y = u \tau_{yx} + v \tau_{yy} + w \tau_{yz} + k \frac{\partial T}{\partial y} \tag{A.10} \]
\[ \Theta_z = u \tau_{zx} + v \tau_{zy} + w \tau_{zz} + k \frac{\partial T}{\partial z} \]
describe the work of the viscous stresses and the heat conduction in the fluid. The terms $\tau_{ij}$ are the components of the viscous stress tensor $\mathbf{\tau}$. For aerospace applications, like in this work, the air can be assumed to behave like a Newtonian fluid, which is defined in equations (2.2) - (2.3).
Appendix B

Conservation laws averaging

When modelling a turbulent fluid, it is advisable to perform the so-called Reynolds averaging (given by (2.12) or (2.13)) to the density $\rho$ and the static pressure $p$, and the Favre averaging (2.15) to the rest of the flow variables. In that case, the compact form of the Navier-Stokes equations (2.9), which is

$$\frac{\partial}{\partial t} \int_\Omega W d\Omega + \int_{\partial\Omega} (F_c - F_v) dS = 0$$

becomes

$$\frac{\partial}{\partial t} \int_\Omega \bar{W} d\Omega + \int_{\partial\Omega} (\bar{F}_c - \bar{F}_v) dS = 0$$

and these equations are usually abbreviated as Unsteady Reynolds Averaged Navier-Stokes (URANS) equations.

The bar over each of the terms means that they have been averaged following the rules given in [5]. For example, the vector of the conservative variables becomes

$$\bar{\bar{W}} = \begin{bmatrix} \bar{\bar{\rho}} \\ \bar{\bar{\rho}\bar{u}} \\ \bar{\bar{\rho}\bar{v}} \\ \bar{\bar{\rho}\bar{w}} \\ \bar{\bar{\rho}\bar{E}} \end{bmatrix} = \begin{bmatrix} \bar{\bar{\rho}} \\ \bar{\bar{\rho}\bar{u}} \\ \bar{\bar{\rho}\bar{v}} \\ \bar{\bar{\rho}\bar{w}} \\ \bar{\bar{\rho}\bar{E}} \end{bmatrix}$$

which is now referred as vector of averaged conservative variables. Note that, as stated previously, the density $\rho$ is Reynolds-averaged (mean value indicated by $\bar{\bar{\rho}}$), while the rest of the variables in $\bar{\bar{W}}$ are Favre-averaged (mean value indicated by $\bar{\bar{\rho}}$). The same
happens for the vector of averaged convective fluxes $\vec{F}_c$,

\[
\vec{F}_c = \begin{bmatrix}
\tilde{\rho} \tilde{V}_r \\
\tilde{\rho} \tilde{u} \tilde{V}_r + n_x \tilde{p} \\
\tilde{\rho} \tilde{v} \tilde{V}_r + n_y \tilde{p} \\
\tilde{\rho} \tilde{w} \tilde{V}_r + n_z \tilde{p} \\
\tilde{\rho} H \tilde{V}_r + \tilde{V}_b \tilde{p}
\end{bmatrix}
\]  

(B.4)

Finally, for the viscous contribution

\[
\vec{F}_v = \begin{bmatrix}
0 \\
n_x \left( \tilde{\tau}_{xx} - \tilde{\rho} u'' u'' \right) + n_y \left( \tilde{\tau}_{xy} - \tilde{\rho} u'' v'' \right) + n_z \left( \tilde{\tau}_{xz} - \tilde{\rho} u'' w'' \right) \\
n_x \left( \tilde{\tau}_{yx} - \tilde{\rho} v'' u'' \right) + n_y \left( \tilde{\tau}_{yy} - \tilde{\rho} v'' v'' \right) + n_z \left( \tilde{\tau}_{yz} - \tilde{\rho} v'' w'' \right) \\
n_x \left( \tilde{\Theta}_x - \tilde{\rho} u'' h'' \right) + n_y \left( \tilde{\Theta}_y - \tilde{\rho} v'' h'' \right) + n_z \left( \tilde{\Theta}_z - \tilde{\rho} w'' h'' \right)
\end{bmatrix}
\]  

(B.5)

where $''$ refers to the fluctuating part of Favre decomposition. The terms $\tilde{\Theta}_x$, $\tilde{\Theta}_y$ and $\tilde{\Theta}_z$ are

\[
\tilde{\Theta}_x = u'' \tilde{\tau}_{xx} + v'' \tilde{\tau}_{xy} + w'' \tilde{\tau}_{xz} + k \frac{\partial \tilde{T}}{\partial x}
\]

\[
\tilde{\Theta}_y = u'' \tilde{\tau}_{yx} + v'' \tilde{\tau}_{yy} + w'' \tilde{\tau}_{yz} + k \frac{\partial \tilde{T}}{\partial y}
\]

\[
\tilde{\Theta}_z = u'' \tilde{\tau}_{zx} + v'' \tilde{\tau}_{zy} + w'' \tilde{\tau}_{zz} + k \frac{\partial \tilde{T}}{\partial z}
\]  

(B.6)

Due to the averaging, there is an additional contribution to the viscous stress tensor $\tau$, which is called Favre-averaged Reynolds-stress tensor. Thanks to the eddy viscosity hypothesis, its components $\tau^F_{ij}$ can be written as

\[
\tau^F_{ij} = -\tilde{\rho} v''_i v''_j = \mu_T \left( \frac{\partial \tilde{v}_i}{\partial x_j} + \frac{\partial \tilde{v}_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial \tilde{v}_k}{\partial x_k} \right) - \frac{2}{3} \delta_{ij} \tilde{p} \tilde{K}
\]  

(B.7)

where $\delta_{ij}$ is the Kronecker delta, $\tilde{K}$ is the Favre-averaged turbulent kinetic energy and $\mu_T$ is the eddy viscosity. The last two variables have to be computed through a turbulence model.
Bibliography


