Development of CFD Algorithms for Transient and Steady Aerodynamics

Fergal Boyle [Thesis]
NUI, Galway

Follow this and additional works at: https://arrow.tudublin.ie/engmecdesignoth

Part of the Mechanical Engineering Commons

Recommended Citation

This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 License
Development of CFD Algorithms for Transient and Steady Aerodynamics

Fergal J. Boyle

A thesis submitted to the National University of Ireland in partial fulfilment of the requirements for the Degree of

DOCTOR OF PHILOSOPHY

in

MECHANICAL ENGINEERING

Department of Mechanical Engineering
National University of Ireland, Galway

December 1999

Supervisor of Research: Dr. J. A. Eaton
Director of Research: Prof. J. F. McNamara
ABSTRACT

Research carried out in the United States and in Europe over the last 20 years has led to the development of the advanced propeller, a small-diameter, highly-loaded, multi-bladed, swept, variable-pitch propeller, that can achieve potential fuel savings of 30% over an equivalent technology turbofan engine at competitive speed and altitudes.

In this work an implicit finite-volume algorithm is developed for predicting the transient flow around an advanced propeller under asymmetric inflow conditions. The development of the unsteady propeller algorithm evolves from a family of methods for progressively more complex applications. In total six algorithms are developed: three steady and three unsteady. Each solves the Euler equations using a cell-centered, central-difference, finite-volume scheme in transformed space. Adaptive artificial dissipation terms are added both for stability and for accuracy. The steady methods employ an explicit, multistage, time-stepping scheme. A fully implicit time discretisation is employed in the time-dependent algorithms to avoid the maximum time step limitation typical of explicit schemes. The implicit equations are iteratively inverted at each physical time step by casting them in a modified steady form and marching to steady-state in a pseudo time. Local time-stepping, implicit residual averaging and multigrid are employed for convergence acceleration.

Results from a range of test cases computed by each of the algorithms are presented and compared with wind tunnel data and with the predictions of other researchers. The unsteady propeller algorithm is used to compute the flowfields around advanced propellers at incidence. The results demonstrate that this family of algorithms is useful for inviscid flowfield analyses and that the unsteady propeller algorithm can provide further insight into the aerodynamics of advanced propellers.
ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor, John Eaton, for providing the excellent facilities needed to complete the complex work reported in this thesis. I would also like to thank him for his continued advice and encouragement during the course of this work.

I am very grateful to Eli Turkel from Tel-Aviv University for his generous help and for always answering my questions so promptly.

Thanks also to the following people for being so receptive to my requests for help: Marcello Amato (CIRA), Graham Holmes (General Electric), David Whitfield and Mark Janus (Mississippi State University), Michael Giles (Oxford University), Rakesh Srivastava (NASA Lewis), Robert Menthe (Hamilton Standard), Norbert Kroll (DLR), V. Venkatakrishnan (Boeing), Jan Vos (Swiss Federal Institute of Technology), Duane Melson (NASA Langley), Feng Liu (University of California at Irvine), Alan Shih (Catalpa Research), Harry Brouwer and Johan Schulten (NLR), Joao Frota (Aerospatiale Matra) and all of the APIAN partners.

I am also grateful to my colleagues, especially Rory Noone, Basil Regan, Ronan Walshe, Micheál O Flaherty, John Taylor, Aonghus Ó Nia, David MacManus, Nathan Quinlan, David Dunleavy, Joseph Kilcullen, Triona Lally, Anthony Reynolds and Paddy Melia. A special word of thanks is due to David Kavanagh for the assistance he has willingly provided on many occasions.

Thanks are also due to my many friends, especially Liam, Tom, Paddy and Mick for the good times we’ve had together.

I am very grateful to my family for their continued support and encouragement over many years.

Finally, I would especially like to thank Louise for being with me through it all and for agreeing to be with me through even more.

This work was partially funded by the European Union within the Fourth Framework programme, under the BRITE-EURAM project APIAN.
TABLE OF CONTENTS

Abstract ........................................................ ii
Acknowledgements ............................................... iii
Table of Contents ............................................... iv
List of Figures ................................................ ix
List of Tables ................................................... xiv
Nomenclature ................................................... xvi

1 Introduction .................................................. 1
   1.1 Background .............................................. 1
   1.2 Advanced Propellers .................................... 1
      1.2.1 Initial Concept .................................... 1
      1.2.2 United States Research .......................... 2
      1.2.3 European Research ............................... 2
      1.2.4 Current Status .................................. 4
   1.3 Previous Numerical Methods for Advanced Propeller Calculations .......... 4
   1.4 Present Research ....................................... 5
      1.4.1 Objective ......................................... 5
      1.4.2 Strategy ......................................... 6
      1.4.3 Layout of Thesis ................................ 6
   1.5 Novel Aspects of Research ..................................... 6
   1.6 Technical Publications .................................... 7

2 Governing Equations of Inviscid Flow ............................ 8
   2.1 Introduction ............................................. 8
   2.2 Formulation of the Governing Equations ........................ 9
      2.2.1 Introduction ..................................... 9
      2.2.2 Integral Form ................................... 10
4.2 Physical and Transformed Computational Domains ......................... 39
  4.2.1 Airfoil Computational Domain .................................. 39
  4.2.2 Wing Computational Domain .................................... 40
  4.2.3 Propeller Computational Domain ................................ 40

4.3 Boundary Conditions ................................................. 46
  4.3.1 Implementation of Boundary Conditions .......................... 46
  4.3.2 Solid Surface Boundary Condition .............................. 47
  4.3.3 Symmetry Plane Boundary Condition ............................ 49
  4.3.4 Periodic Boundary Condition .................................... 50
  4.3.5 Freestream Boundary Condition .................................. 51
  4.3.6 Inflow/Outflow Boundary Condition .............................. 52
    4.3.6.1 Riemann Invariants ....................................... 52
    4.3.6.2 Far-Field Correction for Airfoil Calculations .......... 53
  4.3.7 Radial Equilibrium Boundary Condition .......................... 54
  4.3.8 Singular Axis Boundary Condition ................................ 55

4.4 Initial Conditions .................................................. 56

4.5 Summary ......................................................... 56

5 Convergence Acceleration Techniques .................................... 57
  5.1 Introduction ...................................................... 57
  5.2 Local Time-Stepping .............................................. 57
  5.3 Implicit Residual Averaging ....................................... 58
    5.3.1 General Formulation ....................................... 58
    5.3.2 Improved Smoothing Coefficients for Two-Dimensional
          Calculations ............................................... 59
    5.3.3 Boundary Treatment ....................................... 60
  5.4 Multigrid ........................................................ 60
    5.4.1 Introduction ............................................... 60
    5.4.2 Full Approximation Storage Scheme .......................... 61
    5.4.3 Cycling Strategies ....................................... 62
    5.4.4 Full Multigrid Strategy ................................... 65

5.5 Summary ......................................................... 66
6 Presentation of Results ................................................... 67

6.1 Introduction ......................................................... 67

6.2 Two-Dimensional Fixed Airfoil Calculations ....................... 67

6.2.1 Introduction .................................................... 67

6.2.2 Results of Test Case 1: NACA 0012 Airfoil ..................... 68

6.2.3 Results of Test Case 2: NACA 0012 Airfoil ..................... 70

6.2.4 Results of Test Case 3: NACA 64A010 Airfoil .................. 71

6.3 Two-Dimensional Oscillating Airfoil Calculations ................. 83

6.3.1 Introduction .................................................... 83

6.3.2 Results of Test Case: NACA 64A010 Airfoil ................... 84

6.4 Three-Dimensional Fixed Wing Calculations ...................... 94

6.4.1 Introduction .................................................... 94

6.4.2 Results of Test Case 1 and Test Case 2: ONERA M6 Wing ...... 95

6.5 Three-Dimensional Oscillating Wing Calculations ................. 109

6.5.1 Introduction .................................................... 109

6.5.2 Results of Test Case 1: ONERA M6 Wing ...................... 110

6.5.3 Results of Test Case 2: NACA 64A010 Wing .................... 111

6.6 Steady Propeller Calculations: Axisymmetric Inflow .......... 120

6.6.1 Introduction .................................................... 120

6.6.2 NACA 10-(3)(066)-033 Propeller ............................ 121

6.6.2.1 Introduction ........................................... 121

6.6.2.2 Results of Test Case .................................... 122

6.6.3 APIAN Propeller ................................................ 123

6.6.3.1 Introduction ........................................... 123

6.6.3.2 PIV Measurements ........................................ 123

6.6.3.3 Results of Test Case 1 .................................... 125

6.6.3.4 Results of Test Case 2 .................................... 125

6.6.3.5 Results of Test Case 3 .................................... 126

6.7 Transient Propeller Calculations: Asymmetric Inflow .......... 142

6.7.1 Introduction .................................................... 142

6.7.2 Results of Test Case 1: Take-Off Conditions .................. 143

6.7.3 Results of Test Case 2: Cruise Conditions ..................... 144
Figure 1.1  Single-rotation tractor advanced propeller.
Figure 1.2  Counter-rotation pusher advanced propeller.
Figure 2.1  Finite control volume fixed in space.
Figure 2.2  Physical and transformed computational domains for fixed airfoil flow calculations.
Figure 3.1  Cell-centered finite-volume scheme.
Figure 4.1  Physical computational domain for the steady and unsteady airfoil flowfield calculations.
Figure 4.2  Transformed computational domain and applied boundary conditions for the steady and unsteady airfoil flowfield calculations.
Figure 4.3  Physical computational domain for the steady and unsteady wing flowfield calculations.
Figure 4.4  Transformed computational domain and applied boundary conditions for the steady and unsteady wing flowfield calculations.
Figure 4.5  Physical computational domain for a steady two-bladed propeller flowfield calculation.
Figure 4.6  Transformed computational domain and applied boundary conditions for a steady two-bladed propeller flowfield calculation.
Figure 4.7  Physical computational domain for an unsteady two-bladed propeller flowfield calculation.
Figure 4.8  Transformed computational domain for an unsteady two-bladed propeller flowfield calculation.
Figure 4.9  Ghost cell locations for implementation of boundary conditions.
Figure 4.10 Implementation of the solid surface boundary condition.
Figure 4.11 Implementation of the symmetry plane boundary condition.
Figure 4.12 Implementation of the periodic boundary condition.
Figure 4.13  Implementation of the inflow/outflow boundary condition.

Figure 4.14  Implementation of the singular axis boundary condition.

Figure 5.1  Three-, four- and five-level multigrid V-cycles. E, Euler calculation; R, restriction; P, prolongation.

Figure 5.2  Three-, four- and five-level multigrid W-cycles. E, Euler calculation; R, restriction; P, prolongation.

Figure 6.1  Inner part of the 224x32 cell hyperbolic C-grid around the NACA 0012 airfoil.

Figure 6.2  Inner part of the 224x32 cell hyperbolic C-grid around the NACA 64A010 airfoil.

Figure 6.3  Grid structure in the nose region of the NACA 0012 airfoil.

Figure 6.4  Grid structure in the nose region of the NACA 64A010 airfoil.

Figure 6.5  Convergence history of the steady calculation of the flow around a fixed NACA 0012 airfoil. $Ma_\infty = 0.63$ and $\alpha = 2.0^\circ$.

Figure 6.6  Development of the lift coefficient during the steady calculation of the flow around a fixed NACA 0012 airfoil. $Ma_\infty = 0.63$ and $\alpha = 2.0^\circ$.

Figure 6.7  Variation of computed surface pressure with chordwise position for the NACA 0012 airfoil. $Ma_\infty = 0.63$ and $\alpha = 2.0^\circ$.

Figure 6.8  Variation of computed total pressure loss with chordwise position for the NACA 0012 airfoil. $Ma_\infty = 0.63$ and $\alpha = 2.0^\circ$.

Figure 6.9  Mach number contours around the NACA 0012 airfoil. $Ma_\infty = 0.63$, $\alpha = 2.0^\circ$ and $\Delta Ma = 0.03$.

Figure 6.10  Pressure contours around the NACA 0012 airfoil. $Ma_\infty = 0.63$, $\alpha = 2.0^\circ$ and $\Delta p = 0.03$.

Figure 6.11  Convergence history of the steady calculation of the flow around a fixed NACA 0012 airfoil. $Ma_\infty = 0.8$ and $\alpha = 1.25^\circ$.

Figure 6.12  Development of the lift coefficient during the steady calculation of the flow around a fixed NACA 0012 airfoil. $Ma_\infty = 0.8$ and $\alpha = 1.25^\circ$. 
Figure 6.13 Variation of computed surface pressure with chordwise position for the NACA 0012 airfoil. $M_{\infty} = 0.8$ and $\alpha = 1.25^\circ$.

Figure 6.14 Variation of computed total pressure loss with chordwise position for the NACA 0012 airfoil. $M_{\infty} = 0.8$ and $\alpha = 1.25^\circ$.

Figure 6.15 Mach number contours around the NACA 0012 airfoil. $M_{\infty} = 0.8$, $\alpha = 1.25^\circ$ and $\Delta M_a = 0.05$.

Figure 6.16 Pressure contours around the NACA 0012 airfoil. $M_{\infty} = 0.8$, $\alpha = 1.25^\circ$ and $\Delta p = 0.05$.

Figure 6.17 Convergence history of the steady calculation of the flow around a fixed NACA 64A010 airfoil. $M_{\infty} = 0.796$ and $\alpha = 0.0^\circ$.

Figure 6.18 Development of the lift coefficient during the steady calculation of the flow around a fixed NACA 64A010 airfoil. $M_{\infty} = 0.796$ and $\alpha = 0.0^\circ$.

Figure 6.19 Variation of computed surface pressure with chordwise position for the NACA 64A010 airfoil. $M_{\infty} = 0.796$ and $\alpha = 0.0^\circ$.

Figure 6.20 Variation of total pressure loss with chordwise position for the NACA 0012 airfoil. $M_{\infty} = 0.796$ and $\alpha = 0.0^\circ$.

Figure 6.21 Mach number contours around the NACA 64A010 airfoil. $M_{\infty} = 0.796$, $\alpha = 0.0^\circ$ and $\Delta M_a = 0.03$.

Figure 6.22 Pressure contours around the NACA 64A010 airfoil. $M_{\infty} = 0.796$, $\alpha = 0.0^\circ$ and $\Delta p = 0.03$.

Figure 6.23 Inner part of the 224x32 cell hyperbolic C-grid around the NACA 64A010 airfoil.

Figure 6.24 Computed variation of lift coefficient with non-dimensional time for the NACA 64A010 airfoil. $M_{\infty} = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_o = 1.01^\circ$ and $k = 0.202$.

Figure 6.25 Computed variation of lift coefficient with angle of incidence for the NACA 64A010 airfoil. $M_{\infty} = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_o = 1.01^\circ$ and $k = 0.202$.

Figure 6.26 Computed variation of lift coefficient with angle of incidence for the NACA 64A010 airfoil near the maximum angle of incidence. $M_{\infty} = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_o = 1.01^\circ$ and $k = 0.202$. 
Figure 6.27 Comparison of computed variation of lift coefficient with angle of incidence for the NACA 64A010 airfoil with the measurements of Reference 108. $M_{\infty} = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 1.01^\circ$ and $k = 0.202$.

Figure 6.28 Comparison of computed variation of lift coefficient with angle of incidence for the NACA 64A010 airfoil with Reference 64. $M_{\infty} = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 1.01^\circ$ and $k = 0.202$.

Figure 6.29 Comparison of computed variation of lift coefficient with angle of incidence for the NACA 64A010 airfoil with References 87 and 109. $M_{\infty} = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 1.01^\circ$ and $k = 0.202$.

Figure 6.30 An extract of the convergence history during the unsteady calculation of the flow around a NACA 64A010 airfoil. $M_{\infty} = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 1.01^\circ$ and $k = 0.202$.

Figure 6.31 Comparison of computed and measured surface pressure on the upper surface of the NACA 64A010 airfoil. $M_{\infty} = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 1.01^\circ$ and $k = 0.202$.

Figure 6.32 Comparison of computed and measured surface pressure on the upper surface of the NACA 64A010 airfoil. $M_{\infty} = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 1.01^\circ$ and $k = 0.202$.

Figure 6.33 Comparison of computed and measured surface pressure on the upper surface of the NACA 64A010 airfoil. $M_{\infty} = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 1.01^\circ$ and $k = 0.202$.

Figure 6.34 Comparison of computed and measured surface pressure on the upper surface of the NACA 64A010 airfoil. $M_{\infty} = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 1.01^\circ$ and $k = 0.202$.

Figure 6.35 Comparison of computed and measured surface pressure on the upper surface of the NACA 64A010 airfoil. $M_{\infty} = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 1.01^\circ$ and $k = 0.202$.

Figure 6.36 Comparison of computed and measured surface pressure on the upper surface of the NACA 64A010 airfoil. $M_{\infty} = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 1.01^\circ$ and $k = 0.202$. 

xii
Mach number contours around the NACA 64A010 airfoil during the third period of oscillation. $Ma_0 = 0.796$, $\alpha_m = 0.0\degree$, $\alpha_0 = 1.01\degree$, $k = 0.202$ and $\Delta Ma = 0.03$.

Inner part of the coarse (96x16x16) C-H grid around the ONERA M6 wing.

Inner part of the fine (192x32x32) C-H grid around the ONERA M6 wing.

Convergence history of the steady calculation of the flow around the ONERA M6 wing using the coarse grid. $Ma_0 = 0.699$ and $\alpha = 3.06\degree$.

Convergence history of the steady calculation of the flow around the ONERA M6 wing using the coarse grid. $Ma_0 = 0.84$ and $\alpha = 3.06\degree$.

Convergence history of the steady calculation of the flow around the ONERA M6 wing using the medium grid. $Ma_0 = 0.699$ and $\alpha = 3.06\degree$.

Convergence history of the steady calculation of the flow around the ONERA M6 wing using the medium grid. $Ma_0 = 0.84$ and $\alpha = 3.06\degree$.

Convergence history of the steady calculation of the flow around the ONERA M6 wing using the fine grid. $Ma_0 = 0.699$ and $\alpha = 3.06\degree$.

Convergence history of the steady calculation of the flow around the ONERA M6 wing using the fine grid. $Ma_0 = 0.84$ and $\alpha = 3.06\degree$.

Comparison of computed and measured surface pressure for the ONERA M6 wing using the coarse grid. $Ma_0 = 0.699$ and $\alpha = 3.06\degree$.

Comparison of computed and measured surface pressure for the ONERA M6 wing using the medium grid. $Ma_0 = 0.699$ and $\alpha = 3.06\degree$.

Comparison of computed and measured surface pressure for the ONERA M6 wing using the fine grid. $Ma_0 = 0.699$ and $\alpha = 3.06\degree$.

Comparison of computed and measured surface pressure for the ONERA M6 wing using the coarse grid. $Ma_0 = 0.84$ and $\alpha = 3.06\degree$.

Comparison of computed and measured surface pressure for the ONERA M6 wing using the medium grid. $Ma_0 = 0.84$ and $\alpha = 3.06\degree$.

Comparison of computed and measured surface pressure for the ONERA M6 wing using the fine grid. $Ma_0 = 0.84$ and $\alpha = 3.06\degree$. 


Figure 6.52  Mach number contours on the upper surface of the ONERA M6 wing calculated using the fine grid. $Ma_{\infty} = 0.699$, $\alpha = 3.06^\circ$ and $\Delta Ma = 0.05$.

Figure 6.53  Mach number contours on the upper surface of the ONERA M6 wing calculated using the fine grid. $Ma_{\infty} = 0.84$, $\alpha = 3.06^\circ$ and $\Delta Ma = 0.05$.

Figure 6.54  Inner part of the 144x24x24 C-H grid around the ONERA M6 wing.

Figure 6.55  Inner part of the 144x24x24 C-H grid around the NACA 64A010 wing.

Figure 6.56  Computed variation of lift coefficient with non-dimensional time for the ONERA M6 wing. $Ma_{\infty} = 0.84$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 5.0^\circ$ and $k = 0.1$.

Figure 6.57  Computed variation of lift coefficient with angle of incidence for the ONERA M6 wing. $Ma_{\infty} = 0.84$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 5.0^\circ$ and $k = 0.1$.

Figure 6.58  Variation of computed surface pressure at $\zeta = 0.2$ with angular position for the ONERA M6. $Ma_{\infty} = 0.84$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 5.0^\circ$ and $k = 0.1$.

Figure 6.59  Variation of computed surface pressure at $\zeta = 0.9$ with angular position for the ONERA M6. $Ma_{\infty} = 0.84$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 5.0^\circ$ and $k = 0.1$.

Figure 6.60  Computed variation of lift coefficient with non-dimensional time for the NACA 64A010 wing. $Ma_{\infty} = 0.8$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 1.0^\circ$ and $k = 0.135$.

Figure 6.61  Computed variation of lift coefficient with angle of incidence for the NACA 64A010 wing. $Ma_{\infty} = 0.8$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 1.0^\circ$ and $k = 0.135$.

Figure 6.62  Variation of computed surface pressure at $\zeta = 0.5$ with angular position for the NACA 64A010 wing. $Ma_{\infty} = 0.8$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 1.0^\circ$ and $k = 0.135$.

Figure 6.63  Variation of computed surface pressure at $\zeta = 0.77$ with angular position for the NACA 64A010 wing. $Ma_{\infty} = 0.8$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 1.0^\circ$ and $k = 0.135$.

Figure 6.64  Comparison of computed variation of lift coefficient with angle of incidence for the NACA 64A010 wing with Singh. $Ma_{\infty} = 0.8$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 1.0^\circ$ and $k = 0.135$.

Figure 6.65  The two-bladed NACA 10-(3)(066)-033 propeller.

Figure 6.66  The six-bladed APIAN advanced propeller.

Figure 6.67  A section of a C-H grid in an inter-blade region of the NACA 10-(3)(066)-033 propeller for steady flowfield calculations.
Figure 6.68  A section of a C-H grid in an inter-blade region of the APIAN propeller for steady flowfield calculations.

Figure 6.69  Convergence history of the steady NACA 10-(3)(066)-033 propeller calculation. $Ma_{\infty} = 0.56$, $J = 2.3$ and $\beta_{3/4} = 45.0^\circ$.

Figure 6.70  Development of the thrust coefficient during the steady NACA 10-(3)(066)-033 propeller calculation. $Ma_{\infty} = 0.56$, $J = 2.3$ and $\beta_{3/4} = 45.0^\circ$.

Figure 6.71  Comparison of computed and measured surface pressure for the NACA 10-(3)(066)-033 propeller. $Ma_{\infty} = 0.56$, $J = 2.3$ and $\beta_{3/4} = 45.0^\circ$.

Figure 6.72  Contours of relative Mach number at the cell centres adjacent to the pressure surface of the NACA 10-(3)(066)-033 propeller blade. $Ma_{\infty} = 0.56$, $J = 2.3$, $\beta_{3/4} = 45.0^\circ$ and $\Delta Ma_{rel} = 0.05$.

Figure 6.73  Contours of relative Mach number at the cell centres adjacent to the suction surface of the NACA 10-(3)(066)-033 propeller blade. $Ma_{\infty} = 0.56$, $J = 2.3$, $\beta_{3/4} = 45.0^\circ$ and $\Delta Ma_{rel} = 0.05$.

Figure 6.74  Convergence history of the steady APIAN propeller calculation. $Ma_{\infty} = 0.7$, $J = 3.117$ and $\beta_{3/4} = 57.0^\circ$.

Figure 6.75  Development of the thrust coefficient during the steady APIAN propeller calculation. $Ma_{\infty} = 0.7$, $J = 3.117$ and $\beta_{3/4} = 57.0^\circ$.

Figure 6.76  Computed surface pressure for the APIAN propeller. $Ma_{\infty} = 0.7$, $J = 3.117$ and $\beta_{3/4} = 57.0^\circ$.

Figure 6.77  Contours of relative Mach number at the cell centres adjacent to the suction surface of the APIAN propeller blade. $Ma_{\infty} = 0.7$, $J = 3.117$, $\beta_{3/4} = 57.0^\circ$ and $\Delta Ma_{rel} = 0.05$.

Figure 6.78  Contours of relative Mach number at the cell centres adjacent to the pressure surface of the APIAN propeller blade. $Ma_{\infty} = 0.7$, $J = 3.117$, $\beta_{3/4} = 57.0^\circ$ and $\Delta Ma_{rel} = 0.05$.

Figure 6.79  Schematic of a PIV measurement system.

Figure 6.80  The APIAN propeller in the DNW Low Speed Tunnel.

Figure 6.81  A front view of the APIAN propeller and PIV measurement planes.

Figure 6.82  A plan view of the APIAN propeller and PIV measurement planes.
Figure 6.83 Comparison of computed and measured radial velocity variation with axial position in Plane 1. $Ma_{\infty} = 0.23, J = 1.11$ and $\beta_{3/4} = 40.4^\circ$.

Figure 6.84 Comparison of computed and measured axial velocity variation with axial position in Plane 1. $Ma_{\infty} = 0.23, J = 1.11$ and $\beta_{3/4} = 40.4^\circ$.

Figure 6.85 Comparison of computed and measured axial velocity variation with axial position in Plane 2. $Ma_{\infty} = 0.23, J = 1.11$ and $\beta_{3/4} = 40.4^\circ$.

Figure 6.86 Comparison of computed and measured axial velocity variation with axial position in Plane 2. $Ma_{\infty} = 0.23, J = 1.11$ and $\beta_{3/4} = 40.4^\circ$.

Figure 6.87 Comparison of computed and measured $v$ velocity variation with axial position in Plane 2. $Ma_{\infty} = 0.23, J = 1.11$ and $\beta_{3/4} = 40.4^\circ$.

Figure 6.88 Comparison of computed and measured $v$ velocity variation with axial position in Plane 2. $Ma_{\infty} = 0.23, J = 1.11$ and $\beta_{3/4} = 40.4^\circ$.

Figure 6.89 Comparison of computed and measured radial velocity variation with axial position in Plane 1. $Ma_{\infty} = 0.23, J = 1.2$ and $\beta_{3/4} = 40.4^\circ$.

Figure 6.90 Comparison of computed and measured axial velocity variation with axial position in Plane 1. $Ma_{\infty} = 0.23, J = 1.2$ and $\beta_{3/4} = 40.4^\circ$.

Figure 6.91 Comparison of computed and measured axial velocity variation with radial position in Plane 1. $Ma_{\infty} = 0.23$ and $\beta_{3/4} = 40.4^\circ$.

Figure 6.92 Comparison of computed and measured axial velocity variation with radial position in Plane 1. $Ma_{\infty} = 0.23$ and $\beta_{3/4} = 40.4^\circ$.

Figure 6.93 Inner part of the 96x24x24x6 APIAN propeller grid with $\beta_{3/4} = 40.4^\circ$.

Figure 6.94 Inner part of the 96x24x24x6 APIAN propeller grid with $\beta_{3/4} = 57.0^\circ$.

Figure 6.95 Computed variation of thrust coefficient for blade 1 for the complete unsteady propeller calculation. $Ma_{\infty} = 0.23, J = 1.11$, $\beta_{3/4} = 40.4^\circ$ and $\alpha = 10.0^\circ$.

Figure 6.96 Convergence history for inter-blade region 1 during the unsteady propeller calculation. $Ma_{\infty} = 0.23, J = 1.11$, $\beta_{3/4} = 40.4^\circ$ and $\alpha = 10.0^\circ$.

Figure 6.97 Computed surface pressure at $\eta = 0.3$ on blade 1 during the third revolution. $Ma_{\infty} = 0.23, J = 1.11$, $\beta_{3/4} = 40.4^\circ$ and $\alpha = 10.0^\circ$. 

xvi
**Figure 6.98** Computed surface pressure at $\eta = 0.8$ on blade 1 during the third revolution. $M_{\infty} = 0.23$, $J = 1.11$, $\beta_{3/4} = 40.4^\circ$ and $\alpha = 10.0^\circ$.

**Figure 6.99** Computed variation of thrust coefficient for blade 1 for the complete unsteady propeller calculation. $M_{\infty} = 0.7$, $J = 3.117$, $\beta_{3/4} = 57.0^\circ$ and $\alpha = 3.0^\circ$.

**Figure 6.100** Convergence history for inter-blade region 1 during the unsteady propeller calculation. $M_{\infty} = 0.7$, $J = 3.117$, $\beta_{3/4} = 57.0^\circ$ and $\alpha = 3.0^\circ$.

**Figure 6.101** Computed surface pressure at $\eta = 0.3$ on blade 1 during the third revolution. $M_{\infty} = 0.7$, $J = 3.117$, $\beta_{3/4} = 57.0^\circ$ and $\alpha = 3.0^\circ$.

**Figure 6.102** Computed surface pressure at $\eta = 0.8$ on blade 1 during the third revolution. $M_{\infty} = 0.7$, $J = 3.117$, $\beta_{3/4} = 57.0^\circ$ and $\alpha = 3.0^\circ$.

**Figure 6.103** Contours of relative Mach number at the cell centres adjacent to the propeller blade and nacelle surfaces. $M_{\infty} = 0.23$, $J = 1.11$, $\beta_{3/4} = 40.4^\circ$ and $\alpha = 10.0^\circ$.

**Figure 6.104** Contours of relative Mach number at the cell centres adjacent to the propeller blade and nacelle surfaces. $M_{\infty} = 0.7$, $J = 3.117$, $\beta_{3/4} = 57.0^\circ$ and $\alpha = 3.0^\circ$.

**Figure B.1** Geometrical interpretation of the metric terms.
LIST OF TABLES

Table 2.1 Description of the numerical algorithms developed.

Table 6.1 Airfoil geometry and freestream conditions for the fixed airfoil test cases.

Table 6.2 Computed lift and drag coefficients for the NACA 00121 airfoil. $M_a = 0.63$ and $\alpha = 2.0^\circ$.

Table 6.3 Computed lift and drag coefficients for the NACA0012 airfoil. $M_a = 0.8$ and $\alpha = 1.25^\circ$.

Table 6.4 Computed lift and drag coefficients for the NACA 64A010 airfoil. $M_a = 0.796$ and $\alpha = 0.0^\circ$.

Table 6.5 Freestream Mach number, mean angle of incidence, amplitude of oscillation and reduced frequency for AGARD Test Case CT-6.

Table 6.6 Freestream conditions for the fixed wing test cases.

Table 6.7 Summary of results for Test Case 1 and Test Case 2.

Table 6.8 Test cases for the unsteady wing algorithm.

Table 6.9 Conditions for the NACA 10-(3)(066)-033 propeller steady test case.

Table 6.10 Conditions for the APIAN propeller steady test cases.

Table 6.11 Conditions for the transient propeller test cases.
## Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>A, B, C</td>
<td>inviscid flux Jacobian matrices</td>
</tr>
<tr>
<td>$A_p$</td>
<td>wing planform area</td>
</tr>
<tr>
<td>$c$</td>
<td>speed of sound; chord length</td>
</tr>
<tr>
<td>$c_p$</td>
<td>constant pressure specific heat</td>
</tr>
<tr>
<td>$c_v$</td>
<td>constant volume specific heat</td>
</tr>
<tr>
<td>$C_L$</td>
<td>lift coefficient, $(2D)$ $C_L = \frac{L}{\frac{1}{2}\rho\infty V^2\infty c}$, $(3D)$ $C_L = \frac{L}{\frac{1}{2}\rho\infty V^2\infty A_p}$</td>
</tr>
<tr>
<td>$C_D$</td>
<td>drag coefficient, $(2D)$ $C_D = \frac{D}{\frac{1}{2}\rho\infty V^2\infty c}$, $(3D)$ $C_D = \frac{D}{\frac{1}{2}\rho\infty V^2\infty A_p}$</td>
</tr>
<tr>
<td>$C_p$</td>
<td>static pressure coefficient, $C_p = \frac{p - p\infty}{\frac{1}{2}\rho\infty V^2\infty}$</td>
</tr>
<tr>
<td>$C_T$</td>
<td>thrust coefficient, $C_T = \frac{T}{\frac{1}{2}\rho\infty n^2D^4}$</td>
</tr>
<tr>
<td>CFL</td>
<td>Courant-Friedrichs-Lewy (Courant) number</td>
</tr>
<tr>
<td>$d$</td>
<td>total derivative</td>
</tr>
<tr>
<td>$D$</td>
<td>propeller diameter; drag force</td>
</tr>
<tr>
<td>$e$</td>
<td>internal energy per unit mass</td>
</tr>
<tr>
<td>$E$</td>
<td>total energy per unit mass, $E = \left( e + \frac{</td>
</tr>
<tr>
<td>$E_r$</td>
<td>total roenergy per unit mass, $E_r = \left( e + \frac{</td>
</tr>
<tr>
<td>$f, g, h$</td>
<td>inviscid flux vectors in physical space</td>
</tr>
<tr>
<td>$F$</td>
<td>force applied to system</td>
</tr>
</tbody>
</table>
$F, G, H$ inviscid flux vectors in transformed space

$H$ total enthalpy per unit mass, $H = E + \frac{P}{\rho}$

$H_r$ total rothalpy per unit mass, $H_r = E_r + \frac{P}{\rho}$

$I$ source vector; identity matrix

$J$ advance ratio, $J = \frac{V_\infty}{nD}$

$J^{-1}$ Jacobian of inverse transformation

$k$ dissipation constant; reduced frequency, $k = \frac{\omega e}{2V_\infty}$

$L$ lift force

$m$ system mass

$Ma$ Mach number, $Ma = \frac{V}{c}$

$n$ rotational speed (revolutions per second); integer counter

$p$ static pressure

$P$ total pressure, $P = p\left[1 + \frac{1}{2}(\gamma - 1)Ma^2\right]^{\gamma/(\gamma - 1)}$

$Q$ heat added to system

$Q$ vector of conserved variables

$r$ radius

$R$ residual

$R^\pm$ Riemann invariant

$t$ time in physical space

$T$ thrust force

$U, V, W$ contravariant velocity components

$u, v, w$ Cartesian velocity components
\( V \) fluid speed

\( \mathbf{V} \) absolute fluid velocity vector

\( W \) work done by system

\( \mathbf{W} \) relative fluid velocity vector

\( x, y, z \) Cartesian coordinates

\( \Delta t \) time step

**Greek Symbols**

\( \alpha \) angle of incidence; basic parameter of Runge-Kutta time-stepping scheme; mesh aspect ratio scaling factor

\( \alpha_m \) mean angle of incidence

\( \alpha_o \) amplitude of oscillation

\( \beta \) weighting factor of Runge-Kutta time-stepping scheme for convective term; smoothing coefficient of implicit residual averaging; propeller blade angle

\( \partial \) partial derivative

\( \varepsilon \) dissipation coefficient

\( \gamma \) ratio of specific heats; weighting factor of Runge-Kutta time-stepping scheme for dissipative terms

\( \lambda \) eigenvalue or spectral radius of inviscid flux Jacobian matrix; ratio of fictitious time step to physical time step

\( \nu \) dissipation switching function

\( \xi, \eta, \zeta \) curvilinear coordinates

\( \pi \) pi

\( \rho \) fluid density
\( \tau \)  
\text{time in transformed space}

\( \omega \)  
\text{angular velocity vector; radian frequency}

\( \Omega \)  
\text{cell volume}

\( \psi \)  
\text{implicit residual smoothing parameter}

### Operators

- \( C \)  
\text{convective operator}

- \( D \)  
\text{dissipation operator; backward time difference operator}

- \( I \)  
\text{multigrid solution restriction operator}

- \( L \)  
\text{multigrid interpolation operator}

- \( T \)  
\text{multigrid residual restriction operator}

- \( \nabla \)  
\text{backward difference operator; gradient operator}

- \( \Delta \)  
\text{forward difference operator}

- \( \nabla \Delta \)  
\text{second difference operator}

### Subscripts

- \( i, j, k \)  
\text{cell centre location}

- \( x, y, z, t \)  
\text{partial differentiation; dependence on a particular coordinate}

- \( \xi, \eta, \zeta, \tau \)  
\text{partial differentiation; dependence on a particular coordinate}

- ref  
\text{reference quantity}

- \( \infty \)  
\text{freestream quantity}

### Superscripts

- \( n \)  
\text{time level counter}

- \( 2 \)  
\text{second difference dissipation quantity}
fourth difference dissipation quantity
modified quantity; scaled quantity
dimensional quantity
normalised quantity
inverse
pseudo-time values

**Symbols**

- vector dot product
- vector cross product
- vector magnitude
- summation
- vector update
- less than
INTRODUCTION

1.1 Background

Computational fluid dynamics (CFD) is the application of numerical algorithms to solving fluid flow problems. With the development of accurate and efficient numerical algorithms and the continuing improvements in computer hardware, CFD has now matured to the point where it is widely accepted as a key tool for aerodynamic analysis and design.

This thesis describes the application of CFD to predicting the aerodynamics of advanced propellers. The history of the advanced propeller is first outlined, together with research carried out in this field and the current status of advanced propeller technology. The objective of the present research, the strategy undertaken and the novel aspects of the work are also described.

1.2 Advanced Propellers

1.2.1 Initial Concept

In 1973 the Organisation of Petroleum Exporting Countries imposed an oil embargo on many nations worldwide, including the United States. The serious effects of this embargo led the United States government to initiate an intensive national effort to investigate new sources of energy and also means of reducing fuel consumption throughout the country. In 1975 the Senate Committee on Aeronautical and Space Science directed NASA to investigate every potential fuel saving concept that aviation technology could produce. In response, the engineers at NASA Lewis proposed the advanced propeller, a novel and controversial concept, as a possible replacement for the turbofan engine for high-speed flight. At that time it was a well known fact that the propulsive efficiency of the turboprop, a propeller driven by a gas turbine powerplant, was far superior to that of a turbofan engine.
for cruise Mach numbers up to 0.6. At higher cruise speeds the propulsive efficiency of the turboprop dropped sharply due to compressibility losses near the blade tips. The NASA Lewis engineers claimed that a turboprop designed specifically for high-speed flight would offer a fuel saving of up to 50% over an equivalent technology turbofan engine operating at competitive speeds and altitudes. The proposed high-speed turboprop, also known as the advanced propeller, consisted of a small-diameter, highly-loaded, multi-bladed, swept, variable-pitch propeller driven by a gas turbine engine.\textsuperscript{1-6}

### 1.2.2 United States Research

The large potential fuel savings of this new concept led to a massive NASA funded and co-ordinated research project on advanced propellers, named the Advanced Turboprop Project.\textsuperscript{1} This project started in 1978 and included aircraft, engine and propeller manufactures, research institutes and universities. The project lasted 10 years and culminated with the flight of three demonstrator airplanes equipped with advanced propellers in both single-rotation (i.e., a single blade row) tractor (see Figure 1.1) and counter-rotation (i.e., a pair of counter-rotating blades rows) pusher (see Figure 1.2) configurations in 1987 and 1988. These flight tests demonstrated that fuel savings of up to 30% could be achieved over an equivalent technology turbofan engine. Having proven the efficiency of the advanced propeller and having developed the technology necessary to make it a viable propulsion concept, only limited research on advanced propellers has since been performed in the United States.

### 1.2.3 European Research

The European Union, recognising the serious potential of the advanced propeller and the lead built up by the United States in this technology, funded its first, large-scale, collaborative research project on advanced propellers in 1990. The project was named GEMINI. Since 1990 three other advanced propeller projects have also been funded: SNAAP, GEMINI II and APIAN. These projects have included aircraft companies, propeller manufacturers, national research centres and universities from the various member countries. In contrast to the American effort, the European research has focused on wind tunnel testing and on the development of numerical tools only, with no flight testing of large-scale hard-
ware.\textsuperscript{7,8} The APIAN project is currently on-going and is due for completion in the year 2000.

\textbf{Figure 1.1} Single-rotation tractor advanced propeller.

\textbf{Figure 1.2} Counter-rotation pusher advanced propeller.
1.2.4 Current Status

Contrary to what one might expect, even after such massive research efforts both in the United States and in Europe, no advanced propeller driven aircraft has ever been developed to production status or is in the process of being developed. The two main reasons for this are the return to relatively low fuel prices and the potential cabin noise problems associated with advanced propeller driven aircraft. Meanwhile, propellers showing blade shape characteristics clearly influenced by advanced propeller research have been introduced on regional aircraft (Saab 2000 and Bombardier Q400 Dash 8) and on military transport aircraft (Lockheed Martin C-130J).

Considerable interest still remains in the advanced propeller both for commercial and for military use, because fuel efficiency will inevitably become of crucial importance in the development of future propulsion systems. In order to further improve the aerodynamic and especially the acoustic performance of the advanced propeller, it is necessary to fully understand the very complex flow patterns occurring on the blade and nacelle surfaces and in the surrounding flowfield. This insight can be achieved using wind tunnel testing and numerical techniques. Wind tunnel testing, especially for a complex configuration such as an advanced propeller, is expensive, difficult and time-consuming. Numerical prediction techniques on the other hand offer an attractive alternative. Using a numerical approach the flowfield around an advanced propeller can be calculated over the full range of operating conditions (i.e., low subsonic to transonic freestream Mach numbers and all inflow angles). Data such as density, pressure and velocity can be obtained as well as propeller performance coefficients such as thrust, torque and efficiency. Complex flow phenomena, like blade leading-edge and tip vortices and shock waves can also be accurately resolved.

1.3 Previous Numerical Methods for Advanced Propeller Calculations

Several numerical algorithms have been developed specifically for the computation of advanced propeller flows. In the present study, only algorithms that solve the Euler or Navier-Stokes equations were reviewed. Algorithms that employed simpler mathematical models (e.g., panel and potential field methods) have now been surpassed. Descriptions of a number of prolific Euler and Navier-Stokes algorithms and corresponding predictions can
be found in References 9-30. A review of some of these techniques is given by Zondervan and is not replicated here.

There are two important points to note about previous methods. The first is that the vast majority of these algorithms were developed for, and are restricted to, advanced propeller flowfield calculations with an inflow which is axisymmetric (i.e., zero angle of incidence and zero angle of yaw). Under realistic flight conditions, the inflow is more often than not asymmetric due both to non-zero angles of incidence and yaw and to installation interaction effects. Even at cruise conditions, the propeller is at incidence to the freestream. Of all the algorithms developed so far, only three are capable of predicting advanced propeller flowfields under asymmetric inflow conditions. Each of these algorithms solves the Euler equations. The three methods are by Srivastava, Whitfield et al., and Janus and Whitfield. Of this select few only two have actually been used for this particular purpose. In 1987 Whitfield et al. presented results from a calculation of the flow around the SR-3 propeller at 4° angle of incidence. In the early 1990’s Nallasamy et al. used the algorithm developed by Janus and Whitfield to calculate the flow around the SR-7L propeller for different angles of incidence. The second point is that these flow calculations involving asymmetric inflows have been very computationally demanding (i.e., long run times). To demonstrate the scale of CPU time required, a calculation by Nallasamy et al. in 1990 of the flow around a two-bladed version of the SR-7L propeller required 33 CPU hours on a Cray YMP supercomputer at NASA Ames Research Centre.

A need therefore exists for a computationally efficient algorithm for the calculation of advanced propeller flowfields under the conditions of asymmetric inflow. This need is addressed in this work.

1.4 Present Research

1.4.1 Objective

The objective of this research is to develop an algorithm capable of predicting the inviscid flowfield around a propeller with an asymmetric inflow, and to demonstrate the capability of this algorithm by calculating the flowfield around an advanced propeller at incidence. With an asymmetric inflow, a propeller blade experiences an inflow velocity direction that varies with angular position and therefore the flowfield is unsteady. As a result the asym-
metric propeller algorithm must solve the Euler equations in a time-accurate manner. Note that in this thesis the expressions time-accurate, transient, unsteady and time-dependent are used synonymously.

1.4.2 Strategy

The development of the unsteady propeller algorithm is achieved by first developing a series of simpler algorithms, each designed to predict a progressively more complex inviscid flowfield, starting with the prediction of the two-dimensional, steady, inviscid flow around a fixed airfoil. Common to each algorithm is the use of a cell-centered finite-volume scheme with central differencing and blended artificial dissipation terms. Explicit time-stepping is employed in the steady algorithms and implicit time-stepping in the unsteady algorithms. Well proven convergence acceleration techniques, typical of explicit schemes, are used to accelerate the approach to steady-state.

The results of test cases computed using each algorithm are presented and, where possible, compared with wind tunnel measurements and with the results of other researchers.

1.4.3 Layout of Thesis

The layout of the thesis is as follows: Chapter 2 discusses the formulation of the governing equations for each flowfield modelled. The complete set of governing equations for each flowfield are then presented. In Chapter 3 the numerical formulation, which includes the spatial discretisation, the added dissipation terms and the time-stepping schemes, is described. The formulation and implementation of the boundary conditions employed in each algorithm is discussed in Chapter 4. The convergence acceleration techniques are presented in Chapter 5. Results of test cases computed using each algorithm are presented and analysed in Chapter 6. In Chapter 7 the conclusions drawn from the present research are presented and, finally, in Chapter 8 recommendations are made for possible future work.

1.5 Novel Aspects of Research

This research is novel in three respects:

This is the first time that:
1. A steady algorithm for the calculation of the flow around a propeller with an axisymmetric inflow has been developed incorporating the present solution strategy.

2. A comparison of the computed flowfield downstream of an advanced propeller under axisymmetric inflow conditions has been compared with particle image velocimetry (PIV) measurements made in a large-scale wind tunnel.

3. An unsteady propeller algorithm for asymmetric inflow flowfield calculations has been developed incorporating the present scheme.

1.6 Technical Publications

Some of the research work described in this thesis has been published in international conference papers and in European Union technical reports.\(^{8,32-36}\)
CHAPTER 2

GOVERNING EQUATIONS OF INVISCID FLOW

2.1 Introduction

The ultimate goal of the present research is to develop a numerical algorithm for the prediction of the time-dependent inviscid flowfield around a propeller under asymmetric inflow conditions. This goal is achieved by developing six numerical algorithms consecutively that are used to model progressively more complex flowfields, starting with the prediction of the two-dimensional, steady, inviscid flow around a fixed airfoil. The six algorithms that are developed, and the flowfields modelled, are listed in Table 2.1

<table>
<thead>
<tr>
<th>Number</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FLO2DS</td>
<td>Two-dimensional steady algorithm for fixed airfoil flow calculations</td>
</tr>
<tr>
<td>2</td>
<td>FLO2DU</td>
<td>Two-dimensional unsteady algorithm for oscillating airfoil flow calculations</td>
</tr>
<tr>
<td>3</td>
<td>FLO3DS</td>
<td>Three-dimensional steady algorithm for fixed wing flow calculations</td>
</tr>
<tr>
<td>4</td>
<td>FLO3DU</td>
<td>Three-dimensional unsteady algorithm for oscillating wing flow calculations</td>
</tr>
<tr>
<td>5</td>
<td>PROP3DS</td>
<td>Steady propeller algorithm for calculating the flowfield around a propeller with an axisymmetric inflow</td>
</tr>
<tr>
<td>6</td>
<td>PROP3DU</td>
<td>Unsteady propeller algorithm for calculating the flowfield around a propeller under asymmetric inflow conditions</td>
</tr>
</tbody>
</table>

*Table 2.1* Description of the numerical algorithms developed.

This chapter presents the formulation of the governing equations that describe the fluid motion in each of these six flowfields.
2.2 Formulation of the Governing Equations

2.2.1 Introduction

The flow is assumed to be inviscid in all flowfields modelled. Therefore, the dissipative transport phenomena of friction, thermal conduction and mass diffusion are neglected when formulating the equations of motion. The governing equations of inviscid flow are called the Euler equations.

The equations solved by each of the numerical algorithms are formulated following a similar methodology. The equations are initially formulated in their most fundamental form by applying three principles of classical mechanics to a fixed finite control volume model of the flow in an Eulerian frame of reference. A finite control volume is defined as a reasonably large finite region of the flow that is bounded by a closed control surface as shown in Figure 2.1. The equations directly obtained from a finite control volume model of the flow are in integral form. These equations are then re-written in partial differential equation form, non-dimensionalised and finally transformed to a body-fitted curvilinear coordinate system. Each of these steps in the formulation of the governing equations is described in the following sections.

![Diagram of a finite control volume](image)

*Figure 2.1* Finite control volume fixed in space.
2.2.2 Integral Form

The three principles of classical mechanics are written for a system of fixed mass \( \tilde{m} \), and are:

**Conservation of Mass**

\[
\frac{d}{dt} \tilde{m} = 0
\]  
(2.1)

**Newton’s Second Law**

\[
\tilde{F} = \frac{d}{dt} (\tilde{m} \tilde{V})
\]  
(2.2)

**Conservation of Energy**

\[
\frac{d}{dt} \tilde{Q} - \frac{d}{dt} \tilde{W} = \frac{d}{dt} \tilde{E}
\]  
(2.3)

In Equations (2.1) to (2.6) \( \tilde{F} \) is the force applied to the system, \( \tilde{V} \) is the velocity, \( \tilde{Q} \) is heat added to the system, \( \tilde{W} \) is work done by the system and \( \tilde{E} \) is the system energy. Although these principles are written specifically for a system, they can be rewritten to apply to a finite control volume using Reynolds’ transport theorem.

For all algorithms, excluding the steady propeller algorithm PROP3DS, the finite control volume model is assumed fixed in an inertial (i.e., at rest or moving at constant velocity) frame of reference. It is also assumed that there are no body forces acting on the fluid as it moves through the control volume and that there are no sources of volumetric heat. The equations obtained by applying each of the three principles in turn to the flow model are called the continuity equation, the momentum equation and the energy equation, respectively.

The Euler equations for an inertial and a non-inertial reference frame are presented in the following sections.
2.2.2.1 Inertial Reference Frame

The Euler equations in an inertial reference frame are:

*Continuity Equation*

\[
\frac{\partial}{\partial t} \int \tilde{\rho} \tilde{V} d\tilde{V} + \oint \tilde{p} \tilde{S} \cdot d\tilde{S} = 0
\]  \hspace{1cm} (2.4)

*Momentum Equation*

\[
\frac{\partial}{\partial t} \int \tilde{\rho} \tilde{V} d\tilde{V} + \oint (\tilde{p} \tilde{V} \cdot d\tilde{S}) \tilde{V} = -\oint \tilde{p} d\tilde{S}
\]  \hspace{1cm} (2.5)

*Energy Equation*

\[
\frac{\partial}{\partial t} \int \tilde{\rho} \tilde{E} d\tilde{V} + \oint \tilde{\rho} \tilde{E} (\tilde{V} \cdot d\tilde{S}) \tilde{V} = -\oint \tilde{p} \tilde{V} \cdot d\tilde{S}
\]  \hspace{1cm} (2.6)

In Equations (2.4) to (2.6) \( \tilde{\rho} \) is the fluid density, \( \tilde{V} \) is the fluid velocity, \( \tilde{p} \) is the static pressure, \( d\tilde{S} \) is an elemental area of surface, \( d\tilde{V} \) is an element of volume and \( \tilde{E} \) is the total energy per unit mass. The total energy per unit mass is defined as:

\[
\tilde{E} = (\tilde{e} + \frac{|\tilde{V}|^2}{2})
\]  \hspace{1cm} (2.7)

where \( \tilde{e} \) is the internal energy per unit mass.

2.2.2.2 Non-Inertial Reference Frame

For PROP3DS, which is used to predict the flow around a propeller under axisymmetric inflow conditions, the equations are formulated following the methodology developed by Holmes and Tong.\(^{38}\) The finite control volume is assumed fixed in a non-inertial reference frame that is attached to the propeller. The non-inertial reference frame rotates steadily with the angular velocity \( \tilde{\omega} \) of the propeller and the origin of the non-inertial reference frame is fixed relative to an inertial reference frame. This particular formulation of the Euler equations allows the flow around a steadily-rotating propeller with an axisymmetric inflow to
be treated as steady relative to the blades, resulting in an algorithm that is far more efficient than one that solves for the unsteady flowfield in an inertial reference frame.

Equations (2.4) to (2.6) can be rewritten for a steadily-rotating non-inertial reference frame by replacing the absolute velocity \( \vec{V} \) by the relative velocity \( \vec{W} = \vec{V} - \vec{\omega} \times \vec{r} \), where \( \vec{r} \) is the local position vector, and by adding Coriolis and centrifugal acceleration terms to the right-hand side of Equation (2.5). A total energy \( \tilde{E} \) in Equation (2.6) is replaced by the total roenergy \( \tilde{E}_r \), defined as:

\[
\tilde{E}_r = \left( \tilde{c} + \frac{|\vec{W}|^2}{2} - \frac{|\vec{\omega} \times \vec{r}|^2}{2} \right)
\]

or, alternatively, as:

\[
\tilde{E}_r = \left( \tilde{c} + \frac{|\vec{V}|^2}{2} - \vec{V} \cdot (\vec{\omega} \times \vec{r}) \right)
\]

The Euler equations in a non-inertial reference frame can then be written as:

**Continuity Equation**

\[
\frac{\partial}{\partial t} \int \rho \vec{V} d\vec{S} + \int \rho \vec{V} \cdot \vec{d} \vec{S} = 0
\]

**Momentum Equation**

\[
\frac{\partial}{\partial t} \int \rho \vec{W} d\vec{V} + \int \rho \vec{W} \cdot \vec{d} \vec{S} = -\int \rho \vec{W} d\vec{S} - \int \rho (2\vec{\omega} \times \vec{W} + \vec{\omega} \times (\vec{\omega} \times \vec{r})) d\vec{V}
\]

**Energy Equation**

\[
\frac{\partial}{\partial t} \int \rho \tilde{E}_r d\vec{V} + \int \rho \tilde{E}_r (\vec{W} \cdot \vec{d} \vec{S}) = -\int \rho \vec{W} \cdot \vec{d} \vec{S}
\]

When formulating the Euler equations in the inertial and non-inertial reference frames no specific Eulerian coordinate system was implied. A right-handed Cartesian coordinate system is employed for all algorithms developed here. The Cartesian velocity components are \( u, v \) and \( w \) along the \( x, y \) and \( z \) axes respectively.
2.2.3 Differential Form

Having formulated the governing equations in integral form, the equations are then re-written in partial differential equation form by applying the divergence and gradient theorems. The resulting sets of differential equations are said to be in either strong or weak conservation-law form.\(^{41,42}\)

The equations for PROP3DS are also re-written in terms of the absolute flow velocity components instead of the relative flow velocity components. As pointed out by Holmes and Tong,\(^{38}\) this formulation of the Euler equations provides more accurate solutions for a propeller with an axisymmetric inflow where the absolute far-field flow is uniform but the relative flow is non-uniform. The roenergy equation is retained in favour of the energy equation; this helps to maintain the rothalpy \(\tilde{H}_r\), defined as:

\[
\tilde{H}_r = \tilde{E}_r + \frac{\tilde{p}}{\tilde{\rho}}
\]  \hspace{1cm} (2.13)

constant in the computational domain at a value equal to the freestream rothalpy. The value of rothalpy is constant along a streamline in a steady inviscid flowfield with no body forces in a non-inertial reference frame.\(^{40}\) If the inflow is also assumed uniform, as is done here, the value of rothalpy is then constant throughout the domain. Note that in the present research it is also assumed that the non-inertial reference frame rotates around the \(x\) axis of the inertial reference frame with an angular velocity \(\tilde{\omega} = \tilde{\omega}_x \mathbf{i}\).

2.2.4 Non-dimensionalisation

The variables appearing in the equations are non-dimensionalised using a standard non-dimensionalisation procedure.\(^{43}\) Non-dimensionalising the governing equations removes any restriction to a particular system of units and also results in more user-friendly numerical algorithms. The details of the non-dimensionalisation employed are presented in Appendix A. The form of the governing equation remains invariant under the non-dimensionalising procedure.
2.2.5 Transformation to a Curvilinear Coordinate System

The Euler equations in partial differential equation form are finally transformed to a body-fitted curvilinear coordinate system. There are two significant advantages of applying this transformation.\(^44\) The first is that boundary surfaces in the physical domain can be mapped onto rectangular surfaces in the transformed domain, simplifying the implementation of boundary conditions. The second is that unsteady body motion, such as an oscillating airfoil, can be incorporated into the equations by assuming that the body-fitted curvilinear coordinate system is time-dependent. The physical and transformed computational domains for the prediction of the two-dimensional steady flow around a fixed airfoil are shown in Figure 2.2.

*Figure 2.2* Physical and transformed computational domains for fixed airfoil flow calculations.

For a fixed, three-dimensional, curvilinear coordinate system the curvilinear axes are defined as:

\[
\begin{align*}
\xi &= \xi(x, y, z) \\
\eta &= \eta(x, y, z) \\
\zeta &= \zeta(x, y, z) \\
\tau &= t
\end{align*}
\]

\hspace{1em}(2.14)

while for a time-dependent, three-dimensional, curvilinear coordinate system the axes are:
The transformation of the three-dimensional Euler equations in Cartesian coordinates in an inertial reference frame to a time-dependent curvilinear coordinate system is presented in Appendix B. This transformation represents the most complex of the six transformations that are undertaken. The transformed set of equations describe the unsteady flow around an oscillating wing and are the equations solved by FLO3DU. The unsteady motion of the wing is accounted for by the time-dependency of the curvilinear coordinate system.

The final form of the governing equations for each of the six flowfields is presented in §2.3.

2.3 Presentation of the Euler Equations

The Euler equations that describe the flow in each of the six flowfields modelled are presented in detail for completeness. An equation of state is required in order to close each equation set. The equation of state used here assumes a calorically perfect (i.e., a perfect gas with constant specific heats).

2.3.1 Governing Equations for Fixed Airfoil Calculations

The Euler equations are:

$$\frac{\partial}{\partial t} (J^{-1} \mathbf{Q}) + \frac{\partial \mathbf{F}}{\partial \xi} + \frac{\partial \mathbf{G}}{\partial \eta} = 0$$

(2.16)

where the vector of conserved variables, \( \mathbf{Q} \), and the vectors of inviscid flux terms, \( \mathbf{F} \) and \( \mathbf{G} \), are:

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{bmatrix}$$

(2.17)
\[
F = J^{-1} \begin{bmatrix}
\rho U \\
\rho uU + \xi_x p \\
\rho vU + \xi_y p \\
\rho HU
\end{bmatrix}
\]

(2.18)

\[
G = J^{-1} \begin{bmatrix}
\rho V \\
\rho uV + \eta_x p \\
\rho vV + \eta_y p \\
\rho HV
\end{bmatrix}
\]

(2.19)

The Jacobian of the inverse transformation, \(J^{-1}\), is:

\[
J^{-1} = x_{\xi} y_\eta - y_{\xi} x_\eta
\]

(2.20)

and the metrics of the transformation are:

\[
\begin{align*}
\xi_x &= Jy_\eta \\
\xi_y &= -Jx_\eta \\
\eta_x &= -Jy_\xi \\
\eta_y &= Jx_\xi
\end{align*}
\]

(2.21)

\(U\) and \(V\) are the contravariant velocity components in the \(\xi\) and \(\eta\) directions respectively and are defined as:

\[
\begin{align*}
U &= \xi_x u + \xi_y v \\
V &= \eta_x u + \eta_y v
\end{align*}
\]

(2.22)

The system of equations is closed using the equation of state:

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

(2.23)

where \(\gamma\) is defined as:

\[
\gamma = \frac{c_p}{c_v}
\]

(2.24)

and \(c_p\) and \(c_v\) are the constant pressure and constant volume specific heats respectively.
2.3.2 Governing Equations for Oscillating Airfoil Calculations

The Euler equations are:

\[
\frac{\partial}{\partial \tau} (J^{-1} \mathbf{Q}) + \frac{\partial \mathbf{F}}{\partial \xi} + \frac{\partial \mathbf{G}}{\partial \eta} = 0
\]  

(2.25)

where the vector of conserved variables, \( \mathbf{Q} \), and the vectors of inviscid flux terms, \( \mathbf{F} \) and \( \mathbf{G} \), are:

\[
\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{bmatrix}
\]

(2.26)

\[
\mathbf{F} = J^{-1} \begin{bmatrix} \rho U \\ \rho u U + \xi_x \rho \\ \rho v U + \xi_y \rho \\ \rho H U - \xi_t \rho \end{bmatrix}
\]

(2.27)

\[
\mathbf{G} = J^{-1} \begin{bmatrix} \rho V \\ \rho u V + \eta_x \rho \\ \rho v V + \eta_y \rho \\ \rho H V - \eta_t \rho \end{bmatrix}
\]

(2.28)

The Jacobian of the inverse transformation, \( J^{-1} \), is:

\[
J^{-1} = x_\xi y_\eta - y_\xi x_\eta
\]

(2.29)

and the metrics of the transformation are:

\[
\xi_x = J y_\eta \\
\xi_y = -J x_\eta \\
\eta_x = -J y_\xi \\
\eta_y = J x_\xi \\
\xi_t = -x_\tau \xi_x - y_\tau \xi_y \\
\eta_t = -x_\tau \eta_x - y_\tau \eta_y
\]

(2.30)
U and V are the contravariant velocity components in the ξ and η directions respectively and are defined as:

\[
U = \xi_x u + \xi_y v + \xi_t \\
V = \eta_x u + \eta_y v + \eta_t
\]  
(2.31)

The system of equations is closed using the equation of state:

\[
p = (\gamma - 1) \rho \left( E - \frac{u^2 + v^2}{2} \right)
\]  
(2.32)

### 2.3.3 Governing Equations for Fixed Wing Calculations

The Euler equations are:

\[
\frac{\partial}{\partial \tau} (J^{-1} Q) + \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} + \frac{\partial H}{\partial \zeta} = 0
\]  
(2.33)

where the vector of conserved variables, \( Q \), and vector of inviscid flux terms, \( F \), \( G \) and \( H \), are:

\[
Q = \begin{bmatrix}
\rho \\
\rho u \\
\rho v \\
\rho w \\
\rho E
\end{bmatrix}
\]  
(2.34)

\[
F = J^{-1} \begin{bmatrix}
\rho U \\
\rho uU + \xi_x \rho \\
\rho vU + \xi_y \rho \\
\rho wU + \xi_z \rho \\
\rho HU
\end{bmatrix}
\]  
(2.35)
The Jacobian of the inverse transformation, $J^{-1}$, is:

$$J^{-1} = x_\zeta(y_\eta z_\zeta - z_\eta y_\zeta) - y_\zeta(x_\eta z_\zeta - z_\eta x_\zeta) + z_\zeta(x_\eta y_\zeta - y_\eta x_\zeta)$$

and the metrics of the transformation are:

$$\xi_x = J(y_\eta z_\zeta - z_\eta y_\zeta)$$
$$\xi_y = J(z_\eta x_\zeta - x_\eta z_\zeta)$$
$$\xi_z = J(x_\eta y_\zeta - y_\eta x_\zeta)$$
$$\eta_x = J(y_\zeta z_\xi - z_\zeta y_\xi)$$
$$\eta_y = J(z_\xi x_\zeta - x_\xi z_\zeta)$$
$$\eta_z = J(x_\xi y_\zeta - y_\xi x_\zeta)$$
$$\zeta_x = J(y_\xi z_\eta - z_\xi y_\eta)$$
$$\zeta_y = J(z_\xi x_\eta - x_\xi z_\eta)$$
$$\zeta_z = J(x_\xi y_\eta - y_\xi x_\eta)$$

$U$, $V$ and $W$ are the contravariant velocity components in the $\xi$, $\eta$ and $\zeta$ directions respectively and are defined as:

$$U = \xi_x u + \xi_y v + \xi_z w$$
$$V = \eta_x u + \eta_y v + \eta_z w$$
$$W = \zeta_x u + \zeta_y v + \zeta_z w$$

The system of equations is closed using the equation of state:
\[ p = (\gamma - 1)p \left( E - \frac{(u^2 + v^2 + w^2)}{2} \right) \]  

(2.41)

2.3.4 Governing Equations for Oscillating Wing Calculations

The Euler equations are:

\[ \frac{\partial}{\partial \tau} (J^{-1} Q) + \frac{\partial F}{\partial \zeta} + \frac{\partial G}{\partial \eta} + \frac{\partial H}{\partial \zeta} = 0 \]  

(2.42)

where the vector of conserved variables, \( Q \), and inviscid flux terms, \( F, G \) and \( H \), are:

\[
Q = \begin{bmatrix}
\rho \\
\rho u \\
\rho v \\
\rho w \\
\rho E
\end{bmatrix}
\]

(2.43)

\[
F = J^{-1} \begin{bmatrix}
\rho U \\
\rho u U + \zeta_x p \\
\rho v U + \zeta_y p \\
\rho w U + \zeta_z p \\
\rho H U - \zeta_{\zeta} p
\end{bmatrix}
\]

(2.44)

\[
G = J^{-1} \begin{bmatrix}
\rho V \\
\rho u V + \eta_x p \\
\rho v V + \eta_y p \\
\rho w V + \eta_z p \\
\rho H V - \eta_{\eta} p
\end{bmatrix}
\]

(2.45)

\[
H = J^{-1} \begin{bmatrix}
\rho W \\
\rho u W + \zeta_x p \\
\rho v W + \zeta_y p \\
\rho w W + \zeta_z p \\
\rho H W - \zeta_{\zeta} p
\end{bmatrix}
\]

(2.46)
The Jacobian of the inverse transformation, $J^{-1}$, is:

$$J^{-1} = \eta_\xi (y_\eta z_\xi - z_\eta y_\xi) - \xi_\eta (x_\eta z_\xi - z_\eta x_\xi) + z_\xi (x_\eta y_\xi - y_\eta x_\xi)$$  \hspace{1cm} (2.47)

and the metrics of the transformation are:

$$\xi_\xi = J(y_\eta z_\xi - z_\eta y_\xi)$$  
$$\xi_\eta = J(x_\eta y_\xi - y_\eta x_\xi)$$  
$$\xi_\zeta = J(x_\eta z_\xi - z_\eta x_\xi)$$  
$$\eta_\xi = J(y_\xi z_\eta - z_\xi y_\eta)$$  
$$\eta_\eta = J(z_\xi x_\eta - x_\xi z_\eta)$$  
$$\eta_\zeta = J(z_\xi y_\eta - y_\xi z_\eta)$$  
$$\zeta_\xi = J(y_\xi z_\eta - z_\xi y_\eta)$$  
$$\zeta_\eta = J(z_\xi x_\eta - x_\xi z_\eta)$$  
$$\zeta_\zeta = J(x_\xi y_\eta - y_\xi x_\eta)$$  \hspace{1cm} (2.48)

$\xi_t = x_\xi \xi_x - y_\xi \xi_y - z_\xi \xi_z$
$$\eta_t = -x_\xi \eta_x - y_\xi \eta_y - z_\xi \eta_z$$
$$\zeta_t = -x_\xi \zeta_x - y_\xi \zeta_y - z_\xi \zeta_z$$

$U, V$ and $W$ are the contravariant velocity components in the $\xi, \eta$ and $\zeta$ directions respectively and are defined as:

$$U = \xi_\xi u + \xi_\eta v + \xi_\zeta w + \xi_t$$
$$V = \eta_\xi u + \eta_\eta v + \eta_\zeta w + \eta_t$$
$$W = \zeta_\xi u + \zeta_\eta v + \zeta_\zeta w + \zeta_t$$  \hspace{1cm} (2.49)

The system of equations is closed using the equation of state:

$$p = (\gamma - 1)\rho \left( \frac{E - (u^2 + v^2 + w^2)}{2} \right)$$  \hspace{1cm} (2.50)
2.3.5 Governing Equations for Propeller with Axisymmetric Inflow Calculations

The Euler equations (in the non-inertial reference frame) are:

\[
\frac{\partial}{\partial \tau} (J^{-1} \mathbf{Q}) + \frac{\partial \mathbf{F}}{\partial \xi} + \frac{\partial \mathbf{G}}{\partial \eta} + \frac{\partial \mathbf{H}}{\partial \zeta} = \mathbf{I} \tag{2.51}
\]

where the vector of conserved variables, \( \mathbf{Q} \), and the vectors of inviscid flux terms and source term, \( \mathbf{F}, \mathbf{G}, \mathbf{H} \) and \( \mathbf{I} \), are:

\[
\mathbf{Q} = \begin{bmatrix}
\rho \\
\rho u \\
\rho v \\
\rho w \\
\rho E_c
\end{bmatrix}
\tag{2.52}
\]

\[
\mathbf{F} = J^{-1} \begin{bmatrix}
\rho U \\
\rho u U + \xi_x p \\
\rho v U + \xi_y p \\
\rho w U + \xi_z p \\
\rho H_t U
\end{bmatrix}
\tag{2.53}
\]

\[
\mathbf{G} = J^{-1} \begin{bmatrix}
\rho V \\
\rho u V + \eta_x p \\
\rho v V + \eta_y p \\
\rho w V + \eta_z p \\
\rho H_t V
\end{bmatrix}
\tag{2.54}
\]

\[
\mathbf{H} = J^{-1} \begin{bmatrix}
\rho W \\
\rho u W + \zeta_x p \\
\rho v W + \zeta_y p \\
\rho w W + \zeta_z p \\
\rho H_t W
\end{bmatrix}
\tag{2.55}
\]
\[
I = J^{-1} \begin{bmatrix}
0 \\
0 \\
\rho \omega_x w \\
-\rho \omega_x v \\
0
\end{bmatrix}
\]  \hspace{1cm} (2.56)

The Jacobian of the inverse transformation, \(J^{-1}\), is:

\[
J^{-1} = \sum_i x_i \frac{y_i z_j - z_i y_j}{J(z J y - y J z)} - y_i \frac{x_i z_j - z_i x_j}{J(z J y - y J z)} + z_i \frac{x_i y_j - y_i x_j}{J(z J y - y J z)}
\]  \hspace{1cm} (2.57)

and the metrics of the transformation are:

\[
\begin{align*}
\xi_x &= J(y \eta z - z \eta y) \\
\xi_y &= J(z \eta x - x \eta z) \\
\xi_z &= J(x \eta y - y \eta x) \\
\eta_x &= J(y \zeta z - z \zeta y) \\
\eta_y &= J(z \zeta x - x \zeta z) \\
\eta_z &= J(x \zeta y - y \zeta x) \\
\zeta_x &= J(y \zeta \eta - z \zeta y) \\
\zeta_y &= J(z \zeta \eta - x \zeta z) \\
\zeta_z &= J(x \zeta \eta - y \zeta x) \\
\xi_l &= -\xi_x \xi_y - \xi_y \xi_z - \xi_z \xi_x \\
\eta_l &= -\eta_x \eta_y - \eta_y \eta_z - \eta_z \eta_x \\
\zeta_l &= -\zeta_x \zeta_y - \zeta_y \zeta_z - \zeta_z \zeta_x
\end{align*}
\]  \hspace{1cm} (2.58)

\(U, V\) and \(W\) are the contravariant velocity components in the \(\xi, \eta\) and \(\zeta\) directions respectively and are defined as:

\[
\begin{align*}
U &= \xi_u + \xi_v + \xi_w + \xi_t \\
V &= \eta_u + \eta_v + \eta_w + \eta_l \\
W &= \zeta_u + \zeta_v + \zeta_w + \zeta_l
\end{align*}
\]  \hspace{1cm} (2.59)

The system of equations is closed using the equation of state:

\[
p = (\gamma - 1) \rho \left( E - \frac{u^2 + v^2 + w^2}{2} + v y + w z \right)
\]  \hspace{1cm} (2.60)
Equation (2.60) is obtained in exactly the same fashion as the previous equations of state, except that total roenergy, $E_r$, is now used instead of the total energy, $E$, and by using the fact that $\omega = \omega_x i$ in Equation (2.9).

As outlined in §2.2.3 the roenergy equation is employed in favour of the energy equation to help maintain a constant value of roenthalpy (equal to the freestream roenthalpy due to the assumption of uniform inflow) in the computational domain. Note however that the absolute velocity components are solved for in Equation (2.51) (i.e., $u$, $v$ and $w$ are the absolute Cartesian velocity components in the $x$, $y$ and $z$ directions respectively in the non-inertial reference frame).

### 2.3.6 Governing Equations for Propeller with Asymmetric Inflow Calculations

The Euler equations are:

$$\frac{\partial (J^{-1} \mathbf{Q})}{\partial \tau} + \frac{\partial \mathbf{F}}{\partial \xi} + \frac{\partial \mathbf{G}}{\partial \eta} + \frac{\partial \mathbf{H}}{\partial \zeta} = 0$$  \hspace{1cm} (2.61)

where the vector of conserved variables, $\mathbf{Q}$, and the vectors of inviscid flux terms, $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{H}$, are:

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{bmatrix}$$  \hspace{1cm} (2.62)

$$\mathbf{F} = J^{-1} \begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho w U + \xi_z p \\ \rho HU - \xi_t p \end{bmatrix}$$  \hspace{1cm} (2.63)
\[ G = J^{-1} \begin{bmatrix} \rho V \\ \rho uV + \eta_x \rho \\ \rho vV + \eta_y \rho \\ \rho wV + \eta_z \rho \\ \rho HV - \eta_t \rho \end{bmatrix} \] (2.64)

\[ H = J^{-1} \begin{bmatrix} \rho W \\ \rho uW + \zeta_x \rho \\ \rho vW + \zeta_y \rho \\ \rho wW + \zeta_z \rho \\ \rho HW - \zeta_t \rho \end{bmatrix} \] (2.65)

The Jacobian of the inverse transformation, \( J^{-1} \), is:

\[ J^{-1} = x_\xi (y_\eta z_\zeta - z_\eta y_\zeta) - y_\xi (x_\eta z_\zeta - z_\eta x_\zeta) + z_\xi (x_\eta y_\zeta - y_\eta x_\zeta) \] (2.66)

and the metrics of the transformation are:

\[ \begin{align*}
\xi_x &= J(y_\eta z_\zeta - z_\eta y_\zeta) \\
\xi_y &= J(z_\eta x_\zeta - x_\eta z_\zeta) \\
\xi_z &= J(x_\eta y_\zeta - y_\eta x_\zeta) \\
\eta_x &= J(y_\zeta z_\xi - z_\zeta y_\xi) \\
\eta_y &= J(z_\zeta x_\xi - x_\zeta z_\xi) \\
\eta_z &= J(x_\zeta y_\xi - y_\zeta x_\xi) \\
\zeta_x &= J(y_\xi z_\eta - z_\xi y_\eta) \\
\zeta_y &= J(z_\xi x_\eta - x_\xi z_\eta) \\
\zeta_z &= J(x_\xi y_\eta - y_\xi x_\eta) \\
\xi_t &= -x_\tau \xi_x - y_\tau \xi_y - z_\tau \xi_z \\
\eta_t &= -x_\tau \eta_x - y_\tau \eta_y - z_\tau \eta_z \\
\zeta_t &= -x_\tau \zeta_x - y_\tau \zeta_y - z_\tau \zeta_z
\end{align*} \] (2.67)

U, V and W are the contravariant velocity components in the \( \xi \), \( \eta \) and \( \zeta \) directions respectively and are defined as:
\[
\begin{align*}
U &= \xi_x u + \xi_y v + \xi_z w + \xi_t \\
V &= \eta_x u + \eta_y v + \eta_z w + \eta_t \\
W &= \zeta_x u + \zeta_y v + \zeta_z w + \zeta_t
\end{align*}
\] (2.68)

The system of equations is closed using the equation of state:

\[
p = (\gamma - 1)p \left( E - \frac{(u^2 + v^2 + w^2)}{2} \right)
\] (2.69)

2.4 Summary

In this chapter the methodology for the formulation of the governing equations that are solved by each of the numerical algorithms has been presented. The governing equations have also been presented in full. In the next chapter the spatial and time discretisation that is employed is described, together with the form of the artificial dissipation that is used.
NUMERICAL FORMULATION

3.1 Introduction

The chapter describes the spatial and time discretisations of the six sets of Euler equations presented in Chapter 2. The spatial discretisation is the same for each equation set. The time discretisation depends on whether the time integration is simply a means to obtain a steady-state solution, or whether a time-accurate solution is sought. The form of the artificial dissipation that is added explicitly to the discretised equation sets is also presented.

3.2 Spatial Discretisation

A finite-volume formulation is employed to discretise the equations of motion. The spatial and time discretisations are performed separately following the method of lines.\(^4\) This allows flexibility in the choice of the time-stepping scheme that is used to integrate the conserved variables in time.

A structured, body-fitted, curvilinear mesh is generated in each physical computational domain. In the corresponding transformed computational domains, the mesh points are equally spaced in the $\xi$, $\eta$, and $\zeta$ directions with mesh intervals $\Delta \xi = \Delta \eta = \Delta \zeta = 1$. The transformed computational domain is discretised into a set of contiguous cells; these cells are quadrilaterals in a two-dimensional mesh and hexahedrons in a three-dimensional mesh. The Euler equations (Equations (2.16), (2.25), (2.33), (2.42), (2.51) and (2.61)) are integrated over each cell in the transformed computational domain yielding a system of ordinary differential equations. It is important to note that the cells in the transformed computational domains are fixed in time. The solution variables are stored at the cell centres as shown in Figure 3.1 and represent cell-averaged quantities. The fluxes at the cell faces are evaluated using central differencing applied to the values of the flow variables at the cell centres on both sides of a cell face.
The cell-centered finite-volume scheme with central differencing leads to a second-order accurate space discretisation on a regularly spaced Cartesian mesh.

![Figure 3.1 Cell-centered finite-volume scheme.](image-url)

As an example, the semi-discretisation is applied to Equation (2.42), which describes the unsteady flow around an oscillating wing. The following set of ordinary differential equations is obtained for a cell with centre denoted \( i, j, k \), where \( i, j \) and \( k \) are the indices in the \( \xi, \eta, \) and \( \zeta \) directions respectively:

\[
\frac{d}{d\tau}(\mathcal{O}_{i,j,k}^T \mathbf{Q}_{i,j,k}) + (\mathbf{F}_{i+1/2,j,k} - \mathbf{F}_{i-1/2,j,k}) \\
+ (\mathbf{G}_{i,j+1/2,k} - \mathbf{G}_{i,j-1/2,k}) \\
+ (\mathbf{H}_{i,j,k+1/2} - \mathbf{H}_{i,j,k-1/2}) = 0
\]  

(3.1)

\( \mathbf{F}_{i \pm 1/2,j,k} \), \( \mathbf{G}_{i,j \pm 1/2,k} \) and \( \mathbf{H}_{i,j,k \pm 1/2} \) are the face-averaged \( \mathbf{F} \), \( \mathbf{G} \) and \( \mathbf{H} \) flux vectors evaluated at the cell faces. The flux vectors are calculated using central differencing:

\[
\mathbf{F}_{i \pm 1/2,j,k} = \frac{1}{2}(\mathbf{F}_{i,j,k} + \mathbf{F}_{i \pm 1,j,k})
\]

(3.2)

\[
\mathbf{G}_{i,j \pm 1/2,k} = \frac{1}{2}(\mathbf{G}_{i,j,k} + \mathbf{G}_{i,j \pm 1,k})
\]

(3.3)

\[
\mathbf{H}_{i,j,k \pm 1/2} = \frac{1}{2}(\mathbf{H}_{i,j,k} + \mathbf{H}_{i,j,k \pm 1})
\]

(3.4)
3.3 Artificial Dissipation

3.3.1 Introduction

The central-difference finite-volume discretisation leads to a non-dissipative approximation of the Euler equations with odd and even point decoupling. Artificial dissipation terms must be added to the discretised equations for two reasons. Firstly, there is the possibility of undamped oscillatory modes with alternate signs at odd and even points. These high-frequency modes, if not damped, can have an adverse effect on the convergence rate of a numerical scheme to steady-state. The second reason is for the clean capture of shock waves and of stagnation points without unwanted oscillations.

3.3.2 Dissipation Model

The artificial dissipation model employed is called the Jameson, Schmidt and Turkel model (i.e., the JST model).\textsuperscript{45-47} The dissipation terms consist of blended second and fourth differences of the conserved variables for the continuity and momentum equations and either energy, enthalpy or rothalpy for the energy equation, depending on the particular flowfield being modelled.

The dissipation terms that are added to Equation (3.1) are presented here. The dissipation terms that are added to the other sets of discretised equations can be defined in similar manner. Equation (3.1) with the dissipation terms added can be written in the operator form:

\[
\frac{d}{dt}\left(T_{i,j,k}^{-1}Q_{i,j,k}\right) + C(Q_{i,j,k}) + D(Q_{i,j,k}) = 0
\] (3.5)

where the discrete operator C accounts for spatial discretisation of the inviscid flux terms and D is the operator for the artificial dissipation. The operator D is defined as:

\[
D(Q_{i,j,k}) = -(D_\xi^2 - D_\eta^4 + D_\eta^2 - D_\eta^4 + D_\eta^2 - D_\eta^4)Q_{i,j,k}
\] (3.6)

where:

\[
D_\xi^2(Q_{i,j,k}) = \nabla_\xi^2[\left(\lambda_{i+1/2,j,k} + \epsilon_{i+1/2,j,k}^{(2)}\right)\Delta_\xi^2]Q_{i,j,k}
\] (3.7)

and:
\[ D_\xi^4(Q_{i,j,k}) = \nabla_\xi [(\lambda_{i+1/2,j,k}^{(4)} + \frac{1}{2}, j, k) \Delta_\xi \nabla_\xi \Delta_\xi] Q_{i,j,k} \quad (3.8) \]

\[ \Delta_\xi \text{ and } \nabla_\xi \text{ are the forward and backward difference operators in the } \xi \text{ direction, respectively, i.e.:} \]

\[ \Delta_\xi Q_{i,j,k} = Q_{i+1,j,k} - Q_{i,j,k} \quad (3.9) \]

\[ \nabla_\xi Q_{i,j,k} = Q_{i,j,k} - Q_{i-1,j,k} \quad (3.10) \]

The scaling factor \( \lambda \) is used to give the dissipation terms the proper scale. The original isotropic scaling factor in the JST model enabled the accurate prediction of complex inviscid flows using inviscid type meshes with cell aspect ratios of the order of one.\(^{48-51}\) However, this scaling factor can provide too much dissipation on meshes with higher cell aspect ratios and is not used in this work. Following Martinelli\(^{52}\) and Swanson and Turkel,\(^{53}\) an anisotropic scaling factor is employed, and is defined as:

\[ \lambda_{i+1/2,j,k} = \frac{1}{2} \left[ (\bar{\lambda}_\xi)_{i,j,k} + (\bar{\lambda}_\xi)_{i+1,j,k} \right] \quad (3.11) \]

where:

\[ \bar{\lambda}_\xi = \lambda_\xi \left[ 1 + \left( \frac{\lambda_\eta}{\lambda_\xi} \right)^\alpha + \left( \frac{\lambda_\zeta}{\lambda_\xi} \right)^\alpha \right] \quad (3.12) \]

\( \lambda_\xi, \lambda_\eta \) and \( \lambda_\zeta \) are the spectral radii of the inviscid flux Jacobian matrices in the three coordinate directions. The determination of the spectral radii of Equation set (2.42) is presented in Appendix C. The spectral radii are given by:

\[ \lambda_\xi = |U| + c_s \sqrt{\zeta_\xi^2 + \zeta_y^2 + \zeta_z^2} \quad (3.13) \]

\[ \lambda_\eta = |V| + c_s \sqrt{\eta_\eta^2 + \eta_y^2 + \eta_z^2} \quad (3.14) \]

\[ \lambda_\zeta = |W| + c_s \sqrt{\zeta_\zeta^2 + \zeta_y^2 + \zeta_z^2} \quad (3.15) \]

where U, V and W are the contravariant velocity components defined in Chapter 2 and \( c \) is the speed of sound. The exponent \( \alpha \) depends on cell aspect ratios and is set to a value in the range of 0.4-0.667 in the calculations reported herein.
The coefficients \( \varepsilon^{(2)} \) and \( \varepsilon^{(4)} \) use the normalised second difference of the pressure as a sensor for shocks and stagnation points, and are defined as:

\[
\varepsilon_{i + 1/2, j, k}^{(2)} = k^{(2)} \max(v_{i - 1, j, k}, v_{i, j, k}, v_{i + 1, j, k}, v_{i + 2, j, k})
\]

\[
\varepsilon_{i + 1/2, j, k}^{(4)} = \max[0, (k^{(4)} - \varepsilon_{i + 1/2, j, k}^{(2)})]
\]

where the switching function \( v \) is:

\[
v_{i, j, k} = \frac{|p_{i - 1, j, k} - 2p_{i, j, k} + p_{i + 1, j, k}|}{|p_{i - 1, j, k} + 2p_{i, j, k} + p_{i + 1, j, k}|}
\]

\( k^{(2)} \) and \( k^{(4)} \) are constants input by the user and control the amount of dissipation added. Typical values for \( k^{(2)} \) and \( k^{(4)} \) are 1/4 to 1/2 and 1/64 to 1/32, respectively. The dissipation terms in the \( \eta \) and \( \zeta \) directions are defined in a similar manner noting that:

\[
\tilde{\lambda}_\eta = \lambda_\eta \left[ 1 + \left( \frac{\lambda_\zeta}{\lambda_\eta} \right)^{\alpha} + \left( \frac{\lambda_\eta}{\lambda_\zeta} \right)^{\alpha} \right]
\]

and:

\[
\tilde{\lambda}_\zeta = \lambda_\zeta \left[ 1 + \left( \frac{\lambda_\eta}{\lambda_\zeta} \right)^{\alpha} + \left( \frac{\lambda_\eta}{\lambda_\zeta} \right)^{\alpha} \right]
\]

There are two dissipation mechanisms at work in this dissipation model. In smooth regions of the flowfield \( v \) is small and the dissipation is dominated by a fourth difference term which is third-order accurate in comparison to the convective terms. The fourth difference term damps the high-frequency modes that the central-difference scheme does not damp and allows convergence to a steady-state. In a region of high pressure gradients (i.e., in the neighbourhood of a shock wave or of a stagnation point) \( v \) is of the order of one. The second difference dissipation term is then activated and the scheme behaves locally like a first-order scheme. The fourth difference term is cut off to prevent it from causing oscillations.

### 3.3.3 Dissipation Terms for the Energy Equation

The dissipation terms for the energy equation depend on the flowfield being modelled and the corresponding equation set. In the steady algorithms FLO2DS and FLO3DS, the dissi-
dissipation terms for the energy equation are based on differences of total enthalpy. The total enthalpy, $H$, is constant in a steady inviscid flowfield in an inertial reference frame. By computing differences of total enthalpy, a constant value is maintained in the computational domain. In the steady propeller algorithm, PROP3DS, the dissipation terms for the energy equation are based on differences of total rothalpy, which is constant in a steadily-rotating non-inertial reference frame. The dissipation terms for the energy equation in the unsteady algorithms (i.e., FLO2DU, FLO3DU and PROP3DU) are based on differences of total energy.

### 3.3.4 Boundary Treatment of the Dissipation Terms

The implementation of boundary conditions in the finite-volume method is facilitated through the use of a layer of ghost cells exterior to the flow domain. The values of the conserved variables in the ghost cells are updated after the flowfield variables are calculated, and are not part of the flow solution. These ghost cells ensure that the second differences of the conserved variables can be calculated in the first and last cells interior cells in a given coordinate direction. However, the calculation of the fourth difference terms in the first and last interior cells requires information at two cells on either side of the cell under consideration. Therefore, special treatment of the fourth-difference dissipation term is required at the boundaries. Moreover, special treatment both of the second and of the fourth difference terms is also required at solid surface boundaries. The boundary dissipation stencils used in this work were recommended by Swanson and Turkel\(^53\) and have proven satisfactory.

### 3.3.5 Alternative Dissipation Models

The JST dissipation model is easy to implement and is computationally economical. However, the disadvantage of this model is that computed shock waves have a thickness of three or four cells, whereas a real shock wave has a thickness of the order of one micrometer, a small fraction of a single cell width. An increase in the popularity of upwind schemes, coupled with their higher level of accuracy, led to the design of alternative dissipation models that would make the central-difference scheme comparable in accuracy to an upwind scheme. Two such models are the matrix-valued dissipation model (MATD) of Turkel\(^54-56\) and the convective upwind and split pressure (CUSP) dissipation model of Jameson.\(^57-59\) These models lead to improved accuracy; the CUSP scheme is also capable of computing
a discrete shock wave with one interior point. However, they require more computational effort than the JST model and also lead to a decrease in convergence rate due to the lower levels of artificial dissipation. For these reasons they are not implemented here.

3.4 Time-Stepping Schemes

3.4.1 Introduction

The system of ordinary differential equations obtained by integrating any one of the six sets of Euler equations over the fixed cells in the discretised transformed computational domain, and adding the dissipation terms, can be written as:

\[
\frac{d}{dt}(Q_{i,j,k}) + R(Q_{i,j,k}) = 0 \tag{3.21}
\]

where \( R(Q_{i,j,k}) \) is the residual:

\[
R(Q_{i,j,k}) = \frac{1}{J_{i,j,k}}(C + D)Q_{i,j,k} \tag{3.22}
\]

The operators \( C \) and \( D \) are the spatial and artificial dissipation operators. Note that \( \tau \) in Equation (3.21) is replaced by \( t \), which is now used to denote time both in the physical and in the transformed computational domains.

The system of ordinary differential equations defined by Equation (3.21) can be integrated in time using either an explicit or implicit time-stepping scheme. With explicit time-stepping the values of the conserved variables at the current time step are calculated using values of the variables at the end of the previous time step. With implicit time-stepping the new values of the variables are calculated using the as yet unknown values at the current time step, so that a set of coupled equations must be solved. The maximum permissible time step for an explicit scheme is limited by the Courant-Friedrichs-Lewy (CFL) stability condition which states that the domain of dependence of the discretised equations must contain the domain of dependence of the corresponding differential equations. Implicit schemes do not have any limit on the size of the permissible time step, but have the disadvantage that a set of coupled equations must be solved at each time step, which can be very expensive, especially in three-dimensional flow calculations.
An explicit time-stepping scheme is chosen to integrate the semi-discretised equations in the steady algorithms. There are three reasons for this choice. The first is that an explicit scheme requires less computational effort per time step than an implicit scheme. The second is that when the explicit scheme employed here is coupled with the convergence acceleration devices (i.e., local time-stepping, implicit residual averaging and multigrid) it leads to an extremely efficient algorithm for steady flow calculations. The third is that an explicit scheme is relatively straightforward to implement.

An implicit time-stepping scheme is, however, employed in the unsteady algorithms. If an explicit scheme was used to calculate the unsteady flows, the maximum time step for stability would be so small in comparison with the characteristic time scales that an excessively large number of time steps would be required. An implicit scheme is therefore employed here so that the time step can be chosen based on the physics to be resolved and on the desired accuracy of the flow solution. The explicit and implicit time-stepping schemes employed are now described in the following sections.

### 3.4.2 Explicit Time-Stepping Scheme

A multistage Runge-Kutta scheme is used to integrate Equation (3.21) in time in the steady algorithms. In general a multistage scheme is designed to have a high order of accuracy. However, because the time-stepping is performed solely to obtain a steady solution, the order of accuracy in time is unimportant. There are two basic requirements for the multistage scheme. The first is that it must allow rapid convergence to a steady-state with the minimum amount of computational effort. The second is that it must be effective at damping high frequency error components, so that it can be used in a multigrid procedure. To design a multistage scheme with good stability and damping properties, the convective and dissipative terms are treated separately. This leads to a class of hybrid multistage scheme.

Let $Q^{(n)}$ denote the value of the conserved variable vector after $n$ time steps (i.e., at time level $n\Delta t$) and let $R^{(q)} = R(Q^{(q)})$. After dropping the $i, j, k$ subscripts, the hybrid $m$-stage scheme to advance by a time $\Delta t$ is:

\[
\begin{align*}
Q^{(0)} &= Q^{(n)} \\
Q^{(1)} &= Q^{(0)} - \alpha_1 \Delta t R^{(0)} \\
Q^{(2)} &= Q^{(0)} - \alpha_2 \Delta t R^{(1)}
\end{align*}
\]
\[
\begin{align*}
Q^{(m-1)} &= Q^{(0)} - \alpha_{m-1} \Delta t R^{(m-2)} \\
Q^{(m)} &= Q^{(0)} - \alpha_m \Delta t R^{(m-1)} \\
Q^{(n+1)} &= Q^{(m)}
\end{align*}
\]

where the residual at stage \((q+1)\) is:

\[
R^{(q)} = \frac{1}{J_{i,j,k}} \left[ \sum_{r=0}^{q} \{ \beta_{qr} C(Q^{(r)}) + \gamma_{qr} D(Q^{(r)}) \} \right]
\]

subject to the consistency constraints that:

\[
\sum_{r=0}^{q} \beta_{qr} = 1 \quad \text{and} \quad \sum_{r=0}^{q} \gamma_{qr} = 1
\]

The basic parameters \(\alpha\) and the weighting factors, \(\beta\) and \(\gamma\), that define the \(m\)-stage scheme are obtained from the desired stability and damping requirements. A five-stage scheme with three evaluations of the dissipation terms, on the first, third and fifth stages, has been found to be very effective\(^{62,63}\) and it is used here. The basic parameters are:

\[
\alpha_1 = \frac{1}{4}, \quad \alpha_2 = \frac{1}{6}, \quad \alpha_3 = \frac{3}{8}, \quad \alpha_4 = \frac{1}{2} \quad \text{and} \quad \alpha_5 = 1
\]

with the weighting factors:

\[
\beta_{qr} = \begin{cases} 
1 & q = r \\
0 & q \neq r 
\end{cases}
\]

and:

\[
\begin{align*}
\gamma_{00} &= 1 \\
\gamma_{10} &= 1, \quad \gamma_{11} = 0 \\
\gamma_{20} &= \Gamma_3, \quad \gamma_{21} = 0, \quad \gamma_{22} = \bar{\gamma}_3 \\
\gamma_{30} &= \Gamma_3, \quad \gamma_{31} = 0, \quad \gamma_{32} = \bar{\gamma}_3, \quad \gamma_{33} = 0 \\
\gamma_{40} &= \Gamma_3 \Gamma_5, \quad \gamma_{41} = 0, \quad \gamma_{42} = \bar{\gamma}_3 \Gamma_5, \quad \gamma_{43} = 0, \quad \gamma_{44} = \bar{\gamma}_5
\end{align*}
\]
where \( \Gamma_3 = (1 - \bar{\gamma}_3), \quad \Gamma_5 = (1 - \bar{\gamma}_5), \quad \bar{\gamma}_3 = 0.56 \) and \( \bar{\gamma}_5 = 0.44 \). The stability and damping characteristics of this scheme have been presented in the literature and are not repeated here.\(^{43,47}\) The maximum CFL number of this explicit scheme is 4.

### 3.4.3 Implicit Time-Stepping Scheme

The implicit scheme of Jameson in employed for the time-accurate integration of Equation (3.21).\(^{64}\) The set of coupled equations at each time step is solved using the multistage explicit scheme in an inner iteration. This is sometimes referred to as the dual time approach.

An implicit time-stepping scheme is obtained by approximating Equation (3.21) at time level \( (n+1)\Delta t \) by:

\[
D_t Q^{(n+1)} + R(Q^{(n+1)}) = 0
\]  
(3.29)

where the i, j, k subscripts have been dropped. In Equation (3.29) \( D_t \) is a \( k^{th} \) order accurate backward difference operator in time of the form:

\[
D_t = \frac{1}{\Delta t} \sum_{q=1}^{k} \frac{1}{q} (\nabla)^q
\]  
(3.30)

where \( \nabla \) is defined as:

\[
\nabla Q^{(n+1)} = Q^{(n+1)} - Q^{(n)}
\]  
(3.31)

In the present work a second order (\( k=2 \)) accurate backward time discretisation is employed, so that Equation (3.29) can be written as:

\[
\frac{3Q^{(n+1)} - 4Q^{(n)} + Q^{(n-1)}}{2\Delta t} + R(Q^{(n+1)}) = 0
\]  
(3.32)

Equation (3.32) is treated as a modified steady-state problem in a pseudo time \( t^* \) by expressing it as:

\[
\frac{dQ}{dt^*} + R^*(Q) = 0
\]  
(3.33)

where the modified residual \( R^*(Q) \) is defined as:
\[ R^*(Q) = \frac{3Q^{(n+1)} - 4Q^{(n)} + Q^{(n-1)}}{2\Delta t} + R(Q^{(n+1)}) \]  

(3.34)

The term:

\[ -\frac{4Q^{(n)} + Q^{(n-1)}}{2\Delta t} = S(Q^{(n)}, Q^{(n-1)}) \]  

(3.35)

is treated as a fixed source term in the multigrid procedure described later and can be omitted from the calculation of the coarse grid residuals. The solution of Equation (3.29) is then equivalent to marching Equation (3.33) to a steady-state in pseudo time. Equation (3.33) is solved using the multistage explicit time-stepping scheme described in §3.4.2.

If Equation (3.33) is solved directly using the multistage time-stepping scheme, a stability problem may occur when the ratio of the local fictitious time step \( \Delta t^* \) to the physical time step \( \Delta t \) is large. Arnone et al.\(^65-67\) and Gaitonde\(^68-70\) overcame this stability problem by limiting the value of the local time fictitious step \( \Delta t^* \) in the computational domain. Melson et al.\(^71-72\) demonstrated that the stability problem lies in the fact that all terms in the approximation of the physical time derivative are treated explicitly in the modified residual in the multistage scheme. They modified the scheme, and eliminated the stability problem, by treating the \( 3Q^{(n+1)}/2\Delta t \) term implicitly. The multistage scheme at stage \( m \) is then rewritten as:

\[
\left(1 + \frac{3}{2} \alpha_m \lambda\right)Q^{(m)} = Q^{(0)} + \frac{3}{2} \alpha_m \lambda Q^{(m-1)} - \alpha_m \Delta t^* \left[ R^{(m)} + \left(\frac{3Q^{(m-1)} - 4Q^{(n)} + Q^{(n-1)}}{2\Delta t}\right)\right]
\]  

(3.36)

where the ratio of the fictitious to the physical time step, \( \lambda \), is:

\[
\lambda = \frac{\Delta t^*}{\Delta t}
\]  

(3.37)

This modification has no negligible effect on convergence but increases considerably the robustness of the implicit scheme. The modification has been employed by several researchers using this implicit method, \(^73,74\) by Jameson and his co-workers\(^75,76\) and is also implemented here.
Third-order differencing has also been used by some researchers\textsuperscript{73,76,77} with the possible advantage that fewer time steps may be required in comparison with a second-order scheme to obtain a desired level of solution accuracy. However, third-order differencing suffers the penalty that the flow solution and the grid points at an extra time level must be stored, thus increasing the computer memory demands. Third-order differencing has therefore not been implemented here, but may well be worth employing in the future.

### 3.5 Summary

In this chapter, the basic numerical formulation has been presented. The spatial and time discretisations have been described together with the form of the artificial dissipation added to the discretised equations. The next chapter describes the full set of boundary conditions and their implementation in the numerical algorithms.
CHAPTER 4

BOUNDARY AND INITIAL CONDITIONS

4.1 Introduction

The selection and implementation of an appropriate set of boundary conditions is extremely important in the development of an accurate and efficient numerical algorithm. It is the applied boundary conditions that make a solution of the governing equations unique to a particular problem. This chapter describes the complete set of boundary conditions implemented in the six numerical algorithms. The initial conditions used in each algorithm to initialise the flow solution are also detailed.

4.2 Physical and Transformed Computational Domains

In external flow aerodynamic calculations the extent of the physical computational domain is dictated by the model around which the flow is being calculated, the type of body-fitted grid employed and by accuracy requirements. In the far-field the infinite domain is truncated at a finite distance from the primary region of interest and suitable boundary conditions are applied along this artificial boundary.

4.2.1 Airfoil Computational Domain

A C-type grid is employed here to define the physical computational domain for the two-dimensional airfoil flowfield calculations. The far-field boundary is placed at approximately 20 chord lengths from the airfoil surface. This is a sufficiently large distance for obtaining accurate flow solutions only if the far-field vortex effect is incorporated in the implementation of the far-field boundary condition. The location of the boundaries in physical space are shown in Figure 4.1. The corresponding transformed computational domain and the applied boundary conditions are presented in Figure 4.2. It is important to note
that in Figures 4.1, 4.3 and 4.5 the far-field boundary is shown unrealistically close to the airfoil, wing and propeller surfaces for illustration purposes.

### 4.2.2 Wing Computational Domain

A C-H mesh topology is employed in the wing flowfield calculations, with the C-type mesh in the streamwise direction. The far-field boundary is placed closer to the wing surface than in the airfoil calculations. This is due to the fact that disturbances decay more rapidly in a three-dimensional flow than in a two-dimensional flow. The far-field boundary is placed at a distance of approximately 10 root chords from the wing surface in the streamwise direction and at a distance of one wing span beyond the wing tip in the spanwise direction. These distances are typical for wing flowfield calculations.\textsuperscript{81-83} The physical and transformed computational domains and the boundary conditions applied are presented in Figures 4.3 and 4.4.

### 4.2.3 Propeller Computational Domain

The extent of computational domain for the calculation of the flowfield around a propeller depends on whether the inflow is axisymmetric or asymmetric. For the calculation of the steady flow around a propeller with an axisymmetric inflow, the flow is periodic from one inter-blade region to the next. It is only necessary, therefore, to model the flow in one inter-blade region and then to apply periodicity at the periodic boundaries. For the transient propeller calculations (i.e., asymmetric inflow) the flow is not periodic in each inter-blade region, which means that the complete annulus (i.e., all N inter-blade regions, where N is the number of propeller blades) must be modelled.

A C-H grid system is employed for all propeller calculations. The C-type grid is in the streamwise direction and the H-type grid in the circumferential direction. The far-field boundary is placed at a distance of one propeller diameter beyond the blade tip and one propeller diameter downstream of the blade trailing edge. These locations are typical for inviscid propeller flowfield calculations.\textsuperscript{26}

The boundaries of the computational domain in physical space for a steady two-bladed propeller calculation are presented in Figure 4.5. The computational domain spans 180° in the
circumferential direction. The corresponding transformed computational domain and the boundary conditions are shown in Figure 4.6.

The physical computational domain for an unsteady two-bladed propeller flowfield calculation is shown in Figure 4.7. The computational domain contains the two inter-blade regions. Figure 4.8 shows the corresponding computational domain in transformed space. The boundary conditions applied are the same as for the steady propeller calculation, except that the periodic and singular axis boundary conditions are no longer required as continuity is maintained between the inter-blade regions.
**Physical Space**

Figure 4.1 Physical computational domain for the steady and unsteady airfoil flowfield calculations.

**Transformed Space**

Figure 4.2 Transformed computational domain and applied boundary conditions for the steady and unsteady airfoil flowfield calculations.
Physical computational domain for the steady and unsteady wing flowfield calculations.

Transformed computational domain and applied boundary conditions for the steady and unsteady wing flowfield calculations.
Figure 4.5  Physical computational domain for a steady two-bladed propeller flowfield calculation.

Figure 4.6  Transformed computational domain and applied boundary conditions for a steady two-bladed propeller flowfield calculation.
**Figure 4.7** Physical computational domain for an unsteady two-bladed propeller flowfield calculation.

**Figure 4.8** Transformed computational domain for an unsteady two-bladed propeller flowfield calculation.
4.3 Boundary Conditions

4.3.1 Implementation of Boundary Conditions

The standard method for the implementation of boundary conditions in a cell-centered finite-volume algorithm is to assign values of the flowfield variables to ghost cells exterior to the flow domain. The ghost cells variables are then used when updating the flow solution in the solution process. The ghost cell variables are themselves updated after new values of the conserved variables and the pressure in the flow domain are obtained. In the numerical algorithms developed in this work a single layer of ghost cells is employed. The only geometrical information that is required about the ghost cells are the Cartesian coordinates of the cell centres which maybe used for the three-point pressure extrapolation. The coordinates of the cell centres can be obtained by linear extrapolation of the first interior cell centre locations adjacent to the boundary.

The location of the ghost cells in a two-dimensional computational domain are shown in Figure 4.9.

![Diagram of ghost cell locations](image)

*Figure 4.9* Ghost cell locations for implementation of boundary conditions.

The use of the ghost cell approach to implement boundary conditions requires the division of the transformed computational domain for a transient propeller calculation into sub-domains in order to implement the solid surface boundary conditions. Each sub-domain contains a single inter-blade region, as in steady propeller calculations. Continuity is
maintained between the sub-domains using the ghost cells. This is referred to as a multi-block strategy.

4.3.2 Solid Surface Boundary Condition

The airfoil, wing and propeller blade and nacelle surfaces are treated as solid surfaces. To implement the solid surface boundary condition, appropriate values of the conserved variables (i.e., \( \rho, \rho_u, \rho_v, \rho_w \) and \( \rho_E \)) and the pressure are assigned to the ghost cells adjacent to the surface. The implementation of the boundary conditions is described for a three-dimensional computational domain.

The ghost cell density is extrapolated from the nearest cell inside the flow domain. With the points defined as in Figure 4.10 the density in ghost cell 0 is:

\[
\rho_0 = \rho_1
\]

(4.1)

![Figure 4.10](image)

**Figure 4.10** Implementation of the solid surface boundary condition.

The ghost cell velocity is set to enforce flow tangency at the body surface. For the \( \eta = \text{constant} \) surface in Figure 4.10 the velocity components in the ghost cell are calculated as:

\[
\begin{align*}
    u_0 &= u_1 - 2 \hat{\eta}_x \hat{V}_l \\
    v_0 &= v_1 - 2 \hat{\eta}_y \hat{V}_l \\
    w_0 &= w_1 - 2 \hat{\eta}_z \hat{V}_l
\end{align*}
\]

(4.2)
where \( \hat{\eta}_x \), \( \hat{\eta}_y \) and \( \hat{\eta}_z \) are the normalised metrics (i.e., components of the unit normal vector to the surface) defined as:

\[
\begin{align*}
\hat{\eta}_x &= \frac{\eta_x}{\sqrt{\eta_x^2 + \eta_y^2 + \eta_z^2}} \\
\hat{\eta}_y &= \frac{\eta_y}{\sqrt{\eta_x^2 + \eta_y^2 + \eta_z^2}} \\
\hat{\eta}_z &= \frac{\eta_z}{\sqrt{\eta_x^2 + \eta_y^2 + \eta_z^2}}
\end{align*}
\] (4.3)

and \( \hat{V}_1 \) is the normalised contravariant velocity at cell centre 1 given by:

\[
\hat{V}_1 = u_1 \hat{\eta}_x + v_1 \hat{\eta}_y + w_1 \hat{\eta}_z + \hat{\eta}_t
\] (4.4)

Using the expression \( \hat{\eta}_x^2 + \hat{\eta}_y^2 + \hat{\eta}_z^2 = 1 \) it can be shown that \( \hat{V} \) is zero at the surface, as required. It is important to note that for fixed body calculations (as opposed to moving) the \( \xi_t \), \( \eta_t \) and \( \zeta_t \) metric terms are zero.

The normal momentum equation is used to obtain the ghost cell pressure in the steady airfoil and wing algorithms. The most general form of this equation is obtained by substituting the expression obtained from the streamline differentiation of the wall boundary condition for a fixed surface into the inner product of the inviscid momentum equation and the unit normal vector to the surface.\(^{84,85}\) In this derivation it is also assumed that the flow is not subject to any body forces. The normal momentum equation is given as:

\[
\frac{\partial p}{\partial \eta} = \frac{1}{|J^{-1}\nabla \eta|^2} \left\{ (\rho \mathbf{V} \cdot J^{-1}\nabla \xi) \cdot \left( \mathbf{V} \cdot \frac{\partial}{\partial \xi} (J^{-1}\nabla \eta) \right) \right.
\]
\[
+ (\rho \mathbf{V} \cdot J^{-1}\nabla \zeta) \cdot \left( \mathbf{V} \cdot \frac{\partial}{\partial \zeta} (J^{-1}\nabla \eta) \right) \right.
\]
\[
- (J^{-1}\nabla \xi \cdot J^{-1}\nabla \eta) \frac{\partial p}{\partial \xi} - (J^{-1}\nabla \zeta \cdot J^{-1}\nabla \eta) \frac{\partial p}{\partial \zeta} \right\}
\] (4.5)

where \( \mathbf{V} \) is the velocity vector and \( J^{-1}\nabla k \) is the directed area of the cell face normal to a \( k = \) constant surface, where \( k = \xi, \eta \) or \( \zeta \) (see Appendix B, §B.4). Equation (4.5) is numerically approximated at the first interior cell using the average area method.\(^{85}\) A simple first-
order finite-difference approximation is used to approximate the pressure derivative. The ghost cell pressure can then be easily calculated.

The normal momentum equation presented above can be modified to include body force terms (e.g., Coriolis and centrifugal forces) and time-dependent body motion.\textsuperscript{44,86} However, the extra terms obtained by the inclusion of these modifications makes the implementation of the normal momentum equation more difficult.\textsuperscript{87} A simpler, but slightly less accurate approach,\textsuperscript{79} (i.e., the shape of the surface is not accounted for) is to extrapolate the pressure in the ghost cells from the pressure in the cells inside the flow domain. This method is implemented here in the transient algorithms and also in the steady propeller algorithm.

Referring to Figure 4.10 the ghost cell pressure, using three-point extrapolation, is given by:

\[
p_0 = \left( \frac{s_3}{s_3 - s_1} \right) p_1 - \left( \frac{s_3}{s_3 - s_2} \right) p_2 + \left( \frac{s_2}{s_3 - s_1} \right) p_3 \tag{4.6}
\]

where \(s_1, s_2\) and \(s_3\) are the distances between the cell centres 1, 2 and 3 and cell centre 0.

Once the values of the primitive variables (i.e., \(p, \rho, u, v\) and \(w\)) are known in the ghost cell, the values of the conserved variables can then be calculated.

### 4.3.3 Symmetry Plane Boundary Condition

As the name implies, the symmetry plane boundary condition is used to apply symmetry across a plane and is employed in the wing calculations only. In Figure 4.11 the symmetry plane is a surface of constant \(\xi\). The density and pressure at ghost cell centre 0 are set equal to the values at the mirror point 1, i.e.:

\[
p_0 = \rho_1 \tag{4.7}
\]

\[
p_0 = \rho_1 \tag{4.8}
\]

The velocity components in the ghost cell are calculated so that the contravariant velocity components normal to the \(\xi\) = constant surface in the ghost and flow cells are symmetric. This is achieved by setting the ghost cell velocities as:
\[ u_0 = u_1 - 2\hat{x}_x \hat{U}_1 \]
\[ v_0 = v_1 - 2\hat{y}_y \hat{U}_1 \]
\[ w_0 = w_1 - 2\hat{z}_z \hat{U}_1 \]

where \( \hat{x}_x \), \( \hat{y}_y \) and \( \hat{z}_z \) are the normalised metrics and \( \hat{U}_1 \) is the normalised contravariant velocity component at point 1. Since \( \hat{x}_x^2 + \hat{y}_y^2 + \hat{z}_z^2 = 1 \), \( \hat{U}_0 = -\hat{U}_1 \) as required.

The values of the conserved variables can be obtained from the primitive variables.

\[ \xi = \text{constant surface} \]

**Figure 4.11** Implementation of the symmetry plane boundary condition.

### 4.3.4 Periodic Boundary Condition

This boundary condition is employed only in the calculation of the flowfield around a propeller with an axisymmetric inflow. To demonstrate the implementation of the periodic boundary condition it is assumed here that the periodicity exists around the x axis in the physical space and that the magnitude of the periodic angle is \( \theta_p \). The corresponding transformed domain is shown in Figure 4.12. The density and pressure in the ghost cell 0 are set as:

\[ \rho_0 = \rho_3 \]  
\[ p_0 = p_3 \]

The Cartesian velocity components are evaluated as:
\[ u_0 = u_3 \\
\[ v_0 = w_3 \sin \theta_p + v_3 \cos \theta_p \\
\[ w_0 = w_3 \cos \theta_p - v_3 \sin \theta_p \] (4.12)

The sign of \( \theta_p \) is dictated by the direction in which the flow variables are rotated. The values of the flow variables in the ghost cell 4 at the opposing boundary can be obtained from the values in the flow cell 1.

![Figure 4.12](image-url) Implementation of the periodic boundary condition.

### 4.3.5 Freestream Boundary Condition

The freestream boundary condition is employed for the wing calculations only. The values of the conserved variables and the pressure in a ghost cell are obtained from the freestream values of the primitive variables which, for the wing calculations, are:

\[ \rho_\infty = 1.0 \]
\[ p_\infty = 1.0 \]
\[ u_\infty = M_{\infty} c_\infty \cos \alpha \] (4.13)
\[ v_\infty = M_{\infty} c_\infty \sin \alpha \]
\[ w_\infty = 0.0 \]

where \( M_{\infty} \) is the freestream Mach number, \( c_\infty \) is the freestream speed of sound and \( \alpha \) is the flow angle of incidence.
4.3.6 Inflow/Outflow Boundary Condition

4.3.6.1 Riemann Invariants

The implementation of the inflow/outflow boundary condition is based on the introduction of the Riemann invariants for a one-dimensional homentropic flow normal to the boundary.

Let the inflow/outflow boundary be a surface of constant $\eta$ and let the direction of increasing $\eta$ point to the exterior of the computational domain as shown in Figure 4.13.

$\eta$ is constant surface

**Figure 4.13** Implementation of the inflow/outflow boundary condition.

The Riemann invariants are defined as:

$$R^\pm = \hat{\mathbf{v}} \pm \frac{2c}{\gamma - 1}$$  \hspace{1cm} (4.14)

where $c$ is the speed of sound, $\gamma$ is the ratio of constant specific heats and $\hat{\mathbf{v}}$ is the absolute normal velocity defined as:

$$\hat{\mathbf{v}} = \hat{\mathbf{V}} - \hat{\eta}_t$$  \hspace{1cm} (4.15)

$\hat{\mathbf{V}}$ is the normalised contravariant velocity given by:

$$\hat{\mathbf{V}} = u\hat{\eta}_x + v\hat{\eta}_y + w\hat{\eta}_z + \hat{\eta}_t$$  \hspace{1cm} (4.16)

The Riemann invariants are considered constant along the characteristics defined by:

$$\left(\frac{d\eta}{dt}\right)^\pm = (\hat{\mathbf{V}} \pm c) \cdot |\nabla \eta|$$  \hspace{1cm} (4.17)
The incoming Riemann invariant, $R^-$, is calculated from conditions outside the computational domain (i.e., freestream conditions) while the outgoing Riemann invariant, $R^+$, is calculated from values in the interior cells adjacent to the boundary.

The absolute normal velocity and the speed of sound in ghost cell 0 are obtained by adding and subtracting the Riemann invariants, i.e.:

$$
\hat{v}_0 = \frac{1}{2}(R^+ + R^-)
$$

$$
c = \frac{\gamma - 1}{4}(R^+ - R^-)
$$

The sign of the relative normal velocity $\hat{V}_0 = \hat{v}_0 + \hat{n}_t$ determines whether the flow is an inflow ($\hat{V}_0 < 0$) or an outflow ($\hat{V}_0 > 0$). If the flow is an inflow the tangential velocity and the entropy in the ghost cell are extrapolated from the exterior while if it is an outflow these variables are extrapolated from the first interior cell. It is important to note that entropy itself is not actually used, but a variable that has the same functional dependence as entropy.

The Cartesian velocity components in the ghost cell can then be obtained by decomposing the normal and tangential velocities to give:

$$
u_0 = u_{\text{ref}} + \hat{n}_x(\hat{v}_0 - \hat{v}_{\text{ref}})
$$

$$
v_0 = v_{\text{ref}} + \hat{n}_y(\hat{v}_0 - \hat{v}_{\text{ref}})
$$

$$
w_0 = w_{\text{ref}} + \hat{n}_z(\hat{v}_0 - \hat{v}_{\text{ref}})
$$

where $\text{ref}$ denotes the freestream for an inflow and the first interior cell for an outflow. Note that the use of Equations (4.20) avoids the need for the explicit calculation of the tangential velocity components.

The density and the pressure in the ghost cells can be obtained by combining the extrapolated entropy and the calculated speed of sound.

### 4.3.6.2 Far-Field Correction for Airfoil Calculations

The circulation at a far-field boundary in a two-dimensional flow due to the presence of a lifting airfoil can be accounted for by viewing the airfoil as a point vortex and adding the
perturbation velocities at the boundary to the freestream velocity. The effective Cartesian velocity components at the far-field boundary are then given by:

\[ u_{\text{eff}} = u_{\infty} + \frac{V_{\infty} C_L c \sin \phi \sqrt{1 - M_{\infty}^2}}{4\pi r (1 - M_{\infty}^2 \sin^2 (\phi - \alpha))} \]  
\[ v_{\text{eff}} = v_{\infty} + \frac{V_{\infty} C_L c \cos \phi \sqrt{1 - M_{\infty}^2}}{4\pi r (1 - M_{\infty}^2 \sin^2 (\phi - \alpha))} \]  

where \( V_{\infty} \) and \( M_{\infty} \) are the freestream speed and Mach number respectively, \( c \) is the airfoil chord, \( C_L \) is the lift coefficient, \( \alpha \) is the flow angle of incidence, and \( r \) and \( \phi \) are the polar coordinates to the point of application on the outer boundary from an origin at the quarter chord point on the airfoil centre line. The effective velocity components are used in the implementation of the inflow/outflow boundary condition for all airfoil flowfield calculations. Using these velocity components means that the far-field boundary can be placed at approximately 20 chords from the airfoil surface, instead of a minimum of 50 chords that would be required if the vortex effect was not incorporated.

The effects of the presence of the wing and propeller surfaces at the far-field boundary are not accounted for in the three-dimensional calculations. The outer boundaries are located sufficiently far away from the models that the application of freestream boundary conditions should not affect the computed flowfield.

### 4.3.7 Radial Equilibrium Boundary Condition

The radial equilibrium boundary condition is applied at the downstream boundary in the transformed computational domain for the steady and unsteady propeller flowfield calculations. This boundary condition is implemented by extrapolating the density and the Cartesian velocity components from the first interior point and by solving the simple radial momentum equation:

\[ \frac{dp}{dr} = \frac{\rho V_{\theta}^2}{r} \]  

for the pressure. In Equation (4.23) \( r \) is the radial distance from the axis of rotation and \( V_{\theta} \) is the tangential velocity. The simple radial equilibrium equation is solved for the radial
pressure distribution by a trapezoidal rule integration, starting from a specified value of the pressure at the far-field boundary. Since the integration is applied along every radial line at the boundary, the pressure can vary circumferentially as well as radially.

### 4.3.8 Singular Axis Boundary Condition

The singular axis boundary condition is employed in the calculation of the flowfield around a propeller with an axisymmetric inflow, and so is implemented in the steady propeller algorithm only. For all steady propeller calculations the singular axis coincides with the x axis, the axis of rotation (i.e., \( \omega = \omega_x \mathbf{i} \)). When mapped to the transformed space the singular axis opens out to an entire side of the computational domain as shown in Figures 4.5 and 4.6. In Figure 4.14 the singular axis is mapped to a \( \xi = \text{constant} \) plane. The ghost cell 0 density and pressure are extrapolated directly from the interior cell 1:

\[
\rho_0 = \rho_1 \quad (4.24)
\]
\[
p_0 = p_1 \quad (4.25)
\]

The Cartesian velocity components are extrapolated as follows, noting that the singular axis coincides with the x axis:

\[
u_0 = -v_1 \]
\[
w_0 = -w_1 \quad (4.26)
\]

![Figure 4.14](image)

*Figure 4.14* Implementation of the singular axis boundary condition.
4.4 Initial Conditions

The equation sets solved by the numerical algorithms are hyperbolic with respect to time. To integrate these equations forward in time, proper initial conditions must be specified. For the steady calculations the initial values of the conserved variables and the pressure are set equal to appropriate freestream values (see §4.3.5).

Each transient calculation performed in this work is preceded by a steady calculation. The converged solution of the steady calculation is then used as the initial unsteady solution. This has proven to be an effective strategy when performing transient flow calculations.

4.5 Summary

The boundary conditions employed by the algorithms and their implementation have been described in the chapter. In Chapter 5 the convergence acceleration devices (i.e., local time-stepping, variable coefficient implicit residual averaging and multigrid) used to accelerate solution convergence to steady-state are detailed.
5.1 Introduction

Explicit time-stepping is employed in the steady and transient algorithms developed in this work. In the steady algorithms the explicit time-stepping is used to obtain steady solutions of the Euler equations. It is also employed in an inner loop at each physical time step in the unsteady algorithms to obtain steady solutions of modified equation sets in a fictitious time $t^*$. As described in §3.4.2, the explicit time-stepping is performed using a multistage scheme.

Well-proven convergence acceleration techniques can be applied with the explicit time-stepping scheme to significantly increase the rate of convergence to a steady-state. Three convergence acceleration techniques are employed concurrently here: local time-stepping, implicit residual averaging and multigrid. Each of these techniques is described in this chapter for implementation in a three-dimensional algorithm.

5.2 Local Time-Stepping

The most obvious way to accelerate convergence to a steady-state using an explicit scheme is to increase the time step. The maximum time step, however, that can be used when updating an individual cell is limited by the CFL stability criterion. If global time stepping is employed, every cell in the computational domain is updated using the same time step. The maximum global time step that can be used is determined by the shortest time step in the domain; this occurs for the smallest cell. For external flow calculations that employ highly-stretched meshes the global time step is then governed by the very small cells near the solid surfaces that are much smaller that those in the far-field. To avoid this limitation each cell can be updated with its own maximum time step. This techniques is called local time-stepping and, according to Turkel,\textsuperscript{89} was first introduced by Li.\textsuperscript{90} The time accuracy of the flow
solution is destroyed but a significant increase in the rate of convergence is obtained. The
time step is each cell is set as:

$$\Delta t = \frac{\text{CFL}}{\lambda_\zeta + \lambda_\eta + \lambda_\zeta}$$

(5.1)

where CFL is the Courant number of the explicit scheme and $\lambda_\zeta$, $\lambda_\eta$ and $\lambda_\zeta$ are the spectral
radii of the inviscid flux Jacobian matrices defined in §3.3.2.

5.3Implicit Residual Averaging

5.3.1General Formulation

An implicit smoothing of the residuals can be used to extend both the stability range and
the robustness of the basic explicit time-stepping scheme. This technique was first intro­
duced for Runge-Kutta schemes by Jameson.\textsuperscript{91} The residual smoothing is applied in the
factored form:

$$(1 - \beta_\zeta \nabla_\zeta \Delta_\zeta)(1 - \beta_\eta \nabla_\eta \Delta_\eta)(1 - \beta_\zeta \nabla_\zeta \Delta_\zeta)\bar{R}^{(m)}_{i,j,k} = \bar{R}^{(m)}_{i,j,k}$$

(5.2)

for cell $i,j,k$, where $\nabla \Delta$ is a standard second difference operator given by:

$$\nabla_\zeta \Delta_\zeta \bar{R}^{(m)}_{i,j,k} = \bar{R}^{(m)}_{i-1,j,k} - 2\bar{R}^{(m)}_{i,j,k} + \bar{R}^{(m)}_{i+1,j,k}$$

(5.3)

$\bar{R}^{(m)}_{i,j,k}$ is the residual of the unsmoothed scheme computed at Runge-Kutta stage $m$ and is
defined as:

$$\bar{R}^{(m)}_{i,j,k} = \left( \alpha_m \frac{\Delta t}{\beta_{i,j,k}} \left[ \sum_{r=0}^{q} \{ \beta_{i,j,k} C(Q^{(r)}) + \gamma_{i,j,k} D(Q^{(r)}) \} \right] \right)$$

(5.4)

(see §3.4.2 for more details about the calculation of the residual). $\bar{R}^{(m)}_{i,j,k}$ is the smoothed
residual at stage $m$ after the sequence of smoothings in the $\zeta$, $\eta$ and $\zeta$ directions. The
variables $\beta_\zeta$, $\beta_\eta$, and $\beta_\zeta$ are the smoothing coefficients. A tridiagonal system of equations is
solved using Thomas’ algorithm for each coordinate direction to obtain the unknown residu­
als $\bar{R}^{(m)}_{i,j,k}$. 
In the original formulation Jameson used constant smoothing coefficients. These coefficients proved satisfactory for inviscid type meshes, and also for highly stretched meshes provided that enthalpy damping was used for additional support. In the present work variable coefficient smoothing coefficients are used that account for variations in mesh cell aspect ratio. Variable smoothing coefficients were originally introduced by Martinelli for two-dimensional calculations and these were later extended to three-dimensions by other researchers. The smoothing coefficients for the three coordinate directions are defined as:

\[ \beta_\xi = \max\left\{ \frac{1}{4}\left[ \left( \frac{CFL}{CFL^*} \cdot \frac{\bar{\lambda}_\xi}{\lambda_\xi + \lambda_\eta + \lambda_\zeta} \right)^2 - 1 \right], 0 \right\} \]

\[ \beta_\eta = \max\left\{ \frac{1}{4}\left[ \left( \frac{CFL}{CFL^*} \cdot \frac{\bar{\lambda}_\eta}{\lambda_\xi + \lambda_\eta + \lambda_\zeta} \right)^2 - 1 \right], 0 \right\} \] (5.5)

\[ \beta_\zeta = \max\left\{ \frac{1}{4}\left[ \left( \frac{CFL}{CFL^*} \cdot \frac{\bar{\lambda}_\zeta}{\lambda_\xi + \lambda_\eta + \lambda_\zeta} \right)^2 - 1 \right], 0 \right\} \]

and the scaled eigenvalues \( \bar{\lambda}_\xi, \bar{\lambda}_\eta, \) and \( \bar{\lambda}_\zeta \) have already been defined in §3.3.2. CFL and CFL* are the smoothed and unsmoothed Courant numbers of the explicit time-stepping scheme. In the calculations reported here CFL and CFL* are set equal to 7.5 and 4 respectively.

### 5.3.2 Improved Smoothing Coefficients for Two-Dimensional Calculations

Swanson has defined new variable smoothing coefficients for two-dimensional calculations. These coefficients result in a significant improvement in convergence rate on the Martinelli coefficients with standard inviscid meshes. The smoothing coefficients are defined as:
\[
\beta_\xi = \max \left\{ \frac{1}{4} \left[ \left( \frac{\text{CFL}}{\text{CFL}_*} \right)^2 - 1 \right], 0 \right\}
\]

\[
\beta_\eta = \max \left\{ \frac{1}{4} \left[ \left( \frac{\text{CFL}}{\text{CFL}_*} \right)^2 - 1 \right], 0 \right\}
\]

(5.6)

where \( r_{\eta\xi} = \frac{\lambda_\eta}{\lambda_\xi} \) and \( \psi \) is a user defined parameter, generally set as 0.25. These coefficients are used for all the two-dimensional calculations reported herein.

### 5.3.3 Boundary Treatment

To apply the implicit residual averaging at the first and last interior cells in each coordinate direction an assumption must be made regarding values of the residuals in the ghost cells. A Dirichlet condition can be used and the residuals in the ghost cells set to zero. Alternatively, using the Von Neumann condition, the ghost cell residuals can be set equal to the residuals in the first interior cells. Both conditions have been tested in this work. The Dirichlet condition was found to be the more robust and efficient and is the condition now employed in all the algorithms.

### 5.4 Multigrid

#### 5.4.1 Introduction

Significant improvements in the rate of convergence to a steady-state can be achieved using multigrid time-stepping. The theory behind the operation of multigrid methods is well-developed for elliptic equations and is based on the concept that the iterative scheme eliminates effectively the high-frequency error components on each grid. This theory does not hold for hyperbolic equations. Nevertheless, multigrid has been applied very successfully in the solution of the Euler and the Navier-Stokes equations. Jameson attributes the increase in convergence rates to the use of large time steps on the coarse-grid levels so that disturbances are expelled more rapidly out of the computational domain.

The five-stage explicit time-stepping scheme presented in §3.4.2 is very effective at damping high-frequency error components. When applied for the integration of the solutions of Equations (3.21) and (3.33) to a steady-state, very fast convergence is initially obtained as...
the high-frequency error components are damped. However, once these error components are eliminated the convergence slows down significantly because the time-stepping scheme is not as effective at damping the low-frequency error components. These low-frequency components can be removed by approximating the fine-grid problem on a sequence of successively coarser grids; this technique is called the multigrid method. With suitable coarse-grid approximations of the fine-grid problem, the low-frequency error components on the fine grid appear as high-frequency error components on the coarser grids. Thus an effective multigrid scheme can be constructed using a good high-frequency damping scheme as a driver.

5.4.2 Full Approximation Storage Scheme

A full approximation storage (FAS) multigrid scheme based on the work of Brandt\textsuperscript{97} and Jameson\textsuperscript{98} is employed here. Let $G_N$ represent the generated mesh and let $G_{N-1}, G_{N-2}$ be a sequence of coarser grids obtained by eliminating alternate points in each mesh direction of the next finer grid. The solution vector on a coarse grid $k$ is then initialised as:

$$Q_k^{(0)} = I_{k+1}^k Q_{k+1}$$  \hspace{1cm} (5.7)

where $Q_{k+1}$ is the current solution on $G_{k+1}$ and $I_{k+1}^k$ is a volume-weighted restriction operator of the form:

$$I_{k+1}^k Q_{k+1} = \frac{\sum_{m=1}^{8} (\Omega_{k+1} Q_{k+1})_m}{\sum_{m=1}^{8} (\Omega_{k+1})_m}$$  \hspace{1cm} (5.8)

where $\Omega$ represents the volume of a fine-grid cell. The summation in Equation (5.8) is over the eight fine-grid cells on $G_{k+1}$ that comprise the coarse-grid cell on $G_k$. In order to properly define the coarse-grid problem a forcing function is required so that the solution on $G_k$ is driven by the residuals calculated on $G_{k+1}$. The forcing function on $G_k$ is defined as:

$$P_k = T_{k+1}^k R_{k+1}(Q_{k+1}) - R_k(I_{k+1}^k Q_{k+1})$$  \hspace{1cm} (5.9)
where $T_{k+1}^k$ is the residual restriction operator and $\bar{R}_{k+1}(Q_{k+1})$ is the sum of the residual $R_{k+1}$ and the forcing function $P_{k+1}$ on $G_{k+1}$. The residual restriction operator is defined as:

$$T_{k+1}^k R_{k+1} = \frac{1}{\Omega_k} \sum_{m=1}^{8} (\Omega_{k+1}^m R_{k+1}^m)_{\Omega_k}$$

(5.10)

and the summation is again over the eight fine-grid cells. The restriction operators for the conserved variables and the residuals conserve mass, momentum and energy. The multi-stage scheme is then reformulated as:

$$Q_k^{(1)} = Q_k^{(0)} - \alpha_1 \Delta t_k [R_k^{(0)} + P_k]$$

$$.........$$

$$Q_k^{(m)} = Q_k^{(m-1)} - \alpha_m \Delta t_k [R_k^{(m-1)} + P_k]$$

(5.11)

The solution on $G_k$ can be used to provide initial data for $G_{k+1}$. Finally, the accumulated correction on $G_k$ is prolonged back to $G_{k+1}$ using a trilinear interpolation operator $L_k^{k+1}$. The solution on $G_{k+1}$ is then updated to:

$$Q_{k+1} \leftarrow Q_{k+1} + L_k^{k+1}(Q_k - L_k^k Q_{k+1})$$

(5.12)

### 5.4.3 Cycling Strategies

The work split between the various grid levels can be achieved using either fixed or adaptive cycling strategies. With an adaptive cycling strategy a variable number of time steps are performed on each grid level until the convergence rate has decreased to a prescribed level. In a fixed cycle strategy a fixed number of time steps are performed on each level irrespective of the convergence rate.

A fixed cycle strategy is employed in this work. Two alternative fixed cycles are implemented in each algorithm: a $V$-cycle and a $W$-cycle. The structure of a three-, four- and five-level $V$-cycle with one Runge-Kutta time step on each grid level is shown in Figure 5.1. Figure 5.2 shows the corresponding $W$-cycles. The structure of the $W$-cycle becomes quite complex with an increasing number of grid levels but it has a recursive definition and
Figure 5.1 Three-, four- and five-level multigrid V-cycles. E, Euler calculation; R, restriction; P, prolongation.
Figure 5.2 Three-, four- and five-level multigrid W-cycles. E, Euler calculation; R, restriction; P, prolongation.
as such is essentially as easy to program as the V-cycle.\textsuperscript{43} The upper work bound for both of these cycles occurs for two-dimensional calculations and is as follows:

\[
\text{Work}_{\text{Multigrid}} < \frac{4}{3} \text{Work}_{\text{Fine}} \text{ for V-cycle}
\]

\[
\text{Work}_{\text{Multigrid}} < 2 \text{Work}_{\text{Fine}} \text{ for W-cycle}
\]

where \textit{Fine} refers to one Runge-Kutta time step on the fine-grid level, and the cost of the inter-grid transfer operations have been neglected. In comparison to a fine-grid time step, the cost of the inter-grid transfers are negligible. The relative cost of both cycles is even less in three-dimensional calculations.

To achieve a desired level of solution accuracy, a computation using the W-cycle generally requires fewer time steps than the same computation using the corresponding V-cycle. However, the overall CPU time for a full computation using either cycle is about the same due to the higher computational cost of a W-cycle. The W-cycle is the preferred cycle in the present work as it is generally accepted that it provides improved robustness.\textsuperscript{43} Multiple coarse-grid time steps can also be employed on the coarse grids primarily to increase robustness but also to increase the convergence rate.

The same boundary conditions are applied on all grid levels. On the coarse-grid levels the fine-grid artificial dissipation model is replaced by a simpler constant-coefficient second-difference dissipation model. This reduces the computational effort of the coarse-grid calculations and also increases the convergence rate on the coarse grids.

The robustness of the multigrid scheme is significantly enhanced when using highly-stretched or non-uniform grids by the smoothing of the coarse-grid corrections before addition to the fine-grid solution. Smoothing reduces high-frequency oscillations introduced by trilinear interpolation of the coarse-mesh corrections. The factored scheme of Equation (5.2) for the implicit residual averaging with constant coefficients (\(\beta_\xi = 0.1\text{–}0.2, \beta_\eta = 0.1\text{–}0.2, \beta_\zeta = 0.1\text{–}0.2\)) is the most efficient way of achieving this.\textsuperscript{62}

\subsection*{5.4.4 Full Multigrid Strategy}

A full multigrid strategy (FMG) is employed in the all the steady numerical algorithms to provide a well-conditioned starting solution for the finest mesh being considered.\textsuperscript{43,63}
FMG is not required in the unsteady algorithms as each unsteady solution is initialised using the converged solution of a preliminary steady flow calculation.

With the FMG strategy the solution is initialised on the coarsest grid of a basic sequence of grids and iterated for a prescribed number of multigrid cycles using the FAS multigrid scheme. The solution is then interpolated to the next finer grid level using trilinear interpolation. This process is repeated until the finest grid level is reached. Freestream conditions are used as the starting solution on the coarsest grid level (see §4.4).

5.5 Summary

The convergence acceleration devices employed to accelerate the rate of convergence to a steady-state have been presented in this chapter. This ends the description of the components of the numerical algorithms. In the next chapter results of test cases computed using each of these algorithms are presented and analysed.
CHAPTER 6

PRESENTATION OF RESULTS

6.1 Introduction

A large number of test cases were computed during the development of each of the six numerical algorithms in order to verify results, validate the methods, and to measure accuracy and efficiency. The test cases were for various combinations of model geometry and operating conditions. The results of a small subset of test cases are presented in this chapter. These cases are chosen to demonstrate the ability of each algorithm to predict complex flowfields. Where possible predictions are compared with measurements for validation purposes, and also with the computed data of other researchers. A novel comparison is made between the computed flowfield downstream of an advanced propeller under axisymmetric inflow conditions and state-of-the-art PIV measurements from wind tunnel tests. Results from test cases using the new unsteady propeller algorithm are also presented.

The results are presented in the order in which the algorithms were developed. Results from the two-dimensional fixed airfoil calculations are presented first.

6.2 Two-Dimensional Fixed Airfoil Calculations

6.2.1 Introduction

The steady airfoil algorithm designated FLO2DS was used for the fixed airfoil flowfield calculations. Results of three test cases are presented here. The first two cases were for the subsonic and transonic flow around a NACA 0012 airfoil at zero and non-zero angle of incidence respectively. The third test case was for the transonic flow around a NACA 64A010 airfoil at zero angle of incidence. The freestream conditions for each test case are listed in Table 6.1.
The NACA 0012 airfoil is a standard 4-digit series airfoil with a 12% thickness to chord ratio and zero camber. The NACA 64A010 airfoil is a 6A-series airfoil with a 10% thickness to chord ratio, also with zero camber. Ordinates for both airfoils were calculated using the program AIRFOLS.

The three test cases were computed using C-type meshes with 224 cells in the streamwise direction and 32 cells in the surface normal direction. This grid density has been found by other researchers to be adequate for accurate inviscid airfoil flowfield calculations. A hyperbolic grid generator was employed to generate the grids around both airfoils. Grid clustering was used at the airfoil leading and trailing edges in the streamwise direction and at the body surface in the normal direction. The outer boundary was placed at 20 chord lengths from the airfoil surface. The inner parts of both grids are shown in Figures 6.1 and 6.2. Corresponding leading edge grid structures are shown in Figures 6.3 and 6.4. As can be seen from these figures the variation in cell size is smooth and the cells are almost orthogonal in the entire domain.

The same dissipation constants were employed for the three test cases. The second and fourth difference dissipation constants were set equal to 1/2 and to 1/32 respectively. The scaling factor $\alpha$ was set equal to 0.667. The multigrid strategy consisted of a five-grid W-cycle with multiple Runge-Kutta time steps on the coarse grids; two on the first coarse-grid level and three on all coarser grid levels. Three FMG refinements levels were used with two and three multigrid levels on the first and second levels. 25 multigrid cycles were performed on the first and second FMG levels.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Airfoil Type</th>
<th>$M_{\infty}$</th>
<th>$\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NACA 0012</td>
<td>0.63</td>
<td>2.0°</td>
</tr>
<tr>
<td>2</td>
<td>NACA 0012</td>
<td>0.8</td>
<td>1.25°</td>
</tr>
<tr>
<td>3</td>
<td>NACA 64A010</td>
<td>0.796</td>
<td>0.0°</td>
</tr>
</tbody>
</table>

*Table 6.1* Airfoil geometry and freestream conditions for the fixed airfoil test cases.

### 6.2.2 Results of Test Case 1: NACA 0012 Airfoil

Test Case 1 was for the subsonic flow over a NACA 0012 airfoil for a freestream Mach number of 0.63 and for an angle of incidence of 2.0°. The results of this test case are pre-
sented in Figures 6.5 to 6.10. The convergence history is shown in Figures 6.5. In this work convergence is measured by the logarithm of the root-mean-square of the residual of the continuity equation. The convergence to steady-state was very rapid, with convergence to engineering accuracy (i.e., four orders reduction in the residual) achieved in 44 multigrid cycles, and to machine zero in double precision (i.e., 13 orders reduction in the residual) in approximately 140 multigrid cycles. The development of the lift coefficient is shown in Figure 6.6 and is very rapid. The lift coefficient converged to five decimal places in only 40 multigrid cycles.

Figure 6.7 shows the surface pressure distribution around the airfoil. The total pressure change at the airfoil surface is shown in Figure 6.8. The total pressure change is defined as $1 - P/P_\infty$ where P is the local total pressure and $P_\infty$ is the freestream total pressure. The total pressure is theoretically constant in an inviscid flowfield in the absence of shock waves, as in this test case, and hence the total pressure change is zero. Therefore the computed total pressure change is a measure of the accuracy of the numerical algorithm. As can be seen in the Figure 6.8 the total pressure change is less that 0.05% over most of the airfoil, except at the leading and trailing edges where the differences are around 1.0%. This level of accuracy is very good.\textsuperscript{79} Figures 6.9 and 6.10 show contours of Mach number and pressure in the region around the airfoil. The stagnation and minimum pressure points can be easily identified near the airfoil leading edge.

The computed lift and drag coefficients are presented in Table 6.2 along with those computed by Kroll using a similar finite-volume algorithm.\textsuperscript{79} Good agreement is obtained between the computed performance coefficients. The higher value of drag coefficient predicted by FLO2DS can be attributed to the higher level of dissipation employed in the present calculation. Ideally, in this inviscid shock free flow, the drag coefficient should be zero.

<table>
<thead>
<tr>
<th>Method</th>
<th>Grid Type</th>
<th>Mesh Size</th>
<th>$C_L$</th>
<th>$C_D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLO2DS</td>
<td>C</td>
<td>224 x 32</td>
<td>0.3347</td>
<td>0.00024</td>
</tr>
<tr>
<td>Kroll\textsuperscript{79}</td>
<td>O</td>
<td>160 x 32</td>
<td>0.3339</td>
<td>0.00001</td>
</tr>
</tbody>
</table>

\textit{Table 6.2} Computed lift and drag coefficients for the NACA 00121 airfoil. $Ma_\infty = 0.63$ and $\alpha = 2.0^\circ$. 

69
6.2.3 Results of Test Case 2: NACA 0012 Airfoil

This test case was for the transonic flow over a NACA 0012 airfoil. The freestream Mach number was 0.8 and the angle of incidence was 1.25°. This test case was recommended as a benchmark test case for inviscid flowfield methods by AGARD Working Group 07 due to the complexity of the flowfield. A strong shock wave occurs on the airfoil upper surface and a weak shock wave on the lower surface. The results of this test case are presented in Figures 6.11 to 6.16. The convergence history and the development of the lift are shown in Figure 6.11 and 6.12. Very rapid convergence was again obtained with convergence to engineering accuracy achieved in 33 multigrid cycles and to machine zero in approximately 140 multigrid cycles. The lift coefficient converged to five decimal places in 28 multigrid cycles. A strong shock wave is predicted on the upper surface at x/c = 0.6 as shown in Figures 6.13 and 6.14. The pressure increases across the shock wave and the total pressure decreases. The shock wave has a discrete thickness of three points as anticipated. However, the lower surface shock wave, which occurs around x/c = 0.32, is smeared out due to the high level of dissipation employed in the present calculations and to the coarseness of the grid in this region. Contours of Mach number and pressure are shown in Figures 6.15 and 6.16. The deceleration of the flow from supersonic to subsonic by the upper surface shock wave can be seen in Figure 6.15.

Table 6.3 presents the computed lift and drag coefficients along with those of other researchers. The overall agreement between computations is good. It should be noted that the mesh used in the present calculation was not as fine as those used by the other researchers; nevertheless the agreement is quite good.

<table>
<thead>
<tr>
<th>Method</th>
<th>Grid Type</th>
<th>Mesh Size</th>
<th>C_L</th>
<th>C_D</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLO2DS</td>
<td>C</td>
<td>224 x 32</td>
<td>0.3656</td>
<td>0.0232</td>
</tr>
<tr>
<td>Kroll⁷⁹</td>
<td>O</td>
<td>360 x 64</td>
<td>0.3594</td>
<td>0.0228</td>
</tr>
<tr>
<td>Jameson⁷⁶</td>
<td>O</td>
<td>320 x 64</td>
<td>0.3632</td>
<td>0.0230</td>
</tr>
<tr>
<td>Pulliam⁷⁷</td>
<td>C</td>
<td>560 x 64</td>
<td>0.3618</td>
<td>0.0236</td>
</tr>
</tbody>
</table>

Table 6.3 Computed lift and drag coefficients for the NACA0012 airfoil. \(\text{Ma}_\infty = 0.8\) and \(\alpha = 1.25°\).
6.2.4 Results of Test Case 3: NACA 64A010 Airfoil

Test Case 3 was for the transonic flow over a NACA 64A010 airfoil for a freestream Mach number of 0.796 and zero angle of incidence. The NACA 64A010 is a symmetric airfoil and so theoretically the computed lift should be zero and the computed flowfield symmetric above and below the airfoil. The results of this test case are presented in Figures 6.17 to 6.22. The convergence history and the development of the lift coefficient are shown in Figures 6.17 and 6.18. Convergence to engineering accuracy was again achieved in 33 multigrid cycles. Convergence to machine zero took 200 multigrid cycles which was more than in the previous two test cases. The computed lift coefficient was constant and equal to zero to at least six decimal places for the duration of the calculation. Figures 6.19 and 6.20 show the surface pressure and total pressure distributions. A shock wave is predicted at approximately \( \frac{x}{c} = 0.5 \) on the upper and lower surfaces. Contours on Mach number and pressure are shown in Figures 6.21 and 6.22. The computed flowfield is perfectly symmetrical as anticipated. The computed lift and drag coefficients are presented in Table 6.4.

<table>
<thead>
<tr>
<th>Method</th>
<th>Grid Type</th>
<th>Mesh Size</th>
<th>( C_L )</th>
<th>( C_D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLO2DS</td>
<td>C</td>
<td>224 x 32</td>
<td>0.0000</td>
<td>0.0009</td>
</tr>
</tbody>
</table>

*Table 6.4* Computed lift and drag coefficients for the NACA 64A010 airfoil. \( \text{Ma}_{\infty} = 0.796 \) and \( \alpha = 0.0^\circ \).
Figure 6.1  Inner part of the 224x32 cell hyperbolic C-grid around the NACA 0012 airfoil.

Figure 6.2  Inner part of the 224x32 cell hyperbolic C-grid around the NACA 64A010 airfoil.
Figure 6.3  Grid structure in the nose region of the NACA 0012 airfoil.

Figure 6.4  Grid structure in the nose region of the NACA 64A010 airfoil.
Figure 6.5 Convergence history of the steady calculation of the flow around a fixed NACA 0012 airfoil. \( \text{Ma}_\infty = 0.63 \) and \( \alpha = 2.0^\circ \).

Figure 6.6 Development of the lift coefficient during the steady calculation of the flow around a fixed NACA 0012 airfoil. \( \text{Ma}_\infty = 0.63 \) and \( \alpha = 2.0^\circ \).
Figure 6.7 Variation of computed surface pressure with chordwise position for the NACA 0012 airfoil. \( Ma_\infty = 0.63 \) and \( \alpha = 2.0^\circ \).

Figure 6.8 Variation of computed total pressure loss with chordwise position for the NACA 0012 airfoil. \( Ma_\infty = 0.63 \) and \( \alpha = 2.0^\circ \).
**Figure 6.9** Mach number contours around the NACA 0012 airfoil. $M_{x\infty} = 0.63$, $\alpha = 2.0^\circ$ and $\Delta Ma = 0.03$.

**Figure 6.10** Pressure contours around the NACA 0012 airfoil. $M_{x\infty} = 0.63$, $\alpha = 2.0^\circ$ and $\Delta p = 0.03$. 

76
Figure 6.11 Convergence history of the steady calculation of the flow around a fixed NACA 0012 airfoil. $Ma_\infty = 0.8$ and $\alpha = 1.25^\circ$.

Figure 6.12 Development of the lift coefficient during the steady calculation of the flow around a fixed NACA 0012 airfoil. $Ma_\infty = 0.8$ and $\alpha = 1.25^\circ$. 
Figure 6.13 Variation of computed surface pressure with chordwise position for the NACA 0012 airfoil. $M_{\infty} = 0.8$ and $\alpha = 1.25^\circ$.

Figure 6.14 Variation of computed total pressure loss with chordwise position for the NACA 0012 airfoil. $M_{\infty} = 0.8$ and $\alpha = 1.25^\circ$. 
**Figure 6.15** Mach number contours around the NACA 0012 airfoil. $Ma_{∞} = 0.8$, $\alpha = 1.25^\circ$ and $\Delta Ma = 0.05$.

**Figure 6.16** Pressure contours around the NACA 0012 airfoil. $Ma_{∞} = 0.8$, $\alpha = 1.25^\circ$ and $\Delta p = 0.05$. 

Figure 6.17 Convergence history of the steady calculation of the flow around a fixed NACA 64A010 airfoil. $M_a = 0.796$ and $\alpha = 0.0^\circ$.

Figure 6.18 Development of the lift coefficient during the steady calculation of the flow around a fixed NACA 64A010 airfoil. $M_a = 0.796$ and $\alpha = 0.0^\circ$. 
Figure 6.19  Variation of computed surface pressure with chordwise position for the NACA 64A010 airfoil. $M_{\infty} = 0.796$ and $\alpha = 0.0^\circ$.

Figure 6.20  Variation of total pressure loss with chordwise position for the NACA 0012 airfoil. $M_{\infty} = 0.796$ and $\alpha = 0.0^\circ$. 
Figure 6.21 Mach number contours around the NACA 64A010 airfoil. $M_{\infty} = 0.796$, $\alpha = 0.0^\circ$ and $\Delta M_a = 0.03$.

Figure 6.22 Pressure contours around the NACA 64A010 airfoil. $M_{\infty} = 0.796$, $\alpha = 0.0^\circ$ and $\Delta p = 0.03$. 
6.3 Two-Dimensional Oscillating Airfoil Calculations

6.3.1 Introduction

The two-dimensional unsteady algorithm, FLO2DU, was used to calculate the unsteady transonic flow around a NACA 64A010 airfoil undergoing forced pitching oscillation about its quarter chord point. This type of flowfield occurs when considering a two-dimensional section of an aircraft wing in torsional flutter. The operating conditions for the test case are listed in Table 6.5. These conditions correspond to AGARD Test Case CT-6 for which unsteady flow measurements were made by Davis at NASA Ames.\textsuperscript{108} Because comparisons can be made with wind tunnel data, this is a standard test case for unsteady airfoil algorithms.\textsuperscript{64,73,77,109-112} The pitching motion of the airfoil was governed by the relation:

$$\alpha(t) = \alpha_m + \alpha_0 \sin(\omega t) \quad (6.1)$$

where $\alpha(t)$ is the instantaneous airfoil angle of incidence, $\alpha_m$ is the mean angle of incidence, $\alpha_0$ is the pitching amplitude and $\omega$ is the radian frequency. The radian frequency $\omega$ is related to the reduced frequency $k$ by the expression:

$$k = \frac{\omega c}{2V_\infty} \quad (6.2)$$

where $c$ is the airfoil chord and $V_\infty$ is the freestream speed.

A 224x32 C-type mesh, generated using a hyperbolic mesh generator, was employed for the unsteady calculations. Figure 6.23 shows the inner part of the grid. The location of the far-field boundaries was the same as for the grid used in the steady calculation described in §6.2.4. In comparison with that grid, the grid clustering was tighter near the mid-chord on the upper and lower surfaces of the airfoil, where a shock wave was known to occur, and was relaxed near the trailing edge so that the grid spacing did not become excessive anywhere on the airfoil. The dissipation constants and multigrid strategy employed were the same as in the steady calculations. Note however, that the dissipation for the energy equation in the unsteady algorithm is based on differences of total energy, not enthalpy (see §3.3.3). Each unsteady solution was initialised using the converged solution of a preceding
steady calculation at the mean angle of incidence. The airfoil was then set impulsively into motion and three cycles of oscillation calculated to establish a periodic solution.

<table>
<thead>
<tr>
<th>$\text{Ma}_\infty$</th>
<th>$\alpha_m$</th>
<th>$\alpha_o$</th>
<th>$k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.796</td>
<td>0.0°</td>
<td>1.01°</td>
<td>0.202</td>
</tr>
</tbody>
</table>

*Table 6.5* Freestream Mach number, mean angle of incidence, amplitude of oscillation and reduced frequency for AGARD Test Case CT-6.

### 6.3.2 Results of Test Case: NACA 64A010 Airfoil

The test case was computed using 18, 36 and 72 time steps per period to assess the effect of the number of time steps on the accuracy of the flow solution. The results of the calculations are presented in Figures 6.24 to 6.37. A reduction of three orders of magnitude in the residual of the continuity equation was specified as the convergence criterion for the 18 and 36 time step calculations and a two orders of magnitude reduction for the 72 time step calculation. These convergence criteria resulted in very well-converged solutions, with the lift coefficient at each time step in each calculation generally converged to five decimal places. To achieve these convergence criteria on average 20, 17 and 11 multigrid cycles were required at each time step in the 18, 36 and 72 time step calculations respectively. It can be seen, therefore, that the 72 time step only took approximately 2.2 the CPU time of the 18 time step calculation, instead of 4 times as one might expect. This is a very important characteristic of the multigrid implicit method, i.e., there is a trade-off between the time step size and the number of multigrid cycles needed for convergence of the implicit equations.

Figure 6.24 shows the variation of lift coefficient with non-dimensional time for the three calculations. A periodic solution was established during the third period of oscillation for the three time step sizes. Figures 6.25 and 6.26 show the variation of lift coefficient with angle of incidence during the third period of oscillation. Note that the computed values lie on a slanting oval because of the phase lag between lift and angle of incidence. The differences between the lift coefficients calculated using the three time step sizes is small, with the 36 time step calculation providing almost identical results to the 72 time step calculation. The remaining results that are presented here are for the 36 time step calculation only.
The computed variation of lift coefficient with angle of incidence is compared with measurements\textsuperscript{108} and with the computed values of Jameson\textsuperscript{64} and Venkatakrishnan\textsuperscript{87,109} in Figures 6.27 to 6.29. As can be seen in Figure 6.27 the measurements show a slightly smaller total variation of lift and lie on a broader oval. The agreement between computed and measured values can be considered good. The differences may be attributed to experimental error and to omission of viscous effects in the calculations. The three sets of computed values show excellent agreement. Jameson also used the multigrid implicit method for his calculation while Venkatakrishan used an explicit multistage scheme.

An extract from the convergence history during the third cycle is shown in Figure 6.30. As mentioned earlier a reduction of three orders of magnitude was specified at the convergence criterion. The convergence of the solution at each time step to a steady-state was very rapid with convergence generally achieved in 17 multigrid cycles.

Figures 6.31 to 6.36 show comparisons of computed upper surface pressure with measurements at six angular positions during the third cycle. The overall agreement is good. However, discrepancies exist between computed and measured values in the forward portion of the airfoil. These discrepancies also exist in the inviscid calculations of Venkatakrishan\textsuperscript{87,109} and Chyu,\textsuperscript{110} and the inviscid and viscous calculations of Liu\textsuperscript{111} and can be attributed to the fact that the profile of the NASA Ames model differed slightly from a NACA 64A010 airfoil.\textsuperscript{108,109}

Mach number contours in the field around the airfoil are presented in Figure 6.37 at six instances in time during the third cycle of oscillation. The development, movement and decay of the upper and lower shock waves can be clearly identified near the mid-chord region of the airfoil.
Figure 6.23  Inner part of the 224x32 cell hyperbolic C-grid around the NACA 64A010 airfoil.

Figure 6.24  Computed variation of lift coefficient with non-dimensional time for the NACA 64A010 airfoil. \( \text{Ma}_\infty = 0.796, \alpha_m = 0.0^\circ, \alpha_0 = 1.01^\circ \) and \( k = 0.202 \).
**Figure 6.25** Computed variation of lift coefficient with angle of incidence for the NACA 64A010 airfoil. \( M_a = 0.796, \alpha_m = 0.0^\circ, \alpha_0 = 1.01^\circ \) and \( k = 0.202 \).

**Figure 6.26** Computed variation of lift coefficient with angle of incidence for the NACA 64A010 airfoil near the maximum angle of incidence. \( M_a = 0.796, \alpha_m = 0.0^\circ, \alpha_0 = 1.01^\circ \) and \( k = 0.202 \).
Figure 6.27  Comparison of computed variation of lift coefficient with angle of incidence for the NACA 64A010 airfoil with the measurements of Reference 108. $M_{\infty} = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_o = 1.01^\circ$ and $k = 0.202$.

Figure 6.28  Comparison of computed variation of lift coefficient with angle of incidence for the NACA 64A010 airfoil with Reference 64. $M_{\infty} = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_o = 1.01^\circ$ and $k = 0.202$. 
Figure 6.29  Comparison of computed variation of lift coefficient with angle of incidence for the NACA 64A010 airfoil with References 87 and 109. $M_{\infty} = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 1.01^\circ$ and $k = 0.202$.

Figure 6.30  An extract of the convergence history during the unsteady calculation of the flow around a NACA 64A010 airfoil. $M_{\infty} = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 1.01^\circ$ and $k = 0.202$. 
Figure 6.31 Comparison of computed and measured\textsuperscript{108} surface pressure on the upper surface of the NACA 64A010 airfoil. $\text{Ma}_{\infty} = 0.796$, $\alpha_{m} = 0.0^\circ$, $\alpha_{0} = 1.01^\circ$ and $k = 0.202$.

Figure 6.32 Comparison of computed and measured\textsuperscript{108} surface pressure on the upper surface of the NACA 64A010 airfoil. $\text{Ma}_{\infty} = 0.796$, $\alpha_{m} = 0.0^\circ$, $\alpha_{0} = 1.01^\circ$ and $k = 0.202$. 
**Figure 6.33** Comparison of computed and measured surface pressure on the upper surface of the NACA 64A010 airfoil. $M_{\infty} = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 1.01^\circ$ and $k = 0.202$.

**Figure 6.34** Comparison of computed and measured surface pressure on the upper surface of the NACA 64A010 airfoil. $M_{\infty} = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_0 = 1.01^\circ$ and $k = 0.202$. 
Figure 6.35 Comparison of computed and measured surface pressure on the upper surface of the NACA 64A010 airfoil. $Ma_\infty = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_o = 1.01^\circ$ and $k = 0.202$.

Figure 6.36 Comparison of computed and measured surface pressure on the upper surface of the NACA 64A010 airfoil. $Ma_\infty = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_o = 1.01^\circ$ and $k = 0.202$. 
Figure 6.37  Mach number contours around the NACA 64A010 airfoil during the third period of oscillation. $Ma_{\infty} = 0.796$, $\alpha_m = 0.0^\circ$, $\alpha_o = 1.01^\circ$, $k = 0.202$ and $\Delta Ma = 0.03$. 
6.4 Three-Dimensional Fixed Wing Calculations

6.4.1 Introduction

The three-dimensional steady wing algorithm, FLO3DS, was used to compute two fixed wing test cases. The test cases were for the subsonic and transonic flow over the ONERA M6 swept wing at incidence. These test cases are standard ones for the evaluation of inviscid and viscous three-dimensional wing algorithms\textsuperscript{47,62,63,114-120} as comparisons of surface pressure can be made with wind tunnel measurements.\textsuperscript{113} The flow conditions for the two test cases are listed in Table 6.6. Test Cases 1 and 2 correspond to the experimental test cases 2312 and 2308 respectively.

The ONERA M6 wing geometry is comprised of symmetrical airfoil sections with a leading edge sweep of 30°, a trailing edge sweep of 15.8°, an aspect ratio of 3.8 and a taper ratio of 0.562. A C-H mesh topology was used in all computations, with the C-type mesh in the streamwise vertical direction. The C-H meshes were generated by simply stacking a series of two-dimensional cross sections in the spanwise direction; the sections were stacked at equal intervals along the wing span. The two-dimensional grids were generated using a hyperbolic grid generator and had increased grid clustering at the leading edge. Meshes of this type contain badly distorted cells in the region of the singular line where it passes into the flowfield beyond the wing tip. These cells, which have a very high aspect ratio and a triangular cross section, present a severe test of the robustness of the multigrid scheme. The farfield boundaries were located at a distance of 10 root chords from the wing surface for each airfoil section, and at a distance of one wing span beyond the wing tip (see §4.2).

Three different meshes were used to compute each test case. A coarse 24,576 (96×16×16) cell grid, a medium 82,944 (144×24×24) cell grid and a fine 196,608 (192×32×32) cell grid. These different densities were employed to show the effect of grid density on the computed flowfield. Figures 6.38 and 6.39 show the inner part of the coarse and fine grids respectively.

The same dissipation constants as used in the two-dimensional calculations were employed in these calculations also, i.e., the second and fourth difference dissipation constants were set equal to 1/2 and to 1/32 respectively. The scaling factor $\alpha$ was set equal to 0.5.
The multigrid strategy consisted of a W-multigrid cycle, with the maximum feasible number of grid levels employed in the multigrid cycle: four for the coarse and medium grids and five for the fine grid. Multiple sub-iterations were also used on the coarse grid levels: two on the first coarse grid level and three on coarser grid levels. Three FMG refinement levels were used in all calculations with 50 multigrid cycles performed on the first and second levels.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Wing</th>
<th>Ma_∞</th>
<th>α</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ONERA M6</td>
<td>0.699</td>
<td>3.06°</td>
</tr>
<tr>
<td>2</td>
<td>ONERA M6</td>
<td>0.84</td>
<td>3.06°</td>
</tr>
</tbody>
</table>

*Table 6.6* Freestream conditions for the fixed wing test cases.

### 6.4.2 Results of Test Case 1 and Test Case 2: ONERA M6 Wing

Test Case 1 was for a subsonic freestream Mach number of 0.699 while Test Case 2 was for a transonic Mach number of 0.84. The angle of incidence in both test cases was 3.06°. The results of the calculations for both test cases are presented in Figures 6.40 to 6.53.

Convergence histories are shown in Figures 6.40 to 6.45. For the subsonic test case convergence to engineering accuracy was obtained in 42, 36 and 39 multigrid cycles for the coarse, medium and fine grids respectively. Convergence to machine zero was obtained in 181, 164 and 168 cycles for the same grids. For the transonic case convergence to engineering accuracy was obtained in 33, 34 and 41 multigrid cycles and to machine zero in 154, 150 and 174 cycles for the coarse, medium and fine grids respectively. It is important to note that the number of multigrid cycles required to converge a solution to either engineering accuracy or machine zero in these calculations was relatively insensitive to the number of grid cells, thus indicating that the multigrid scheme is working successfully. The convergence to steady-state was very rapid in all calculations. These convergence rates compare very well with those of other researchers using similar type schemes.\(^{47,120}\)

Comparisons of computed and measured surface pressure distributions at six spanwise locations for Test Case 1 are shown in Figures 6.46 to 6.48 for the coarse, medium and fine grid calculations. Note that the spanwise locations (ζ) have been non-dimensionalised by the wing span. The computed pressures were obtained at the experimental spanwise locations by cubic interpolation.\(^{121}\) The comparison of computed and measured values is good
for the coarse and medium grid calculations and is excellent for the fine grid calculation. Only the fine grid properly resolved the sharp leading edge suction peak on the upper surface along the full wing span. The correct rounded wing tip geometry was not modelled by the C-H mesh topology and this leads to discrepancies between computed and measured value at the outermost station \( \zeta = 0.95 \). A C-O mesh that wraps around the wing tip would lead to more accurate resolution of the flow in this region.\(^99\)

Similar comparisons of computed and measured surface pressure distributions are shown in Figures 6.49 to 6.51 for Test Case 2. The measured pressure distributions show the existence of a so-called \( \lambda \) shock wave on the upper surface of the wing under these conditions: an oblique shock wave emanates from near the root leading edge and a near normal shock wave emanates from near the root mid-chord. Both shocks merge in the region between 80 and 90\% of the wing span to form a single and much stronger shock wave that continues to the wing tip. This is a very complex flowfield to predict. The coarse grid resolution was not sufficient to accurately compute the upper surface shock system and the shock waves are totally smeared out in this calculation. The accuracy of the medium grid calculation is much better, but it is only using the fine grid that the shock waves are resolved to a reasonable level of accuracy. Further grid refinement in the streamwise and spanwise direction is required in order to accurately capture the shock waves especially in the mid-span region. The differences between the computed and measured values using the fine grid are consistent with those of other researchers using similar size and finer grids.\(^{50,99,115,117}\)

Overall the agreement between the computed and measured surface pressure is good for the fine grid calculation, thus demonstrating the ability of this algorithm to accurately and efficiently predict complex flowfields.

Figure 6.52 and 6.53 show computed upper surface Mach number contours from the fine grid calculation for both test cases. For the subsonic test case the flow accelerates to supersonic speeds near the leading edge. The computed shock waves can be clearly identified in Figure 6.53 for the transonic test case.

The computed lift and drag coefficients are presented in Table 6.7 for both test cases, along with the number of multigrid cycles required for convergence both to engineering accuracy and to machine zero.
<table>
<thead>
<tr>
<th>Test Case</th>
<th>Grid</th>
<th>Number of Multigrid Cycles to Converge to</th>
<th>$C_L$</th>
<th>$C_D$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Engineering Accuracy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>96x16x16</td>
<td>42</td>
<td>0.25900</td>
<td>0.01098</td>
</tr>
<tr>
<td></td>
<td>144x24x24</td>
<td>36</td>
<td>0.25645</td>
<td>0.00980</td>
</tr>
<tr>
<td></td>
<td>192x32x32</td>
<td>39</td>
<td>0.25571</td>
<td>0.00794</td>
</tr>
<tr>
<td>2</td>
<td>96x16x16</td>
<td>33</td>
<td>0.29563</td>
<td>0.01629</td>
</tr>
<tr>
<td></td>
<td>144x24x24</td>
<td>34</td>
<td>0.30230</td>
<td>0.01499</td>
</tr>
<tr>
<td></td>
<td>192x32x32</td>
<td>41</td>
<td>0.30203</td>
<td>0.01340</td>
</tr>
</tbody>
</table>

Table 6.7 Summary of results for Test Case 1 and Test Case 2.
**Figure 6.38** Inner part of the coarse (96x16x16) C-H grid around the ONERA M6 wing.

**Figure 6.39** Inner part of the fine (192x32x32) C-H grid around the ONERA M6 wing.
Figure 6.40 Convergence history of the steady calculation of the flow around the ONERA M6 wing using the coarse grid. $Ma_{\infty} = 0.699$ and $\alpha = 3.06^\circ$.

Figure 6.41 Convergence history of the steady calculation of the flow around the ONERA M6 wing using the coarse grid. $Ma_{\infty} = 0.84$ and $\alpha = 3.06^\circ$. 
**Figure 6.42** Convergence history of the steady calculation of the flow around the ONERA M6 wing using the medium grid. $Ma_{\infty} = 0.699$ and $\alpha = 3.06^\circ$.

**Figure 6.43** Convergence history of the steady calculation of the flow around the ONERA M6 wing using the medium grid. $Ma_{\infty} = 0.84$ and $\alpha = 3.06^\circ$. 
Figure 6.44  Convergence history of the steady calculation of the flow around the ONERA M6 wing using the fine grid. $\text{Ma}_\infty = 0.699$ and $\alpha = 3.06^\circ$.

Figure 6.45  Convergence history of the steady calculation of the flow around the ONERA M6 wing using the fine grid. $\text{Ma}_\infty = 0.84$ and $\alpha = 3.06^\circ$. 
Figure 6.46  Comparison of computed and measured\textsuperscript{113} surface pressure for the ONERA M6 wing using the coarse grid. \(\text{Ma}_{\infty} = 0.699\) and \(\alpha = 3.06^\circ\).
Figure 6.47  Comparison of computed and measured\textsuperscript{113} surface pressure for the ONERA M6 wing using the medium grid. $Ma_\infty = 0.699$ and $\alpha = 3.06^\circ$. 

103
Figure 6.48  Comparison of computed and measured\textsuperscript{113} surface pressure for the ONERA M6 wing using the fine grid. $Ma_{\infty} = 0.699$ and $\alpha = 3.06^\circ$. 
Figure 6.49  Comparison of computed and measured\textsuperscript{113} surface pressure for the ONERA M6 wing using the coarse grid. $Ma_{\infty} = 0.84$ and $\alpha = 3.06^\circ$. 

105
Figure 6.50 Comparison of computed and measured\textsuperscript{113} surface pressure for the ONERA M6 wing using the medium grid. $M_{\infty} = 0.84$ and $\alpha = 3.06^\circ$. 
Figure 6.51  Comparison of computed and measured surface pressure for the ONERA M6 wing using the fine grid. $M_a = 0.84$ and $\alpha = 3.06^\circ$. 
Figure 6.52 Mach number contours on the upper surface of the ONERA M6 wing calculated using the fine grid. $Ma_\infty = 0.699$, $\alpha = 3.06^\circ$ and $\Delta Ma = 0.05$.

Figure 6.53 Mach number contours on the upper surface of the ONERA M6 wing calculated using the fine grid. $Ma_\infty = 0.84$, $\alpha = 3.06^\circ$ and $\Delta Ma = 0.05$. 
6.5 Three-Dimensional Oscillating Wing Calculations

6.5.1 Introduction

Two test cases were computed using the three-dimensional transient wing algorithm FLO3DU. Both test cases were for the time-dependent transonic flow over a wing undergoing forced pitching oscillation about the mid-chord point of the wing root section. Details of both test cases are given in Table 6.8. The pitching motion of the wing in each test case was given by the expression:

\[ \alpha(t) = \alpha_m + \alpha_0 \sin(\omega t) \]  

(6.3)

where, similar to §6.3, \( \alpha(t) \) is the instantaneous wing angle of incidence, \( \alpha_m \) is the mean angle of incidence, \( \alpha_0 \) is the pitching amplitude and \( \omega \) is the radian frequency. The radian frequency \( \omega \) is related to the reduced frequency \( k \) by Equation (6.2) except that the airfoil chord is now replaced by the wing root chord.

The first test case was for the unsteady transonic flow over a pitching ONERA M6 wing. The geometry of this swept wing is described in detail in §6.4. The pitching amplitude was 5°, which is quite large, and the reduced frequency was 0.1. This test case was also computed by Jameson and results presented with his original description of the multigrid implicit method.\(^{64}\) This is a complex test case to compute due to the large pitching amplitude.

The second test case was for the unsteady transonic flow over a rectangular wing with NACA 64A010 airfoil sections. The wing is unswept and has an aspect ratio of two. The pitching amplitude and the reduced frequency were 1° and 0.135 respectively. As comparisons can be made with measurements obtained by Mabey \textit{et al.},\(^{122}\) this is a standard test case for validation purposes.\(^{70,123-125}\)

A 144x24x24 C-H grid was used for performing each test case. The grid around the ONERA M6 wing was the same as one used in the steady calculations and was referred to in §6.4 as the medium grid. The grid for the NACA 64A010 wing was generated by applying the same methodology used in the generation of the grids for the steady wing calculations. This grid density was found to be sufficient in the steady calculations to obtain solutions of reasonable accuracy. The inner parts of both grids are shown in Figures 6.54 and 6.55. The
same dissipation constants and multigrid strategy that were employed very successfully in the steady wing predictions were used in these calculations also.

The unsteady calculations were performed using 36 time steps per period and for three periods of oscillation. This number of time steps was found to be sufficient in the two-dimensional calculations. Each unsteady solution was initialised using the converged solution of a preceding steady calculation at the mean angle of incidence. Convergence was deemed to have been achieved at each time physical time step when the residual of the continuity equation had dropped by two orders of magnitude. To achieve this level of convergence it was necessary to run the algorithm in single precision mode only.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Wing</th>
<th>$M_{a\infty}$</th>
<th>$\alpha_{m}$</th>
<th>$\alpha_{o}$</th>
<th>$k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ONERA M6</td>
<td>0.84</td>
<td>0.0°</td>
<td>5.0°</td>
<td>0.1</td>
</tr>
<tr>
<td>2</td>
<td>NACA 64A010</td>
<td>0.8</td>
<td>0.0°</td>
<td>1.0°</td>
<td>0.135</td>
</tr>
</tbody>
</table>

*Table 6.8* Test cases for the unsteady wing algorithm.

### 6.5.2 Results of Test Case 1: ONERA M6 Wing

The results of the test cases are presented in Figures 6.56 to 6.63, with the results of Test Case 1 presented first. The variation of the lift coefficient with time is shown in Figure 6.56 and with angle of incidence during the third cycle in Figure 6.57. The values of lift coefficient in Figure 6.57 lie on a slanted oval, as in the two-dimensional results, because of the phase lag between lift coefficient and time. Computed surface pressures at four instances during the third period are shown in Figures 6.58 and 6.59 for two spanwise locations; the first near the root at $\zeta = 0.2$ and the second near the wing tip at $\zeta = 0.9$. As can be seen from these figures there is a large variation in the surface pressure over a complete cycle due the large amplitude of oscillation. The convergence at each time step was very rapid with, on average, only 22 multigrid cycles required to reduce the residual by two orders of magnitude. With this convergence criterion, the lift coefficient was converged to at least five decimal places at each time step. The maximum CFL number for this test case was approximately 2000 and the complete calculation, including the initial steady part, took 12 hours using a single 300 MHz processor on a SUN ULTRA SPARC 450 workstation.
6.5.3 Results of Test Case 2: NACA 64A010 Wing

The results of the NACA 64A010 wing test case are shown in Figures 6.60 to 6.63. The variation of lift coefficient with time is shown in Figure 6.60. An important point to note from this figure, and also the corresponding figure for the previous test case, is that a periodic solution is established more quickly in the three-dimensional calculations than in the two-dimensional calculations. The variation of lift coefficient with angle of incidence is shown in Figure 6.61. Figures 6.62 and 6.63 show the computed surface pressure during the third cycle at two spanwise locations; $\zeta = 0.5$ and $\zeta = 0.77$. The variation of surface pressure during the cycle is not as large as in the previous test case due to the smaller pitching amplitude. The convergence at each time step was very rapid for this test case also with, on average, 20 multigrid cycles required at each time step to achieve the convergence criterion. The lift coefficient was converged to five decimal places after only 5-6 cycles at each time step and the maximum CFL number exceeded 800. This calculation required 11 hours of CPU time, 1 hour less than the previous calculation. Indeed, a convergence criterion of one order of magnitude reduction would have been sufficient to compute this test case due to the small pitching amplitude. This would lead to a significant reduction of between 40 to 50% in CPU time.

The results of this test case compare well with the computations of other researchers.\textsuperscript{70,123-125} This is illustrated in Figure 6.64 which shows a comparison of the computed variation of the lift coefficient with the computed values of Singh.\textsuperscript{123} Note that Singh’s values were read off a graph in Reference 123 because tabulated values were not available. In his calculation, Singh used an unstructured mesh with 40,533 cells and an explicit time-stepping scheme. His calculation required 23.6 hours of CPU time on a Cray YMP computer to compute an initial steady solution and three complete cycles of oscillation. A direct comparison of run times between the present algorithm and Singh’s cannot be made due to the fact that Singh incurred the extra work of grid adaptation at each time step.
Figure 6.54  Inner part of the 144x24x24 C-H grid around the ONERA M6 wing.

Figure 6.55  Inner part of the 144x24x24 C-H grid around the NACA 64A010 wing.
Figure 6.56 Computed variation of lift coefficient with non-dimensional time for the ONERA M6 wing. $M_{\infty} = 0.84$, $\alpha_m = 0.0^\circ$, $\alpha_o = 5.0^\circ$ and $k = 0.1$.

Figure 6.57 Computed variation of lift coefficient with angle of incidence for the ONERA M6 wing. $M_{\infty} = 0.84$, $\alpha_m = 0.0^\circ$, $\alpha_o = 5.0^\circ$ and $k = 0.1$. 
Figure 6.58 Variation of computed surface pressure at $\zeta = 0.2$ with angular position for the ONERA M6. $M_{\infty} = 0.84$, $\alpha_m = 0.0^\circ$, $\alpha_o = 5.0^\circ$ and $k = 0.1$. 
Figure 6.59 Variation of computed surface pressure at $\zeta = 0.9$ with angular position for the ONERA M6. $Ma_{\infty} = 0.84$, $\alpha_m = 0.0^\circ$, $\alpha_o = 5.0^\circ$ and $k = 0.1$. 
Figure 6.60 Computed variation of lift coefficient with non-dimensional time for the NACA 64A010 wing. $M_{\infty} = 0.8$, $\alpha_m = 0.0^\circ$, $\alpha_o = 1.0^\circ$ and $k = 0.135$.

Figure 6.61 Computed variation of lift coefficient with angle of incidence for the NACA 64A010 wing. $M_{\infty} = 0.8$, $\alpha_m = 0.0^\circ$, $\alpha_o = 1.0^\circ$ and $k = 0.135$. 
Figure 6.62 Variation of computed surface pressure at $\zeta = 0.5$ with angular position for the NACA 64A010 wing. $Ma_{\infty} = 0.8$, $\alpha_m = 0.0^\circ$, $\alpha_o = 1.0^\circ$ and $k = 0.135$. 
Figure 6.63 Variation of computed surface pressure at $\zeta = 0.77$ with angular position for the NACA 64A010 wing. $Ma_{in} = 0.8$, $\alpha_{m} = 0.0^\circ$, $\alpha_{o} = 1.0^\circ$ and $k = 0.135$. 
Figure 6.64 Comparison of computed variation of lift coefficient with angle of incidence for the NACA 64A010 wing with Singh.\textsuperscript{123} \( M_a = 0.8 \), \( \alpha_m = 0.0^\circ \), \( \alpha_o = 1.0^\circ \) and \( k = 0.135 \).
6.6 Steady Propeller Calculations: Axisymmetric Inflow

6.6.1 Introduction

The steady propeller algorithm, PROP3DS, was applied to the prediction of the flowfields around two very different propeller geometries under axisymmetric inflow conditions; a two-bladed propeller and a six-bladed propeller shown in Figures 6.65 and 6.66 respectively. These propeller geometries were selected because predictions can be compared with high quality wind tunnel measurements to validate and to demonstrate the accuracy of the algorithm.

The computational domain for all steady propeller calculations spanned one inter-blade region only, since the flow was spatially periodic from one inter-blade region to the next (see §4.2). Grids for all propeller test cases were generated using the turbomachinery interactive grid generation program TIGER. A C-H mesh topology was employed for all steady calculations, with the C part in the streamwise direction along the nacelle and the H part in the circumferential direction. Sections from two sample grids are shown in Figures 6.67 and 6.68. Grid clustering was applied near the blade leading and trailing edges in the streamwise direction and near the nacelle and blade tip in the radial direction. The grid was also clustered near the blade surfaces in the circumferential direction.

The design of an effective multigrid strategy for the propeller calculations was not easy. Originally, it was planned to use the same multigrid strategy for the propeller calculations that was used very successfully for the airfoil and wing calculations, both steady and time-accurate. The multigrid strategy for these calculations consisted of using a W-multigrid cycle with the maximum number of grid levels, or very close to the maximum number, and sub-iterations on the coarse-grid levels. This strategy was tried in the propeller calculations also but led to convergence difficulties, especially when using four or five multigrid levels.

Initially, these convergence problems were attributed to a coding error in the multigrid routines as no difficulties were ever encountered with the single-grid scheme. Several exhaustive checks of the algorithm were carried out but no error was identified. It is worth pointing out that the same multigrid routines worked very successfully in the wing algorithms and led to rapid convergence. The blame was then assigned to the fact that periodicity was not accounted for in the implementation of the implicit residual averaging. However, the incor-
poration of periodicity and the use of a periodic tridiagonal solver made no significant difference. The current theory is that high aspect ratios and distorted cells in the computational domain were the cause of the convergence difficulties. Eventually, after a large number of calculations using different grid densities, an effective strategy was established that consisted of a W-multigrid cycle with a maximum of three multigrid grid levels and one Runge-Kutta time step on each level. This was the multigrid strategy that was used to obtain the results presented here. This multigrid scheme leads to a significant speed-up in convergence over a single-grid scheme and, very importantly, it is also robust. FMG was used with two or three grid levels and 50 to 100 multigrid cycles performed on the coarse-grid levels.

The second and fourth difference dissipation constants were set equal to 1/2 and 1/32 respectively. These values led to rapid convergence and accurate solutions in the airfoil and wing calculations and hence were employed here also. The scaling factor $\alpha$ used in the dissipation model and the implicit residual averaging was found to be more problem dependent and one value could not be used for all calculations. The value used in each test case is given below.

### 6.6.2 NACA 10-(3)(066)-033 Propeller

#### 6.6.2.1 Introduction

The two-bladed NACA 10-(3)(066)-033 propeller, is composed of NACA 16 series airfoil sections and has a rectangular planform. The geometry of the propeller is shown in Figure 6.65. A series of high speed wind tunnel tests were carried out by NACA around 1950 on full scale propellers, including a 10-ft diameter NACA 10-(3)(066)-033 propeller, to determine blade section characteristics by measuring the pressure distributions on the airfoil sections under operating conditions. These tests were conducted to fulfil a need for propeller blade section characteristics in the transonic regime. The measurements for the NACA 10-(3)(066)-033 propeller are extensive and are useful for validating the accuracy of a propeller algorithm.

One test case was computed using this propeller geometry. The flow conditions are given in Table 6.9. The freestream Mach number was 0.56 and the advance ratio $J$, defined as:
J = \frac{V_\infty}{nD} \quad (6.4)

where $V_\infty$ is the freestream speed, $n$ is the number of revolutions per second and $D$ is the propeller diameter, was blade angle at 75% of the blade radius was 45°. A medium density grid with 96x24x64 cells in the axial, radial and circumferential directions respectively was used with 48x16 cells on the blade surfaces. The scaling factor $\alpha$ was set equal to 0.4.

<table>
<thead>
<tr>
<th>$Ma_\infty$</th>
<th>$J$</th>
<th>$\beta_{3/4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.56</td>
<td>2.3</td>
<td>45.0°</td>
</tr>
</tbody>
</table>

Table 6.9  Conditions for the NACA 10-(3)(066)-033 propeller steady test case.

6.6.2.2 Results of Test Case

The results of the test case are presented in Figures 6.69 to 6.73. Figure 6.69 shows the convergence history. Convergence was rapid with engineering accuracy achieved in 91 multi-grid cycles and machine zero in 467 cycles. The development of the thrust coefficient is shown in Figure 6.70. The thrust coefficient, defined as:

$$C_T = \frac{T}{\rho_\infty n^2 D^4} \quad (6.5)$$

where $T$ is the thrust force and $\rho_\infty$ is the freestream density, was converged to four decimal places after only 51 cycles. A comparison of computed and measured surface pressure is shown in Figure 6.71 for six radial ($\eta$) locations. As in the three-dimensional wing calculations, the computed pressures were obtained at the experimental locations by cubic polynomial interpolation. The comparison between computed and measured values is excellent considering the complexity of the flowfield. A shock wave can be clearly identified on the suction surface spanning from about 65% of the blade radius out to the blade tip. The disparity between the predictions and measurements can be attributed to the omission of viscous effects and also to the fact that the blade deformation due to centrifugal and aerodynamic loading was not accounted for. The undeformed blade shape, also called the "cold" blade shape, was used in the present calculation. Contours of relative Mach number at the cell centres adjacent to the pressure and suction surfaces are shown in Figures 6.72
and 6.73 respectively. The supersonic flow region and the shock wave on the suction surface can be seen in Figure 6.73.

### 6.6.3 APIAN Propeller

#### 6.6.3.1 Introduction

This six-bladed advanced propeller was designed as part of a major collaborative, Fourth Framework, EU Research and Technology project, APIAN, on advanced propeller integration, aerodynamics and noise. The blades of the propeller incorporate very thin airfoil sections and are highly swept and twisted to minimise noise generation and to reduce compressibility losses. The APIAN propeller geometry is shown in Figure 6.66. Three test cases were computed for this propeller; one for a high subsonic/low transonic freestream Mach number and two for a low subsonic freestream Mach number. The conditions for each test case are listed in Table 6.10. The low speed test cases were computed so that a novel comparison could be made between predictions and state-of-the-art particle image velocimetry (PIV) measurements. As mentioned previously in Chapter 1 this is the first time that a comparison of the computed flowfield downstream of an advanced propeller under axisymmetric inflow conditions has been compared with PIV measurements made in a large-scale wind tunnel.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>$M_a$</th>
<th>$J$</th>
<th>$\beta_{3/4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.7</td>
<td>3.117</td>
<td>57.0°</td>
</tr>
<tr>
<td>2</td>
<td>0.23</td>
<td>1.11</td>
<td>40.4°</td>
</tr>
<tr>
<td>3</td>
<td>0.23</td>
<td>1.2</td>
<td>40.4°</td>
</tr>
</tbody>
</table>

*Table 6.10* Conditions for the APIAN propeller steady test cases.

#### 6.6.3.2 PIV Measurements

The PIV measurements were obtained on an APIAN propeller model in the DNW Low Speed Tunnel (DNW LST) in Holland in December 1997 as part of the APIAN project. The test campaign was conducted jointly by the German and Italian national aerospace research centres, otherwise known as DLR and CIRA, respectively.
PIV is a recently developed non-intrusive measurement technique. With PIV the flow is seeded with tiny tracer particles that follow the flow streamlines. The most common seeding particles for gaseous flow investigations are oil droplets with a diameter of the order of one micrometer. A plane in the flow is illuminated at two instances in time by means of a pulse laser, with a short time delay $\Delta t$ between the first and second pulses. The positions of the tracer particles in the flow at times $t$ and $t+\Delta t$ are recorded by a camera. A comparison of the two images gives a displacement vector for the movement of a particle between the two illuminations. The velocity of the tracer particles is obtained by dividing by time interval $\Delta t$. A schematic of a PIV system in a wind tunnel is shown in Figure 6.79.

DNW LST is a continuous wind tunnel with a 3.0x2.25m test section, and a Mach number range of 0.0 to 0.23. The propeller model had a diameter of 0.5m and was mounted on the external force balance of the wind tunnel. The propeller model included an internal six component rotating balance and the blade angle was set at 40.4° for all tests.

Tests were conducted at two advance ratios, 1.11 and 1.2, and for three different angles of incidences and yaw, 0°, -10°, +10°. All the tests were conducted for a freestream velocity of approximately 80m/s, corresponding to a freestream Mach number of 0.23. The variation in advance ratio was achieved by changing the rotational speed of the propeller from approximately 8650 to 8000 rpm.

The PIV measurement planes were located in the propeller wake and had the same dimensions of approximately 0.2x0.2m. In total about 1700 image pairs, related to 28 different test conditions, were acquired. For each test condition, at least 40 pairs of images of the instantaneous flowfield were obtained. The velocity vectors obtained from the post-processing of image pairs were interpolated on to a uniform 62x61 point grid with a spatial resolution of 0.3 mm. A complete description of the PIV measurements, including the level of experimental error, is given in References 132 to 134. Figure 6.80 shows the APIAN propeller model in the DNW wind tunnel during the tests.

Comparisons are made of circumferentially averaged predictions and PIV measurements for the propeller at zero angle of incidence and zero yaw (i.e., axisymmetric inflow conditions). The comparisons are made for the measurement planes shown in Figures 6.81 and 6.82.
6.6.3.3 Results of Test Case 1

The first test case was for a freestream Mach number of 0.7, an advance ratio of 3.117 and a blade angle of 57°. A fine grid with 128x48x48 cells was used with 64x32 cells on the blade surfaces. The scaling factor $\alpha$ was set equal to 0.5, the value also used in the three-dimensional wing calculations. Figures 6.74 to 6.78 show the results of this test case. The convergence history is shown in the first of the figures. For this test case convergence to engineering accuracy was achieved in 113 multigrid cycles and to machine zero in 618 cycles. Small oscillations can be seen in the convergence history that can be attributed to a less than optimum value for the scaling factor $\alpha$. The convergence rate was still very good, however. The thrust coefficient, the development of which is shown in Figure 6.75, converged to four decimal places in only 88 multigrid cycles. Computed surface pressures at six radial stations are shown in Figure 6.76. The variation of pressure with chordwise positions is very smooth over most of the blade chord. The pressure jumps at the trailing edges are due to the inadequate grid resolution in this area. Figures 6.77 and 6.78 show contours of relative Mach number at the cell centres adjacent to the suction and pressure surfaces respectively. For these test case conditions, a region of supersonic flow is established on the suction surface near the blade tip.

6.6.3.4 Results of Test Case 2

Test Case 2 was for a freestream Mach number of 0.23, an advance ratio of 1.11 and a blade angle of 40.4°. This test case was computed using a very fine grid with 136x64x64 cells and 48x32 cells on the blade surfaces. This was the finest grid used to date by the author for the calculation of the flowfield around an advanced propeller under axisymmetric inflow conditions. The scaling factor $\alpha$ was set equal to 0.5 for this test case also. The results of the test case, shown in Figures 6.83 to 6.88, are compared with PIV measurement for Planes 1 and 2 in the propeller wake. It is important to note that Cartesian coordinate system used in the presentation of the results of Test Cases 2 and 3 corresponds to the coordinate system used in the PIV measurements shown in Figures 6.81 and 6.82.

A comparison of the computed and measured variation of radial velocity with axial position in Plane 1 is shown in Figure 6.83. The comparison is very good along the length of the plane. A similar comparison of axial velocity at four radial locations is shown in Figure 6.84. The algorithm predicts the magnitude of the axial velocity very well at all radial lo-
It is important to note that the \( y \) direction and the \( v \) velocity component do not correspond to radial values as for Plane 1. The comparison of axial and \( v \) velocity components with measurements at two \( y \) locations is shown in Figures 6.85 to 6.88. The overall agreement between computed and measured values is very good.

### 6.6.3.5 Results of Test Case 3

The conditions for Test Case 3 are the same as the previous test case with the exception of the advance ratio which was 1.2. This test case was computed using the same grid and scaling factor as in the previous test case. Comparisons are made of computed data with PIV measurements for Plane 1 only. Figure 6.89 shows the comparison of computed and measured radial velocity variation just beyond the blade tip. A comparison of computed and measured axial velocity at four radial locations is shown in Figure 6.90. The comparisons are very good for this test case also.

The computed variation of axial velocity with radial position is compared with measurements for Test Case 2 and 3 in Figures 6.91 and 6.92 for Plane 1. The two figures are for different axial locations. The computed values are in very good agreement with measurements for both advance ratios and at both axial locations. The change in magnitude of axial velocity due to the change in advance ratio is similar both in the computed and in the measured data.

In summary the overall agreement between the predictions using PROP3DS and the PIV measurements is very good. This generates a high level of confidence in the steady propeller algorithm for the calculation of the flow around a propeller with an axisymmetric inflow.
Figure 6.65  The two-bladed NACA 10-(3)(066)-033 propeller.

Figure 6.66  The six-bladed APIAN advanced propeller.
Figure 6.67  A section of a C-H grid in an inter-blade region of the NACA 10-(3)(066)-033 propeller for steady flowfield calculations.

Figure 6.68  A section of a C-H grid in an inter-blade region of the APIAN propeller for steady flowfield calculations.
Figure 6.69  Convergence history of the steady NACA 10-(3)(066)-033 propeller calculation. $\text{Ma}_{\infty} = 0.56$, $J = 2.3$ and $\beta_{3/4} = 45.0^\circ$.

Figure 6.70  Development of the thrust coefficient during the steady NACA 10-(3)(066)-033 propeller calculation. $\text{Ma}_{\infty} = 0.56$, $J = 2.3$ and $\beta_{3/4} = 45.0^\circ$. 

129
Figure 6.71 Comparison of computed and measured surface pressure for the NACA 10-(3)(066)-033 propeller. $M_{a_{\infty}} = 0.56$, $J = 2.3$ and $\beta_{3/4} = 45.0^\circ$. 
**Figure 6.72** Contours of relative Mach number at the cell centres adjacent to the pressure surface of the NACA 10-(3)(066)-033 propeller blade. $Ma_\infty = 0.56$, $J = 2.3$, $\beta_{3/4} = 45.0^\circ$ and $\Delta Ma_{rel} = 0.05$.

**Figure 6.73** Contours of relative Mach number at the cell centres adjacent to the suction surface of the NACA 10-(3)(066)-033 propeller blade. $Ma_\infty = 0.56$, $J = 2.3$, $\beta_{3/4} = 45.0^\circ$ and $\Delta Ma_{rel} = 0.05$. 
Figure 6.74  Convergence history of the steady APIAN propeller calculation. \( \text{Ma}_\infty = 0.7, \ J = 3.117 \) and \( \beta_{3/4} = 57.0^\circ \).

![Convergence History Graph](image1)

Figure 6.75  Development of the thrust coefficient during the steady APIAN propeller calculation. \( \text{Ma}_\infty = 0.7, \ J = 3.117 \) and \( \beta_{3/4} = 57.0^\circ \).

![Thrust Coefficient Graph](image2)
Figure 6.76  Computed surface pressure for the APIAN propeller. $Ma_{\infty} = 0.7$, $J = 3.117$ and $\beta_{3/4} = 57.0^\circ$. 
Figure 6.77  Contours of relative Mach number at the cell centres adjacent to the suction surface of the APIAN propeller blade. $Ma_{in} = 0.7$, $J = 3.117$, $\beta_{3/4} = 57.0^\circ$ and $\Delta Ma_{rel} = 0.05$.

Figure 6.78  Contours of relative Mach number at the cell centres adjacent to the pressure surface of the APIAN propeller blade. $Ma_{in} = 0.7$, $J = 3.117$, $\beta_{3/4} = 57.0^\circ$ and $\Delta Ma_{rel} = 0.05$. 
Figure 6.79  Schematic of a PIV measurement system.

Figure 6.80  The APIAN propeller in the DNW Low Speed Tunnel.
Figure 6.81  A front view of the APIAN propeller and PIV measurement planes.

Figure 6.82  A plan view of the APIAN propeller and PIV measurement planes.
Figure 6.83 Comparison of computed and measured radial velocity variation with axial position in Plane 1. $M_{\infty} = 0.23$, $J = 1.11$ and $\beta_{3/4} = 40.4^\circ$.

Figure 6.84 Comparison of computed and measured axial velocity variation with axial position in Plane 1. $M_{\infty} = 0.23$, $J = 1.11$ and $\beta_{3/4} = 40.4^\circ$. 
Figure 6.85  Comparison of computed and measured axial velocity variation with axial position in Plane 2. $Ma_{\infty} = 0.23$, $J = 1.11$ and $\beta_{3/4} = 40.4^\circ$.

Figure 6.86  Comparison of computed and measured axial velocity variation with axial position in Plane 2. $Ma_{\infty} = 0.23$, $J = 1.11$ and $\beta_{3/4} = 40.4^\circ$. 
Figure 6.87  Comparison of computed and measured v velocity variation with axial position in Plane 2. \( \text{Ma}_{\infty} = 0.23, J = 1.11 \) and \( \beta_{3/4} = 40.4^\circ \).

Figure 6.88  Comparison of computed and measured v velocity variation with axial position in Plane 2. \( \text{Ma}_{\infty} = 0.23, J = 1.11 \) and \( \beta_{3/4} = 40.4^\circ \).
Figure 6.89 Comparison of computed and measured radial velocity variation with axial position in Plane 1. \( M_a = 0.23, J = 1.2 \) and \( \beta_{3/4} = 40.4^\circ \).

Figure 6.90 Comparison of computed and measured axial velocity variation with axial position in Plane 1. \( M_a = 0.23, J = 1.2 \) and \( \beta_{3/4} = 40.4^\circ \).
**Figure 6.91** Comparison of computed and measured axial velocity variation with radial position in Plane 1. $M_a = 0.23$ and $\beta_{3/4} = 40.4^\circ$.

**Figure 6.92** Comparison of computed and measured axial velocity variation with radial position in Plane 1. $M_a = 0.23$ and $\beta_{3/4} = 40.4^\circ$. 
6.7 Transient Propeller Calculations: Asymmetric Inflow

6.7.1 Introduction

As outlined at the beginning of this thesis, the ultimate goal of the present work was to develop an algorithm capable of predicting the unsteady flowfield around a propeller under asymmetric inflow conditions. The unsteady propeller algorithm was successfully developed and is designated PROP3DU.

The results of two test cases computed using PROP3DU are presented here. Both test cases were for the APIAN advanced propeller, but for very different flow conditions. The first test case was for a low subsonic freestream Mach number of 0.23, a large angle of incidence of $10^\circ$ and a blade angle of $40.4^\circ$. This test case corresponded to take-off conditions. The second test case was for cruise conditions with a freestream Mach number of 0.7, an angle of incidence of $3^\circ$ and a blade angle of $57^\circ$. The complete set of conditions for each test case are listed in Table 6.11. Note that due to the change in blade angle, the propeller geometry differed significantly between these two test cases.

The complete propeller must be modelled in the prediction of the flow around a propeller with an asymmetric inflow (see §4.2). Each test case was computed using a C-H grid with a density of 96x24x24 cells in each inter-blade region. The grids were generated for a single inter-blade region using TIGER, and then replicated for the other five inter-blade regions by the propeller algorithm. Each grid contained a total of 331,776 cells. Figures 6.93 and 6.94 show the surface grids for both test cases (note the increased grid clustering in the different regions).

The same dissipation constants and multigrid strategy that were employed successfully in the steady propeller calculations were used for these test cases also. The second and fourth difference constants were set equal to $1/2$ and $1/32$ respectively and a W-multigrid cycle with three grid levels and one Runge-Kutta step on each level was employed as the multigrid strategy. The scaling factor $\alpha$ was set equal to 0.4 for both calculations.

The test cases were performed using 36 time steps per revolution of the propeller and for three complete revolutions. A two orders of magnitude reduction in the residual of the continuity equation was specified as the convergence criterion at each time step. This convergence criterion was found to provide well converged solutions in the unsteady wing
calculations (see §6.5). The unsteady solution in each inter-blade region was initialised using the converged solution of a preliminary steady flow calculation (i.e., with an axisymmetric inflow) for a single inter-blade region only.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>( M_{\infty} )</th>
<th>( J )</th>
<th>( \beta_{3/4} )</th>
<th>( \alpha )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.23</td>
<td>1.11</td>
<td>40.4°</td>
<td>10.0°</td>
</tr>
<tr>
<td>2</td>
<td>0.7</td>
<td>3.117</td>
<td>57.0°</td>
<td>3.0°</td>
</tr>
</tbody>
</table>

*Table 6.11* Conditions for the transient propeller test cases.

### 6.7.2 Results of Test Case 1: Take-Off Conditions

The results of this test case are presented in Figures 6.95 to 6.98 and in Figure 6.103. Figure 6.95 shows the variation of thrust coefficient with phase angle for a single blade, denoted blade 1, for the complete unsteady calculation. The blades are notionally numbered in a clockwise direction looking downstream, with blade 1 in the 12 o’clock position at the beginning of the unsteady calculation. The variation in thrust coefficient is quite large with the maximum and minimum values occurring when the blade is close to the 9 and 3 o’clock positions respectively, as would be expected. The thrust coefficient for a single blade from the preliminary steady calculation was 0.0824. An extract from the convergence history for an inter-blade region is shown in Figure 6.96. The inter-blade region is between blades 1 and 2 and is designated inter-blade region 1. The convergence to a pseudo steady-state was fast with approximately 50 multigrid cycles required at each time step to achieve the convergence criterion in all inter-blade regions. The computed surface pressure on blade 1 at two radial locations is shown in Figure 6.97 and 6.98 respectively for four instances during the third revolution of the propeller. The variation in surface pressure during the revolution is significant at both locations, with a very large suction peak occurring on the upper surface at \( \eta = 0.3 \) when the blade is around the 9 o’clock position. Contours of relative Mach number from the last time step are shown in Figure 6.103 at the cell centres adjacent to the propeller blade and nacelle surfaces. Even with a low subsonic freestream Mach number, the relative flow becomes supersonic near the leading edges of the blades. This calculation, including the initial steady computation, required approximately 115 hours of CPU time using a single 300 MHz processor on a SUN ULTRA SPARC 450 workstation. PROP3DU was run in single precision mode only for these unsteady test cases. The maximum CFL number in this calculation reached approximately 1780.
6.7.3 Results of Test Case 2: Cruise Conditions

The results of the second test case are presented in Figures 6.99 to 6.102 and in Figure 6.104. The variation of thrust coefficient with phase angle for blade 1 is shown in Figure 6.99 and again is quite large. The single blade thrust coefficient computed during the steady calculation was 0.0533. Figure 6.100 shows an extract from the convergence history for inter-blade region 1. Approximately 32 multigrid cycles were required at each time step to achieve convergence, less than the previous test case. The convergence appears to be monotonic at each time step. The computed surface pressure at $\eta = 0.3$ and $\eta = 0.8$ on blade 1 during the third revolution are shown in Figures 6.101 and 6.102 respectively. As in the previous test case the pressure changes during the revolution are significant both near the blade root and near the blade tip. Contours of relative Mach number at the last time step are shown in Figure 6.104. This calculation required approximately 74 hours of CPU time on the SUN SPARC workstation with the maximum CFL number reaching a value of around 1720.

These test cases demonstrate the ability of PROP3DU to successfully predict the complex flowfield around a propeller at incidence. However, these test cases also demonstrate that the calculation of the flowfield around a propeller under asymmetric inflow conditions is still computationally demanding using a standard workstation, even using the efficient multigrid implicit method.
Figure 6.93  Inner part of the 96x24x24x6 APIAN propeller grid with $\beta_{3/4} = 40.4^\circ$.

Figure 6.94  Inner part of the 96x24x24x6 APIAN propeller grid with $\beta_{3/4} = 57.0^\circ$. 
Figure 6.95 Computed variation of thrust coefficient for blade 1 for the complete unsteady propeller calculation. $M_{\infty} = 0.23$, $J = 1.11$, $\beta_{3/4} = 40.4^\circ$ and $\alpha = 10.0^\circ$.

Figure 6.96 Convergence history for inter-blade region 1 during the unsteady propeller calculation. $M_{\infty} = 0.23$, $J = 1.11$, $\beta_{3/4} = 40.4^\circ$ and $\alpha = 10.0^\circ$. 
Figure 6.97  Computed surface pressure at $\eta = 0.3$ on blade 1 during the third revolution. $Ma_{\infty} = 0.23$, $J = 1.11$, $\beta_{3/4} = 40.4^\circ$ and $\alpha = 10.0^\circ$. 
Figure 6.98  Computed surface pressure at $\eta = 0.8$ on blade 1 during the third revolution. $M_{\infty} = 0.23$, $J = 1.11$, $\beta_{3/4} = 40.4^\circ$ and $\alpha = 10.0^\circ$. 
Figure 6.99  Computed variation of thrust coefficient for blade 1 for the complete unsteady propeller calculation. \( \text{Ma}_\infty = 0.7, J = 3.117, \beta_{3/4} = 57.0^\circ \) and \( \alpha = 3.0^\circ \).

Figure 6.100  Convergence history for inter-blade region 1 during the unsteady propeller calculation. \( \text{Ma}_\infty = 0.7, J = 3.117, \beta_{3/4} = 57.0^\circ \) and \( \alpha = 3.0^\circ \).
Figure 6.101  Computed surface pressure at $\eta = 0.3$ on blade 1 during the third revolution. $M_{\infty} = 0.7$, $J = 3.117$, $\beta_{3/4} = 57.0^\circ$ and $\alpha = 3.0^\circ$. 
Figure 6.102 Computed surface pressure at \( \eta = 0.8 \) on blade 1 during the third revolution. \( \text{Ma}_{\infty} = 0.7, J = 3.117, \beta_{3/4} = 57.0^\circ \) and \( \alpha = 3.0^\circ \).
Figure 6.103 Contours of relative Mach number at the cell centres adjacent to the propeller blade and nacelle surfaces. $M_\infty = 0.23$, $J = 1.11$, $\beta_{3/4} = 40.4^\circ$ and $\alpha = 10.0^\circ$.

Figure 6.104 Contours of relative Mach number at the cell centres adjacent to the propeller blade and nacelle surfaces. $M_\infty = 0.7$, $J = 3.117$, $\beta_{3/4} = 57.0^\circ$ and $\alpha = 3.0^\circ$. 
Conclusions

The aim of this work was to develop an algorithm capable of predicting the unsteady inviscid flowfield around a propeller under asymmetric inflow conditions (i.e., at incidence, at non-zero yaw or even a combination of both). This goal was successfully reached by first developing a series of simpler algorithms, each designed to predict a progressively more complex flowfield. In total six algorithms were developed: three steady and three unsteady. Each algorithm solves the Euler equations using a cell-centered, central-differencing, finite-volume method in a transformed computational domain. A controlled amount of artificial dissipation is added to the discrete set of equations both for stability and for accuracy. Explicit multistage time-stepping is employed by the steady algorithms to reach a steady-state. The unsteady algorithms use an implicit method so that there is no theoretical limitation on the maximum time step size. The implicit set of equations are solved at each time step by reformulating the equations as a modified steady problem and marching to steady-state in a pseudo time using the explicit multistage scheme. Local time-stepping, variable coefficient implicit residual averaging and a full approximation storage multigrid scheme are used to accelerate the progression to steady-state.

The results of a range of complex test cases computed using the steady algorithms have been presented. Excellent convergence rates were achieved in these calculations, mainly due to the use of very effective multigrid strategies. Very good agreement was obtained in comparisons of predictions with wind tunnel measurements and with the computed data from other researchers. The steady algorithms have also demonstrated the ability to compute complex shock structures with good accuracy.
The results of test cases computed using the transient algorithms have also been presented, including the results of two unsteady advanced propeller flowfield test cases. A small number of time steps were used to compute each test case and the convergence rate to a pseudo steady-state was very good at each time step. The unsteady flowfields were accurately resolved, including the motion of shock waves. The results of the test cases demonstrate the efficiency of the multigrid implicit method for unsteady analyses.

In conclusion, the steady and unsteady algorithms developed in this work are useful engineering tools for aerodynamic analyses. Furthermore, the unsteady propeller algorithm can be used to provide improved understanding of the complex aerodynamics of advanced propellers.
CHAPTER 8

RECOMMENDATIONS FOR FUTURE WORK

8.1 Recommendations

The following recommendations are made for the improvement of the inviscid algorithms developed during the present research:

1. The mesh topology employed by the two-dimensional airfoil algorithms should be changed from C-type to O-type.\textsuperscript{105} An O-type mesh requires fewer cells than a C-type mesh for the same grid resolution near the airfoil surface. Currently, both algorithms are limited to C-type meshes only, but the modification required to allow them to use O-type meshes is straightforward. The change of mesh topology would make the steady and unsteady airfoil algorithms even more efficient.

2. Likewise, the mesh topology employed by the three-dimensional wing algorithms should be changed from C-H type to C-O type or ideally to O-O type. By using an O-type mesh in the spanwise direction, the geometry of the wing tip could be properly modelled. This would lead to improved solution accuracy in the wing tip region and to a more accurate resolution of the wing tip vortex. As in the previous recommendation, this modification would also lead to more efficient wing algorithms.

3. Alternative dissipation schemes, such as the MATD scheme by Turkel\textsuperscript{54-56} and the CUSP scheme by Jameson,\textsuperscript{57-59} should be implemented and tested in each algorithm with the ultimate aim of replacing the scalar dissipation model with one capable of producing more accurate flow solutions (and, if possible, discrete shock waves with one interior point).

4. Third-order differencing of the physical time derivative should be evaluated in the unsteady algorithms to see if solutions comparable in accuracy to those obtained using
second-order differencing can be obtained but with significantly fewer time steps, as claimed by some researchers.\textsuperscript{73,76,77} If true, this would further improve the efficiency of the transient algorithms, at the affordable expense of an increase in computer memory requirements.

5. Appropriate forms of the normal momentum equation should be derived and implemented in the unsteady algorithms and in the steady propeller algorithm. This would lead to improved solution accuracy, especially on coarse grids, at the expense of a small increase in run time.

6. A thorough investigation should be performed to establish the reasons behind the convergence difficulties when using the same multigrid strategy that was very successful in the airfoil and wing calculations for the propeller calculations.

7. More accurate far-field boundary conditions should be investigated for each algorithm.

8. Time-accurate propeller predictions should be compared with wind tunnel measurements to assess the level of accuracy of the unsteady propeller algorithm.
REFERENCES


Proceedings No. 374, Transonic Unsteady Aerodynamics and Its Aeroelastic Applications.


NON-DIMENSIONALISATION OF THE EULER EQUATIONS

A.1 Non-Dimensionalisation

To ensure that the numerical algorithms developed in this work are not restricted to any particular system of units, and also to make these algorithms user-friendly, the variables appearing in the Euler equations are non-dimensionalised using a standard non-dimensionalisation procedure.\(^4^3\)

The independent variables \(\tilde{x}, \tilde{y}, \tilde{z}\) and \(\tilde{t}\) (i.e., Cartesian coordinates and time) are non-dimensionalised using reference values:

\[
x = \frac{\tilde{x}}{l_{\text{ref}}}, \quad y = \frac{\tilde{y}}{l_{\text{ref}}}, \quad z = \frac{\tilde{z}}{l_{\text{ref}}} \quad \text{and} \quad t = \frac{\tilde{t}}{t_{\text{ref}}}
\]  

(A.1)

The tilde superscript (\(\sim\)) denotes dimensional values and the subscript ref denotes reference values. \(l_{\text{ref}}\) is a reference length which is taken as the airfoil chord for the two-dimensional airfoil flow calculations, the wing root chord for the three-dimensional wing flow calculations and the propeller blade tip radius for the propeller flow calculations. \(t_{\text{ref}}\) is a reference time defined as:

\[
\frac{\tilde{t}_{\text{ref}}}{\tilde{t}_{\text{ref}}} = \frac{l_{\text{ref}}}{V_{\text{ref}}}
\]  

(A.2)

where \(V_{\text{ref}}\) is a reference velocity calculated using the reference static pressure, \(p_{\text{ref}}\), and density, \(\rho_{\text{ref}}\), i.e.:

\[
\frac{V_{\text{ref}}}{\sqrt{\frac{p_{\text{ref}}}{\rho_{\text{ref}}}}}
\]  

(A.3)
The Cartesian velocity components are non-dimensionalised as follows:

\[ u = \frac{\tilde{u}}{V_{\text{ref}}} , v = \frac{\tilde{v}}{V_{\text{ref}}} \text{ and } w = \frac{\tilde{w}}{V_{\text{ref}}} \]  

(A.4)

where \( u, v \) and \( w \) are the non-dimensional Cartesian velocity components in the \( x, y \) and \( z \) directions respectively.

Static pressure, density and temperature are non-dimensionalised by their corresponding reference quantities:

\[ p = \frac{\tilde{p}}{p_{\text{ref}}}, \rho = \frac{\tilde{\rho}}{\rho_{\text{ref}}} \text{ and } T = \frac{\tilde{T}}{T_{\text{ref}}} \]  

(A.5)

where \( T_{\text{ref}} \) is a reference temperature. The reference static pressure, density and temperature are set equal to freestream values.

The total energy, \( \tilde{E} \), and total enthalpy, \( \tilde{H} \), per unit mass are non-dimensionalised as follows:

\[ E = \frac{\tilde{E}}{V_{\text{ref}}^2} \text{ and } H = \frac{\tilde{H}}{V_{\text{ref}}^2} \]  

(A.6)

The total roenergy, \( \tilde{E}_r \), and total roenthalpy, \( \tilde{H}_r \), per unit mass are also non-dimensionalised using \( V_{\text{ref}}^2 \):

\[ E_r = \frac{\tilde{E}_r}{V_{\text{ref}}^2} \text{ and } H_r = \frac{\tilde{H}_r}{V_{\text{ref}}^2} \]  

(A.7)

Angular velocity components \( \tilde{\omega}_x, \tilde{\omega}_y \) and \( \tilde{\omega}_z \) about the \( \tilde{x}, \tilde{y} \) and \( \tilde{z} \) axes are non-dimensionalised using \( t_{\text{ref}}^{-1} \):

\[ \omega_x = \frac{\tilde{\omega}_x}{t_{\text{ref}}^{-1}}, \omega_y = \frac{\tilde{\omega}_y}{t_{\text{ref}}^{-1}} \text{ and } \omega_z = \frac{\tilde{\omega}_z}{t_{\text{ref}}^{-1}} \]  

(A.8)

The structure of the six sets of governing equations has been verified to be invariant under this non-dimensionalising transformation.
APPENDIX B

TRANSFORMATION TO A CURVILINEAR COORDINATE SYSTEM

B.1 Introduction

To demonstrate the application of the curvilinear transformation, the three-dimensional Euler equations in Cartesian coordinates in an inertial reference frame are transformed to a time-dependent curvilinear coordinate system. The final set of equations describe the unsteady flow around an oscillating wing.

B.2 Euler Equations in Cartesian Coordinates

The partial differential equation form of the non-dimensional, three-dimensional, Euler equations in Cartesian coordinates in an inertial reference frame, neglecting volumetric heat addition and body forces, is:

\[
\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} + \frac{\partial \mathbf{h}}{\partial z} = 0
\] (B.1)

where the vector of conserved variables, \( \mathbf{q} \), and the vectors of inviscid flux terms, \( \mathbf{f} \), \( \mathbf{g} \) and \( \mathbf{h} \), are:

\[
\mathbf{q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{bmatrix}
\] (B.2)
\[
\begin{align*}
\mathbf{f} &= \begin{bmatrix} 
\rho u \\
\rho u^2 + p \\
\rho uv \\
\rho uw \\
\rho uH 
\end{bmatrix} \\
\mathbf{g} &= \begin{bmatrix} 
\rho v \\
\rho uv \\
\rho v^2 + p \\
\rho vw \\
\rho vH 
\end{bmatrix} \\
\mathbf{h} &= \begin{bmatrix} 
\rho w \\
\rho uw \\
\rho vw \\
\rho w^2 + p \\
\rho wH 
\end{bmatrix}
\end{align*}
\]  

(B.3)  

(B.4)  

(B.5)

where \( \rho \) is the density, \( p \) is the static pressure, \( u, v \) and \( w \) are the Cartesian velocity components in the \( x, y \) and \( z \) directions respectively, \( E \) is the total energy per unit mass and \( H \) is the total enthalpy per unit mass.

### B.3 Applying the Curvilinear Transformation

The general curvilinear axes for a time-dependent curvilinear coordinate system are:

\[
\begin{align*}
\xi &= \xi(x, y, z, t) \\
\eta &= \eta(x, y, z, t) \\
\zeta &= \zeta(x, y, z, t) \\
\tau &= t
\end{align*}
\]  

(B.6)

Using the chain rule for a function of multiple variables, the Cartesian derivatives can be written in terms of the curvilinear derivatives as:
\[
\frac{\partial}{\partial x} = \xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta} + \zeta_x \frac{\partial}{\partial \zeta} + \tau_x \frac{\partial}{\partial \tau} \\
\frac{\partial}{\partial y} = \xi_y \frac{\partial}{\partial \xi} + \eta_y \frac{\partial}{\partial \eta} + \zeta_y \frac{\partial}{\partial \zeta} + \tau_y \frac{\partial}{\partial \tau} \\
\frac{\partial}{\partial z} = \xi_z \frac{\partial}{\partial \xi} + \eta_z \frac{\partial}{\partial \eta} + \zeta_z \frac{\partial}{\partial \zeta} + \tau_z \frac{\partial}{\partial \tau} \\
\frac{\partial}{\partial t} = \xi_t \frac{\partial}{\partial \xi} + \eta_t \frac{\partial}{\partial \eta} + \zeta_t \frac{\partial}{\partial \zeta} + \tau_t \frac{\partial}{\partial \tau} 
\]

(B.7)

where \( \xi_x, \xi_y, \xi_z, \eta_x, \eta_y, \eta_z, \zeta_x, \zeta_y, \zeta_z \) and \( \zeta_t \) are the metrics of the transformation (noting that \( \tau_x = \tau_y = \tau_z = 0 \) and \( \tau_t = 1 \)). Equation set (B.7) can be written in the following matrix form:

\[
\begin{bmatrix}
\frac{\partial}{\partial x} \\
\frac{\partial}{\partial y} \\
\frac{\partial}{\partial z} \\
\frac{\partial}{\partial t}
\end{bmatrix} =
\begin{bmatrix}
\xi_x & \eta_x & \zeta_x & 0 \\
\xi_y & \eta_y & \zeta_y & 0 \\
\xi_z & \eta_z & \zeta_z & 0 \\
\xi_t & \eta_t & \zeta_t & 1
\end{bmatrix}
\begin{bmatrix}
\frac{\partial}{\partial \xi} \\
\frac{\partial}{\partial \eta} \\
\frac{\partial}{\partial \zeta} \\
\frac{\partial}{\partial \tau}
\end{bmatrix}
\]

(B.8)

It is also possible to expand the curvilinear derivatives in terms of the Cartesian derivatives with the aid of the chain rule:

\[
\frac{\partial}{\partial \xi} = x_\xi \frac{\partial}{\partial x} + y_\xi \frac{\partial}{\partial y} + z_\xi \frac{\partial}{\partial z} + t_\xi \frac{\partial}{\partial t} \\
\frac{\partial}{\partial \eta} = x_\eta \frac{\partial}{\partial x} + y_\eta \frac{\partial}{\partial y} + z_\eta \frac{\partial}{\partial z} + t_\eta \frac{\partial}{\partial t} \\
\frac{\partial}{\partial \zeta} = x_\zeta \frac{\partial}{\partial x} + y_\zeta \frac{\partial}{\partial y} + z_\zeta \frac{\partial}{\partial z} + t_\zeta \frac{\partial}{\partial t} \\
\frac{\partial}{\partial \tau} = x_\tau \frac{\partial}{\partial x} + y_\tau \frac{\partial}{\partial y} + z_\tau \frac{\partial}{\partial z} + t_\tau \frac{\partial}{\partial t}
\]

(B.9)

Note again that \( t_\xi = t_\eta = t_\zeta = 0 \) and \( t_\tau = 1 \). Equation set (B.9) can also be written in matrix form:
Comparing Equation sets (B.8) and (B.10) it is evident that the following holds true:

\[
\begin{bmatrix}
\frac{\partial}{\partial \xi} & \frac{\partial}{\partial \eta} & \frac{\partial}{\partial \zeta} & \frac{\partial}{\partial \tau}
\end{bmatrix}
\begin{bmatrix}
x & y & z & 0 \\
x & y & z & 0 \\
x & y & z & 0 \\
x & y & z & 1
\end{bmatrix}
= \begin{bmatrix}
x & y & z & 0 \\
x & y & z & 0 \\
x & y & z & 0 \\
x & y & z & 1
\end{bmatrix}
\]  \quad (B.10)

If the 4x4 matrix on the right-hand side is inverted, it is then possible to solve for the metrics of the transformation. The inverse of a matrix \( \mathbf{D} \) can be obtained using the following expression:

\[
\mathbf{D}^{-1} = \frac{1}{\text{det}(\mathbf{D})} \text{adj}(\mathbf{D})  \quad (B.12)
\]

The determinant of the inverse matrix is:

\[
\begin{vmatrix}
x & y & z & 0 \\
x & y & z & 0 \\
x & y & z & 0 \\
x & y & z & 1
\end{vmatrix} = x_\xi (y_\eta z_\zeta - z_\eta y_\zeta) - y_\xi (x_\eta z_\zeta - z_\eta x_\zeta) + z_\xi (x_\eta y_\zeta - y_\eta x_\zeta)  \quad (B.13)
\]

The Jacobian of the inverse transformation is defined as:

\[
J^{-1} = \frac{\partial(x, y, z, 1)}{\partial(\xi, \eta, \zeta, \tau)} = \begin{vmatrix}
x & y & z & 0 \\
x & y & z & 0 \\
x & y & z & 0 \\
x & y & z & 1
\end{vmatrix}
\]  \quad (B.14)

Following the matrix inversion the metrics of the transformation are evaluated as:
\[
\begin{align*}
\dot{\xi}_x &= J(y_\eta z_\eta - z_\eta y_\eta) \\
\dot{\xi}_y &= J(z_\eta x_\eta - x_\eta z_\eta) \\
\dot{\xi}_z &= J(x_\eta y_\eta - y_\eta x_\eta) \\
\eta_x &= J(y_\xi z_\xi - z_\xi y_\xi) \\
\eta_y &= J(z_\xi x_\xi - x_\xi z_\xi) \\
\eta_z &= J(x_\xi y_\xi - y_\xi x_\xi)
\end{align*}
\]  
\( (B.15) \)

To apply the transformation to the Euler equations, substitute Equation \((B.7)\) into Equation \((B.1)\) and multiply by \(J^{-1}\) to get:

\[
J^{-1} \frac{\partial \mathbf{q}}{\partial \tau} + J^{-1} \frac{\partial \mathbf{q}}{\partial \xi} \dot{\xi}_t + J^{-1} \frac{\partial \mathbf{q}}{\partial \eta} \eta_t + J^{-1} \frac{\partial \mathbf{q}}{\partial \zeta} \zeta_t 
\]

\[
+ J^{-1} \frac{\partial \mathbf{f}}{\partial \xi} + J^{-1} \frac{\partial \mathbf{f}}{\partial \eta} \eta + J^{-1} \frac{\partial \mathbf{f}}{\partial \zeta} \zeta 
\]

\[
+ J^{-1} \frac{\partial \mathbf{g}}{\partial \xi} + J^{-1} \frac{\partial \mathbf{g}}{\partial \eta} \eta + J^{-1} \frac{\partial \mathbf{g}}{\partial \zeta} \zeta 
\]

\[
+ J^{-1} \frac{\partial \mathbf{h}}{\partial \xi} + J^{-1} \frac{\partial \mathbf{h}}{\partial \eta} \eta + J^{-1} \frac{\partial \mathbf{h}}{\partial \zeta} \zeta 
= 0
\]  
\( (B.16) \)

The differentiation of a product gives the following relation:

\[
(J^{-1} k_m) \frac{\partial c}{\partial k} = \frac{\partial}{\partial k} (J^{-1} k_m c) - c \frac{\partial}{\partial k} (J^{-1} k_m) \]  
\( (B.17) \)

where \(k = \xi, \eta, \zeta\) or \(\tau\), \(m = x, y, z\) or \(t\) and \(c = f, g, h\) or \(q\). Using this relation in Equation \((B.16)\) yields:
\frac{\partial}{\partial \tau}(J^{-1} \mathbf{q}) + \frac{\partial}{\partial \xi}(J^{-1} (\xi_1 \mathbf{q} + \xi_x \mathbf{f} + \xi_y \mathbf{g} + \xi_z \mathbf{h})) + \frac{\partial}{\partial \eta}(J^{-1} (\eta_1 \mathbf{q} + \eta_x \mathbf{f} + \eta_y \mathbf{g} + \eta_z \mathbf{h})) + \frac{\partial}{\partial \zeta}(J^{-1} (\zeta_1 \mathbf{q} + \zeta_x \mathbf{f} + \zeta_y \mathbf{g} + \zeta_z \mathbf{h})) = 0 \tag{B.18}

The coefficients, shown in square brackets, of \( \mathbf{q}, \mathbf{f}, \mathbf{g} \) and \( \mathbf{h} \) in the fifth to the eighth terms can be shown to be zero. Consider the sixth term, the coefficient of \( \mathbf{f} \), in Equation (B.18):

\begin{align*}
\frac{\partial}{\partial \xi}(J^{-1} \xi_1) &= \frac{\partial}{\partial \xi}(y_\eta z_\xi - z_\eta y_\xi) = y_\eta z_\xi + y_\eta z_\xi - z_\eta y_\xi - z_\eta y_\xi \\
\frac{\partial}{\partial \eta}(J^{-1} \eta_1) &= \frac{\partial}{\partial \eta}(z_\xi y_\eta - y_\xi z_\eta) = z_\xi y_\eta + z_\xi y_\eta - y_\xi z_\eta - y_\xi z_\eta \\
\frac{\partial}{\partial \zeta}(J^{-1} \zeta_1) &= \frac{\partial}{\partial \zeta}(y_\xi z_\eta - z_\xi y_\eta) = y_\xi z_\eta + y_\xi z_\eta - z_\xi y_\eta - z_\xi y_\eta \tag{B.19}
\end{align*}

The summation of the three terms in Equation (B.19) is zero. The coefficient of the \( \mathbf{g} \) and \( \mathbf{h} \) terms can be shown analytically to be zero in a similar manner. To prove that the coefficient of the fifth term, \( \mathbf{q} \), is zero we refer to the differential form of the geometric conservation law in curvilinear coordinates:\textsuperscript{135}

\frac{\partial}{\partial \tau}(J^{-1} \mathbf{q}) + \frac{\partial}{\partial \xi}(J^{-1} \xi_1) + \frac{\partial}{\partial \eta}(J^{-1} \eta_1) + \frac{\partial}{\partial \zeta}(J^{-1} \zeta_1) = 0 \tag{B.20}

For a rigid grid (i.e., the mesh cells do not deform in time) Equation (B.20) is satisfied automatically. However, for a deforming grid, Equation (B.20) must also be solved along with the governing equations in order to eliminate spurious source terms that can effect accuracy and convergence of the numerical algorithm. Equation (B.20) thus demonstrates that the coefficient of the \( \mathbf{q} \) term is zero. Therefore Equation (B.18) can now be expressed as:
where the vector of conserved variables, \( Q \), and the vector of inviscid flux terms, \( F \), \( G \) and \( H \), are:

\[
Q = q = \begin{bmatrix} 
\rho \\
\rho u \\
\rho v \\
\rho w \\
\rho E 
\end{bmatrix}
\]  
(B.23)

\[
F = J^{-1}(\xi_t q + \xi_x f + \xi_y g + \xi_z h) = J^{-1} 
\begin{bmatrix} 
\rho U \\
\rho u U + \xi_x p \\
\rho v U + \xi_y p \\
\rho w U + \xi_z p \\
\rho H U - \xi_t p 
\end{bmatrix}
\]  
(B.24)

\[
G = J^{-1}(\eta_t q + \eta_x f + \eta_y g + \eta_z h) = J^{-1} 
\begin{bmatrix} 
\rho V \\
\rho u V + \eta_x p \\
\rho v V + \eta_y p \\
\rho w V + \eta_z p \\
\rho H V - \eta_t p 
\end{bmatrix}
\]  
(B.25)

\[
H = J^{-1}(\zeta_t q + \zeta_x f + \zeta_y g + \zeta_z h) = J^{-1} 
\begin{bmatrix} 
\rho W \\
\rho u W + \zeta_x p \\
\rho v W + \zeta_y p \\
\rho w W + \zeta_z p \\
\rho H W - \zeta_t p 
\end{bmatrix}
\]  
(B.26)
U, V and W are the contravariant velocity components in the \( \xi, \eta \) and \( \zeta \) directions and are defined as:

\[
U = \xi_x u + \xi_y v + \xi_z w + \xi_t \\
V = \eta_x u + \eta_y v + \eta_z w + \eta_t \\
W = \zeta_x u + \zeta_y v + \zeta_z w + \zeta_t
\]  

(B.27)

### B.4 Numerical Evaluation of the Metric Terms

A geometric interpretation of the metric terms in Equation (B.22) results in a direct analogy with the integral form of the Euler equations in the physical domain. The vector \( J^{-1} \nabla k \) is the directed area of the cell face normal to a \( k = \text{constant} \) surface (\( k = \xi, \eta \) or \( \zeta \)). The direction of the vector is in the direction of increasing \( k \) as shown in Figure B.1. An area vector, and hence \( J^{-1} \nabla k \), is calculated as one-half of the cross product of the diagonal line elements of the cell face. The inverse Jacobian of the transformation, \( J^{-1} \), is equal to the volume of the cell in the physical domain and is calculated using the efficient method of Kordulla and Vinokur.\(^{136}\)

![Figure B.1](Image)

Figure B.1 Geometrical interpretation of the metric terms.
Determination of the Eigenvalues of the Flux Jacobian Matrices

C.1 Introduction

In this appendix the eigenvalues of the inviscid flux Jacobian matrices of the three-dimensional Euler equations that describe the unsteady flow around an oscillating wing, Equation set (2.42), are obtained. The determination of the eigenvalues is based on the methodology employed by Whitfield and Janus.\textsuperscript{137,138} The eigenvalues of the other five equations sets are obtained in a similar manner, but this work is not reported here.

C.2 Determination of the Eigenvalues

The three-dimensional Euler equations in a time-dependent curvilinear coordinate system in an inertial reference frame, neglecting body forces and volumetric heat addition, can be written as:

\[
\frac{\partial}{\partial t}(J^{-1}Q) + \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} + \frac{\partial H}{\partial \zeta} = 0 \tag{C.1}
\]

where the vector of conserved variables, \(Q\), and the vectors of inviscid flux terms, \(F\), \(G\) and \(H\), are as defined in Appendix B. Equation set (C.1) can also be written in the following quasi-linear form:

\[
\frac{\partial}{\partial t}(J^{-1}Q) + A \frac{\partial Q}{\partial \xi} + B \frac{\partial Q}{\partial \eta} + C \frac{\partial Q}{\partial \zeta} = 0 \tag{C.2}
\]

where the matrices \(A\), \(B\) and \(C\) are called the inviscid flux Jacobian matrices and are defined as:
\[ A = \frac{\partial F}{\partial Q}, \quad B = \frac{\partial G}{\partial Q} \text{ and } C = \frac{\partial H}{\partial Q} \]  

(C.3)

A general expression for the inviscid flux Jacobian matrices can be obtained and is:

\[
\mathbf{K} = \begin{bmatrix}
    k_t & k_x & k_y & k_z & 0 \\
    k_x \phi - u \theta_k & k_x (2 - \gamma) u + \beta_k & k_y u - k_x (\gamma - 1) v & k_z u - k_x (\gamma - 1) w & k_x (\gamma - 1) \\
    k_y \phi - v \theta_k & k_y v - k_y (\gamma - 1) u & k_y (2 - \gamma) v + \beta_k & k_z v - k_y (\gamma - 1) w & k_y (\gamma - 1) \\
    k_z \phi - w \theta_k & k_z w - k_z (\gamma - 1) u & k_z w - k_z (\gamma - 1) v & k_z (2 - \gamma) w + \beta_k & k_z (\gamma - 1) \\
    (2\phi - \gamma \mathbf{E}) \theta_k & k_x (\gamma \mathbf{E} - \phi) & k_y (\gamma \mathbf{E} - \phi) & k_z (\gamma \mathbf{E} - \phi) & \gamma \theta_k + k_t
\end{bmatrix}
\]  

(C.4)

where the terms \( \phi \), \( \beta_k \) and \( \theta_k \) are as follows:

\[
\phi = \frac{\gamma - 1}{2} (u^2 + v^2 + w^2)
\]  

(C.5)

\[
\beta_k = k_x u + k_y v + k_z w + k_t
\]  

(C.6)

\[
\theta_k = k_x u + k_y v + k_z w
\]  

(C.7)

and \( \mathbf{K} \) is equal to \( \mathbf{A}, \mathbf{B} \) and \( \mathbf{C} \) for \( k = \xi, \eta \) and \( \zeta \) respectively. The eigenvalues of the inviscid flux Jacobian matrices are difficult to determine using Equation (C.4) directly because of the number of non-zero elements. To obtain the eigenvalues in a simpler manner we consider the non-conservative vector form of the governing equations in curvilinear coordinates:

\[
\frac{\partial}{\partial \tau} (J^{-1} \mathbf{q}) + a \frac{\partial \mathbf{q}}{\partial \xi} + b \frac{\partial \mathbf{q}}{\partial \eta} + c \frac{\partial \mathbf{q}}{\partial \zeta} = 0
\]  

(C.8)

where:

\[
\mathbf{q} = \begin{bmatrix}
    \rho \\
    u \\
    v \\
    w \\
    p
\end{bmatrix}
\]  

(C.9)
Note that Equation (C.2) can also be written as:

\[ M \frac{\partial}{\partial \tau} (J^{-1} q) + A \frac{\partial q}{\partial \xi} + B \frac{\partial q}{\partial \eta} + C \frac{\partial q}{\partial \zeta} = 0 \]  

(C.10)

where \( M \) is the matrix \( \partial Q / \partial q \). Multiplying Equation (C.10) on the left by \( M^{-1} \) gives:

\[ I \frac{\partial}{\partial \tau} (J^{-1} q) + M^{-1} A \frac{\partial q}{\partial \xi} + M^{-1} B \frac{\partial q}{\partial \eta} + M^{-1} C \frac{\partial q}{\partial \zeta} = 0 \]  

(C.11)

where \( I \) is the identity matrix. Then from Equations (C.8) and (C.11) it is evident that:

\[ a = M^{-1} A, \quad b = M^{-1} B \quad \text{and} \quad c = M^{-1} C \]  

(C.12)

The matrices \( a, b \) and \( c \) are said to be similar to the matrices \( A, B \) and \( C \). The matrix \( M \) is the similarity matrix. With similar matrices having the same eigenvalues, the eigenvalues of the matrices \( A, B \) and \( C \) are known by determining the eigenvalues of \( a, b \) and \( c \). The matrices \( a, b \) and \( c \) are simpler to operate on than matrices \( A, B \) and \( C \) as they contain several zero elements as demonstrated below.

The matrices \( a, b \) and \( c \) are now determined. The matrix \( M \), defined as \( \partial Q / \partial q \), is:

\[
M = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\phi & u & v & w & \frac{1}{\gamma - 1} \\
\end{bmatrix}
\]  

(C.13)

The inverse of this matrix is obtained using the method described in Appendix B and is:

\[
M^{-1} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
\frac{1}{u} & \frac{1}{\rho} & 0 & 0 & 0 \\
\frac{1}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 \\
\frac{1}{\rho} & 0 & 0 & \frac{1}{\rho} & 0 \\
\frac{\phi}{\gamma - 1} & \frac{-u(\gamma - 1)}{\rho} & \frac{-v(\gamma - 1)}{\rho} & \frac{-w(\gamma - 1)}{\rho} \\
\end{bmatrix}
\]  

(C.14)
Then, using Equation (C.12), the matrices \( \mathbf{a}, \mathbf{b} \) and \( \mathbf{c} \) can be written in the generic form:

\[
\mathbf{P} = \begin{bmatrix}
\beta_k & \rho k, & \rho k_y & \rho k_z & 0 \\
0 & \beta_k & 0 & 0 & k_x \\
0 & 0 & \beta_k & 0 & k_y \\
0 & 0 & 0 & \beta_k & k_z \\
0 & k_x \rho c^2 & k_y \rho c^2 & k_z \rho c^2 & \beta_k
\end{bmatrix}
\]  

(C.15)

where \( \mathbf{P} \) is equal to \( \mathbf{a}, \mathbf{b} \) and \( \mathbf{c} \) for \( k = \xi, \eta \) and \( \zeta \) respectively. Note that the \( c \) in the above matrix is the speed of sound. The eigenvalues of the matrix \( \mathbf{P} \) are then the solutions of the characteristic equation:

\[
\det(\mathbf{P} - \lambda \mathbf{I}) = 0
\]  

(C.16)

The eigenvalues can easily be evaluated as:

\[
\begin{align*}
\lambda_k^1 &= \beta_k \\
\lambda_k^2 &= \beta_k \\
\lambda_k^3 &= \beta_k \\
\lambda_k^4 &= \beta_k + c|\nabla k| \\
\lambda_k^5 &= \beta_k - c|\nabla k|
\end{align*}
\]

(C.17)

where \( \lambda_k^1, \lambda_k^2, \lambda_k^3, \lambda_k^4 \) and \( \lambda_k^5 \) are the eigenvalues of \( \mathbf{a}, \mathbf{b} \) and \( \mathbf{c} \) for \( k = \xi, \eta \) and \( \zeta \) respectively, and:

\[
|\nabla k| = (k_x^2 + k_y^2 + k_z^2)^{1/2}
\]  

(C.18)