



Wind turbine aerodynamics using an incompressible overset grid method

Zahle, Frederik

Publication date:
2006

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Zahle, F. (2006). *Wind turbine aerodynamics using an incompressible overset grid method*. Imperial College of Science, Technology and Medicine.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

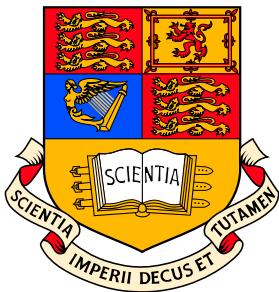
- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Wind Turbine Aerodynamics Using an Incompressible Overset Grid Method

by

Frederik Zahle



Department of Aeronautics
Imperial College of Science, Technology and Medicine
Prince Consort Road
London SW7 2BY

This thesis is submitted for the degree of Doctor
of Philosophy of the University of London

2006

Abstract

This thesis presents 3D Navier-Stokes simulations of unsteady flow around the two bladed NREL Phase VI wind turbine. The computations are carried out using the structured grid, incompressible, finite volume flow solver EllipSys3D, which has been extended to include the use of overset grids. To handle the relative movement between the rotor and the tower, the domain is decomposed into a number of topologically simple grids that overlap each other arbitrarily. Relative movement of bodies is thus possible without the need for re-meshing. Coupling between overlapping grids is achieved through non-conservative interpolation of the flow variables onto the internal overset boundaries. To satisfy the implicit incompressibility constraint of the pressure-correction equation an explicit correction of the mass fluxes is employed. The grid assembly and exchange of flow field information is carried out in a fully parallelised environment using MPI.

Results are presented on the isolated rotor, isolated tower, and on the downwind configuration of the turbine, which includes modelling of the rotor, tower and tunnel boundary layer. The results are compared to previously published computations as well as experimental data from the NREL UAE wind tunnel tests, and are shown to agree very well with both. It is demonstrated that the solver is capable of accurately capturing the unsteady interaction between the rotor and the tower for the downwind configuration. The aerodynamic response of the rotor is characterised by high transient loads on the blades that are strongly influenced by the occurrences of blade-vortex interaction. The rotor has a strong effect on the tower shedding, causing vortex lock-in to take place on the upper part of the tower. This lock-in is characterised by a synchronisation of the shedding frequency with the blade passage frequency and a spanwise correlation of the wake.

Acknowledgements

This research work was carried out mainly in the Wind Energy Department at Risø National Laboratory, Roskilde, Denmark combined with a number of stays at Imperial College under the joint supervision of Niels N. Sørensen and Jeppe Johansen from Risø together with Mike Graham from Imperial College in the time period from September 2003 to October 2006. The PhD programme was conducted under the Imperial College Public Research Institute - Industrial Research Laboratory (PRI-IRL) Scheme.

I would like to express my sincere gratitude to Niels Sørensen and Jeppe Johansen for providing me with professional and dedicated supervision throughout my three years as Ph.D. student. Likewise, I would like to thank Mike Graham for his efforts in making a Ph.D. at Imperial College possible, and for sharing his extensive knowledge on aerodynamics with me during my visits to London.

Lastly, I would like to thank my co-workers and friends in the Department of Wind Energy at Risø for the both academically and socially open and inspiring environment.

CONTENTS

Abstract	2
Acknowledgements	3
Contents	4
List of Figures	6
List of Tables	9
List of Algorithms	10
1 Introduction	11
1.1 Background and Motivation	11
1.2 The Present Study	14
2 Numerical Methods for Overset Grids	17
2.1 Introduction	17
2.2 Solution Method for the Base Solver	23
2.2.1 Finite Volume Equations	26
2.2.2 Boundary conditions	27
2.3 Solution Method on Overset Grids	29
2.3.1 Interface boundary conditions	32
2.3.2 Overset Grid Interpolation Scheme	33
2.3.3 Mass Flux Correction	34
2.3.4 Turbulence Modelling	36
2.3.5 Moving Grid Issues	37
2.4 Mesh Generation	38
2.5 Domain Connectivity	39
2.5.1 Hole-cutting	42
2.5.2 Fringe Cell Identification	44
2.5.3 Donor Cell Search	44
2.6 Code Parallelisation	45
3 Numerical Validation of the Overset Grid Method	51
3.1 Introduction	51
3.2 Flow over a Circular Cylinder	52
3.2.1 Computational Meshes	53
3.2.2 Steady Laminar Flow at $R_e=40$	54

3.2.3	Unsteady Turbulent Flow at $R_e=1.84 \times 10^5$	60
3.3	Flow Over the NACA 63 ₂ – 415 Aerofoil	68
3.3.1	Computational Meshes	68
3.3.2	Results and Discussion	69
3.4	Dynamic Stall of the NACA 0015 Aerofoil	74
3.5	Steady Flow Over a 6:1 Prolate Spheroid	77
3.5.1	Computational Meshes	78
3.5.2	Results and Discussion	79
3.6	Parallel Performance	80
3.6.1	Computational Approach	81
3.6.2	Results and Discussion	82
3.7	Conclusions	86
4	Wind Turbine Rotor-Tower Interaction	88
4.1	Introduction	88
4.2	Test Configuration	92
4.3	Computational Meshes	96
4.4	Computational Framework	99
4.5	Isolated Rotor Computation	100
4.6	Isolated Tower Computation	102
4.7	Rotor-Tower Computation	107
4.7.1	Computational Framework	107
4.7.2	Flow Field	107
4.7.3	Blade Response	112
4.7.4	Tower Response	127
4.7.5	Discussion	129
4.7.6	Computational Cost	134
4.8	Conclusions	136
5	Conclusions	138
5.1	Outlook	141
	Bibliography	144

LIST OF FIGURES

2.1	Example of an overset grid mesh configuration	19
2.2	Detailed view explaining the overset grid terminology	20
2.3	Overset grid interpolation stencil	34
2.4	Relative movement between overset grids	37
2.5	Example of 3D mesh configuration	38
2.6	Schematic of the boundary ray method.	43
2.7	Schematic of binary tree subdivision	44
3.1	Grid C2 cylinder mesh	54
3.2	Grid C3 cylinder mesh	55
3.3	Pressure and skin friction on cylinder	57
3.4	Solution residual reduction	57
3.5	Effect of flux correction on convergence rate	58
3.6	Time history of the non-dimensionalised divergence error	59
3.7	Flow contours on cylinder	60
3.8	$k - \omega$ SST turbulence model zonal functions.	63
3.9	2D cylinder unsteady wake deficit	64
3.10	Velocity flow contours on cylinder	65
3.11	Pressure gradient flow contours on cylinder	66
3.12	Vorticity flow contours on cylinder	67
3.13	Eddy viscosity flow contours on cylinder	67
3.14	Mesh configuration A2 and A3 for the NACA 63-415 aerofoil. .	69
3.15	Mesh configuration A4 for the NACA 63-415 aerofoil.	69
3.16	Solution dependence on overlap size	70
3.17	Convergence rate dependence on overlap size	71
3.18	Solution dependence on mesh coarseness	72
3.19	Solution dependence on mesh topology	72
3.20	Comparison of computations and experimental data	73
3.21	Hysteresis curves of flow over the NACA 0015 aerofoil	76
3.22	Hysteresis curves of flow over the NACA 0015 aerofoil	76
3.23	Comparison of overset boundary interpolation schemes	77
3.24	Instantaneous vorticity contours for $k = 0.154$	77
3.25	Grid S1 spheroid mesh	78
3.26	Spheroid C_f and C_p	79
3.27	Spheroid contour plots	80
3.28	Multibody spheroid mesh	82
3.29	Parallel speedup for the spheroid problem.	84
3.30	Load balance for the spheroid problem using 14 processors. .	85

3.31 Distribution of fringe cells for the spheroid problem	85
4.1 The NREL UAE Phase VI wind turbine	93
4.2 Pressure tap locations on the S809 profile	94
4.3 Planform of the NREL UAE Phase VI blade	94
4.4 Definitions of the aerodynamic force coefficients	94
4.5 Definitions of the frame of reference for the NREL experiment .	95
4.6 Definition of the local blade velocities.	95
4.7 Mesh for the rotor-only configuration.	97
4.8 Mesh for the rotor-tower configuration.	98
4.9 Side and back view of the rotor-tower mesh	99
4.10 Detail of the rotor and tower grids.	100
4.11 C_N and C_T on isolated rotor for $w=7$ m/s	101
4.12 C_p on isolated rotor for $w=7$ m/s	102
4.13 Iso-vorticity plot for isolated rotor	103
4.14 Isolated tower C_x at various heights	104
4.15 Isolated tower mean C_p and C_f at $y=8.17$ m	105
4.16 Isolated tower wake axial velocity profile	105
4.17 Isolated tower wake tangential velocity profile	105
4.18 Isolated tower iso-vorticity plot	106
4.19 Isolated tower vorticity plot	106
4.20 Rotor-tower iso-vorticity plot	108
4.21 Rotor-tower vorticity contour plot	109
4.22 Rotor-tower axial velocity contour plot	109
4.23 Rotor-tower BVI type 1	110
4.24 Rotor-tower BVI type 2	110
4.25 Rotor-tower vorticity plot	111
4.26 Blade 1 C_N	114
4.27 Blade 2 C_N	114
4.28 Blade 1 C_T	115
4.29 Blade 2 C_T	115
4.30 Blade 1 probe velocity traces at 34% and 84% span	117
4.31 Blade 2 probe velocity traces at 34% and 84% span	117
4.32 Blade 1 probe LFA at 34% and 84% span	118
4.33 Blade 2 probe LFA at 34% and 84% span	118
4.34 Aerodynamic 2D force coefficients for the S809 aerofoil	119
4.35 Detail of blade 1 C_N , C_M and C_p at at 80% span	121
4.36 Detail of blade 1 C_N , C_M and C_p at at 80% span	121
4.37 Surface contour plot of C_p on blade 1 at 80% span	123
4.38 Surface contour plot of C_p on blade 2 at 80% span	123
4.39 Comparison of computed and experiment C_N	124
4.40 Blade 1 computed and experiment C_N , C_T and C_M	126
4.41 Blade 2 computed and experiment C_N , C_T and C_M	126
4.42 Computed and experiment LFA at 84% span	127
4.43 Blade 1 computed and experiment LFA at 84% span	127
4.44 Blade 2 computed and experiment LFA at 84% span	127
4.45 Tower C_X at various heights	128
4.46 Frequency content of tower C_X	129

LIST OF TABLES

3.1	Spatial convergence rate	56
3.2	Steady flow cylinder recirculation length	56
3.3	Divergence error for the three grid levels for grid C2	59
3.4	Dependence on overset interpolation scheme	61
3.5	Temporal convergence rate	62
3.6	C_l , C_d and St on cylinder	64
3.7	Variation of integrated divergence error with overlap size	71
3.8	Spheroid lift and drag coefficients	79
3.9	Computational cost (wall-clock time) of spheroid problem.	83
4.1	Wall-clock timings of rotor-tower problem	135

LIST OF ALGORITHMS

1	Search routine: locate a point in general curvilinear grid	46
2	Connect0G : Top level of connectivity routine	48
3	Connect0Gparallel : Parallel handling of search requests	49
4	Exchange3D.f : Asynchronous communication of data	50

1.1 Background and Motivation

The aerodynamics of wind turbines is highly complex due to its inherently unsteady, three-dimensional nature. A turbine will generally operate in conditions where the flow is dominated by separation towards the inner sections of the blade whereas it is largely two-dimensional on the middle sections of the blade. The three-dimensionality of the flow is rooted in the effects of finite span of the blade as well as rotational effects. The former gives rise to a loss of loading near the root and tip, whereas the latter accounts for an additional large degree of spanwise flow near the root of the blades. This effect has been found to significantly augment the aerodynamic loading in the root sections by delaying stall due to centrifugal and Coriolis boundary layer effects. Additionally, effects such as dynamic stall due to the shear of the atmospheric boundary layer, yawed inflow conditions and atmospheric turbulence increase the complexity of wind turbine aerodynamics.

There exists an extensive body of experimental work on non-rotating wing sections, since the cost and complexity of establishing such experiments is relatively low. There are, however, very few experiments done on wind turbine rotors where it has been possible to investigate the complex three-dimensional effects discussed above . It is indeed in this respect that Computational Fluid Dynamics (CFD) has become an increasingly invaluable tool for improving the understanding of wind turbine aerodynamics since it provides less costly and less time consuming means of investigating wind turbine rotor flows. Although CFD is by no means mature enough to replace experimental aerodynamics, it has been shown to be very useful both with respect to the understanding of the basic aerodynamic properties of rotating blades and the development of new blade designs and it has, in recent years, been subject to a significant increase of interest from both the research community and the industry.

Engineering design codes based on the Blade Element Momentum theory (BEM) are predominantly used in the design of wind turbine blades, since they provide very accurate and fast predictions for traditional rotor flows. Their main drawback is their inherent dependence on accurate aerodynamic coefficients, which are usually derived from 2D wind tunnel measurements on blade sections. To account for the three-dimensionality of the flow various empirical corrections have to be made, which are well-established for traditional rotor configurations. However, the continuous increase in the demand of aerodynamic efficiency has led to a degree of complexity of blade designs which in certain areas lies beyond the capabilities of these codes.

For flows dominated by a large degree of unsteadiness or complex effects due to new blade designs, CFD is a much more comprehensive tool. As discussed above, the aerodynamic coefficients on a wind turbine blade differ significantly from two-dimensional force coefficients extracted from experimental data. CFD is therefore increasingly being used to provide design codes with more accurate data and it therefore acts as an important complement to these codes since it can be used to increase their accuracy.

In CFD, the flow solution is calculated using the Navier-Stokes equations. However, due to the high cost of resolving all length and time scales in the flow by Direct Numerical Simulation (DNS), high Reynolds number flows are typically solved using the Reynolds-averaged formulation of the Navier-Stokes equations (RANS) which arises from the decomposition of the velocity field into a mean flow component, which describes the large scale time dependency of the flow, and a fluctuating component which accounts for the small-scale turbulent fluctuations in the flow. The recast of the Navier-Stokes equations gives rise to a new term known as the Reynolds stress term. This term is modelled using an appropriate turbulence model. It is indeed the modelling of this term which is the main drawback of RANS methods, since it introduces a degree of empiricism into the equations describing the flow. The most common class of models known as eddy-viscosity models are generally recognised as not to be fully adequate for describing large scale separation since the small scale fluctuations which dominate these types of flows are not accounted for appropriately. Additionally, the modelling of transition between laminar and turbulent flow is a major source of uncertainty in CFD, where models to a large extent also rely on empirical models. With the increase in computational power, methods commonly referred to as Large Eddy Simulation (LES) are becoming increasingly popular, since higher proportions of the flow field

(namely the large fluctuating scales) are simulated directly, although it still relies on a turbulence model that describes the sub-grid scales of the flow. These models have been shown to produce a much better representation of the three-dimensionality inherent to complex flows, but are still only of limited use due to the lack of computational power. An alternative to LES is Detached Eddy Simulation (DES) [62], which is a combination of RANS and LES, where only large eddies in regions of separated flow are simulated, whereas the remaining parts of the flow are modelled using RANS with an appropriate turbulence model.

The main drawback of CFD is the considerable computational cost of achieving an acceptable resolution of the flow. For this reason, simulation of wind turbine aerodynamics has so far mostly been limited to modelling the isolated rotor [14, 29, 36, 61, 76, 77], which meant that the effects of the tower and nacelle were ignored. Only one simulation of a complete turbine has been published by Duque et al. [15] who used overset grids to handle the relative movement between the rotor and tower. Additionally, the influence of the atmospheric boundary layer shear, inflow turbulence and the terrain in which it is situated have not been included in the modelling. It is, however, commonly known that all of these effects have a profound impact on the performance and structural loads of the turbine. The effect of the tower has a fairly limited significance on a turbine where the rotor is placed upwind of the tower, whereas on a downwind turbine the rotor interacts strongly with the unsteady tower wake. This interaction has consequences for both the aeroelastic responses of the turbine as well as its acoustic signature. A detailed understanding of the interaction between the rotor and the tower could help to improve the engineering tower shadow models used at present, and possibly also provide the grounds for mitigation of some of the critical issues such as noise and high transient loads associated with rotor-tower interaction.

The modelling of an entire wind turbine poses a number of challenges to the flow solver, most notably that the relative movement between the rotor and tower would have to be handled in some appropriate manner. A number of numerical methods are available for solving this problem: deforming structured and unstructured meshes can be used for relatively small movements, whereas for large relative motions of bodies, re-meshing would be necessary at certain intervals, which would be very time consuming. Sliding meshes is a more simple alternative, which requires the meshes that move relative to each other to have surfaces in common on which the flow solution is ex-

changed between the two grids. This method can be made conservative and is perhaps the least computationally expensive of the methods available. The main drawback of this method is that it poses restrictions on the grid topologies used. Another alternative is the overset grid method, also known as the Chimera or composite grid method, which is well-proven and widely used in the aerospace industry for both compressible and incompressible flows, see for example [5, 38] and [11, 12, 24, 25, 68, 79]. This method gives a high degree of flexibility in the grid generation process, since there is no restrictions on grid topologies used. Additionally, bodies can move unrestricted relative to each other as long as the domain is resolved appropriately. The main drawback of this method is that it can be computationally quite costly, particularly in a parallel environment. A more recent alternative is the so-called Dragon grid [80], where regions between component grids are filled with a thin layer of unstructured grids. When bodies move relative to each other these grids have to be re-meshed at certain intervals. This method thus eliminates the need for re-meshing the entire domain, and allows for the use of a combination of unstructured and structured grids of arbitrary topology.

Through a collaboration started in the nineteen nineties between Risø National Laboratory and the Danish Technical University, the two- and three-dimensional incompressible Reynolds-averaged Navier-Stokes flow solvers EllipSys2D and EllipSys3D [42, 43, 59] were developed to further the research in aerodynamics of complex flows. The codes have been used on a large variety of flow problems, which at Risø has been focused around the field of wind turbine aerodynamics. It is the vast experience with this code and the extensive knowledge of wind turbine aerodynamics at Risø that forms the basis for the present work which will be described next.

1.2 The Present Study

In the EllipSys solvers it has so far only been possible to compute the flow around an isolated rotor. As part of the continuous development of the EllipSys codes it was at Risø National Laboratory therefore decided that the codes should be extended such that this relative movement could be modelled directly providing the basis for simulating the unsteady flow over a complete wind turbine configuration including the rotor, tower, nacelle and ground boundary.

A requirement of the method of choice to solve problems involving relative

grid movement was that it had to be implementable in the existing solver. Although sliding grids would be able to handle this problem, it was decided that the higher degree of flexibility of the overset grid method was desirable when solving the flow over a wind turbine.

The principal aim of the present study has been to implement the overset grid method in EllipSys2D and EllipSys3D for the computation of unsteady moving grid problems with the focus on wind turbine aerodynamics.

One challenge, which has so far not been fully met by the wind turbine research community has been to model the unsteady interaction between the rotor and the tower on a downwind turbine. Traditionally, this problem has been investigated under the assumption that the tower wake consisted of a mean flow deficit. It is, however, well-known that this is not adequate since the unsteady nature of the tower wake induces high transient loads on the blades as they pass through the wake.

The second aim of this thesis has therefore been to use the implemented method to provide new insight into the unsteady interaction between the rotor and tower on a downwind turbine, identifying the key physical phenomena that underlie this complex problem, which, to the best knowledge of the author, has not been studied to this degree of detail before.

The thesis is divided into three main chapters:

Chapter 2 discusses the numerical implementation of the overset grid method in the EllipSys2D/3D solvers. A detailed description of solution schemes, boundary conditions, establishment of grid connectivity and parallelisation strategies will be given.

In Chapter 3 a systematic validation of the overset grid method is presented. This involves an investigation of the spatial and temporal order of accuracy of the solver, as well as comparisons to non-overset solutions and experimental results to demonstrate the overall accuracy of the solver for both stationary and moving grid problems. Additionally, the parallel performance of the code will be evaluated and discussed.

Finally computations of wind turbine aerodynamics are presented in Chapter 4. Here the NREL Phase VI turbine will be used as a test case, since extensive experimental and computational results exist for this turbine, making it ideal for validation. The focus in the chapter is on the downwind configuration of the turbine and it will be shown that the overset grid version of EllipSys3D

is fully capable of accurately predicting the complex interaction between the rotor and the tower.

In the last chapter the main findings will be summarised and future areas of research inspired by the present work will be suggested.

2

NUMERICAL METHODS FOR OVERSET GRIDS

2.1 Introduction

The use of structured curvilinear grids has proved to be a very efficient way of resolving thin shear layer flows, which are often encountered in engineering and environmental flows. The ability to solve the Navier-Stokes equations implicitly, and the use of parallel computing has greatly reduced the computational time required to solve these types of problems. The main disadvantage of structured solvers has always been the grid generation, due to requirements of orthogonality and smoothness. Problems involving complex geometries or multiple bodies have thus traditionally been handled using unstructured solvers. These solvers, however, are computationally more expensive and memory intensive, due to the data structures required to handle the unstructured grids. Furthermore, unstructured grids are not as efficient at resolving thin shear layers.

The overset grid method, also known as chimera or composite grid method, addresses many of the limitations of traditional structured grid methods, while at the same time maintaining most of their advantages such as implicit solution strategies and discretisation schemes as well as natural integration into a parallel computing environment. The key advantage of the method is that relative movement of grids can be handled without regeneration of the grid. The domain is decomposed into a number of simpler grids, which overlap each other arbitrarily making the grid generation process considerably easier. Each overset domain can be solved independently using existing structured solvers provided that appropriate boundary conditions are prescribed on overset interfaces. Boundary conditions are based on interpolation of flow variables from neighbouring domains. The interpolation is usually done in an explicit non-conservative manner, which raises the issue of conservation of quantities such

as mass and momentum. This is of particular concern with the incompressible Navier-Stokes equations where conservation is required for the solution of the pressure-Poisson equation to converge and the flow variables to be represented smoothly. The lack of conservation due to the interpolation thus has to be addressed in some manner.

Appropriate body-fitted grids are generated independently around each component, reducing grid distortion towards fixed boundaries. Important flow features can be resolved by refined Cartesian meshes, which together with the body fitted meshes can be embedded in Cartesian background meshes that are successively coarsened towards the farfield where there is no need for high resolution of flow features. To accommodate the solid bodies and Cartesian refinement grids, cells are removed from the background grids where necessary. Since each component is independent of the other components, problems involving multiple bodies with relative movement are naturally handled without the need of re-meshing. The method is therefore widely used for aerospace applications, where problems typically involve a high degree of geometric complexity as well as moving grid problems such as store separation and helicopter aerodynamics. The method is also being used in other fields such as biological and industrial flows where a similar problem complexity is present.

An example of an overset grid system is the 2D multibody configuration featuring an aerofoil downstream of a cylinder depicted in Figure 2.1. The mesh consists of three independently generated grids; two curvilinear meshes for the aerofoil and cylinder, and one Cartesian mesh resolving the rest of the computational domain. Each of these component grids is referred to as a *block group*.

Figure 2.2 shows a detailed view of two overset grid blocks from the grid system showed in Figure 2.1. In a standard grid system without overset boundaries, cells are classified as either *internal discretisation cells*, *physical boundary cells* or *ghost cells*. Ghost cells are used to apply the boundary conditions and to handle the block-block communication in the multiblock grid topology. In an overset grid system two new types of cells are defined: *hole cells* and *fringe cells*. *Hole cells* are cells in the computational domain that are removed from the solution because they lie inside a physical body or are otherwise not needed in the domain. *Fringe cells* are cells along the outer boundaries of individual block groups, or cells along internal boundaries which arise as a result of the hole-cutting. Flow variables are interpolated to the fringe cells from the *donor cells* in the neighbouring grid. Cells qualify as donors if these cells themselves

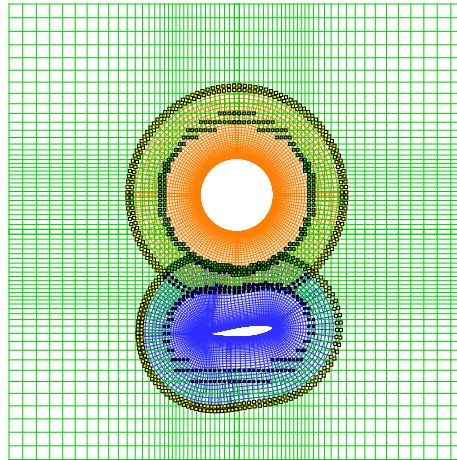


Figure 2.1: Overset grid mesh configuration consisting of three block groups; two curvilinear groups and one Cartesian group. Scatter symbols indicate fringe cells for each group. Scatter symbol colour refers to donor grid identity: Cylinder mesh: green; aerofoil mesh: blue; Cartesian mesh: yellow.

are not fringe cells, hole cells or cells on any physical boundary, that is, they have to be ordinary internal cells. The grid containing the fringe cell is commonly referred to as the *receptor grid* whereas the grid containing the donor cell is referred to as the *donor grid*. The process of locating donor cells is commonly referred to as the grid assembly process. In Figure 2.1 the colour of the scatter symbols refer to the identity of the donor grid.

Using relatively simple mesh topologies it is thus possible to construct meshes for multibody problems, which with conventional structured mesh generation would have required very careful meshing to avoid skewness of the grid. Relative movement of bodies is implemented in the method without needing to regenerate or distort the grid at every time step. When two bodies move relative to each other the only addition to the solution process is that connectivity information has to be updated and holes re-cut at every time step. The component oriented meshing also makes it easy to modify only parts of the grid while leaving the majority of the mesh system untouched during a design process.

With the added flexibility also comes a high cost associated with the need for separate data structures containing information about connectivity of the overset groups and the communication of flow-field data across overset interfaces. Routines are needed for handling the automated process of grid assembly and interpolation of flow field data. Large scale problems have to be

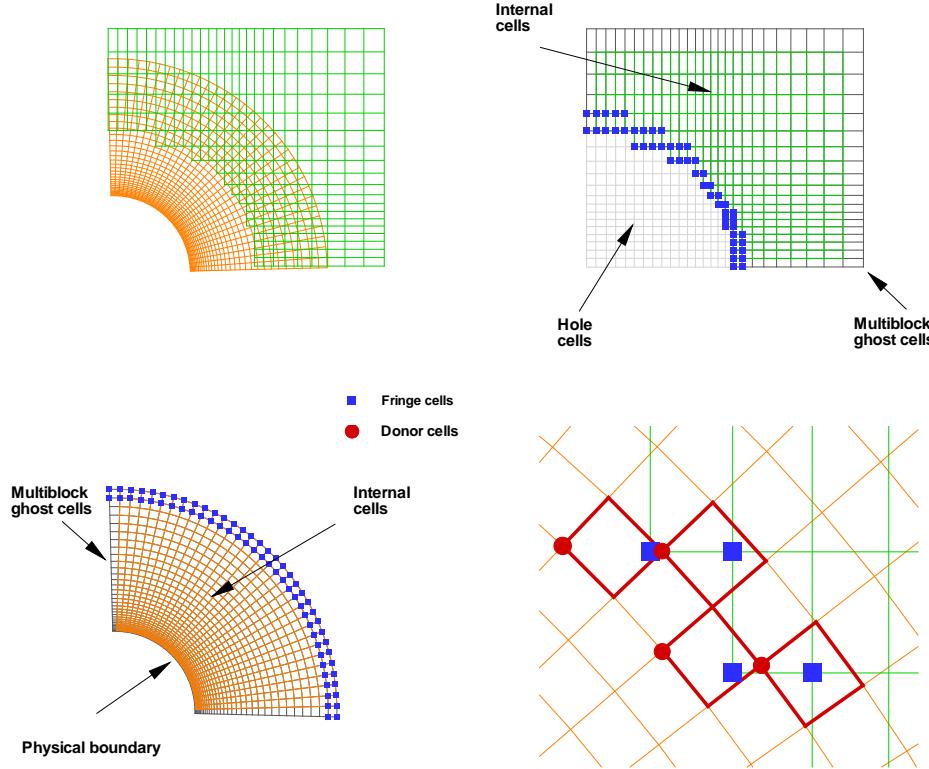


Figure 2.2: Detail of the grid configuration explaining the terminology used in the overset grid method.

solved in a parallel computing environment. For the overset grid method to be efficient it therefore has to be integrated in this parallel environment. Due to the irregular overset boundaries, there will almost inevitably be an imbalance in the data flow between the overset groups, which in a parallel environment is detrimental and can cause the method to be inefficient unless appropriate load balancing strategies are employed.

Over the past three decades much work has been done on overset grid methods. Computations on overset grids were first done for Euler solutions and later on the fully compressible Navier Stokes equations. Among the first to work with this type of solution methods were Atta [2], Steger et al. [63] and Benek et al. [5], the latter starting the development of the well-known OVERFLOW code. This code was later developed into the OVERFLOW-D solver for solution of moving body problems by Meakin [38, 40]. Since the present work deals with the implementation of the overset grid method in an incompressible solver, the following literature overview will focus solely on the main contributors to the incompressible overset grid field. In a later section,

the main contributors to the subject of overset domain connectivity will be discussed.

Rogers et al. [53] presented an incompressible overset grid capable flow solver formulated in the so-called pseudo-compressible formulation. With the pseudo-compressible formulation, strict conservation of mass is not necessary. A non-conservative linear interpolation was therefore used to impose Dirichlet boundary conditions at the overset interfaces using interpolated variables from neighbouring grids.

Chesshire and Henshaw [12] and Henshaw [22] developed a pressure-based composite grid multigrid method to solve the 3D unsteady incompressible Navier-Stokes equations using either a second or fourth-order accurate method. The inter-grid communication of all flow variables u, v, w , and p was done using Lagrange interpolation. In [12] it is showed that for the most general case of an overset grid the order of accuracy of the inter-grid interpolation should be at least one higher than that of the overall solution scheme in order to preserve the order of accuracy of the solution. Therefore, either a third or a fifth-order interpolation scheme was employed. A damping term of the discrete divergence is added to the pressure-correction equation for stability purposes which is proportional to the local cell volume. On the intergrid boundaries Dirichlet conditions are used on the velocities and Neumann conditions are used for the pressure-correction equation. In [23] Henshaw discusses the issues related to the multigrid solution of overset grids. In the method the solution is coupled by interpolation at all multigrid levels. Most other authors choose not to make this coupling, mostly because of the difficulty in doing so. The main advantage of this method is that additional steps can be taken to locally smooth the errors near the interpolation boundaries when solving the pressure-correction equation and thus improve the convergence rates significantly.

Hinatzu and Ferziger [24] investigated the overlapping grid method using both 1D and 2D model problems as well as unsteady flows in complex domains. For the inter-grid interpolation, they used a Coons patch method which is second-order accurate. The authors introduce the terms *incomplete composite multigrid*, ICMG, and *complete composite multigrid*, CCMG. In the ICMG method, interpolation of the flow variables is only done on the finest multigrid level, whereas in the CCMG method communication is done at all levels. However, the method used does not allow for efficient parallelisation, because the grids are solved in a serial manner. The authors emphasise that the ICMG produced accurate solutions for the problems solved, whereas the

CCMG was more costly and a less robust algorithm. The convergence rate was not improved using CCMG.

Hubbard and Chen [25] solve laminar flow over a cylinder and turbulent flow over a submarine configuration. The chimera grid scheme is implemented in an incompressible flow solver, which uses a SIMPLER/PISO algorithm. On overset interfaces the pressure and velocities are specified via interpolation from neighbouring domains. The interpolated contravariant velocities are corrected in each boundary cell such that conservation of mass is maintained before the pressure equation is solved.

Zang and Street [79] developed a second order accurate, fractional step, finite volume, multigrid method for the 3D unsteady incompressible Navier-Stokes equations. They used third-order Lagrangian, bi-quadratic interpolation of the velocities and fluxes between subgrids, which is non-conservative. A correction proportional to the local volume flux is made to the fluxes after interpolating the velocities between subgrids. This ensures that mass conservation is obtained in each domain such that the discrete pressure Poisson equation can be solved. The authors demonstrated that the scheme was second order accurate. On the inter-grid boundaries Dirichlet boundary conditions were used for the velocities and Neumann boundary conditions for the pressure. It was shown that prescribing pressure values as opposed to pressure gradients produced inconsistent solutions.

Tu and Fuchs [68] present an overlapping grid method applied to internal combustion engines. Lagrange interpolation is used for the inter-grid interpolation at the finest grid level and a flux correction is applied to the appropriate velocity components to ensure continuity in the sub-grids.

Burton and Eaton [11] present a fractional step method, which is derived as a matrix factorisation of the spatially and temporally discretised system of equations in general curvilinear coordinates in a staggered grid system. Non-conservative quadratic interpolation is used to interpolate the velocities between sub-grids. A similar correction to that presented by Zang and Street [79] is made to the interpolated velocity components on the inter-grid boundaries to ensure mass conservation such that the discrete continuity equation can be solved. Since they solve the system using a staggered configuration no essential boundary conditions are needed for the pressure, only a numerical Neumann boundary condition is imposed.

Tang [65], Tang et al. [66] present an overset grid method implemented in an incompressible Navier-Stokes solver using the artificial compressibility

method. The velocity components are interpolated non-conservatively from background domains using a so-called mass flux based interpolation (MFBI), where the fringe cell velocities are corrected according to a *local* consideration of conservation. With this strategy conservation cannot be guaranteed since it is not a *global* consideration of conservation. The MFBI approach can more appropriately be thought of as a relaxation of the Dirichlet boundary condition, since the velocity components installed in fringe cells are constructed using information from the previous pseudo-time iteration. A Dirichlet boundary condition is used for the pressure, which in the light of the non-conservative treatment of the velocities is not appropriate for the solution of the incompressible Navier-Stokes equations.

In this chapter the numerical methods used to decompose an incompressible Navier-Stokes solver on non-matching overlapping grids are presented. The method is implemented in the flow solver EllipSys3D by Michelsen [42, 43] and Sørensen [59]. Firstly, the base numerical solution method is presented. This is followed by a description of the modifications needed to integrate the overset method in the code which entails boundary conditions, flux correction scheme and interpolation schemes used. Subsequently, the methods used to identify invalid (or hole points) will be discussed accompanied by a presentation of the search routine used to identify donor cells. The method is implemented in a parallel computing environment, where all processors perform the same tasks simultaneously. To achieve good scalability of the code it is essential to employ efficient strategies of communication between processors. These issues are addressed in the latter parts of the chapter.

2.2 Solution Method for the Base Solver

The governing equations for the flow solver are the three-dimensional incompressible Reynolds Averaged Navier-Stokes equations, which written in Cartesian coordinates take the form

$$\frac{\partial}{\partial x_j}(\rho U_j) = 0 , \quad (2.1)$$

$$\frac{\partial}{\partial t}(\rho U_i) + \frac{\partial}{\partial x_j}(\rho U_i U_j) - \frac{\partial}{\partial x_j} \left[(\mu + \mu_t) \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \right] + \frac{\partial P}{\partial x_i} = S_{\mathbf{v}} , \quad (2.2)$$

$$\frac{\partial}{\partial t}(\rho\varphi) + \frac{\partial}{\partial x_i}(\rho U_i \varphi) - \frac{\partial}{\partial x_i} \left[\left(\mu + \frac{\mu_t}{\sigma_\varphi} \right) \frac{\partial \varphi}{\partial x_i} \right] = S_\varphi , \quad (2.3)$$

where

The Boussinesq approximation, also called the eddy-viscosity technique is

$U_{i=1,2,3}$	Cartesian velocities,
P	pressure,
ρ	density,
μ	molecular dynamic viscosity,
μ_t	eddy viscosity,
S_v	momentum source term,
Φ	scalar quantity,
S_φ	scalar source term.

used to close the problem, which requires modelling of the eddy viscosity using an appropriate turbulence model.

The difficulty in solving the incompressible Navier-Stokes (INS) equations numerically lies in the fact that there is no direct method for determining the pressure. A coupling between the velocity and pressure can be obtained by combining the momentum equations and the continuity equation to obtain a Poisson equation for the pressure. The combined momentum/pressure Poisson equations can then be solved implicitly using a predictor/corrector method where a divergence free solution is obtained iteratively. A commonly used method is the SIMPLE method by Patankar and Spalding [48]. Here, an equation is constructed named the *pressure-correction* equation.

In a single iteration firstly the momentum equations are solved as a predictor step using a guessed pressure, which will yield a velocity field that will not fulfil the continuity equation. The pressure-correction equation is then solved to correct the velocity/pressure field such that it will satisfy continuity. The new field, however, will not satisfy the momentum equations which is why an iterative process is needed where the newly obtained pressure field is used as a guess in the momentum equations. For the SIMPLE algorithm to converge an under-relaxation is needed on both velocities and pressure. Typically, these values are 0.8 for the velocity and 0.2 for the pressure. If the problem is solved in time, a variation of the SIMPLE method is used called the PISO method by Issa [27], where the pressure-correction equation is solved twice. In one time step a number of subiterations are performed to reach a steady state. typically, four to six such predictor/corrector steps are performed within each time step, before a flow field is obtained that satisfies the governing equations.

The PISO method also requires some under-relaxation, with values for the velocity of around 0.8 and 0.2 and 0.9 for first and second pressure correction iterations. The methods are outlined below:

1. Guess the pressure field p^* .
2. Solve the momentum equations for u^* , v^* and w^* .
3. Calculate mass fluxes.
4. Solve the pressure correction equation for p' .
5. Correct p^* by adding the under-relaxed value of p' to obtain p .
6. Correct u^* , v^* and w^* and corresponding fluxes to obtain a divergence free field.
7. If using the PISO algorithm return to 3 using new pressure and fluxes and correct these again (only one loop).
8. Solve for other active scalars such as turbulent quantities.
9. Return to 2 and use corrected pressure and velocities as initial and repeat until sufficiently converged.
10. Advance solution in time and return to 2.

The EllipSys3D code uses the above outlined method to solve the incompressible Reynolds averaged Navier-Stokes (RANS) equations. The code is discretised using the finite volume method, solving for the primitive variables u, v, w , and p , in general curvilinear coordinates. The variables are stored in a collocated grid arrangement, and odd/even pressure decoupling is avoided using the Rhie-Chow interpolation [52].

The momentum equations are solved decoupled from each other using a red/black Gauss-Seidel point solver. The convective terms are discretised using either the second order upwind difference scheme, SUDS [78], or the third order accurate quadratic upstream interpolation for convective kinematics scheme, QUICK [34], and the diffusive terms are discretised using a central difference scheme. The solution is advanced in time using a second-order accurate, three-point backward scheme.

A multiblock decomposition of the domain allows for an efficient parallelisation of the code. A simple geometric decomposition of the solution domain is implemented where each subdomain or block is of equal size and logically cubic. The block-block communication is done through one layer of ghost cells around each block, where second order accuracy is maintained. The cell vertices are required to coincide on interfaces such that conservation is readily maintained. To accelerate the convergence of the pressure-correction equation

a multigrid solution strategy is implemented using a five-level additive Schwarz method, which has the advantage of having no sequentiality in the solution process. The coarse grid problem is solved on one processor and subsequently distributed to all other processors. Combined with a grid sequencing strategy of the outer iteration loop the solution time is reduced significantly.

The turbulence models used in EllipSys will be discussed in Section 2.3.4.

2.2.1 Finite Volume Equations

Momentum Equations

The generic algebraic formulation used to linearise the momentum equations has for the U -momentum the form:

$$A_P U_P^{t+\Delta t} + \sum A_{nb} U_{nb}^{t+\Delta t} = S_{U-mom}, \quad (2.4)$$

where standard compass notation is used to describe the control volume with P as the central node and neighbour cells nb labelled as E, W, S, N, B, and T. This equation can be expanded as follows:

$$\begin{aligned} A_P U_P^{t+\Delta t} + A_W U_W^{t+\Delta t} + A_E U_E^{t+\Delta t} + A_S U_S^{t+\Delta t} + A_N U_N^{t+\Delta t} \\ + A_B U_B^{t+\Delta t} + A_T U_T^{t+\Delta t} = S_T + S_V + S_F + S_P + S_C, \end{aligned} \quad (2.5)$$

where the following definitions are used:

$$\begin{aligned} A_W &= A_W^{dn} + A_W^c, \\ A_E &= A_E^{dn} + A_E^c, \\ A_S &= A_S^{dn} + A_S^c, \\ A_N &= A_N^{dn} + A_N^c, \\ A_B &= A_B^{dn} + A_B^c, \\ A_T &= A_T^{dn} + A_T^c, \\ A_P &= -A_W - A_E - A_S - A_N - A_B - A_T + \frac{1.5\rho_P J_P}{\Delta t}, \\ S_T &= \frac{\rho_P J_P(2U_P^t - 0.5U_P^{t-1})}{\Delta t}, \\ S_V &= 0, \\ S_F &= -I_e^{dc} + I_w^{dc} - I_n^{dc} + I_s^{dc} - I_t^{dc} + I_b^{dc}, \\ S_P &= -I_e^P + I_w^P - I_n^P + I_s^P - I_t^P + I_b^P, \\ S_C &= \text{explicitly treated terms for convection due to higher order sche} \end{aligned} \quad (2.6)$$

where

- A^{dn} normal-diffusion coefficients,
- A^c convective coefficients,
- S_T unsteady source term,
- S_F cross diffusion source term,
- S_P pressure source term.

The equations for the V - and W -momentum equations can be written in the same manner. For further details of the implementation the reader is referred to Sørensen [60].

Pressure-Correction Equation

The corrected pressure needed to obtain a divergence free velocity field can be expressed as

$$p = p^* + p' \quad (2.7)$$

where p^* is the guessed pressure field and p' is the pressure-correction. The corresponding velocity corrections can be written in the same form:

$$u = u^* + u' \quad v = v^* + v' \quad w = w^* + w' \quad (2.8)$$

Using the above definitions an expression can via the momentum equations be derived for the corrected velocity field in terms of the pressure-correction. The pressure-correction equation arises from the substitution of the velocity-correction formulae into the continuity equation. The general finite volume formulation for the pressure-correction equation takes the form:

$$A_P P_P^c = - \sum A_{nb} P_{nb}^c + S_{mass}, \quad (2.9)$$

The derivation of the individual influence terms will not be reproduced here, but can be found in most standard text books on the subject, see for example Patankar [47] and Ferziger and Perić [17].

2.2.2 Boundary conditions

To impose physical boundary conditions in the domain, the most common practise is to introduce fictitious or ghost cells along the boundaries, which hold the necessary information for a given boundary condition. In EllipSys3D ghost cells are also used in conjunction with the multiblock decomposition of

the domain where the ghost cells coincide with the internal cells in the adjacent blocks. Flow variables from adjacent blocks are installed, and, since the interfaces are point-wise continuous, continuity will be maintained. Additional numerical boundary conditions are needed to solve the pressure-correction equation. On internal multiblock interfaces, a so-called Robin boundary condition is prescribed.

For the solution of the INS equations two types of physical boundary conditions can be prescribed. Either the velocities normal to the boundary are given and the pressure unknown, or vice versa with a given pressure and unknown velocities. On inlets and walls, the former is typically used, since for example in most problems the inflow velocities are known. When the flux across all boundaries are known, there can be no flux correction on the boundary, which means that the gradient of the pressure correction across the boundary face must be zero. This corresponds to a Dirichlet condition on the momentum equations and a Neumann boundary condition on the pressure.

On outlets, the velocity is extrapolated across the boundary with a zero gradient condition. With this condition, global mass conservation cannot be guaranteed, which requires the flux on the outlet to be adjusted to match the flux across the inlet. As above, a given flux results in a Neumann boundary condition on the pressure.

With reference to the finite volume formulation of the momentum equation in Equation 2.5, the discrete coefficients for the momentum equations are modified as follows on inlets, taking an east cell face as an example for the U-momentum equation:

$$\begin{aligned} A'_P &= A_P - A_E \\ S'_\phi &= S_U - 2A_E U_e \\ A'_E &= 0 \end{aligned} \tag{2.10}$$

Likewise, the discrete coefficients for the pressure-correction equation are modified such that (again shown for an east face)

$$\begin{aligned} A'_P &= A_P + A_E \\ S'_\phi &= S_\phi - A_E \left(\frac{\partial \phi}{\partial n} \right)_e \\ A_E &= 0 \end{aligned} \tag{2.11}$$

Since a collocated grid arrangement is used, the pressure must be known

at all boundaries to correctly calculate the pressure flux in the momentum equations. A consistent method for obtaining this pressure would be to enforce a normal momentum balance on the boundary and obtain the pressure in the ghost cell. A simpler and very good alternative to this, is to extrapolate the pressure from interior cells using a second order extrapolation. The pressure on an east face is thus given by

$$P_e = \frac{1}{2} (4P_P - 3P_W + P_{WW}) \quad (2.12)$$

The coefficients of the extrapolation correspond to an extrapolation to the ghost cell E with a subsequent interpolation to the face e . The pressure on the boundary is subsequently extrapolated to the ghost cells using a second order extrapolation

$$P_E = 2P_e - P_P \quad (2.13)$$

Since a Neumann boundary condition is used for the pressure on all boundaries, the absolute value of the pressure cannot be determined. This is of no consequence in the solution of the INS equations, since only gradients of the pressure enter the equations. Typically, the initial guess of the pressure is set to zero in a suitable point in the domain.

2.3 Solution Method on Overset Grids

A decomposition of the solution domain into a number of non-matching overlapping sub-domains is possible with no alteration of the basic solution procedures. Each subdomain can be solved independently from others as long as appropriate boundary conditions are specified on internal non-matching interfaces. This could involve, as described in Section 2.2.2, a specification of velocities or pressure (*not* both as this will be discussed subsequently).

The difficulty with overset methods arises from the fact that there is no straight forward method for conserving mass across non-matching interfaces. A conservative method requires that the prescribed velocities and fluxes satisfy continuity. A way to ensure this is by considering the geometry of the local donor cell, making a piece-wise construction of the fluxes across each face, dividing faces intersected by the face of the fringe cell. A procedure for such a method is described by Wright and Shyy [74] for the solution of the INS equations using a SIMPLE-like method on overlaid Cartesian grids. The

implementation of this method in arbitrary 3D geometries, however, is hard to generalise and could be very time consuming particularly in a moving grid computation where connectivity has to be recomputed at every time step.

A non-conservative treatment of flow variables on interfaces is easily implemented since standard interpolation methods such as Lagrange or trilinear interpolation techniques can be used. Such methods will inevitably introduce a mass defect along the boundary, which in general will be proportional to the compatibility between the donor and receptor meshes and the complexity of the flow field. In compressible flows there is no direct requirement for the flow field to be divergence free. A non-conservative treatment at interfaces is therefore tolerable in the solution procedure. The same can be said for incompressible flows solved using an artificial compressibility formulation. In pressure-based incompressible solution strategies, however, the implicit continuity constraint inherent to the pressure-correction equation makes it imperative to the stability and convergence rate of the solution procedure that global conservation of mass is maintained at each time step. This constraint is naturally satisfied in a well-posed finite volume formulation, as described in Section 2.2. However, the error associated with the interpolation across overset interfaces will cause the solution of the pressure-correction equation to be slow and in many cases unstable if no correction of the mass defect is made. For incompressible flows, it is therefore necessary to address the conservation error in some manner, which is usually done using an explicit correction scheme before the pressure-correction equation is solved. This issue is addressed in Section 2.3.3.

The idea behind the current implementation of overset grids in EllipSys3D has been to make the least possible modifications to the original code, implementing the overset grid method as far as possible as an auxiliary module to the solver. This would simplify the implementation and make the code more transparent. A consequence of this decision was that coupling of the overset domains could only take place via the interpolated velocities outside the multigrid solver, since coupling of the solution during the multigrid solve would require the multigrid algorithm to be rewritten entirely.

As pointed out above the existing solution procedure can be used without major modifications. Preparation of the existing solution procedure to be able to handle overset grids can be achieved with the modifications described below.

1. *cut holes to accommodate solid bodies and assemble communication tables for the overset grids*
2. *Modify momentum coefficients a_{nb} , $S_{u,v,w}$ and $A_{Pu,v,w}$ in hole regions and on fringe boundaries*
3. Solve the momentum equations for u^* , v^* and w^*
4. *On overset interfaces interpolate u^* , v^* and w^* from donor grids and place values in fringe cells*
5. *Modify pressure coefficients S , A_P and A_{nb} in hole regions and on fringe boundaries*
6. Calculate mass fluxes
7. *On overset interfaces the mass flux is calculated using the interpolated velocities*
8. *Correct mass fluxes along overset boundaries of each group such that conservation is satisfied to machine accuracy*
9. Solve the pressure correction equation for p'
10. Correct p^* by adding an under-relaxed value of p' to obtain p
11. Correct u^* , v^* and w^* to obtain velocity field, which satisfies continuity
12. *On overset interfaces interpolate u^* , v^* and w^* from donor grids and place values in fringe cells*
13. If using the PISO algorithm return to 8 using new pressure and velocities and correct these again (only one loop)
14. Solve for other active scalars such as turbulent quantities
15. *Interpolate scalars as done for the velocities*
16. Return to 2 and use new pressure as guessed pressure p^* and repeat until sufficiently converged
17. Advance solution in time and return to 1

The explicit treatment of the flow variables on the overset boundaries introduces a time lag compared to interior cells which are solved implicitly. However, since a number of sub-iterations are used in each time step this lag does not have any adverse influence on the temporal accuracy of the solution as is shown in Chapter 3.

The preparation of the overset grid system involves three main routines; a hole-cutting routine which blanks out unwanted cells in the domain, a search routine, which identifies fringe cells, and an interpolation routine, which updates the flow solution in fringe cells during the solution procedure. These routines require special attention — which will be given in Section 2.4.

Firstly, however, the additions needed to the solution procedure must be

addressed. The interface boundary conditions and mass conservation scheme are described in the subsequent section.

2.3.1 Interface boundary conditions

The use of overset grids requires two modifications to be made to the discrete coefficients of the finite volume formulation. In hole cells the coefficients need to be adjusted accordingly, and on overset boundaries the coefficients must be modified according to the boundary conditions applied. As described earlier in this section an overset grid interface can be treated as any other boundary, with boundary conditions either specifying velocity or pressure. Prescribing both requires the boundary conditions to satisfy continuity in each cell, which is not possible when using a non-conservative interpolation scheme.

In the momentum equations, Dirichlet conditions are applied on the velocities. The finite volume formulation for the momentum equations (taking the U -momentum as an example) are modified as follows:

$$(A_P * ib + (1 - ib))U_P^{t+\Delta t} - \sum (A_{nb} * ib)U_{nb}^{t+\Delta t} = S_{mom} * ib + (1 - ib)U_I \quad (2.14)$$

where ib is the iblank array, which is zero in hole and fringe cells and one elsewhere; U_I is the velocity interpolated from the donor cells. In hole regions, the velocity is set to zero.

Since a Neumann boundary condition is applied on the pressure, the coefficients of the cells along fringe boundaries have to be modified according to Equation 2.11. The coefficients in hole cells are modified as follows:

$$A_{nb} = 0 \quad S = 0 \quad A_P = 1 \quad (2.15)$$

The last issue that needs to be addressed is which pressure value one should install in fringe cells. Following from Equation 2.11 a value of the pressure is not needed in fringe cells for the solution of the pressure, because a Neumann boundary condition is imposed. In the momentum equations, however, the pressure is needed to calculate the pressure source term correctly. There are three possible candidates: one is to interpolate the pressure from neighbouring blocks; another is to extrapolate the pressure from interior cells as is done on physical boundaries, and the last is to obtain the pressure by applying

the normal momentum equations on the boundary and combined with the continuity equation obtain the pressure.

Since there is no direct coupling between the pressure in each overset domain, the pressure field will develop independently in each domain differing by an arbitrary constant. The pressure levels in adjacent domains can be corrected to match each other if the pressure is kept fixed in *one* point common to both domains and adjusted in one of the domains. A direct interpolation of the pressure along the entire domain interface would therefore be inconsistent with the boundary conditions applied on the pressure-correction equation. An interpolation of both velocities and pressure would result in poor convergence and discontinuous pressure gradients.

The second alternative is to extrapolate the pressure from interior cells across the boundary according to Equation 2.12 as is done on regular boundaries. In corner fringe cells where the cell shares two faces with the interior domain this method becomes ambiguous since the pressure can be extrapolated from two sides. This is handled by averaging the two values. Given the relative simplicity of this method and its consistency with the standard boundary conditions, this is the method that will be used on overset interfaces.

Although the application of the normal momentum equation combined with the continuity equation might seem the most correct method for obtaining the pressure, the implementation of this method is not straight forward in general curvilinear domains. Additionally, cross terms of the equations must be omitted, which makes it questionable whether this method would yield a higher accuracy than a simple extrapolation.

2.3.2 Overset Grid Interpolation Scheme

Since the solution of the discretised equations is carried out in a transformed coordinate system which is logically Cartesian, the interpolation of flow variables can be carried out using standard interpolation schemes for uniformly spaced grids. Interpolation coefficients of the local curvilinear coordinates are found by inverting the trilinear function for the transformation using the Newton iteration scheme.

Different interpolation schemes can be used to transfer flow field information between grids. Chesshire and Henshaw [12] showed that the formal accuracy of the solver is maintained when the order of the interpolation used is one higher than that of the solver. However, as long as there is an adequately large overlap between component grids, second-order bilinear (trilinear in 2D)

interpolation was shown to be sufficient. The implementation of this scheme is considerably easier than higher order interpolation schemes since the interpolation stencil is much smaller, making the required overlap between grids smaller. A disadvantage of higher order schemes is that they can "over shoot" if the gradients in the flow are very sharp, which is typically seen for the turbulent quantities in wakes. Such schemes can cause the eddy viscosity to become negative, thus some measure must be taken to avoid the over shoot.

A 2D study of interpolation schemes was made (see Chapter 3) and it was found that the second-order accurate bilinear interpolation and the third order Lagrange interpolation produced virtually identical solutions for a large variety of flows. In light of the above comments and the numerical investigation, bilinear (trilinear in 3D) interpolation is thus used for transferring flow variables across overset interfaces.

The interpolation of flow variables must be performed in an explicit manner, that is; cells that are themselves fringe cells must not be donors, or part of an interpolation stencil of neighbour donor cells. If this is not the case the interpolation will become implicit, which could give rise to instabilities. To maintain second order accuracy of the discretisation scheme in cells adjacent to overset boundaries, two layers of fringe cells are needed. A 1D schematic showing the two layers of fringe cells and corresponding donor cells is shown in Figure 2.3.

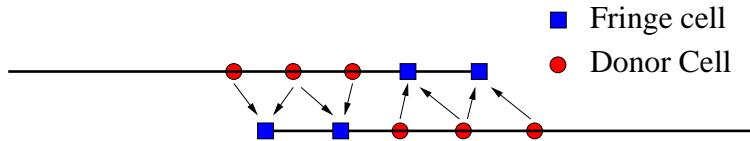


Figure 2.3: 1D Schematic of two overset grids indicating fringe and donor cells. In 1D, two cells are needed in the interpolation stencil. The minimum overlap permissible is that where no cell that is itself a fringe cell is used in the interpolation stencil.

2.3.3 Mass Flux Correction

Introducing a conservation error along the interpolation boundary will inhibit full convergence of the pressure-correction equation. For unsteady problems the lack of conservation will in most cases lead to instability due to a non-physical pressure build-up along the boundary. An explicit correction is therefore needed of the divergence error in each overset group before the pressure correction equation is solved. With the divergence being exactly zero in each

overset group, the solution of the pressure-correction equation will converge to machine accuracy with conservation being satisfied in all internal cells.

Several authors who use a predictor/corrector solution scheme to solve the incompressible Navier-Stokes equations ([11, 25, 68, 79]) have implemented similar corrections, which adjust the face velocities or fluxes on the overset boundaries such that the conservation error integrated over each overset boundary is zero. The difficulty with such a correction is that there is no obvious method for distributing the error in a physically meaningful manner. The divergence in each overset group will not have any direct relation to the divergence in neighbouring domains, because the domains are solved decoupled from each other. The error is most commonly distributed according to the local cell area [11] or the local flux [79]. In [25] normal velocity components on the boundary are corrected such that the divergence is exactly zero in each cell along an overset boundary. The Cartesian velocity components are subsequently derived from the corrected normal velocity, resulting in the velocity contours being slightly discontinuous.

In EllipSys3D a correction scheme similar to that by Zang and Street [79] and Burton and Eaton [11] has been implemented, as is outlined below.

The divergence error is found by integrating the face fluxes along the overset interfaces of each block group in the following way:

$$\varepsilon_n = \sum_{IB_n} F_i \quad (2.16)$$

where F_i is the mass flux across the overset boundary face i and \sum_{IB_n} is the sum over all overset boundaries in each block group n . The correction is proportional to the local flux and the modified flux is expressed as

$$F'_i = F_i - \frac{\varepsilon_n | F_i |}{S_n} \quad (2.17)$$

where S is the absolute sum of fluxes across the overset boundary in block group n

$$S_n = \sum_{IB_n} | F_i | \quad (2.18)$$

Distributing the error evenly or according to cell face area on the boundary was found to lead to instability since fluxes will be created artificially across faces parallel to the flow direction and in regions of low velocity. Although distributing it proportionally to the local flux is not necessarily a more physical representation of the error, it does lead to a more stable scheme.

2.3.4 Turbulence Modelling

For computations of flow over aerofoils and wind turbine blades the EllipSys2D/3D codes use the $k - \omega$ SST model by Menter [41], because of its good performance in wall bounded adverse pressure gradient flows. This model combines the original $k - \omega$ model by Wilcox [71] with the $k - \epsilon$ model by Launder and Spalding [32] through the use of zonal blending functions. In the inner part of the boundary layer the model reduces to the original $k - \omega$ model and gradually switches to the $k - \epsilon$ model (recast into a $k - \omega$ formulation) in the wake and farfield regions. Two blending functions are used: one which is designed to remove the freestream dependence of the original $k - \omega$ model, and another which accounts for the transport of the shear stress.

Using the $k - \omega$ SST model on overset moving grids presents some difficulties because the two blending functions in the model depend on the normal distance to the nearest wall. In the standard solver the distance to the nearest wall is computed once for each cell at the beginning of the simulation. However, in a problem involving relative movement of bodies, this distance will have to be recomputed in each time step to take account of the changes in distance of a given grid point from the different bodies, which is computationally very expensive. At the current stage of development, the computation of normal distances is not performed in background block groups which do not contain any wall boundaries. As a result the two zonal functions F1 and F2 are set to zero in these blocks, which means that the SST model is effectively reduced to the $k - \epsilon$ model in these block groups.

The implications of this issue clearly depend on the particular nature of the problem at hand and the grid configuration used. For bluff body flows such as vortex shedding from a circular cylinder the associated error is significant, whereas for attached flow over aerofoils the effect is less severe. An investigation of the error associated with the above issues is presented in Section 3.2.3.

The boundary conditions for the SST model are of Dirichlet type, which means that direct interpolation of k and ω using bilinear interpolation (trilinear in 3D) is used as boundary conditions. The finite volume equations are modified in holes and on overset boundaries in the same manner as done for

the momentum equations:

$$(A_P * ib + (1 - ib))U_P^{t+\Delta t} - \sum (A_{nb} * ib)U_{nb}^{t+\Delta t} = S * ib + (1 - ib)U_I \quad (2.19)$$

where ib is the iblank array, which is zero in hole and fringe cells and one elsewhere; U_I is k or ω interpolated from the donor cells. In hole regions, k and ω are set to zero.

2.3.5 Moving Grid Issues

When two meshes move relative to each other, the connectivity between them changes. In grids where overset boundaries are created as a result of blanking of internal cells the movement of a component grid will require the hole to be translated accordingly. The process is illustrated schematically in Figure 2.4 for a one-dimensional grid. The hole region travels to the right. On the right-hand side of the hole region cells change identity from internal cells to fringe cells, and fringe cells to hole cells. On the left side, the opposite is the case; hole cells become fringe cells and fringe cells become internal cells. It rarely happens that cells that were previously hole cells will become internal cells, since the movement in each time step is dictated by the flow solver, which for stability reasons has to be kept low.

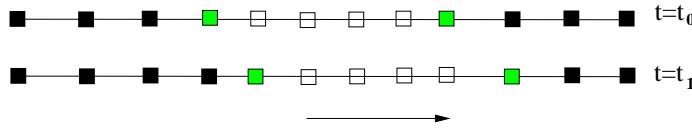


Figure 2.4: Creation of cells as a result of movement of hole boundaries. Filled black squares indicate internal discretisation cells; filled green squares indicate fringe cells; empty squares indicate hole cells.

Cells on the left-hand side of the moving hole region that were previously hole cells contain no valid flow field information. Likewise, in cells that were previously fringe cells and have become internal cells, values of velocity and mass flow from the previous two time steps must be available to correctly calculate the time dependent terms in the momentum and pressure-correction equations. It is therefore necessary to address this issue in some manner to obtain a smooth flow field in cells adjacent to moving hole regions. One possibility is to interpolate the values from neighbouring grids, however, this

approach would require the coordinates as well as connectivity tables to be stored from the previous time levels. The simplest approach is to average the values of velocity from valid neighbouring cells, and on the basis of these values, calculate the mass flow. Although not as accurate, as long as the grid movement is sufficiently small, this simple method has proven to be adequate to handle this issue.

2.4 Mesh Generation

An overset grid system typically consists of a mixture of curvilinear body fitted grids and Cartesian off-body grids. The body fitted grids resolve the boundary layer, and near body flow field, whereas the Cartesian grids resolve the regions between solid bodies and the farfield. High resolution grids are needed in the near-body Cartesian grids for proper resolution of the flow field, whereas coarse grids are used in the far-field. The Cartesian grids are therefore nested in multiple layers successively coarsened towards the farfield. Generation of curvilinear body-fitted grids is done using the in-house hyperbolic grid generator HypGrid by Sørensen [60]. An example of a grid is shown in Figure 2.5 with two spheroids embedded in two layers of Cartesian background grids.

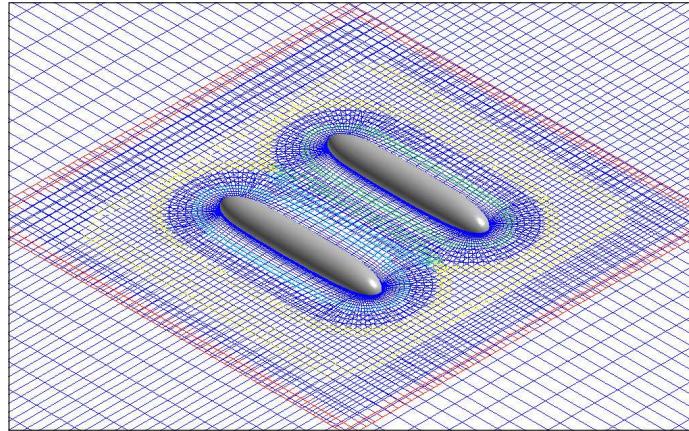


Figure 2.5: A mesh containing two spheroids with a total of four mesh groups. Colours along overset boundaries indicate donor grid identity.

It is important that the grids are dimensioned such that sufficient overlap is obtained between grids, and that the compatibility is good. Compatibility between grids can be regarded as a measure of the combined differences of cell volume, aspect ratio and orientation [64]. If good compatibility is not achieved, the solution will not be smooth in the overlap regions, and instability of the

solver could occur. From experience, it was found that grid sizes in overlap regions should not differ more than about a factor two.

Since EllipSys3D makes use of multiblock partitioning of the solution domain for parallelisation purposes, a grid system will consist of a mixture of connected blocks and overset grids. All blocks, which are patched together in a one-to-one manner will be labelled as part of the same block group. The grid system in Figure 2.5 thus consists of four groups, each composed of a number of logically cubic connected blocks. The process of identifying overset groups is automated using a recursive test of the face-face and interprocessor connections in the mesh, supplied by the base solver.

2.5 Domain Connectivity

The domain connectivity process of an overset grid problem consists of three main parts: firstly, all cells that fall outside the computational domain must be identified and classified as hole cells; secondly, cells along internal hole and internal outer boundaries must be identified and tagged as fringe cells; finally appropriate donor cells must be identified for each fringe cell and interpolation coefficients for these cells must be computed. Arrays are thus assembled containing a list of fringe cells, corresponding donor cell indices and interpolation coefficients.

Domain connectivity algorithms can broadly speaking be grouped into two classes: one which is designed to establish the connectivity externally from the flow solver, which is typically only used for problems not involving moving bodies, and another where the connectivity routine is integrated with the flow solver, and can thus be used for moving body problems. Most of the algorithms characterised in the first class use an iterative approach for determining size of holes and for locating appropriate donor cells. This, however, requires a large amount of communication between each component grid, which makes it very hard to implement efficiently for massively parallel computations. The advantage of such algorithms is that they are typically very robust and produce overset grid systems of high quality. The second class of algorithms are designed to be called at every time step in a simulation, since the relative movement of domains requires the connectivity information to be recomputed each time grids move relatively to each other. Since there is a great demand for efficiency, and more importantly, often a requirement to operate in a parallel computing environment, the iterative approach becomes less attractive since

the communication between processors must be kept at an absolute minimum.

There exists a variety of codes for assembling overset grids. A brief overview of the main codes available is given in the following.

PEGASUS [5, 64] was the first comprehensive package that performed the connectivity tasks. The grid assembly is performed externally to the flow solver and can thus be used with any code, but is less appropriate for moving body problems. The first version of the code required a large amount of user input but has since been replaced by a new version of the code which is highly automated, thus minimising the necessary user input. The hole-cutting task, which identifies points inside solid boundaries is performed using a Cartesian hole-map coupled with a line-of-sight algorithm. Following this step, the best possible receptor/donor pairs in neighbouring domains are found by iteratively adapting the location of outer fringe boundaries and hole boundaries. Donors are found using a binary tree search followed by a stencil jumping technique. This process is likely to be very time consuming, since donors have to be located for all valid cells in the domain. The code is parallelised using the MPI message passing interface with a master/slave distribution of the work load.

The CMPGRD code [12], which is part of the OVERTURE framework [10] provides a comprehensive package, which interactively allows the user to generate an overset grid system via a graphical interface. The CMPGRD algorithm produces an overset grid by firstly marking all points nearest to physical boundaries as hole points. It proceeds by identifying suitable donors for all valid cells in the domain according to a given hierarchy. Points for which no donor is found are marked as hole points. The size of holes and location of outer fringe boundaries are subsequently iteratively adjusted by traversing through each component until a valid grid is obtained with a minimum overlap dictated by the width of the interpolation stencil used to transfer flow field information between grids.

DCF3D [39, 40] was developed mainly with dynamic moving body problems in mind. The hole-cutting process was simplified by using analytic shapes to define hole regions, such that points inside holes could be identified with very little computational expense. The identification of donors for the fringe points was done using Cartesian grid inverse maps of the computational space. Although setting up these maps is expensive it is balanced by the fact that no exhaustive searches were needed. For moving body problems the inverse maps move with the component grids such that maps do not have to be re-

computed at every time step. The code was since parallelised [3] and used for moving body problems. The parallelised version of the code executes the same process on each computational node. Searches are requested using Cartesian bounding boxes and a so-called nth-level-restart which uses information from the previous time step to guess the location of the fringe point.

The Beggar code [37] is also designed to integrate with the flow solver for moving body problems. It uses a Cartesian mark and fill method, similar to that used by PEGASUS. A combination of polygonal mapping (based on octrees) and a stencil jumping technique is used to partition the domain and locate donor cells. The code has been parallelised over the course of four stages of development. In the third stage a combination of front end and back end nodes are used to compute the connectivity and flow solution separately. This process can efficiently be load balanced since each front end node will be assigned an equal number of tasks. The last stage uses a shared memory configuration where all nodes compute both connectivity and flow solution. Again, good load balancing can be achieved, since processors compute the same amount of tasks.

Petersson [49] presents a grid assembly code which combines the methods used in a number of the previously described codes. The method starts out by blanking all points outside the domain. This is done using a mark and fill technique combined with the boundary ray method [44]. Subsequently, all points are classified as either discretisation, interpolation or hole points by locating all possible donors for each point. Points are located using a combination of a bisection technique and Newton's method, where the process is speeded up using the location of neighbouring points as guess for subsequent points. Corrections for mismatches of overset surface grids is made by adjusting the interpolation coefficients. The grid is subsequently trimmed in three steps to produce minimum overlap between grids according to the interpolation stencil and type of the interpolation used.

Noack [45, 46] has with the so-called DiRTlib created a general library to add an overset capability to existing solvers, both structured and unstructured. It uses the SUGGAR grid assembly code, that can be used for both stationary and moving body simulations in a parallel environment. The hole-cutting is done using a binary tree subdivision of the hole-cutting geometries, similar to what is done in the BEGGAR code. Donor cells are identified using a combination of an octree-based search and a stencil-jumping technique.

The current approach has been kept simple to minimise computational cost, while at the same time being sufficiently automated for handling moving grid problems. The fact that the process has to work in a parallel environment means that the procedure must contain as little inter-processor communication as possible. The iterative techniques used in [12], [49] and [64] to optimise the overlaps have thus not been employed since the trimming process would require a considerable amount of communication between processors.

The user specifies via the input file which mesh groups are allowed to cut holes in other specified groups and the size of the overlap for each group. There is no restriction on the number of grids that can be cut by each group. Furthermore, the user specifies which groups are allowed to move, with the rotation/translation being prescribed via the input file. To aid the initial stages of the grid connectivity process, the user also specifies which block groups are allowed to interpolate from each other. This reduces the number of search requests made in vain to block groups which for one reason or another are not adequate donors.

In the following sections the hole-cutting and search routines will be described in detail. This is followed by a description of the current stage of the parallelisation of the overset grid method and the potential improvements that can be made.

2.5.1 Hole-cutting

The purpose of hole-cutting is to remove or blank out cells in the solution domain that fall within the boundaries of solid bodies as is illustrated in Figure 2.1. The method was originally conceived by Benek et al. [6], and later many methods have been developed for cutting holes. To improve compatibility between the component grids and avoid interpolation from high aspect ratio boundary layer cells, it is furthermore necessary to blank out coarse cells from the overlapping grid near the solid boundaries. A closed boundary curve in 2D or surface in 3D is thus defined a certain distance from the surface where good compatibility and sufficient overlap can be achieved. All cells in the background grid that fall within this curve are then blanked out and classified as hole cells.

To identify whether a given node falls inside a closed curve or surface the boundary ray method is used [44], which is also the method applied by Petersson [49]. The current implementation of the method is based on work carried by Bertagnolio [7]. The method counts the number of intersections

with the boundary curve or surface of a horizontal ray projected towards $-\infty$. If the ray intersects the curve or surface an odd number of times the node must be inside the curve, see Figure 2.6.

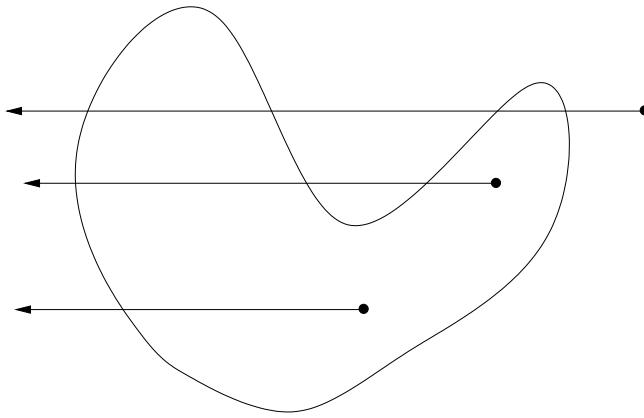


Figure 2.6: Schematic of the boundary ray method.

In 2D, the boundary is described by a polygon, which is typically obtained from the nodes of a constant i - or j -curve in the mesh or a combination of these. The ray intersects the boundary curve if it intersects any of the line segments connecting the nodes on the curve. Details of the implementation of the method can be found in [49]. In 3D the boundary surface will typically consist of a constant i -, j - or k -surface of the mesh or a combination of mesh surfaces. Each quadrilateral element of the surface is divided into two triangular elements, since intersection of a line and a triangular element is geometrically easy to test.

To reduce the computational cost, the curve or surface is subdivided in a binary tree structure, containing min/max values of the bounding boxes of the curve or surface segments. If the ray intersects a line connecting a subdivision of the curve, it must also intersect the bounding box containing the line. A recursive search is made through the tree until the lowest level is reached where the ray intersects the line connecting two consecutive nodes, see Figure 2.7. The same holds for a 3D bounding box containing the triangulated segments of the surface.

In EllipSys the user specifies via the input grid file the outer fringe boundaries of each overset boundary. The hole-cutting surface is then constructed by traversing inward normal to the boundary a given number of cells specified via the input file by the user. The hole-cutting surface of each block group

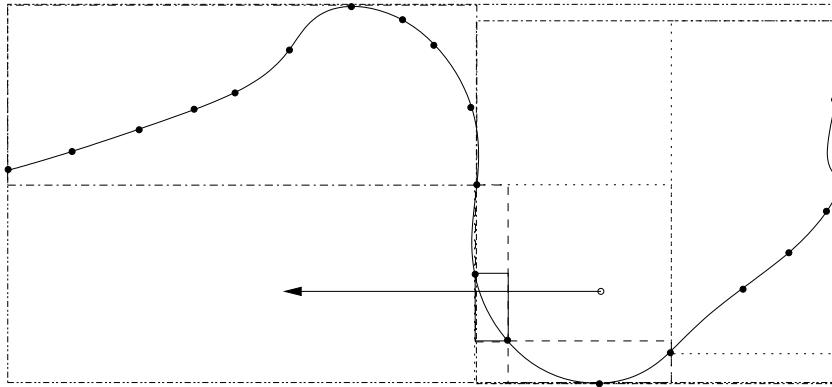


Figure 2.7: Schematic of the binary tree subdivision of the boundary curve during the search for intersections.

will thus consist of a combination of a number of constant i -, j -, and k -curves that form a closed surface. The hole-cutting surfaces are all specified in the cell-vertex based grid, since this grid is what defines the physical boundaries of the domain.

2.5.2 Fringe Cell Identification

The identification of fringe cells is split into two parts: one which identifies fringe cells along the outer boundaries of each group, and one which identifies the fringe cells along the irregular internal boundaries created during the hole-cutting process. The outer boundary fringe cells are located on the basis of the attributes set by the user in the input grid file, whereas the internal boundary fringe cells are found by travelling through the volume array and marking all cells that neighbour cells marked as hole cells.

2.5.3 Donor Cell Search

The routine for locating a point in a curvilinear mesh uses a combination of an octree based search algorithm and the so-called *stencil jumping* technique. If the previous location of the point is not known or no qualified guess is available, the enclosing grid cell of the point is found by partitioning the mesh in an octree structure containing the Cartesian bounding boxes of each block. Each node has eight subnodes which are constructed by diving the grid into eight logically cubic cells. A recursive search through the tree is performed until the bottom level is reached where the point is located inside one cell. To ensure that the located point is within that cell, the trilinear interpolation

coefficients of the cell are found using Newton iteration which maps the point in the curvilinear grid cell to parameter space. If the coefficients are between 0 and 1, the point must be inside the cell. If not found, the integer value of the coefficients are used to jump to neighbour cells, where the Newton method is re-applied.

Since this search method is quite exhaustive, it is desirable to use a qualified guess for the location of the point so the octree search can be skipped and the Newton inversion applied immediately. If there is no time history of the location of the point, a good guess could be a located neighbour fringe point. Using stencil jumping the point is usually located within one jump. However, since the fringe cells are not sorted by geometric location, but the order they were identified, this guess is not very desirable for moving grid computations. A better method is to keep a list of the fringe point locations in the previous time step and use this as a starting point for the stencil jump. Usually, points will only have moved one grid cell, since the flow solver imposes restrictions on the time step allowed for stability. With this method only a small fraction of fringe cells are searched for using a general search, greatly reducing the computational cost. The stencil jumping algorithm will fail if the guessed location exceeds the block boundary and in some cases where the grid is highly stretched. In this case a general search is requested which was described above. In Algorithm 1 the steps taken to locate an interpolation point are outlined using pseudo code.

The search is made in the cell-vertex based grid, which conforms to the physical boundaries of the domain, but since all flow variables are stored in the cell centers of each grid cell, the interpolation coefficients must be recomputed for the cell centered grid to obtain the best accuracy in the interpolation. The recomputed interpolation coefficients can in some cases be invalid, in which case the stencil jumping technique is used to find a valid donor.

2.6 Code Parallelisation

The EllipSys code is fully parallelised using the MPI library. Each node simultaneously performs the same tasks in each step of the solution algorithm. To fully integrate the overset grid method in the solver all overset communication has been constructed in a similar fashion with no sequentiality such that each node performs the same tasks on the set of blocks it has been assigned.

In the non-overset solver there will always be a one-to-one correspondence

Algorithm 1 Search routine: locate a point in general curvilinear grid using stencil jumping

```

for all fringe cells do
    if location of fringe cell in previous time step is known then
        Compute the interpolation coefficients  $d\xi$ ,  $d\eta$  and  $d\zeta$  in cell vertex grid
        based on previous location ( $ipoint$ ,  $jpoint$ ,  $kpoint$ ,  $npoint$ )
         $ijump=ipoint+\text{int}(d\xi)$ 
         $jjump=jpoint+\text{int}(d\eta)$ 
         $kjump=kpoint+\text{int}(d\zeta)$ 
        if predicted location is within block bounds then
            recompute interpolation coefficients using the guessed indices
            ( $ijump$ ,  $jjump$ ,  $kjump$ ,  $npoint$ )
            if  $d\xi$ ,  $d\eta$  and  $d\zeta$  are between 0 and 1 then
                Point has successfully been located
            end if
            Recompute interpolation coefficients in cell-centered grid
        end if
    else
        call locatepoint
    end if
end for

```

in information transfer across block interfaces, and since each processor is assigned an equal portion of the domain, the computation of the flow solution will scale quite well in a parallel environment. In an overset grid system, the tasks associated with grid connectivity and interpolation of flow variables are not guaranteed to be well balanced between all processors. In fact, they very rarely are. This is caused by the fact that block groups in the overset grid system are often of different topology and density, which means that some processors will hold more fringe or donor cells than others. The communication of overset related data taking place between processors is therefore also different to that in the non-overset solver, since the exchange of information is no longer one-to-one. The combination of these two factors could result in poor performance of the code in a parallel environment. To remedy this, a specific strategy can be employed to improve the load balance between processors.

Load balancing can be achieved by redistributing the problem among the processors. Wissink and Meakin [73] presents a dynamic load balancing strategy for overset grids which takes into account the number of fringe cells on each processor and redistributes the blocks if the load balance is poorer than a user specified ratio. This can only be done once the computation is started and the connectivity between each block is established. Additionally, the re-

distribution of blocks has to be performed periodically during the solution of the problem since the number of fringe cells on a given processor will change in a moving grid simulation. If the grid assembly process is carried out on the same processors as the flow solution, all geometric and flow field information has to be relocated onto the new processor, which could be very time consuming. An attempt to improve the load balancing of the overset grid related tasks will almost inevitably degrade the load balancing of the base solver. Indeed, Wissink and Meakin [73] concluded that the dynamic load balancing strategy did not improve the overall solution time for the problems tested, since the strategy severely reduced the parallel performance of the flow solver.

Another possibility is to perform the grid assembly and the flow solution on separate groups of processors. Each processor performing the connectivity tasks thus holds the geometric properties of all grids in the domain. One front end processor can then distribute an equal portion of search requests to each processor, making it possible to achieve very good load balancing of the connectivity process. Additionally, this process can be performed simultaneously with the flow solution, thus reducing the latency significantly. This is the strategy employed by Prewitt et al. [51], who report quite good scaling properties of moving grid problems.

At this stage of development, there has been no changes in the load balancing strategy of the EllipSys2D/3D solvers, which means that blocks are distributed evenly on each processor according to the order given in the input grid file. This will provide near optimal load balancing for the flow solution but as stated above will result in poor load balancing of the overset grid tasks. Each processor only holds geometric properties of blocks local to the processor, which makes it very difficult to load balance the problem.

Looking firstly at the grid assembly there are several additional tasks that are necessary to perform in a parallel environment. The first tasks of cutting holes and setting up the boundary attributes are handled by the routines `HoleCut` and `ComputeOGAttributes`, which were described in detail in Sections 2.5.1 and 2.5.2. The hole-cutting routine is easily parallelised by letting all processors broadcast their respective cutting surfaces to all other processors. This information only has to be communicated once since each processor is supplied with the parameters describing the movement of all block groups. Everything else in this routine can be performed locally with no additional need of communication.

The search routine for locating donor cells, however, is more complicated

Algorithm 2 ConnectOG: Top level of connectivity routine

```

call blockbounds (Compute Cartesian bounds of blocks on each processor)
call HoleCut (Blank invalid cells using the boundary ray method)
call ComputeOGattributes (Assemble list of fringe cells)
if (nstep.eq.1.or.restart) then
    call AssembleOGlocal
    call AssembleOGparallel (Assemble list of local and parallel search requests)
else
    call AssembleOGguess (Assemble list of local and parallel search requests based on previous location)
end if
call ConnectOGparallel (Distribute requests to tagged processors, perform local and incoming searches, return requests)
for all fringe cells not located do
    add fringe cell to reassemble list
end for
if (norphans.gt.0) then
    call ReAssembleOGlocal
    call ReAssembleOGparallel (Assemble list of local and parallel search requests)
end if
call ConnectOGparallel
Assemble final list of donor cells and report number of orphans
call ExchangeOG (Distribute list of donors to donor processors)

```

to parallelise since each processor has to guess which neighbouring processors are most likely to contain valid donors for the fringe cells of the processor. Each node has to assemble a list of fringe cells that need to be located, with separate lists made for local and interprocessor search requests. These tasks are performed in the routines `AssembleOGlocal` and `AssembleOGparallel`. To reduce the number of search requests made the `AssembleOGlocal` routine tests whether a point falls within the Cartesian bounds of any of the blocks locally on the processor that are not in the same block group as the search point, if so, it is tagged for a local search. The `AssembleOGparallel` routine performs a similar search, checking the Cartesian bounds and block group of the blocks on each processor. The `AssembleOGguess` routine is used in moving grid problems where the grid connectivity has to be recomputed at every time step. The method which is used in this routine, referred to as nth-level-restart [3], consults the donor list assembled in the previous time step, and places a search request with the donor processor in question. This reduces the computational cost massively since few search requests are sent in vain,

and because connectivity information only has to be updated between grids that move relative to each other. The requests will be returned unsuccessful at instances where the point has moved across a block boundary in the donor block, into a block placed on another processor. However, most requests are returned successful, since the relative movement between grids is kept small to maintain solver stability. The searches that are unsuccessful will be returned to the requesting processor and a general search will be made, assembled in `ReAssembleOGlocal` and `ReAssembleOGparallel`. Finally, the finished connectivity lists are distributed to the necessary processors. The top level of the connectivity routine is summarised in Algorithm 2.

The `ConnectOGparallel` handles the incoming search requests, calls the search routines, and returns search requests. The routine is outlined below in Algorithm 3.

Algorithm 3 `ConnectOGparallel`: Parallel handling of search requests

```

call ExchangeOG (Each processor receives incoming requests and sends own
requests)
call OGsearch (Searches are performed, see Section 2.5.3)
call ExchangeOG (Own requests are received and searches are returned)
for ii=1,fringecount do
    Incoming successful search results are tagged. Multiple search results are
    prioritised according to grid hierarchy
end for

```

The bulk of the communication taking place is the exchange of connectivity information and flow-field data. The grid connectivity routine `ConnectOG` is by far the most time consuming of all the overset routines, since there are several barriers in the MPI communication where processors have to communicate data globally. The exchange of flow-field velocity data can be done by only using non-blocking MPI calls since all processors have been supplied with the necessary connectivity information from `ConnectOG`. This routine, however, has to be called several times during one time step, and therefore requires the same order of CPU time as the connectivity routines, depending on the problem solved.

The communication between specific processors is handled via non-blocking `MPI_ISEND` and `MPI_IRecv` calls, since this type of call can reduce the load imbalance to a minimum. In Algorithm 4 the `Exchange3D` routine is outlined using pseudo code. This routine is used to communicate all overset data between processors. It contains two non-blocking MPI calls; one that receives all

Algorithm 4 Exchange3D.f: Asynchronous communication of data

```

for np=1,nprocs do
    if (np.ne.myprocnum) then
        if (rcvcount.ne.0) then
            call MPI_IRecv(recv(np),rcvcount(np),...) receive requests
        end if
    end if
end for
for np=1,nprocs do
    if (np.ne.myprocnum) then
        if (sndcount.ne.0) then
            call MPI_ISend(snd(np),sndcount(np),...) send requests
        end if
    end if
end for
call MPI_WAITALL(...)

```

requests from neighbouring processors, and one that sends the processor's own requests. The two calls are in separate loops, since there is not necessarily a one-to-one ratio of communication between processors. The MPI_WAITALL call is blocking and must be imposed when non-blocking MPI calls are used. It imposes a barrier on those processors that have issued receive calls, to ensure that a processor will not exit the routine until the receive process is complete.

An array containing the send and receive counts must be assembled before this routine is called. The sndcount and rcvcount arrays contain this data, and are globally communicated by all processors via an MPI_ALLGATHERV call. Since this type of call is blocking, it is only performed at the beginning of the time step, during the grid assembly process. The only other type of communication taking place in the overset grid routines is the global gathering of the conservation error of each group. Again the MPI_ALLGATHERV call is used for this.

3

NUMERICAL VALIDATION OF THE OVERSET GRID METHOD

3.1 Introduction

The EllipSys3D code has been used on a variety of flow problems throughout its development. Spatial and temporal accuracy associated with mesh quality, solution and discretisation schemes has been investigated extensively, and robust guidelines have been developed for computations of flow over aerofoil sections, wind turbine rotors and terrain [4, 8, 61].

Since the overall solution strategy of the overset solver is identical to that of the original solver, core properties such as discrete spatial and temporal order of accuracy should be maintained. Despite the many similarities between an overset approach and a point-wise continuous approach, the fact that grids do not match on interfaces makes the two approaches fundamentally different. The overset approach developed in this thesis is not conservative due to the introduction of intergrid boundary points, which are updated using simple explicit interpolation of flow variables. This aspect gives rise to a series of questions that need to be addressed to validate the method. Firstly, it must be shown that the non-conservative nature of the method does in fact not compromise the spatial accuracy of the solver. Secondly, it must be shown that the explicit nature of the overset boundary conditions does not deteriorate the temporal accuracy. Additionally, the validation of the overset solver must also address aspects such as solution dependence on interpolation schemes, overlap size and mesh compatibility. In this chapter a series of numerical experiments of flow over a variety of geometries in different flow regimes will be presented to explore the properties of the solver and address the above questions.

Since the accuracy of the original solver is thoroughly documented, the validation of the overset solver is carried out by comparing solutions using the modified solver with solutions obtained using the original solver on equivalent non-overset grids. Additionally, some test cases are also validated against

experimental data. Four test problems were set up:

- Flow over a circular cylinder:

Steady 2D laminar flow at $R_e=40$

Unsteady 2D turbulent flow at $R_e=1.85 \times 10^6$

- Steady 2D turbulent flow over the NACA 63₂–415 aerofoil at $R_e=3 \times 10^6$
- Unsteady 2D turbulent flow over the NACA 0015 aerofoil undergoing oscillatory motion at $R_e=1.5 \times 10^6$
- Steady 3D turbulent flow over a 6:1 prolate spheroid at $R_e=1.42 \times 10^6$

Additionally, a multibody moving grid problem of flow over two spheroids undergoing pitching and rotation was used to evaluate the parallel performance of the 3D solver.

In lack of an exact solution, the spatial and temporal accuracy of a numerical discretisation scheme can be estimated by showing that the rate at which the solution converges is of the same order as the discretisation scheme. The truncation error associated with the discretised equations can be estimated by comparing the solutions ϕ from successively coarsened grids. For grids where the grid sizing (and time step) differs by a factor of two, the order of accuracy σ of integral quantities is found using the following relation

$$\sigma = \frac{\log \left(\frac{\phi_{2h} - \phi_{4h}}{\phi_h - \phi_{2h}} \right)}{\log 2} \quad (3.1)$$

where h is the grid spacing. This estimate is only valid for smooth solutions that converge in a monotonic manner, which can be obtained as long as the grid spacing is sufficiently fine.

3.2 Flow over a Circular Cylinder

The aim of this section is to demonstrate that the overset version of the flow solver maintains both spatial and temporal formal order of accuracy of the base solver, and show that the absolute accuracy of the solver is as good as the original version of the code. The flow over a circular cylinder is investigated in two flow regimes:

- Steady 2D laminar flow at $R_e=40$
- Unsteady 2D turbulent flow at $R_e=1.85 \times 10^5$

The steady laminar flow case is appropriate because the fundamental properties of the solver can be investigated, such as convergence behaviour and conservation properties. This flow is used to demonstrate that the spatial order of accuracy is maintained. The unsteady turbulent flow case is used to test the temporal order of accuracy. Due to its unsteady nature it is also a good test case for an investigation of the properties of two different overset grid interpolation schemes and study the implemented turbulence model.

3.2.1 Computational Meshes

The curvilinear grids used in this work were all generated using the hyperbolic grid generator HypGrid by Sørensen [60]. For the computations using the original solver a non-overset grid was made using a O-grid configuration which consisted of 256 cells in the ξ direction around the cylinder and 256 cells in the η direction normal to the cylinder. The height of the first boundary layer cell was set to 1×10^{-5} and the outer boundary was placed 50 cylinder diameters from the surface. Standard inflow boundary conditions with a prescribed velocity were used. The outflow was specified with a zero gradient condition. This grid will be referred to as grid C1.

Two overset grids were made for the cylinder configuration: one similar to a standard O-grid configuration and another where Cartesian meshes were used to resolve the farfield. Grid C2 consisted of two overset block groups with 256×128 cell in each. The block group around the cylinder surface was identical to grid C1 above and extended 1.2 diameters from the surface. The inner boundary of group 2 was placed 0.3 diameters from the surface of the cylinder and extended 50 chord lengths outwards differing only slightly from the mesh in grid C1. By making the two meshes Grid C1 and C2 as similar as possible, discrepancies due to mesh topology were kept at a minimum in the results, making this test well suited to verify the basic properties of the overset grid method. The grid is shown in Figures 3.1.

In grid C3 the inner block group was identical to grid C2 above also with 256×128 cells, whereas two Cartesian meshes were used to resolve the farfield. The inner Cartesian mesh had physical dimensions of 10×6 cylinder diameters and consisted of 192×128 cells distributed uniformly in the horizontal and vertical directions. The outer mesh consisted of 128×128 cells and was stretched towards the outer boundary that was placed 15 diameters away from the surface in the vertical direction. The inflow was 15 diameters upstream and the outflow 25 diameters downstream. The two overset grids had the same inflow

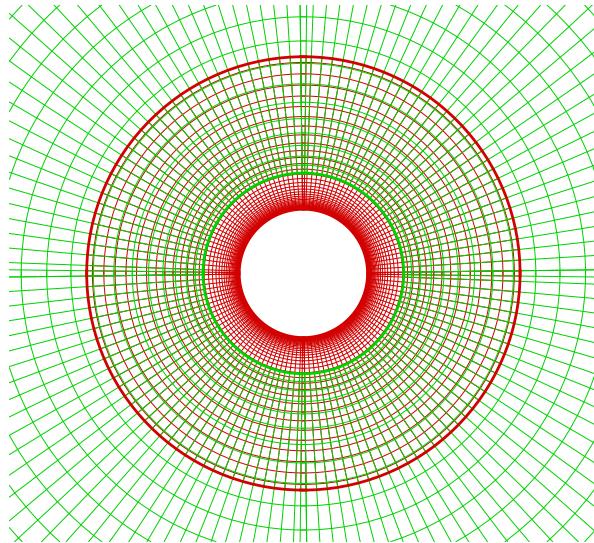


Figure 3.1: Grid C2 cylinder mesh layout showing only every second grid cell for clarity.

and outflow boundary conditions as grid C1.

Using a combination of curvilinear and Cartesian meshes generally requires more care to be taken during the grid generation process to ensure good compatibility between grids and sufficient overlap to guarantee that explicit interpolation is possible. As described in Section 2.4 the overlap size is specified via the input file and is dictated by the grids that are allowed to cut holes. This procedure gives no guarantee that both grids in the overlap region have sufficient overlap if the disparity in grid sizing is large. The process of grid generation therefore requires significant user experience. If the flow problem is solved using a grid sequencing strategy to speed up the solution process the problem becomes more pronounced. This is illustrated in Figure 3.2 where grid C3 is shown at three levels of coarsening. Since two layers of fringe cells are needed, the physical overlap is very large on the coarse grid. A remedy for this could be to reduce the order of accuracy along the fringe boundaries on the coarse grid levels by only using one layer of fringe cells.

3.2.2 Steady Laminar Flow at $R_e=40$

The accuracy of the solution was evaluated by comparing the drag coefficient, skin friction and surface pressure distributions on the cylinder for the overset

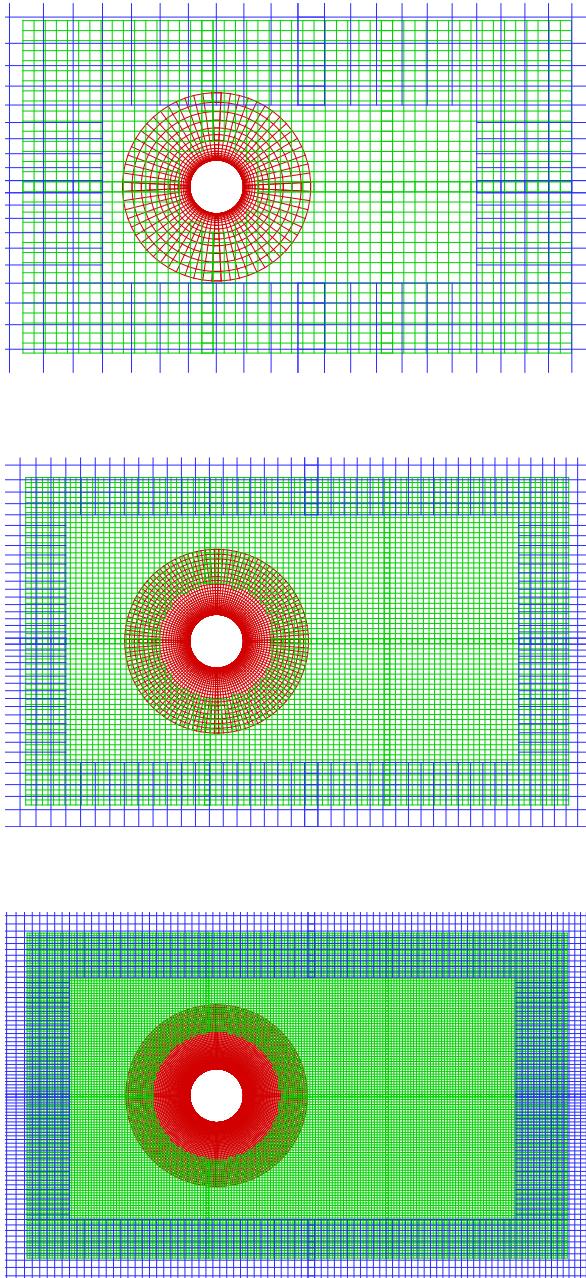


Figure 3.2: Grid C3 cylinder mesh layout showing only every second grid cell for clarity.

and non-overset solutions. The ability of the code to smoothly represent the recirculation region and accurately predict its length was also investigated. Additionally, the behaviour of the flux correction scheme was studied.

The computations were run in a steady state manner. The SUDS scheme was used to discretise the convective terms and the SIMPLE algorithm was used to solve the coupled pressure/velocity equations.

Level	C2 bilinear	C2 Lagrange	C1
3	1.562	1.543	1.539
2	1.530	1.524	1.523
1	1.520	1.520	1.519
σ	2.078	2.23	1.71

Table 3.1: Drag coefficient on the circular cylinder and estimated convergence rate.

A grid dependence study of grids C1 and C2 was conducted where the grids were successively coarsened by halving the number of cells in each coordinate direction with a total of three grid levels. The position of the overset interfaces in grid C2 were kept fixed for all three grid levels. A large overlap was thus needed to ensure that explicit interpolation of flow variables on interfaces could be maintained at all three levels.

Computations were run using either bilinear or Lagrange interpolation of the velocities. Drag coefficients are listed in Table 3.1 for all three grid levels for meshes C1 and C2. The order of accuracy σ of the overset and non-overset solvers was estimated using Equation 3.1.

The two overset solutions compare well with the non-overset solution, with almost identical drag predictions on the finest grid level. Lagrange interpolation of the velocities was slightly more accurate, which is evident on the coarse grid levels. Additionally, the two solutions both approximately show second order convergence in accordance with the non-overset solver. The estimate of the convergence rate was found to depend somewhat on the mesh quality, so the fact that the overset solutions seem to have a higher convergence rate should not be given too much significance.

Level	C2 bilinear	C2 Lagrange	C1
3	2.48	2.48	2.32
2	2.72	2.72	2.64
1	2.74	2.75	2.71

Table 3.2: Recirculation length $\frac{L_{recirc}}{D}$ of wake behind the circular cylinder for three mesh sizes.

The recirculation lengths were estimated graphically using streamlines and are listed in Table 3.2 . The discrepancy on the coarse grid level can be attributed to the slight differences in the farfield grids of mesh C1 and C2 and as can be seen, the agreement is very good on the finer grids. No difference was observed between the two interpolation schemes.

Plots of C_p and C_f solved on grids C1 and C2 are shown in Figure 3.3 for the finest grid level where the bilinear interpolation scheme is used on the overset boundaries.

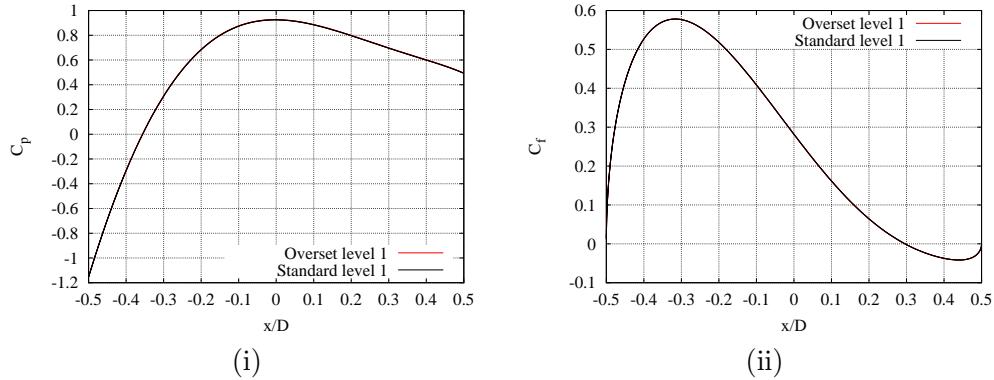


Figure 3.3: (i) pressure distribution and (ii) skin friction coefficient distribution on a circular cylinder at $R_e = 40$ computed on grids C1 and C2 at grid level 1.

The convergence rate of the two solutions were compared at all grid levels. A plot of the residual reduction is shown in Figure 3.4, with the coarsest grid levels starting from the left. The two problems converge at almost the same rate. Any discrepancy is most likely associated with differences in distribution of cells in the farfield. This clearly confirms that the mass flux correction applied along all overset boundaries, ensures that the solution converges to machine accuracy, fulfilling continuity in all internal cells.

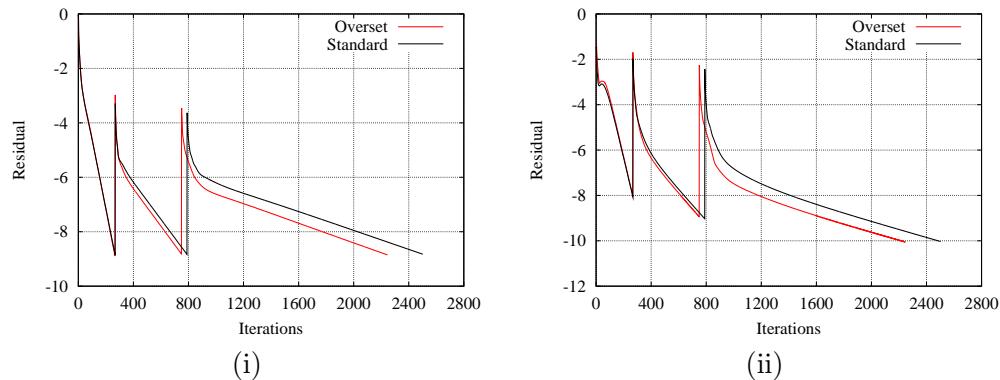


Figure 3.4: (i) u-velocity and (ii) pressure residuals for the overset solution on grid C2 and standard solution on grid C1 for grid levels 1, 2 and 3.

To illustrate the necessity of the mass flux correction (described in Section 2.3.3), the computation was also performed without the correction. Figure 3.5 shows the u-velocity and pressure residual time histories for the two computations. Since the momentum equations do not require the flow field to be divergence free, the velocity residuals are identical with and without the correction. The pressure residual, on the other hand, converges towards a finite value when no correction is applied

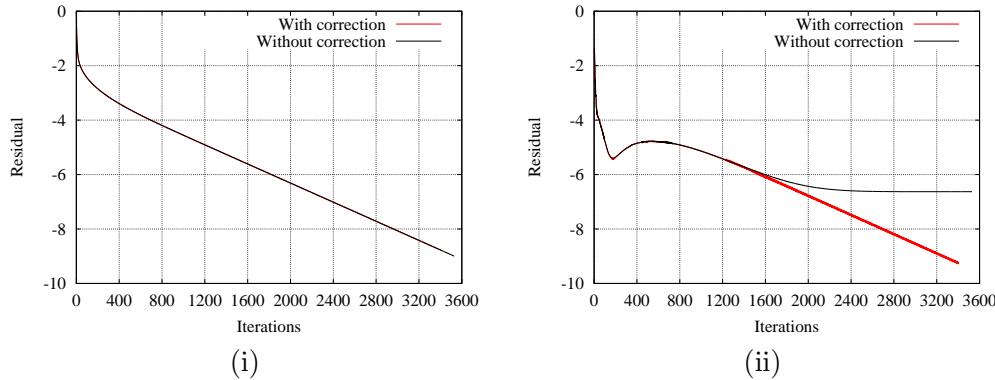


Figure 3.5: (i) u-velocity and (ii) pressure residuals with and without the mass flux correction computed on grid C2.

In the discrete form, the finite volume formulation of the pressure-correction equation converges towards a solution where the integration of all fluxes in the interior of the domain equals the integration of fluxes along the boundary of the domain. Thus, when no correction is applied the discrete divergence along the overset interfaces inhibits further convergence of the pressure-correction equation. For unsteady problems it was found that the solver became unstable unless the correction was applied. This would manifest itself by large pressure build-ups along the overset interfaces due to a high divergence error, making pressure and velocity gradients discontinuous.

The divergence error associated with the non-conservative interpolation was found by integrating the fluxes along all overset boundaries for each block group according to Equation 2.16 before the correction was applied. The time history of the divergence error in each block group is plotted for all grid levels in Figure 3.6. The error quickly reaches a steady level exhibiting no spurious behaviour, indicating that the stability of the base solver is not deteriorated by the use of overset grids.

Table 3.3 shows the absolute values of the divergence errors at the three grid levels for the inner and outer groups. Although one would expect the error reduction, σ , to behave in a second order manner, it is considerably higher

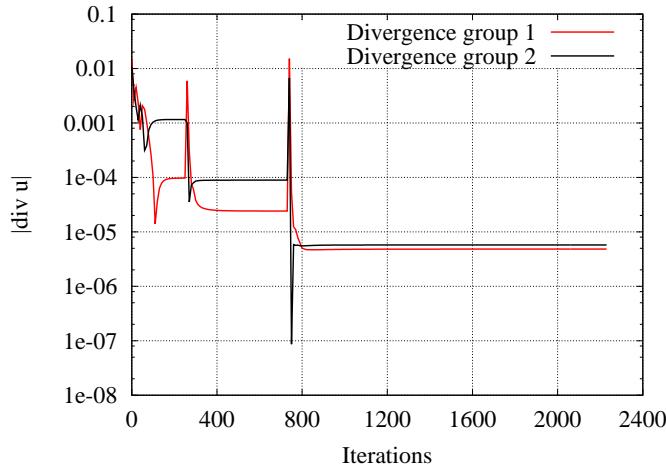


Figure 3.6: Time history of the non-dimensionalised divergence error in each block group computed on grid C2.

for both groups. From a wide range of other computations conducted, it was found that the convergence rate of the divergence error varied in a range of $2 < \sigma < 5$ according to the nature of the problem and compatibility between the overset grids.

Grid level	Divergence Group 1	Divergence Group 2
3	1.076×10^{-4}	8.597×10^{-4}
2	6.292×10^{-6}	1.021×10^{-4}
1	5.965×10^{-7}	8.931×10^{-6}
σ	4.85	3.02

Table 3.3: Absolute value of the divergence error for the three grid levels for grid C2.

Figure 3.7 shows contour plots of the velocity components and pressure gradients for the computation. It is appropriate to plot the gradients of the pressure, since the solver only operates on the gradient and not the actual value of the pressure. Clearly the solutions in the overlap region agree very well both for velocity and pressure. The pressure gradients are smooth across overset interfaces clearly demonstrating that the boundary conditions are appropriately applied.

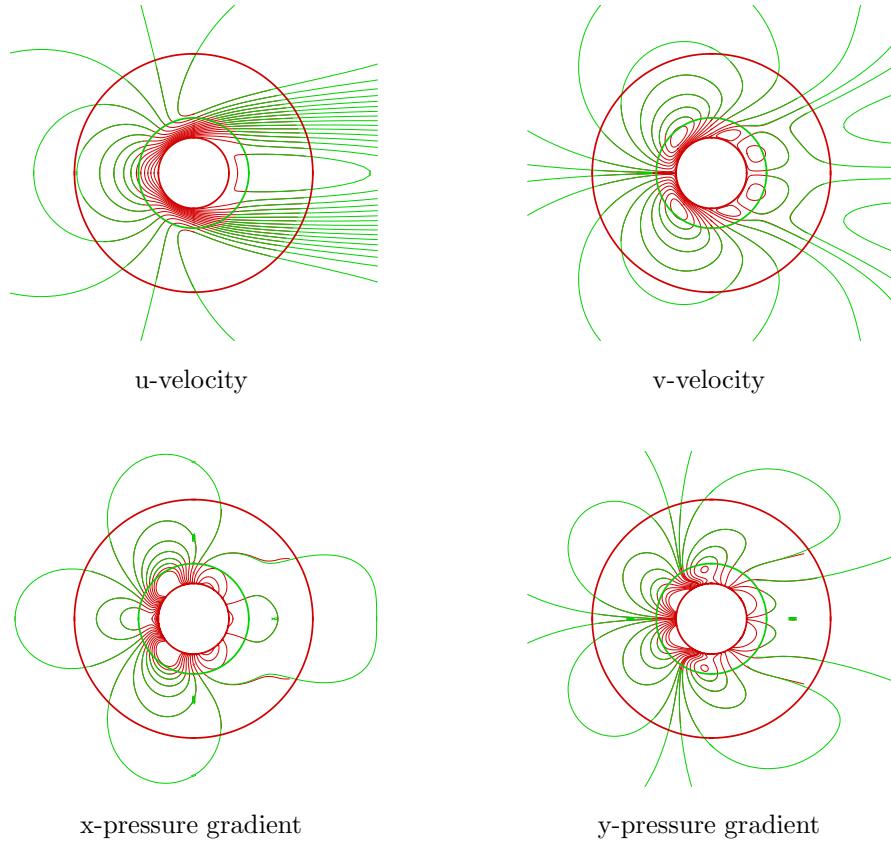


Figure 3.7: Contour plots of velocity and pressure gradients on a circular cylinder at $R_e = 40$ computed using grid C2 on grid level 1. The overset boundaries of the inner and outer block groups are indicated by the red and green curves, respectively.

3.2.3 Unsteady Turbulent Flow at $R_e=1.84 \times 10^5$

In the second test case, the overset grid method is used to compute unsteady turbulent flow over a circular cylinder at a Reynolds number based on the cylinder diameter of $R_e=1.84 \times 10^5$. This Reynolds number was chosen because it corresponds to the Reynolds number encountered in the 3D computations presented in Chapter 4. In this flow regime the wake of the cylinder consists of the well known von Kármán street of alternately shed coherent vortices advected downstream. For $1 \times 10^5 < R_e < 3 \times 10^5$ the flow is in the so-called drag crisis region where the boundary layer undergoes transition from laminar to turbulent flow. Since the purpose of the current computations is mainly to validate the overset grid method against the standard solver, the correspondence to experimental results was less important, and for simplicity the flow was therefore assumed to be fully turbulent. This generally results in

an incorrect prediction of the drag and shedding frequency.

All simulations were run using the SUDS scheme for the convective terms, and the PISO algorithm to solve the velocity/pressure equations using six sub iterations in each time step. The computations were run at three grid levels, each until the residuals were reduced by eight orders of magnitude. Three time steps were investigated: $\Delta t = 1 \times 10^{-2}$, 5×10^{-3} , and 2.5×10^{-3} . All simulations were run using the $k - \omega$ SST model.

Firstly, two different interpolation schemes were tested on grid C2 and compared to results on grid C1 for the coarsest time step of $\Delta t = 0.01$. To verify the temporal accuracy of the solver a time step study was carried out with three time steps of $\Delta t = 1 \times 10^{-2}$, 5×10^{-3} , and 2.5×10^{-3} . For this analysis Grid C1 and C2 were used. The implementation of the $k - \omega$ SST model is as described in Section 2.3.4 not complete in that the zonal functions are not calculated in off-body grids. An investigation was therefore carried out to quantify the errors associated with the current implementation.

Table 3.4 summarises the results for the overset computations using bilinear (which is second-order accurate) and third-order Lagrange interpolation compared to the standard solver. The difference in the solution between the two interpolation schemes was fairly small on all grid levels. The overset solutions underpredict the drag on the two coarsest levels by approximately 6% and 5%, respectively. On the finest grid level, however, the overset and non-overset solutions are in very good agreement, with similar amplitudes of the oscillatory response to both lift and drag. As will be shown subsequently, the apparently better agreement on the fine grid level was an artifact of a lack of time resolution.

Scheme	Grid Level 3			Grid Level 2			Grid Level 1		
	$ C_l $	C_d	St	$ C_l $	C_d	St	$ C_l $	C_d	St
Bilinear	0.242	0.556	0.226	0.229	0.626	0.251	0.601	0.743	0.261
Lagrange	0.242	0.561	0.228	0.229	0.629	0.252	0.601	0.743	0.261
Non-overset	0.322	0.592	0.226	0.300	0.656	0.254	0.579	0.745	0.261

Table 3.4: Dependence on overset interpolation scheme. Lift amplitude, average drag values and Strouhal number (based on the cylinder diameter) at the three grid levels computed on the overset grid C2 and the non-overset grid C1.

These results indicate that the two interpolation schemes perform very similarly, although the Lagrange scheme seems to be slightly more accurate on coarser grids which is consistent with the results of the laminar flow case.

The bilinear interpolation scheme, however, has the clear advantage that it is very compact with a two cell stencil in each coordinate direction, as opposed to the Lagrange scheme which requires three cells in each coordinate direction and is more costly. However, before any final conclusions can be drawn, further studies are required, which are presented in the subsequent sections.

A time step study was conducted on the finest grid level at three time steps: $\Delta t = 1 \times 10^{-2}$, 5×10^{-3} , and 2.5×10^{-3} . The results are summarised in Table 3.5. At the coarsest time step the agreement between the two grids is very good. At the smaller time steps the agreement in the drag coefficient is poorer with a discrepancy of 2.7% and 3.1 %, respectively. The Strouhal number increased slightly with decreasing time step, with good agreement between the two solutions at all time steps.

The temporal order of the numerical scheme, σ was estimated according to Equation 3.1. Both the overset and non-overset solution exhibit second order convergence rates, confirming that the overall temporal accuracy is maintained by the overset solver.

Time step	$ C_l $		C_d		St	
	C1	C2	C1	C2	C1	C2
0.01	0.579	0.601	0.743	0.745	0.261	0.261
0.005	0.427	0.406	0.708	0.689	0.265	0.266
0.0025	0.397	0.351	0.699	0.677	0.265	0.265
σ	-	-	1.96	2.22	-	-

Table 3.5: Dependence on time step. Amplitude of lift, average drag values and Strouhal number (based on the cylinder diameter) at the three time steps.

As discussed in Section 2.3.4 the implementation of the $k - \omega$ SST turbulence model involves the computation of two zonal blending functions, which relate to the normal distance to the nearest solid surface. Since this computation is relatively costly, it has not been implemented in the overset version of the code for moving grid problems where the normal distances would have to be recomputed in every time step. This means that on overset grids, the distances to solid walls will be infinite in block groups that are detached from the solid surfaces, effectively reducing the SST model to a $k - \epsilon$ model. To investigate the influence of the incorrect implementation of the SST model, the code was modified such that the normal distance was computed correctly (possible for this simple geometry, but more costly in the general case). Figure 3.8 shows an instantaneous contour plot the two zonal functions F1 and F2 computed on grid C2. F1 decays very quickly in the wake of the cylinder,

whereas F2 extends a fairly long distance downstream of the cylinder.

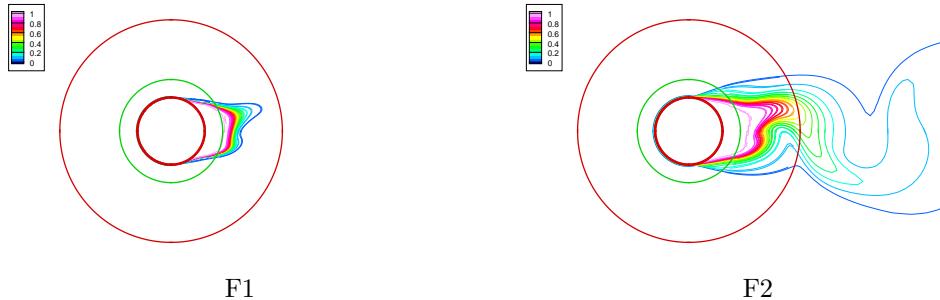


Figure 3.8: $k - \omega$ SST turbulence model zonal functions.

A comparison of the drag coefficients obtained using the incorrect implementation and the corrected implementation revealed that the discrepancy between the overset and non-overset solutions in the time step study was caused by the turbulence modelling. At a time step of $\Delta t=0.005$, the difference between grid C1 and C2 was negligible when using the correctly implemented SST model.

It was also likely that the level of the eddy viscosity would differ in the wake of the cylinder since the blending function F2 as shown above was significant several cylinder diameters downstream. The velocity and eddy viscosity were extracted 3.5 diameters downstream of the cylinder across the wake and averaged over one shedding period to investigate whether any difference could be identified in the development of the wake. The wake profiles are illustrated in Figure 3.9. The correctly implemented SST model was in very good agreement with the non-overset solution, whereas the discrepancy between the incomplete implementation (here referred to as the $k - \epsilon$ model) amounts to about 10%. Reducing the turbulence modelling to a $k - \epsilon$ model results in an increase in the velocity deficit and a higher level of eddy viscosity.

As shown above, computations performed on overset and non-overset grids with identical topologies generally agree very well. The compatibility in the overlap region for grid C2 was optimal, since the cell sizes and orientation were very similar which will increase the accuracy of the solution. When using overset grids, however, it is often more advantageous to combine body fitted grids with background Cartesian grids that resolve the nearbody flow and farfield. With such a grid topology it is generally harder to maintain good compatibility particularly on coarse grids as discussed in Section 3.2.1. To investigate the influence of mesh compatibility the flow was computed using

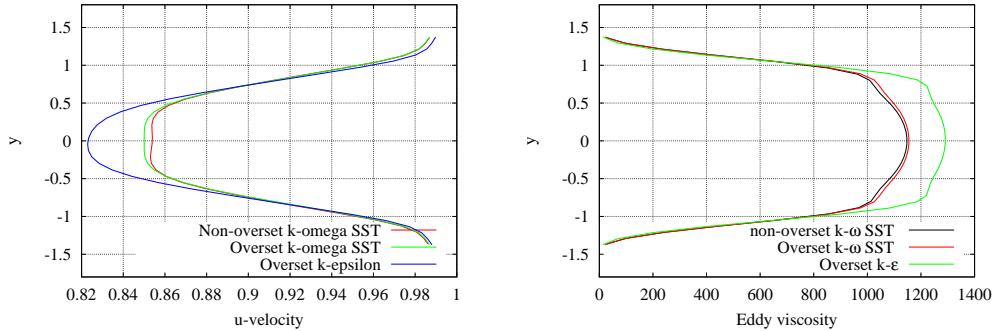


Figure 3.9: Wake profile for incorrect and correct implementation of the $k-\omega$ SST turbulence model. Time averaged over one shedding period.

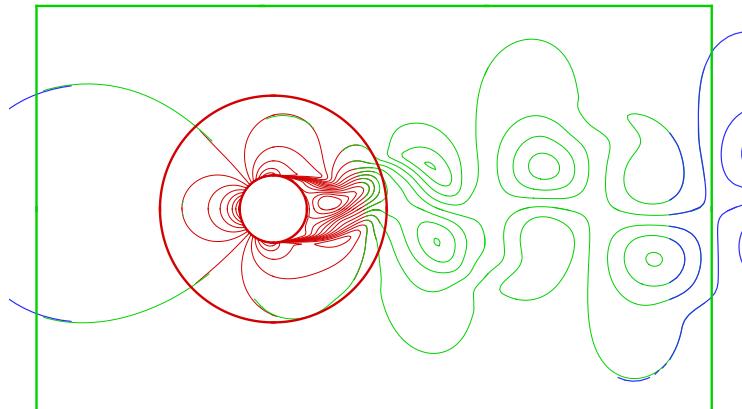
grid C3 which contained three block groups, one curvilinear grid containing the cylinder, and two Cartesian grids to resolve the rest of the domain.

The magnitude of the oscillatory lift coefficient and mean drag coefficient of mesh C3 are listed in Table 3.6 for a time step of $Deltat = 0.005$. Although the two overset solutions differ slightly, they are both within about 3% of the non-overset solution.

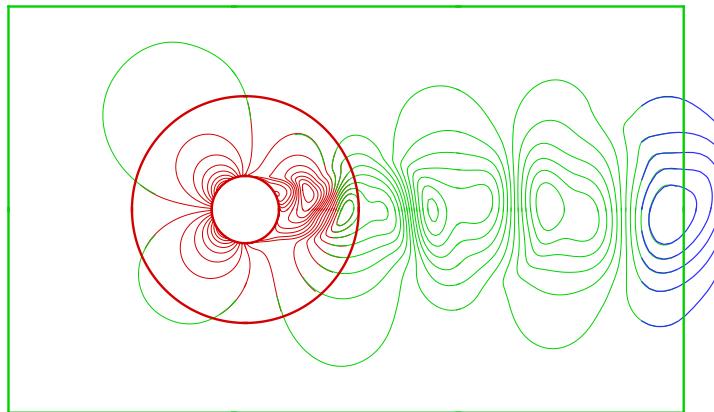
Grid	$ C_l $	C_d	St
C1	0.427	0.708	0.265
C2	0.406	0.689	0.266
C3	0.398	0.728	0.277

Table 3.6: Magnitude of lift, average drag, and Strouhal number at $\Delta t = 0.005$.

The contour plots of velocity and pressure gradients in Figures 3.10 and 3.11 show that the solution is smooth across interfaces. The vorticity and eddy viscosity contour plots in Figures 3.12 and 3.13 illustrate that the unsteady vortex street is propagated downstream across the overset interfaces without any distortion.

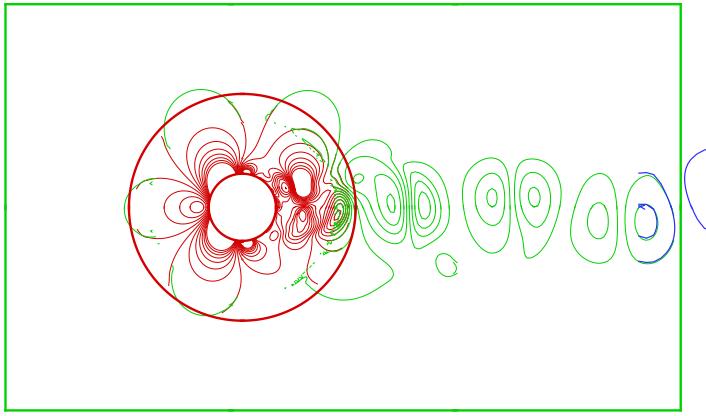


u-velocity

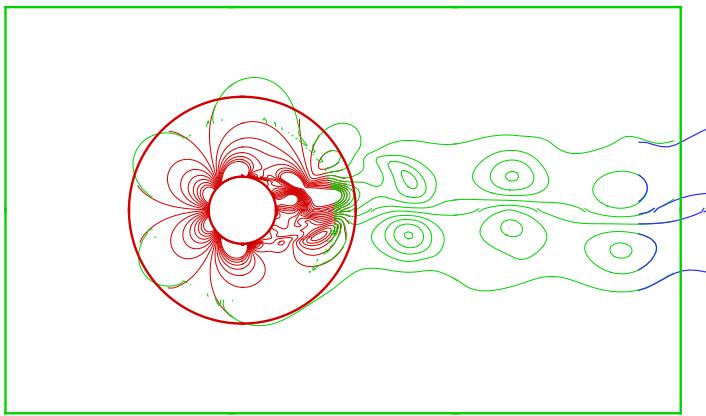


v-velocity

Figure 3.10: Contour plots of velocity on a circular cylinder at $R_e=1.84 \times 10^5$ for grid C3. The outer boundaries of group 1 and 2 are indicated with red and green outlines, respectively.



x-pressure gradient



y-pressure gradient

Figure 3.11: Contour plots of pressure gradients on a circular cylinder at $R_e=1.84 \times 10^5$ for grid C3. The outer boundaries of group 1 and 2 are indicated with red and green outlines, respectively.

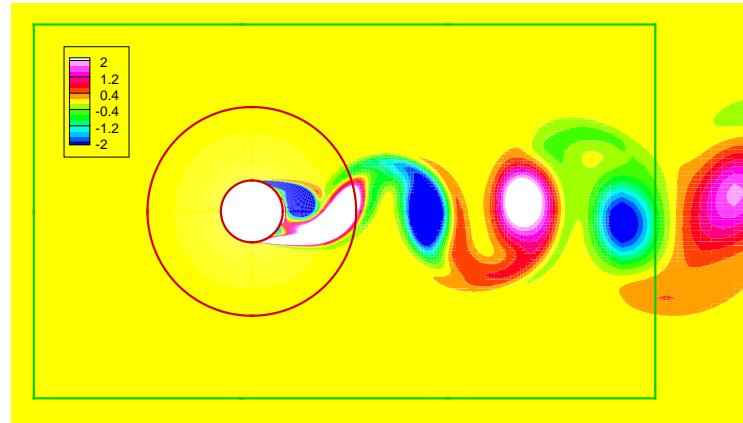


Figure 3.12: Contour plot of vorticity on a circular cylinder at $R_e=1.84 \times 10^5$ for grid C3. The outer boundaries of group 1 and 2 are indicated with red and green outlines, respectively.

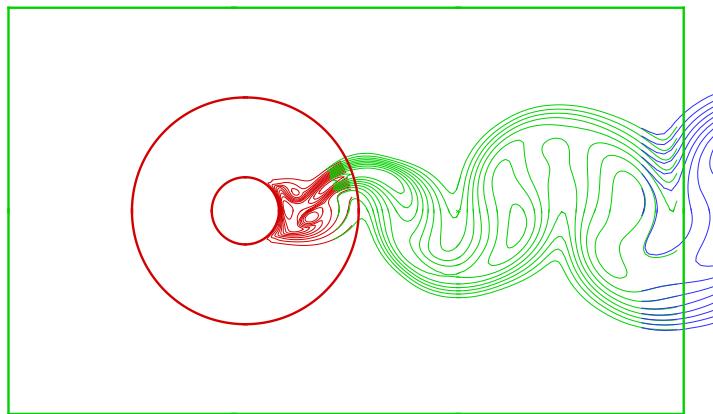


Figure 3.13: Contour plot of eddy viscosity on a circular cylinder at $R_e=1.84 \times 10^5$ for grid C3. The outer boundaries of group 1 and 2 are indicated with red and green outlines, respectively.

3.3 Flow Over the NACA 63₂ – 415 Aerofoil

Another appropriate choice for validation of the overset grid method was the computation of flow over the NACA 63₂ – 415 aerofoil since it has previously been studied computationally using the non-overset version of the present code and validated against various experimental results [8]. It furthermore represents an important class of problems, since the code is used extensively during the design and validation process of wind turbine rotors.

The test case which was chosen for the current study is flow over the NACA 63₂ – 415 aerofoil at $R_e = 3 \times 10^6$, since experimental data is available from the wind tunnel tests by Abbott and von Doenhoff [1]. To reduce the computational cost all computations were done in two dimensions in a steady state manner. The SUDS scheme was used to discretise the convective terms and the SIMPLE algorithm was used to solve the coupled pressure/velocity equations. The $k - \omega$ SST turbulence model by Menter [41] was used due to its good performance in adverse pressure gradient flows.

3.3.1 Computational Meshes

Computations with the original EllipSys2D code were carried out using an O-grid configuration with 256 cells in the circumferential direction and 128 cells in the normal direction. The height of the first cell in the boundary layer was 1×10^{-5} chord lengths with the outer domain placed 20 chord lengths away from the surface. This mesh will be referred to as mesh A1.

Three overset grid configurations were made: The first and most simple one consisted of two overlapped O-meshes, one body fitted mesh containing the aerofoil, and one resolving the farfield. Topologically it was identical to the non-overlapping mesh, except that the meshes do not match on the overset interface. To eliminate as many mesh related solution discrepancies as possible, the curvilinear grid was made identical to the inner part of the non-overset grid, extending one chord length outwards from the surface containing 64 cells in the normal direction. The outer O-mesh contained 256×64 cells with its inner boundary placed 0.3 chord length from the surface of the aerofoil and was grown 20 chord lengths outwards. This mesh will be referred to as mesh A2.

The second overset configuration contained the same curvilinear body fitted mesh, but was embedded in one Cartesian mesh with dimensions 40×40 chord lengths, containing 192×160 cells in the chordwise and normal direc-

tions, respectively. This mesh will be referred to as mesh A3.

The third configuration, mesh A4, consisted of three groups, one curvilinear mesh embedded in two Cartesian meshes; one resolving the nearbody flow field with dimensions 6×4 and a coarser mesh to resolve the farfield flow with dimensions 40×40 . Both Cartesian meshes contained 128×128 cells, and were stretched towards the aerofoil grid to increase the compatibility in the overlap regions. The three overset meshes are shown in Figures 3.14 and 3.15.

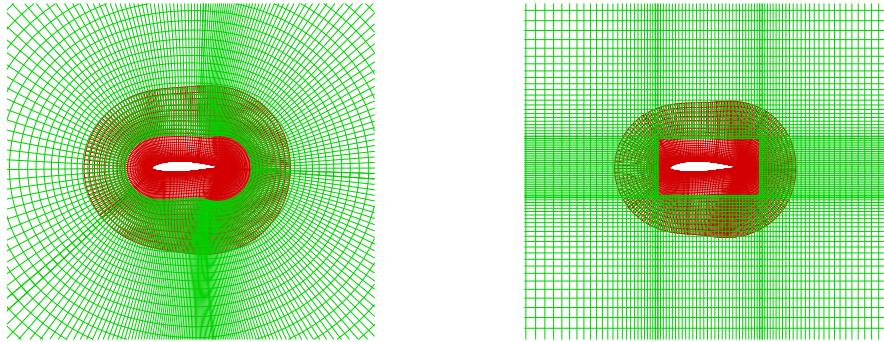


Figure 3.14: Mesh configuration A2 and A3 for the NACA 63-415 aerofoil.

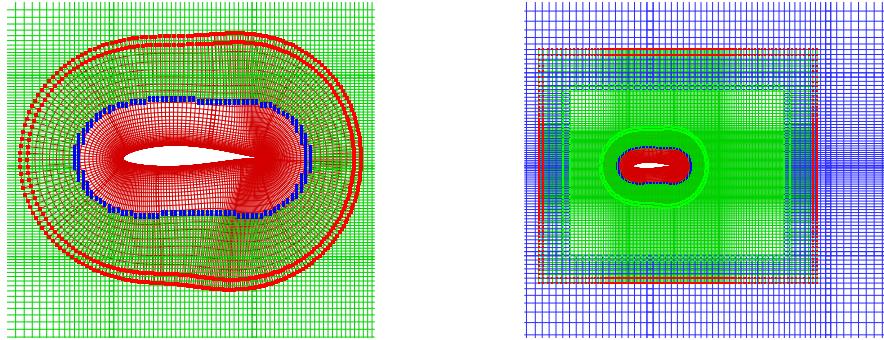


Figure 3.15: Mesh configuration A4 for the NACA 63-415 aerofoil.

3.3.2 Results and Discussion

As discussed in Section 2.3.2, the overlap size is mainly dictated by the size of the interpolation stencil used on overset interfaces. It is necessary that the interpolation is explicit, that is, that no cells that themselves are fringe cells are used to interpolate flow variables. As the divergence is largest near overset boundaries, the solution will also be improved by increasing the overlap, which

is a well known fact from standard domain decomposition techniques.

To investigate these properties a series of computations of steady flow over the NACA 63₂ – 415 aerofoil were made using the double O-mesh configuration (Grid A2). The overlap size was changed by blanking out cells in the outer grid, while keeping the position of the boundary of the inner grid fixed. The results were compared to computations on the equivalent non-overset mesh A1.

Figure 3.16 shows the lift and drag polars for the overset O-mesh configuration for various overlap sizes compared to the reference O-mesh. The overset solution were found to be in very good agreement with the non-overset solution, and no significant difference was observed with variation in overlap size.

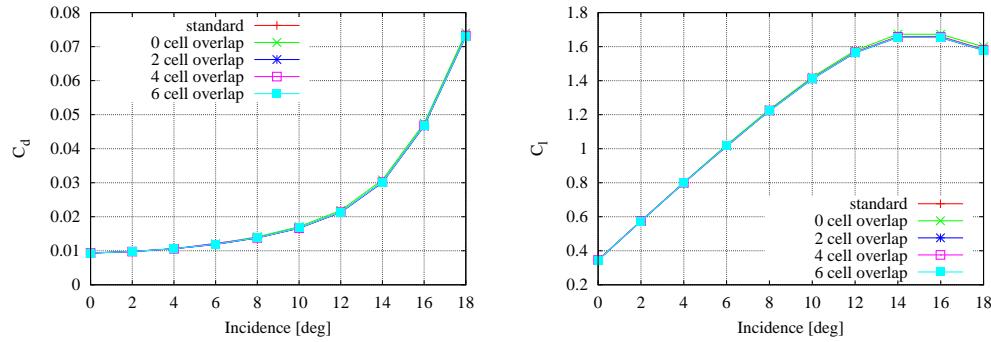


Figure 3.16: Computed lift and drag coefficients for the NACA 63₂ – 415 aerofoil using the non-overset solver compared to the overset solver with various overlap sizes.

Similarly, as shown in Figure 3.17, the residual reduction is equivalent for all overlap sizes, and behaved identically to the non-overset solution.

Table 3.7 shows the variation of the divergence error with overlap size. For a zero cell overlap the interpolation becomes implicit which is clearly reflected in the high divergence error in Group 1. The error in Group 1 does reduce with overlap size, although the reduction is very small for overlaps larger than one cell. For Group 2 the trend is not as consistent, with a rather large error for an overlap of 4 cells. Although no significant difference was observed in the integral forces on the aerofoil with variation of overlap, the above observations indicate that at least a one cell overlap is needed to obtain a good solution.

The second test performed on the overset O-mesh configuration involved coarsening the mesh successively by a factor of two in both coordinate directions with a total of three levels. The finest grid thus consisted of two block groups of 16 patched grids of 32 × 32, the second finest coarsened to blocks

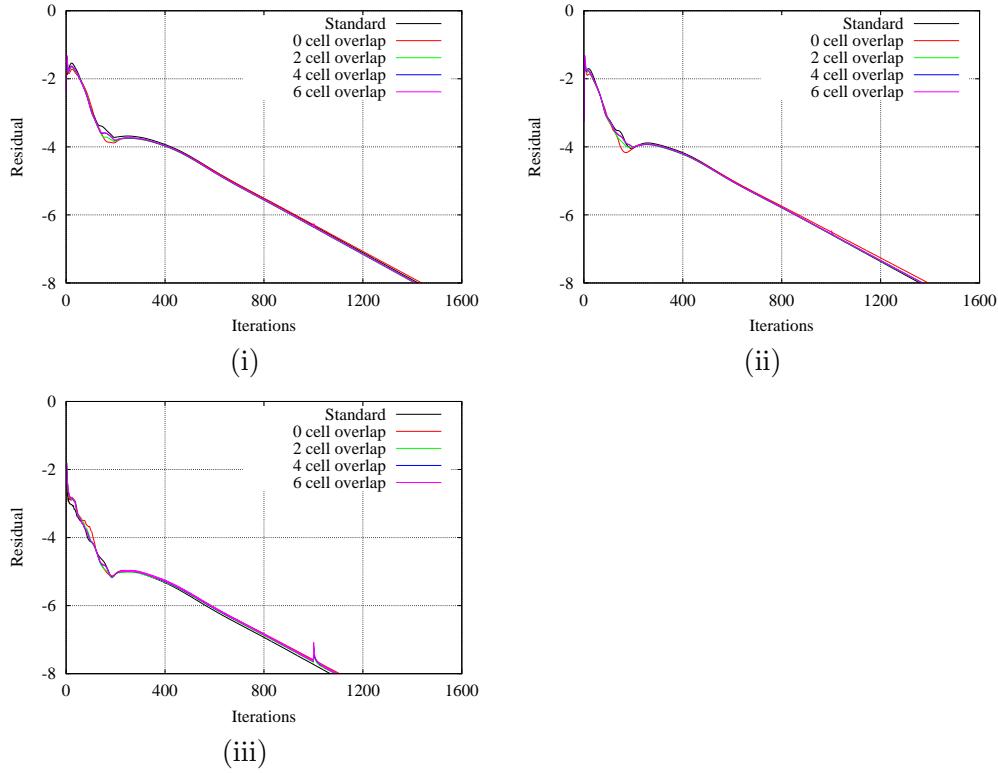


Figure 3.17: Residual history of (i) u-velocity (ii) v-velocity and (iii) pressure for various overlap sizes.

Overlap size	Divergence Group 1	Divergence Group 2
0	2.014×10^{-5}	5.578×10^{-5}
1	7.653×10^{-7}	2.265×10^{-6}
2	6.412×10^{-7}	8.362×10^{-7}
4	3.902×10^{-7}	1.133×10^{-4}
6	2.531×10^{-7}	7.634×10^{-6}

Table 3.7: Variation of integrated divergence error with overlap size for group 1 and group 2.

of 16×16 , and finally the most coarse grid with 8×8 cells in each block. As hole-cutting was not used in the computation, the inner boundary of group two did not move between each level. The number of cells in the overlap was thus reduced by a factor of two between the levels, with a ten cell overlap on the finest level, five on the middle level and one overlap on the coarsest level. Figure 3.18 shows the lift and drag polars for the three grid levels for the standard and overset solvers showing good agreement at all refinement levels.

The third test was made to investigate the solver's ability to handle irregular internal fringe boundaries as a result of cells in the background mesh being blanked. Two overset configurations were investigated; one was the overset

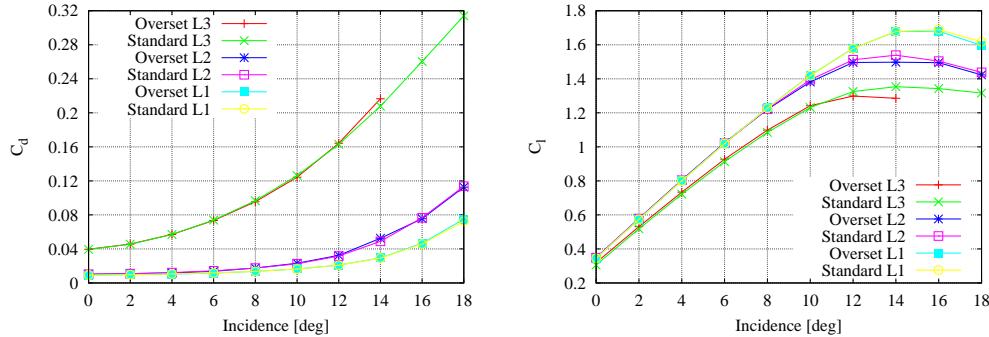


Figure 3.18: Solution dependence on mesh coarseness for the standard and overset solver.

grid A3 which contained two overset block groups, one curvilinear and one Cartesian, and the other was mesh A4 where the farfield was discretised using two Cartesian groups. Mesh A3 was designed such that no irregular boundaries were created due to hole-cutting since an entire block was blanked. This made it logically identical to grid A2. With this as a reference mesh, discrepancies in the solution due to differences in topology between the O-type and Cartesian meshes could be identified. Two solutions were compared to the solution obtained on grid A2.

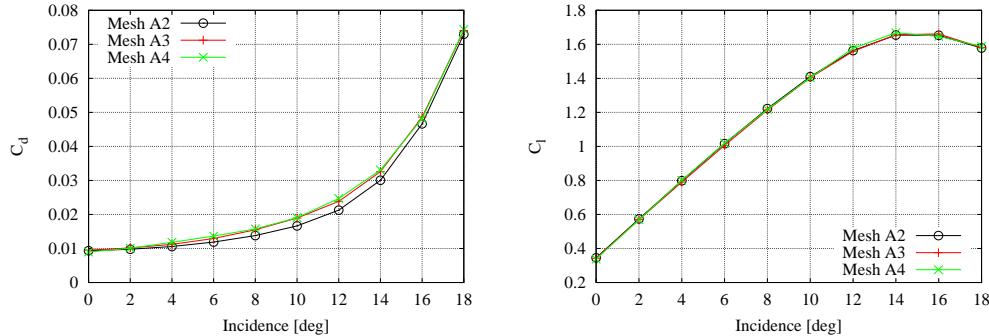


Figure 3.19: Solution dependence on mesh topology

Figure 3.19 shows the lift and drag polars for grid A2, A3 and A4. The lift and drag computed from mesh A3 and A4 agree very well, indicating that the use of hole-cutting where irregular fringe boundaries are created has no adverse influence on the solution. Both differ from the solution on mesh A2 by as much as 6% in the attached flow region for the drag, whereas the lift agrees very well. These findings agree with the results for the circular cylinder with similar mesh configuration presented in Section 3.2.3, where a similar discrepancy was found between the double O-grid configuration and

the combined curvilinear Cartesian mesh. This indicates that the discrepancy is caused by the difference in topology between the overset grids, which will influence the interpolation of flow variables across overset interfaces, and not the blanking of interior cells.

The computations on grid A4 were compared to experimental data from Abbott and von Doenhoff [1]. Since the flow is likely to be transitional on both lower and upper surfaces, the by-pass transition model by Drela and Giles [13] was used in the computation. Figure 3.20 shows the lift and drag polars for the computation and the experiment. For reference the fully turbulent computation was also included in the plot. The agreement of the lift is quite good in the linear region, whereas the stall is not captured as well. Both computations over-predict the lift for incidences above $\alpha = 10^\circ$ with stall not occurring until $\alpha = 16^\circ$. This result is not surprising, since it is well known that 2D computations rarely succeed in predicting the stall characteristics accurately. The agreement in the drag is good for the computations where transition is included, whereas the fully turbulent computations overpredict the drag at low incidences, where most of the flow on the pressure side is likely to be laminar.

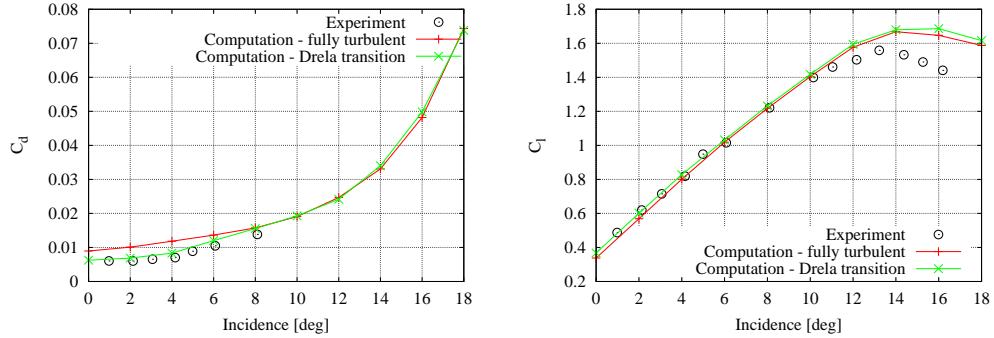


Figure 3.20: Comparison of computations using grid A4 with experimental data [1].

3.4 Dynamic Stall of the NACA 0015 Aerofoil

The flow over a pitching NACA 0015 aerofoil was chosen as the third test case to validate the overset solver for a moving grid unsteady flow problem. Comprehensive experimental studies have been carried out on this profile by Galbraith et al. [19] which have been used extensively in computational validation, see for example Ekaterinaris et al. [16], Sørensen [58] and Johansen [28]. Two cases were chosen from the experiment, one involving light stall and another involving deep stall. The aerofoil motion was prescribed as a sinusoidal pitching motion around the quarter chord with a reduced frequency of $k = 0.102$, where $k = \frac{c\omega}{2U_{rel}}$, where c is the local chord length, ω is the characteristic frequency, and U is the inflow velocity. The amplitude of the motion was $\pm 7.55^\circ$ around a mean incidence of $\alpha = 11.37^\circ$ for Case 1 and for Case 2 a reduced frequency of $k = 0.154$, amplitude of $\pm 6.83^\circ$ around a mean incidence of $\alpha = 19.58^\circ$. The experiment was carried out at a Reynolds number based on the chord length of $R_e = 1.5 \times 10^6$.

The computational meshes had identical dimensions to the non-overset mesh A1 and the overset mesh A4 described in Section 3.3.1 and will therefore not be described in any further detail. The SUDS scheme was used to discretise the convective terms and the PISO algorithm was used to solve the coupled velocity/pressure equations. The non-dimensional time step was set to $Deltat = 0.01$ using six subiterations per time step. Each computation was run for five cycles. The turbulence was modelled using the $k - \omega$ SST turbulence model Menter [41].

In the overset computation, only the body fitted grid around the aerofoil was specified to move, whereas the two Cartesian background meshes were stationary. A consequence of the relative movement between the aerofoil grid and the inner Cartesian grid was that the hole region defined by the aerofoil mesh and its interpolation coefficients had to be recomputed at every time step. The relative grid movement introduces additional issues related to the movement of the hole region. Cells that were previously not part of the solution are un-blanked (and vice versa) during the movement of the hole region. This test case therefore also serves to validate the method employed to handle these issues as described in Section 2.3.5.

Hysteresis curves of normal force, tangential force, and pitching moment coefficients, C_n , C_t , and C_m for the last three cycles are shown in Figures 3.21 and 3.22 for the two cases. Looking firstly at the light stall case illustrated

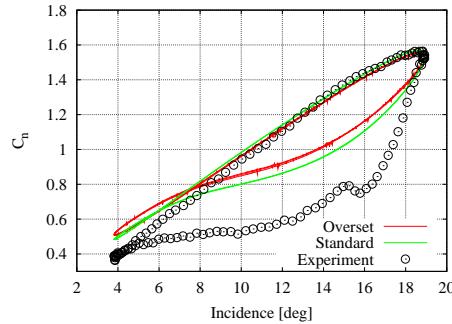
in Figure 3.21 it can be seen that the upstroke is captured quite well by the computations. In the experiment a sharp drop is seen in the normal force in the initial part of the downstroke, which is not captured by the computations. Likewise, the computations fail to predict the drop in the moment coefficient. The likely cause of the poor agreement is that the trailing edge separation is not predicted correctly. The lack of separation in the computations causes the dynamic stall behaviour to become almost linear.

The deep stall case is for the most part captured quite accurately by the computations. The linear behaviour of the upstroke is in good agreement with the experiment. As the trailing edge separation reaches the leading edge, a large vortex forms from the leading edge which travels downstream and causes an increase in the suction on the upper surface. As the vortex leaves the trailing edge a sharp increase in the normal force on the aerofoil is seen. The computations overpredict the increase in lift quite significantly which could be caused by a lack of mesh resolution at the trailing edge. On the downstroke the computations predict the decrease in normal force associated with the fully stalled condition reasonably well. The computations predict vortex shedding to take place during the remainder of the downstroke, causing the normal force to fluctuate excessively, which is not seen to the same extent in the experiment. The same can be said for the tangential and moment coefficients where the upstroke and peak tangential force and pitching moment is captured well.

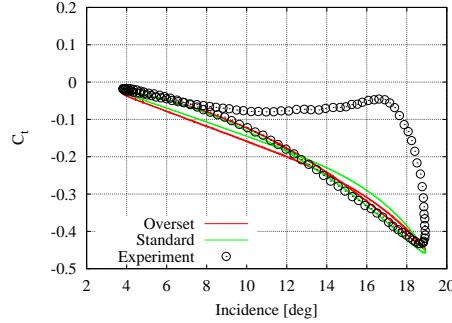
The agreement between the overset and standard solver is favourable, although the downstroke for both cases is predicted slightly differently. In the light stall case the hysteresis is not as pronounced in the overset solution, indicating that the separation bubble at high incidences is smaller. On the downstroke in case 2 the flow is fully separated, which suggests that the incomplete implementation of the $k - \omega$ SST model (as discussed in Section 2.3.4) may also have an influence on the results.

Finally, a comparison was made between the bilinear and Lagrange interpolation schemes on Case 2. As with the previously presented cases, no difference was observed in the solution, as is shown in Figure 3.23.

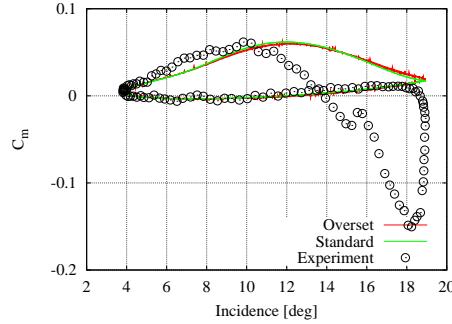
The lack of agreement of the computations in the reattachment phase during the downstroke can mostly be attributed to inadequate turbulence modelling, particularly evident in the light stall case. Additionally, it is well known that massively separated flow over aerofoils is dominated by complex three-dimensional effects which have a significant influence on the separation patterns on a wing section. A two-dimensional representation of the flow



(i)

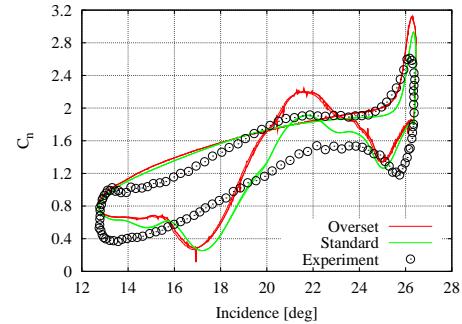


(ii)

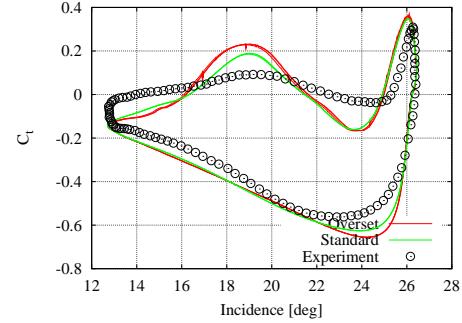


(iii)

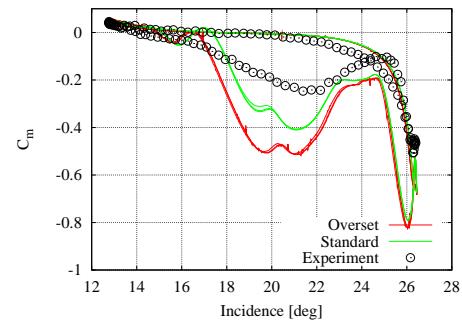
Figure 3.21: Hysteresis curves of unsteady turbulent flow over the NACA 0015 aerofoil at $R_e = 1.5 \times 10^6$. Computations for mesh A1 and A4 compared to experimental data from [19]. Prescribed pitching motion: $\alpha(t) = 11.37^\circ + 7.55^\circ \sin(2kt)$, where $k = 0.102$



(i)



(ii)



(iii)

Figure 3.22: Hysteresis curves of unsteady turbulent flow over the NACA 0015 aerofoil at $R_e = 1.5 \times 10^6$. Computations for mesh A1 and A4 compared to experimental data from [19]. Prescribed pitching motion: $\alpha(t) = 19.6^\circ + 6.86^\circ \sin(2kt)$, where $k = 0.154$

is therefore not likely to capture the physics very well when the aerofoil is fully stalled. The main result of this test case, however, is that the overset solver was in very good agreement with the original solver. The deep stall case involved massively separated flow with large separation bubbles being shed from the aerofoil. Figure 3.24 shows the flow over the pitching aerofoil for the $k = 0.154$ case at two instances during the downstroke where the leading edge vortex detaches from the surface. These flow features appear

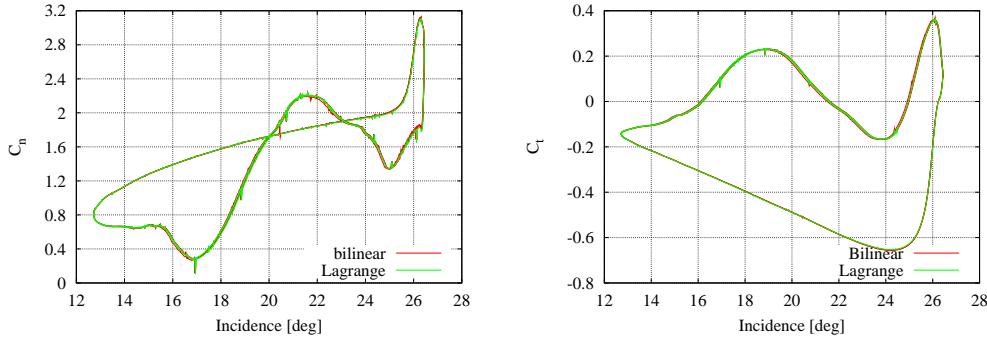


Figure 3.23: Comparison of the bilinear and Lagrange interpolation schemes on Case 2.

to be advected across the overset interfaces without distortion, confirming that a simple interpolation of flow variables does not deteriorate the solution accuracy. The method employed for handling movement of hole regions also appears to perform well for this case.

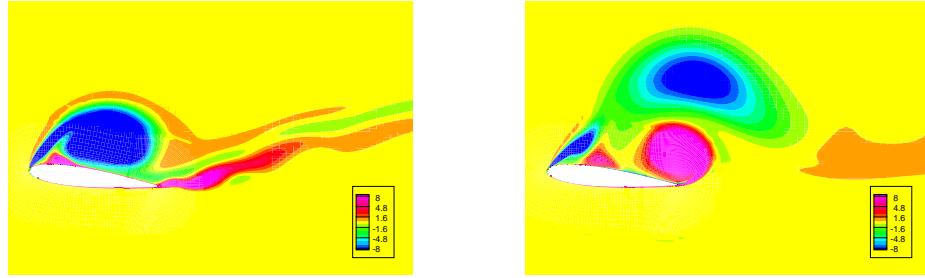


Figure 3.24: Instantaneous vorticity contours for $k = 0.154$ during the downstroke.

3.5 Steady Flow Over a 6:1 Prolate Spheroid

To validate the 3D version of the code turbulent flow over a 6:1 prolate spheroid (length L and diameter D) at 20° incidence was used as a test case. Experiments have previously been carried out on this configuration, see Wetzel [70], which makes it well-suited for the present purpose. In the experiment, the Reynolds number was equal to 4.2×10^6 and the flow was tripped at $x/L=0.2$. The skin friction and the pressure was measured at $x/L=0.729$ and $x/L=0.772$, respectively.

In this numerical experiment, two mesh configurations were used: one non-overset O-grid and a corresponding overset grid with the same configuration.

The computations were carried out assuming laminar flow ahead of the trip wire placed at $x/L=0.2$ and fully turbulent flow aft of this location using the $k-\omega$ turbulence model [71]. The flow was computed in a steady-state manner using the SIMPLE algorithm for the pressure/velocity coupling. The SUDS scheme was used for the convective terms.

3.5.1 Computational Meshes

Two meshes were generated for the present test case: one standard non-overset grid, S1, and an overset grid, S2, with the same topology as the standard mesh. For the two meshes the same surface mesh was generated, which consisted of 256 cells around the circumference of the body, 128 cells in the spanwise direction and two blocks of 64^2 cells on the tips, totalling ten blocks of 64^2 cells. The surface mesh is shown in Figure 3.25.

For mesh S1 the volume grid was generated using a first cell height of $1 \times 10^{-5}D$, grown in the normal direction approximately $3L$ using 128 cells stretched using a tangent hyperbolic function. The mesh thus consisted of a total of 20 blocks of 64^3 cells. The overset grid S2 was generated using two block groups each consisting of 10 blocks; one body-fitted grid, that was grown normal to the body to a distance of 1D again using a first cell height of $1 \times 10^{-5}D$ with a total of 64 cells in the normal direction. The outer block group was identical to the one used in mesh S1 to minimise mesh dependent solution differences. As such the overlap between the two grids was quite large. The grid is shown in Figure 3.25.

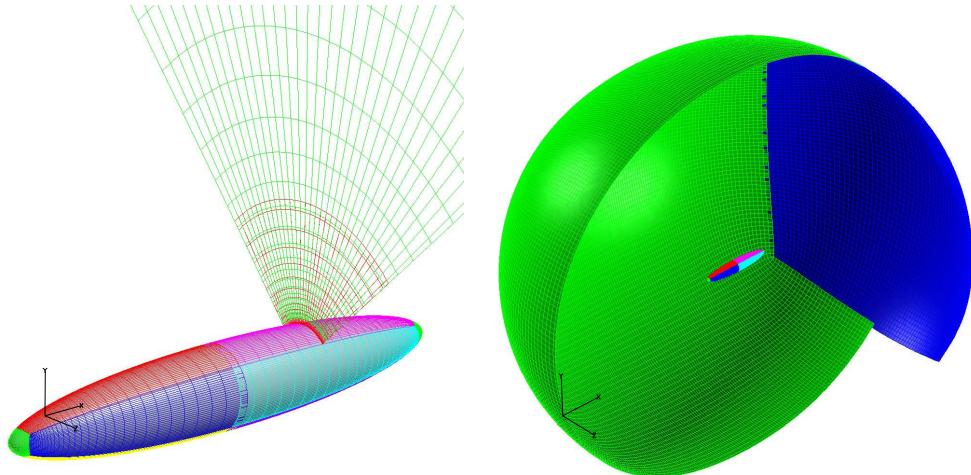


Figure 3.25: Grid S1 spheroid mesh layout showing only every second grid cell for clarity. Inflow is prescribed in the x -direction.

3.5.2 Results and Discussion

Firstly, the integrated lift and drag coefficients were compared for the two solutions obtained using mesh S1 and S2 as shown in Table 3.8. There is good agreement between the two solutions, with the largest error of 1.6% on the drag. The difference in the solution is likely to be associated with the difference in the resolution of the boundary layer in the body fitted meshes of mesh S1 and S2, which for this flow is likely to be quite sensitive to changes due to the complex nature of the separation pattern on the leeside of the body.

Grid	C_l	C_d
Non-overset	0.1524	0.07860
Overset	0.1515	0.07734
% difference	0.6	1.6

Table 3.8: Integrated lift and drag coefficients on the spheroid for the overset and non-overset computations.

Figure 3.26 shows the skin friction coefficient and pressure coefficient at $x/L=0.729$ and $x/L=0.772$, respectively. The two solutions obtained using EllipSys3D are in good agreement with only slight discrepancies on the leeside of the spheroid. The skin friction is under-estimated in the computations, which could be related to the method used to emulate the trip of the flow in the experiment. However, the qualitative agreement is reasonably good, capturing the separation point of the shear layer at approximately 120° from the windward side. The pressure distribution is also in good qualitative agreement, again capturing the separation point on the leeward side well.

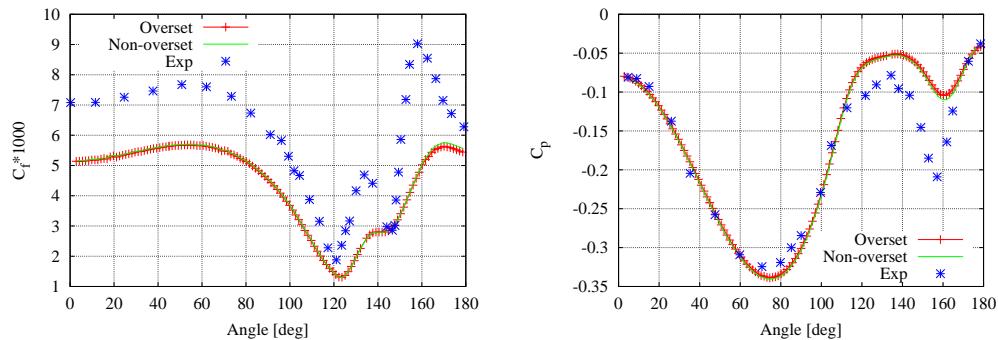


Figure 3.26: Plots of C_f and C_p at $x/L=0.729$ and $x/L=0.772$, respectively, for the overset and non-overset solutions. Angle is measured from the windward side.

Figure 3.27 shows contour plots of vorticity magnitude and pressure gradients at $x/L=0.772$. The rolled up shear layer separating from the surface is

represented smoothly in the overset grid region with little differences in the solution between the two block groups. Also the pressure gradients are transferred correctly between the two grids (note that the discontinuous gradients along the boundaries are due to postprocessing, and has no consequence in the solution).

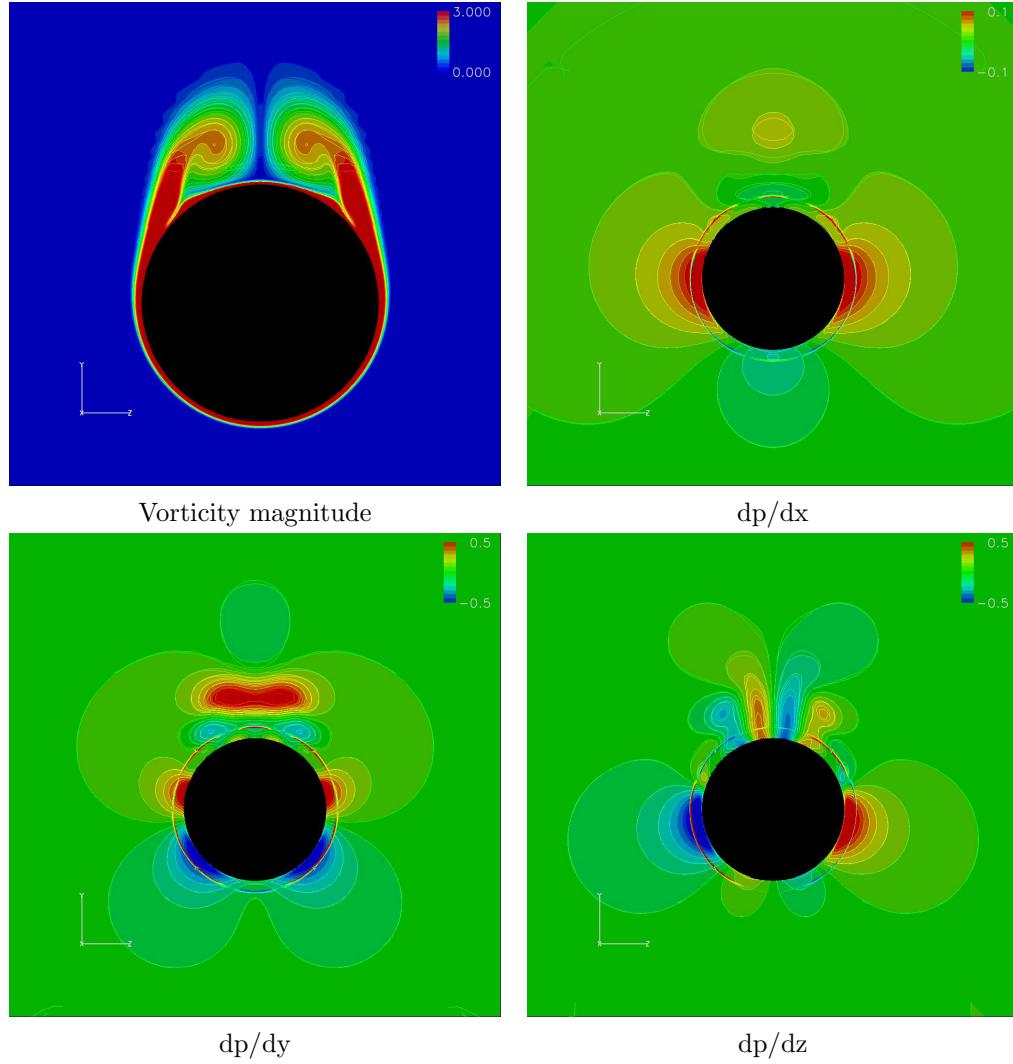


Figure 3.27: Contour plots of vorticity magnitude and pressure gradients at $x/L=0.772$.

3.6 Parallel Performance

In this section a moving grid simulation of flow over a pair of spheroids will be used to investigate the overall performance of the overset grid solver with particular focus on the parallel performance. As discussed in Section 2.5 the

addition of overset grid capabilities to the flow solver presents serious challenges in terms of load balancing and parallel communication strategies. The imbalance between processors is caused by the imbalance in number of fringe cells and donor cells on each processor. The performance of the overset routines is therefore highly problem dependent, but is most of all related to the ratio of fringe cells to total number of grid cells, $n_{fg} = \frac{\sum_{np} \text{fringecells}}{\sum_{np} \text{gridcells}}$ on each processor.

The two most time consuming routines are the ones which handle the overset grid assembly `ConnectOG` and the exchange of flow field information `Copy3D0G`. In the following the parallel performance of these two routines will be assessed on a problem which is representative of a typical overset grid problem involving multiple bodies and a combination of curvilinear and Cartesian block groups. The two spheroids are placed perpendicularly to the freestream directly behind each other, with one of the spheroids rotating with a prescribed angular velocity. The flow is laminar at a Reynolds number of $R_e = 100$.

All computations presented in this thesis were computed on the Risø Central Computer Facility, “Mary”, which consists of 240 Dell PowerEdge 750 compute nodes, each equipped with single processor Intel Pentium 4 (Prescott) 3.2 GHz CPU’s and 1 MB cache. Each machine has 2 GB of memory. The nodes are connected by Gigabit Ethernet by means of a system of D-Link switches. Further details can be found in [31].

3.6.1 Computational Approach

The two spheroids were aligned horizontally with the x-axis with a distance from center to center of 2.5 diameters. The inflow was prescribed parallel to the z-axis. The grid consisted of a total of four block groups; two curvilinear around the bodies and two Cartesian to resolve the farfield. The curvilinear grids contained six blocks of 32×32 on the surface and was grown one diameter outwards using 32 cells totalling 196608 cells each. The nearbody Cartesian block group contained eight blocks of $32 \times 32 \times 32$ cells with physical dimensions $14 \times 14 \times 14$ diameters. The grid was stretched towards the outer boundaries. The outer Cartesian block group had the same number of cells as the nearbody group with physical dimensions $30 \times 30 \times 44$ in the x, y and z directions. The total number of cells in the grid was thus 0.99×10^6 divided in 28 blocks of $32 \times 32 \times 32$ cells. The grid is shown in Figure 3.28.

The solution was computed in an unsteady manner using the SUDS scheme

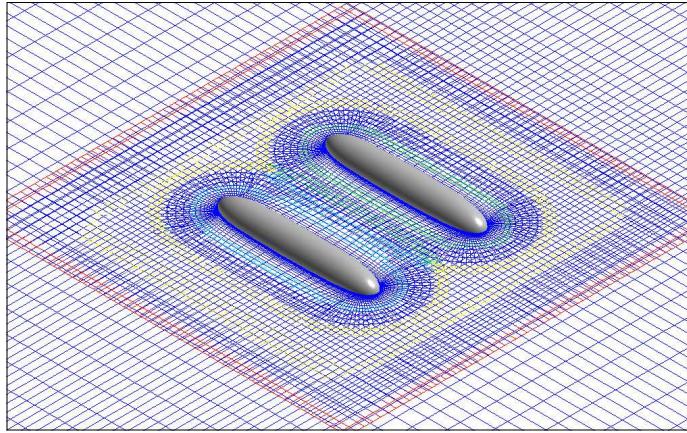


Figure 3.28: A mesh containing two spheroids with a total of four mesh groups. Colours along overset boundaries indicate donor grid identity.

to discretise the convective terms and the PISO algorithm to solve the velocity/pressure equations. The time step was set to $Deltat = 2 \times 10^{-3}$. In each time step six subiterations were performed. The downstream spheroid rotated at an angular velocity of 7.54 rad/s resulting in 0.432° per time step. Additionally, the spheroid was prescribed with a vertical oscillatory sinusoidal motion with an amplitude of 1 diameter and a period of 1.6 seconds. The computation was run for a total of 208 time steps, corresponding to half a revolution.

3.6.2 Results and Discussion

The number of fringe cells for the grid was 91247 resulting in $n_{fg}=0.091$. A load imbalance is caused by the fact that the nearbody Cartesian grid contains fringe cells on both outer face boundaries and internally in the group due to the blanking of interior cells. The curvilinear grids and the farfield groups on the other hand only contain outer boundary fringe cells or hole boundary fringe cells. Additionally, the imbalance was increased since grid connectivity information only had to be recomputed between grids with relative motion.

Table 3.9 gives a summary of the performance of the solver for different number of processors. The computational cost was measured as average wall clock time per node to complete the 208 time steps. The wall clock time spent in each part of the solver is also shown as a percentage of the total time to complete the computation. As described in Section 2.6 the routine `ExchangeOG` handles all interprocessor communication, which means that the time shown in the table comprises the communication performed in both `Copy3DG` and

ConnectOG.

Looking firstly at the serial computation of the problem, it can be seen that the overset routines occupy approximately 25% of the total time, which is dominated by the interpolation of flow field information. The fact that **ConnectOG** occupies such a small percentage of the total time is related to the specific problem. The relatively coarse grid resolution and moderate relative grid movement meant that the stencil jumping technique rarely failed, such that general searches for points were avoided. However, in other overset moving grid problems it was found that the search routine could occupy up to 25% of the total time.

In the parallel computations the **Copy3DOG** routine occupied approximately 22.5% of the total time, which compares reasonably well with the serial computation. The **ConnectOG** routine occupied on average approximately 3.5% of the total time, which is twice the time spent in the serial computation. Although the computational cost was reduced approximately 20 times with the use of the n-th level restart method compared to setting up the table with no prior knowledge of the location of fringe cells, the simple method for assembling search requests was the main cause for the cost increase. Individual processors thus issued a higher number of search requests to processors that did not hold valid donors for the points in question. In particular, processors holding background grids received many search requests for points located in regions that had been blanked out. Since requesting processors are not supplied with the hole boundaries of neighbouring processors, these searches are inevitable with the current method.

Procs	ExchangeOG	% time	Copy3DOG	% time	ConnectOG	% time	Total
1	-	-	6683	23	592	1.8	34707
2	1009	4.9	4256	20.8	512	2.5	20424
4	1295	11.7	2889	26	401	3.6	11047
7	900	12.5	1806	25	290	4	7182
14	528	14.2	887	23.9	135	3.6	3712
28	331	10.9	518	17.1	115	3.6	3023

Table 3.9: Computational cost (wall-clock time) of spheroid problem.

Figure 3.29 shows the parallel speedup of the two overset routines **Copy3DOG** and **ConnectOG** and of the total computational time. The **Copy3DOG** routine scales quite well because the only difference between the parallel and serial execution is one non-blocking MPI send/receive call using the **ExchangeOG**

routine, that only has barriers between individual processors, and not global ones. The `ConnectOG` routine scales more poorly, partly due to the search request method discussed above, and partly because there are a number of global communication barriers, which reduce the scalability of the routine. Since the `ConnectOG` routine comprises only a small percentage of the total computational cost, the overall scalability of the solver is not affected significantly by the relatively poor scalability of the connectivity routine. Additionally, this routine is only called once every time step, once the flow solution has been computed, and will therefore not affect the scalability of the flow solver significantly. The total speedup is more dependent on the speedup of the `Copy3DOG` routine. This is because the routine is called several times during the solution procedure, increasing the latency of the solution procedure.

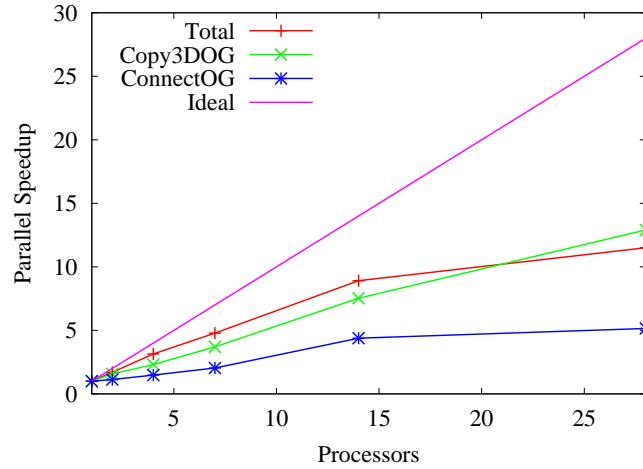


Figure 3.29: Parallel speedup for the spheroid problem.

As shown in Figure 3.30 the load balance of the three overset routines is not very good, with a load imbalance of up to 1.7 between processors. The load imbalance of `Copy3DOG` and `ConnectOG` are directly proportional to the imbalance in the distribution of fringe cells on the processors. Processors 7, 8, 9, and 10 have the highest loading. These processors contain the eight blocks making up block group 2, and as shown in Figure 3.31 this group has the highest number of fringe cells. The neighbouring processors 6 and 11 are also heavily loaded because these processors happen to be subject to many search requests.

From these results it is clear that a significant performance gain of the overset routines can be achieved by redistributing the blocks on the proce-

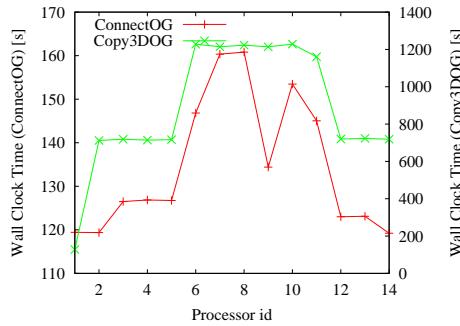


Figure 3.30: Load balance for the spheroid problem using 14 processors.

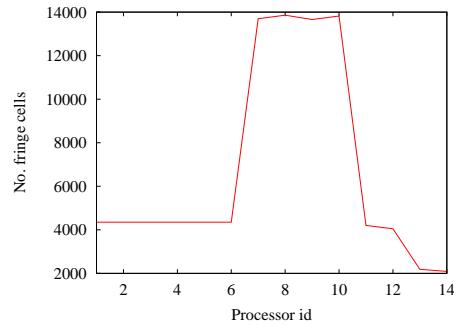


Figure 3.31: Distribution of fringe cells for the spheroid problem using 14 processors.

sors, such that a better load balance is achieved. However, as was found by Wissink and Meakin [73] such a redistribution can result in a significant performance degradation of the core routines in the flow solver. Another source of improvement is a better method for assembling search requests in the `ConnectOG` routine.

On a more general level, it is also worth discussing the strategy used in the grid generation. Since the present work aims to make simulation of bodies in relative motion possible, grids are as shown in Figure 3.28 constructed using a combination of curvilinear body-fitted grids and Cartesian background grids because of the considerable flexibility associated with such a grid layout. The most immediate disadvantage of such a grid assembly is that there is a fairly high proportion of wasted cells in the background grids due to the blanking of cells. Additionally, as is evident from the present results, there is a large load imbalance between processors holding the body fitted curvilinear grids and those holding the Cartesian background grids. Since hole-cutting is in the present problem only used on the Cartesian background grids, the number of fringe cells is therefore greater in the Cartesian background grids since they typically have both outer boundary overset interfaces as well as internal overset interfaces created by the hole-cutting. The cost of firstly cutting the holes, and secondly establishing the connectivity information is therefore much higher on these grids.

The EllipSys3D solver is purely written for the solution of problems on curvilinear grids, which means that there is no strategy in place to take advantage of the simplifications possible in the solver when using purely Cartesian grids. The restrictions inherent to the EllipSys solver with respect to grid topology thus makes it less suited to the present mesh generation strategy.

An obvious improvement of the performance of the solver would therefore be to have a dedicated solver for the solution in the farfield, which through the overlap interfaces is coupled to the solution in the body-fitted grids using the existing EllipSys solver. This could include solution strategies such as adaptive mesh refinement (AMR) which refines the discretised solution domain in regions with high flow gradients and coarsens it in farfield regions where flow gradients are low.

3.7 Conclusions

In this chapter a series of numerical test cases have been presented to validate the overset grid solver against both experimental results and equivalent non-overset solutions in a wide variety of flow regimes. It was found that the use of simple interpolation of the flow variables did not compromise the formal order of accuracy of the solver in both space and time. The application of a flux correction scheme ensured that full convergence could be achieved in the pressure-correction equation, and eliminated instabilities of the solver due to the non-conservative interpolation on overset interfaces.

Two different interface interpolation schemes were investigated for the three test cases. Very little difference was found in the results, although on coarse grids the Lagrange scheme seemed to be slightly more accurate. Chesshire and Henshaw [12] showed that for the most general case of overset grids the order of the interpolation scheme should be one higher than the order of the solution discretisation scheme. However, as long as the overlap is sufficiently large, the order of the interpolation scheme can be the same as the discretisation scheme. On the basis of the current results it is therefore concluded that the second order interpolation scheme is most appropriate for interpolation of flow variables between grids, with its main advantage being that it is more compact and computationally less expensive.

It was found that increasing the size of the overlap between overset grids reduced the divergence error, although for steady flow cases an overlap of one cell seemed sufficient.

It was shown that the overset solver was in very good agreement with the non-overset solver for all test cases presented. However, it was found that the use of a combination of curvilinear body fitted grids and Cartesian background grids produced slight differences particularly pronounced in the drag on the bodies. This discrepancy was not caused by any inconsistency in the solution

algorithm, but was a consequence of the differences in the topology of the grids, which influenced the interpolation of flow variables on overset interfaces.

The $k - \omega$ SST turbulence model was used in all computations although it is not fully implemented on overset grids, since the two blending functions F1 and F2 are not computed on grids not containing any physical boundaries. It was found that in the separated wake behind a circular cylinder this caused a discrepancy of approximately 10% in the velocity deficit and eddy viscosity. Given the general uncertainty associated with turbulence modelling of wake flows, it is reasonable to conclude that this discrepancy will not be of major importance to the accuracy of solutions of such flows. Agreement with available experimental results was also quite good, although the limitations of a two-dimensional modelling of the flow did give rise to discrepancies.

The 3D version of the code was validated using turbulent flow over a 6:1 prolate spheroid. The overset code was in very good agreement with the standard version of the code. These results were compared to experimental results, which showed that the skin friction in general was underestimated. However, both the computed skin friction and pressure was in good qualitative agreement with the experiment.

Lastly, the parallel performance of the overset grid solver was evaluated on a moving grid problem of flow over two spheroids undergoing oscillatory and pitching motion. The overall performance of the solver was limited by the poor load balancing of the overset grid problem caused by a large imbalance in the distribution of fringe cells. It was found that the connectivity routine did not scale as well as the routine exchanging flow field information, which was mainly due to the simple method used to assemble search requests on individual processors.

4

WIND TURBINE ROTOR-TOWER INTERACTION

4.1 Introduction

As most modern wind turbines are configured in an upwind configuration the issue of rotor-tower interaction has largely been focused on this configuration. When the rotor is situated upwind of the tower, the tower perturbation can with reasonable accuracy be modelled using potential flow theory, since the main contributor to the interaction is the reduction in velocity of the oncoming flow as a result of the presence of the tower. To obtain sufficient clearance, the distance between the rotor and the tower on an upwind rotor is always fairly high. The effect of the tower on the loads on the blades is therefore quite limited.

On a downwind wind turbine, however, the interaction between the rotor and the tower is much stronger, because the rotor passes through the tower wake, which is highly unsteady due to the bluff body vortex shedding inherent to high Reynolds number flow over circular cylinders. The interaction of the blade with the tower wake, or in short the blade-wake interaction (BWI), is therefore dominated by two effects: firstly, the presence of the tower creates a velocity deficit, which will cause the incidence on the blade to decrease rapidly when it travels through the the wake. Depending on whether the blade is unstalled or stalled the decrease in incidence will tend to cause either a decease or increase in loading, respectively. Secondly, the blade will occasionally encounter the vortices which are shed from the tower. The interaction between the blade and a vortex depends critically on the sense of rotation, orientation and strength of the vortex. This interaction has some similarities to that seen on helicopter rotor blades during certain types of maneuvering where the retreating blade encounters the tip vortex of the advancing blade. During such an event the vortex core is approximately parallel to the blade axis. This phenomenon is therefore referred to as parallel blade vortex interaction (BVI).

On a downwind turbine the interaction of the rotor blades with the shed vortices from the tower is also parallel in nature, since the blade axis will approximately be aligned with the tower when it passes through its wake. The shed vortices travel through the rotor plane rather than parallel to it, which will give rise to differences in the interaction from that on helicopters. Additionally, there is a considerable difference in the blade speeds on helicopters and wind turbines; however, common to both of them is that the convection velocity of the vortices is considerably lower than the blade velocity. When the wind turbine is operating under skewed inflow conditions the interaction will become oblique, meaning that the blade axis and the vortex core line will not be parallel, but at an angle to each other. Additionally, the vortex street is not necessarily aligned with the vertical axis of the tower in the streamwise direction, increasing the chances of oblique encounters.

The encounter of a blade with a vortex will produce a very sharp change in the incidence on the blade as the vortex passes. The change in incidence is caused by a change in the upwash or downwash on the blade during the vortex passage. The exact magnitude and sign of the change in loading depends strongly on the relative distance between the blade and the vortex, the strength and rotational sense of the vortex as well as the loading on the blade. The BVI on a downwind turbine is likely to be very unpredictable due to the difference in frequencies of the shedding and the rotation of the blade. The exact phasing of the interaction therefore changes from passage to passage, unless the two frequencies happen to coincide.

The interaction between the rotor and the tower is unique on a downwind turbine, because of the proximity of the two components. The passing blade is therefore likely to have a strong influence on the vortex shedding from the tower, possibly modifying both the frequency, phasing and strength of the vortex shedding. This will be caused by a combination of effects: a wind turbine rotor induces significant changes of the axial and tangential velocities due to the induction from the blades. The induced velocities upstream of the rotor disc are highest in the vicinity of the blades, which will result in the rotor exerting periodic velocity perturbations in the wake of the tower equal to the blade passage frequency. Additionally, the viscous interaction between the blades and the shed vortices is likely to have a strong effect on the formation process of the vortex street. Again, this interaction has a forcing frequency equal to the blade passage frequency. The effect from the passing blades will also have an equally significant influence on the tower shedding for

an upwind machine, although here it will not have much effect on the blade response making this phenomenon somewhat less significant with respect to blade loads. On a downwind turbine, however, the blade response is directly related to the shedding strength and frequency. This fact makes the study of this problem very intriguing but also very challenging. To be able to make a comprehensive analysis of the problem by computational means, it is therefore essential that the numerical method is capable of capturing both the effect of the tower wake on the rotor *and* the effect of the rotor on the tower wake.

The high interest in BVI in the helicopter community stems from the undesirable acoustic and vibrational effects associated with it. Likewise, on a downwind turbine, the main concern is the acoustic signature of the wind turbine, which must be kept at a minimum to reduce the disturbance to nearby inhabitants. A number of field studies on downwind turbines [30, 35, 56] reported that, under certain flow conditions, the interaction of the blades and tower wake produced high levels of low frequency noise, audible to the human ear. The apparently unpredictable nature of the occurrences of these so-called thumping noises was attributed to the unsteady nature of the interaction of the rotor with the wake of the tower. However, no precise correlation between flow conditions and noise emission could be made.

The large variation in noise emission is likely to be associated with the characteristics of the tower wake such as shedding frequency and coherence. These mainly depend on the Reynolds number (inflow velocity and tower diameter). Inflow conditions such as turbulence content and shear in the atmospheric boundary layer also have a significant influence on the development of the tower wake. It is therefore not surprising that the occurrences of these thumping noises have not been fully understood, since they seem to be strongly dependent on a large number of parameters, such as the exact flow conditions, turbine geometry and operational state. A better understanding of the aerodynamics of wind turbine rotor-tower interaction is therefore an important prerequisite for understanding and possibly proposing remedies for the acoustic problems associated with downwind turbines.

Since it is well-known that both vortex shedding from bluff bodies and rotor aerodynamics in their own right are dominated by three-dimensional flow features, it is safe to say that the interaction between the tower wake and the rotor on a downwind turbine can be expected also to be a highly three-dimensional phenomenon. The most comprehensive means for investigating the unsteady flow over a wind turbine by computational means is therefore

through modelling of the flow with a numerical solution of the Navier-Stokes equations.

The aerodynamics of isolated wind turbine rotors has been investigated extensively by Duque et al. [14, 15], Xu and Sankar [76, 77], Sørensen et al. [61], Johansen et al. [29], Madsen et al. [36], Le Pape and Lecanu [33] among others. These works showed that CFD tools are capable of accurately capturing the complex flow features on wind turbine blades, in particular the complex three-dimensional features which dominate the flow at high loadings. Additionally, in [15] Duque et al. modelled the flow around a complete wind turbine in a downwind configuration with rotor, tower and nacelle. However, due to computational issues, the unsteady interaction of the tower wake and the rotor was not captured fully. Xu [75] also shows results of rotor-tower interaction on the NREL Phase VI using an overset grid method. However, due to a very coarse grid resolution around the tower, the unsteady nature of the tower wake was not captured. Therefore, to the best of the author's knowledge, no computations on a downwind configuration of a wind turbine have succeeded in fully capturing the complex interaction between the unsteady wake of the tower and the passing rotor blades.

A complete simulation of rotor-tower interaction requires that component grids are allowed to move relative to each other. The overset grid method presented in Chapter 2 was found to be the most appropriate method for handling this problem. It allows for a decomposition of the domain into a number of simpler grids that can overlap each other arbitrarily enabling relative movement of bodies without the need for re-meshing. The validation of the method was carried out on a number of reasonably simple problems and found to be fully capable of solving unsteady flow problems involving high flow gradients and massive separation.

Field measurements on wind turbines are generally quite difficult to use for validation of CFD codes because of the strongly dynamic loading caused by the influence of the atmospheric conditions such as turbulent inflow including large scale veering of the wind direction and the shear of the atmospheric boundary layer that cause a large scatter in the measured data. Additionally, only few full scale experiments include surface pressure measurements on the blades, which makes it difficult to study local flow features on a wind turbine blade. For a summary of available field measurements see Schepers et al. [54]. Wind tunnel experiments have for the most part been restricted to low Reynolds number flows, where the characteristics of the flow are quite different from full

scale turbines. For a thorough review of available wind tunnel experiments see Vermeer et al. [69]. The NREL Unsteady Aerodynamics Experiment (UAE) [18, 55] is to date the most extensive experiment on a wind turbine representative in size and flow regime of full-size wind turbines conducted under controlled inflow conditions. The purpose of the experiment was primarily to study the aerodynamics of wind turbines, in particular the three-dimensional features of the flow, and secondly to provide the wind energy research community with aerodynamic and structural data on a wind turbine operating in a controlled environment, which could be used for validation of aeroelasticity and CFD codes. The experiment was carried out in the 80ft × 120ft (24.4m × 36.6m) NASA Ames wind tunnel on a 10 m diameter stall regulated wind turbine where, among other quantities, blade surface pressures and local flow angles were measured at five spanwise locations. The turbine was run both in an upwind and downwind configuration and under yawed flow conditions at a wide range of wind speeds. The UAE experiment thus provides an excellent possibility to validate the present numerical code and it will be used as a test case in the following analysis of wind turbine rotor-tower interaction.

In this chapter results are presented of unsteady computations of the flow over the NREL turbine with focus on the downwind configuration. Firstly, the flow over an isolated rotor is presented to validate the overset grid method against the original solver and experimental results. Secondly, the flow will be computed on an isolated tower with dimensions equal to that of the UAE turbine to show that the solver successfully captures the unsteady nature of the tower wake. Lastly, the flow over the downwind configuration of the turbine is presented. Through an analysis of the extracted surface pressure distributions, local flow velocities and visualisation of the flow field, a detailed discussion of the interaction between the rotor and the tower will be made, aiming to identify the key features of the interaction. It will thus be demonstrated that the newly developed overset grid version of EllipSys3D successfully captures the unsteady interaction between the rotor blade and the tower, and it will be shown that the computations are in excellent agreement with the experimental data available from the NASA Ames experiment.

4.2 Test Configuration

The NREL Unsteady Aerodynamics Experiment was carried out in the 80ft × 120ft NASA Ames wind tunnel (see Figure 4.1) on a two-bladed 10 m diameter

stall regulated wind turbine operating at 72 rpm. The rotor was mounted at a height of 12.2 m on a tower with a diameter of 0.4 m on the top section with a wider base of 0.6 m. The turbine could be operated in both an upwind and downwind configuration. In both configurations the rotor was mounted with a 1.4 m clearance to the tower, corresponding to 3.5 tower diameters. The rotor could be operated at various teeter and cone angles. The blades were twisted and tapered and fitted with the S809 profile, as is shown in Figure 4.2.

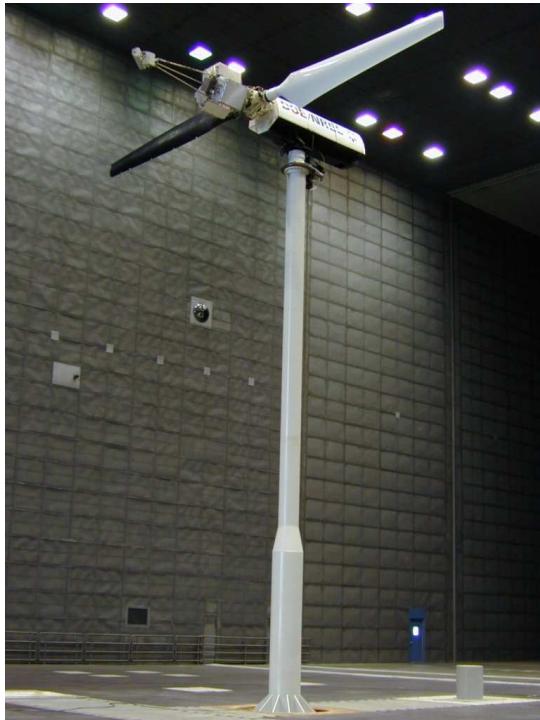


Figure 4.1: The NREL UAE Phase VI wind turbine mounted in the NASA Ames wind tunnel.

One blade was instrumented with pressure probes and pressure taps as shown in Figure 4.2 and 4.3. The pressure taps were placed along the suction and pressure sides of the blade at five spanwise locations on the blade at $r/R=0.3, 0.47, 0.63, 0.80$ and 0.95 . Additionally, five hole pressure probes were placed 80% chord upstream of the blade at $r/R=0.34, 0.51, 0.67, 0.84$ and 0.91 to measure the local flow angles.

The walls of the wind tunnel were smooth to minimise the thickness of the boundary layer, resulting in no shear of the inflow on the turbine and a very small turbulence intensity of less than 1%.

The test was carried out over a wide range of wind speeds ranging from 5 m/s to 25 m/s. With a tip speed of 37.7 m/s, the tip Reynolds number was

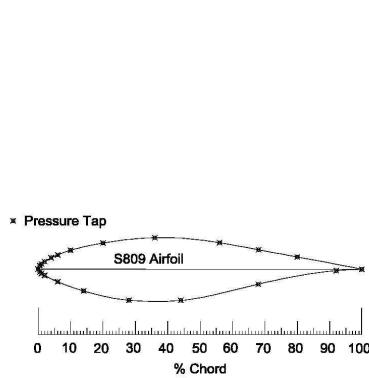


Figure 4.2: Pressure tap locations on the S809 profile [20].

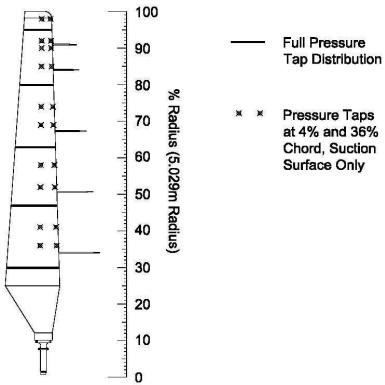


Figure 4.3: Planform of the NREL UAE Phase VI blade showing location of pressure taps and pressure probes [20].

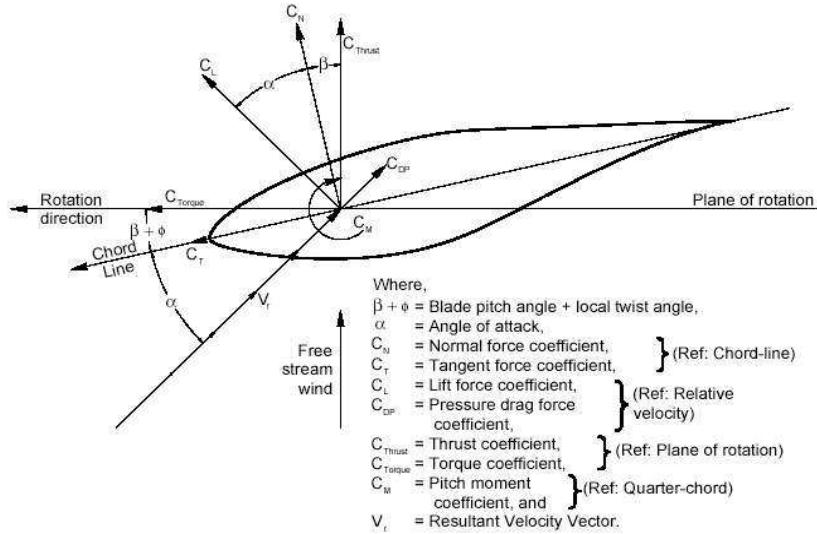


Figure 4.4: Definitions of the aerodynamic force coefficients [21].

approximately $R_e = 1 \times 10^6$ based on a reference chord of 0.4 m. The tower Reynolds number ranged from $R_e = 1.84 \times 10^5$ to $R_e = 6.9 \times 10^5$ based on the tower reference diameter of 0.4 m. For a detailed description of the entire test setup and all measurement campaigns see Hand et al. [21].

The general frame of reference for the turbine is shown in Figure 4.5. For all computations in this chapter, the inflow is prescribed in the z -direction. The azimuth angle is zero when the blade is aligned vertically with the tower and in its top position. The origin of the axis is placed at the tower top, placing the rotor at $z=1.4$ m.

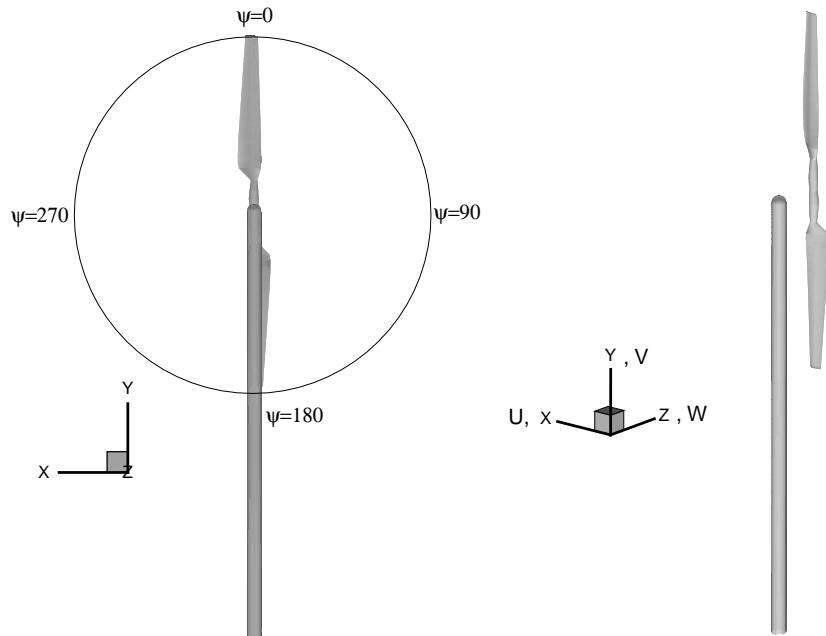


Figure 4.5: Definitions of the frame of reference for the turbine. Inflow direction is in the z -direction.

The blade forces are defined according to Figure 4.4. The integral forces were obtained by integration of the pressure distributions on the surface of the blades at the positions shown in Figure 4.3. The local blade velocities are defined in Figure 4.6.

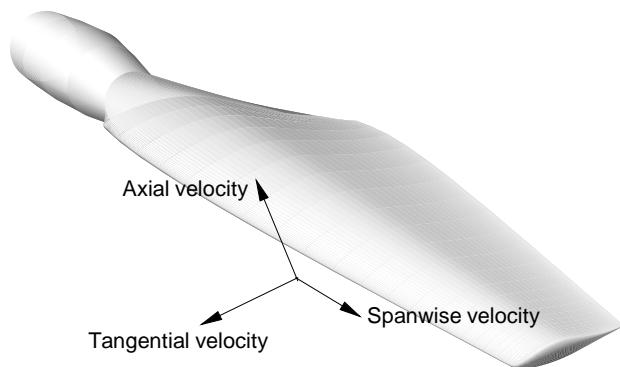


Figure 4.6: Definition of the local blade velocities.

4.3 Computational Meshes

The mesh generation around wind turbine blades can be done in several different ways. One example can be found in Sørensen et al. [61] who used an O-O grid to cover the blades which was grown out as an O-grid to the farfield. The advantage of such an approach is that only one blade needs to be included if the rotor is not yawed since periodic boundaries can be used in place of the second blade.

To model an entire wind turbine including rotor, tower and hub a different approach is needed, since the above approach does not allow for the inclusion of a stationary tower. For such a configuration it is necessary to allow relative movement between grids, by either using sliding or overset grids. The overset grid approach is generally more flexible since there is no requirement of grids to match along lines or planes in the grid. With the possibility of letting grids overlap each other arbitrarily, completely different layouts can therefore be made. Duque et al. [14, 15] used grids based on a combination of curvilinear and Cartesian grids, where the curvilinear grids covered only a small part of the domain around the blades and tower, with the rest of the domain covered by Cartesian background grids. In [15] computations were carried out including both rotor, tower and nacelle, with symmetry conditions along the bottom and side boundaries of the domain.

The same grid generation approach as above will be used in the present study although the turbine geometry has been simplified slightly as will be described subsequently. Curvilinear grids are used around the two-bladed rotor and the tower, whilst Cartesian background grids are used to cover the rest of the domain. The advantage of this approach is that the grid can be more easily designed to resolve regions of high flow gradients in the vicinity of the rotor and tower, without carrying the clustering of cells all the way to the farfield. This coarsening strategy reduces the waste of cells in the farfield typically seen in O-grid or C-grid configurations.

Two different grids were made for the present study. Firstly, a reference mesh, N1, was made where no tower was present, and where the the farfield side boundaries were 4.5 rotor diameters away from the centre of rotation. The inlet was placed 4 rotor diameters upstream of the rotor, and the outlet was 8 diameters downstream. The rotor mesh, which was generated using HypGrid [60], consisted of 40 blocks of 32×32 cells to cover the surface of the two blades with 256 cells in the chord-wise direction, 64 cells in the spanwise

direction, and four blocks of 32×32 to cover each blade tip. The grid was grown out approximately 1 m from the surface with a total of 64 cells in the normal direction stretched using a hyperbolic tangent function. The height of the first cell in the boundary layer was 1×10^{-5} m corresponding to y^+ values of less than 2. This grid was embedded in three Cartesian meshes, successively coarsened towards the farfield consisting of 36, 27, and 12 blocks of 33^3 cells, respectively. The total number of cells in the grid was 5.57M cells. The boundary conditions consisted of an inlet with a prescribed velocity, an outlet with a zero gradient condition, and cyclic boundary conditions on the other four boundaries. On the surface of the blades a no-slip condition was prescribed. The mesh is shown in Figure 4.7.

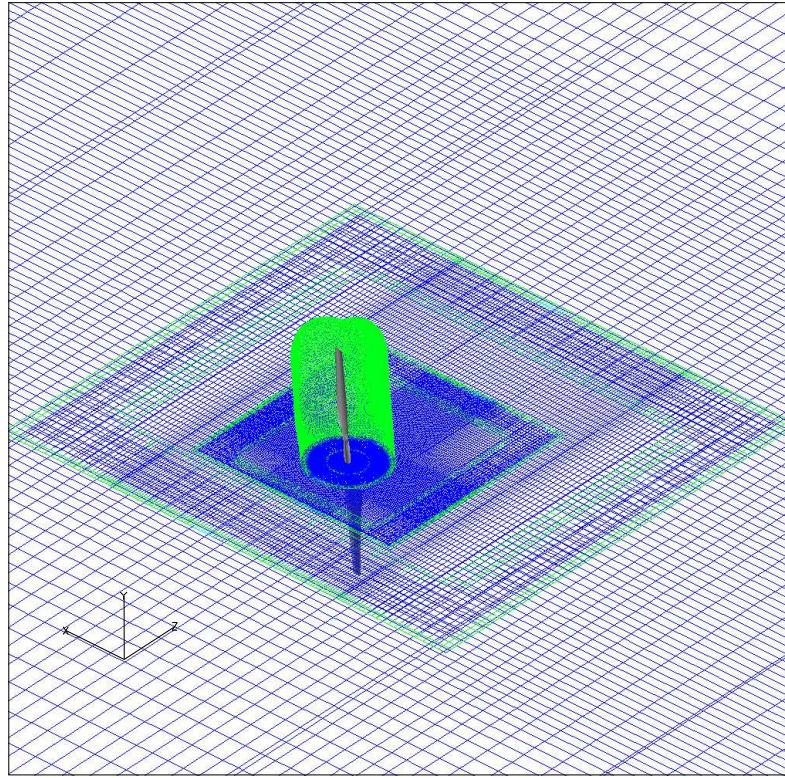


Figure 4.7: Mesh for the rotor-only configuration.

In the present study the geometry of the Phase VI turbine has been simplified somewhat. As can be seen in Figure 4.1 the tower is wider at the bottom with a diameter of 0.6 m, up to a height of 4 m where the diameter is reduced to 0.4 m. In the computational model this feature has been omitted such that the tower has a constant diameter of 0.4 m along the entire height of the tower. Additionally, the nacelle was omitted since it would require a consid-

erable amount of cells to properly resolve its geometry. Although it is likely to have a strong influence on the flow-field in the root section of the blade, its influence was thought to be less pronounced on the outboard sections of the blade. The rotor was modelled without coning and teeter, whereas in fact the Phase VI turbine in its downwind configuration was run with a cone angle of 3.4° and a teeter angle which on average was 3° . The combination of teeter and coning, however, results in the bottom blade being approximately parallel to the tower axis, which justifies this simplification.

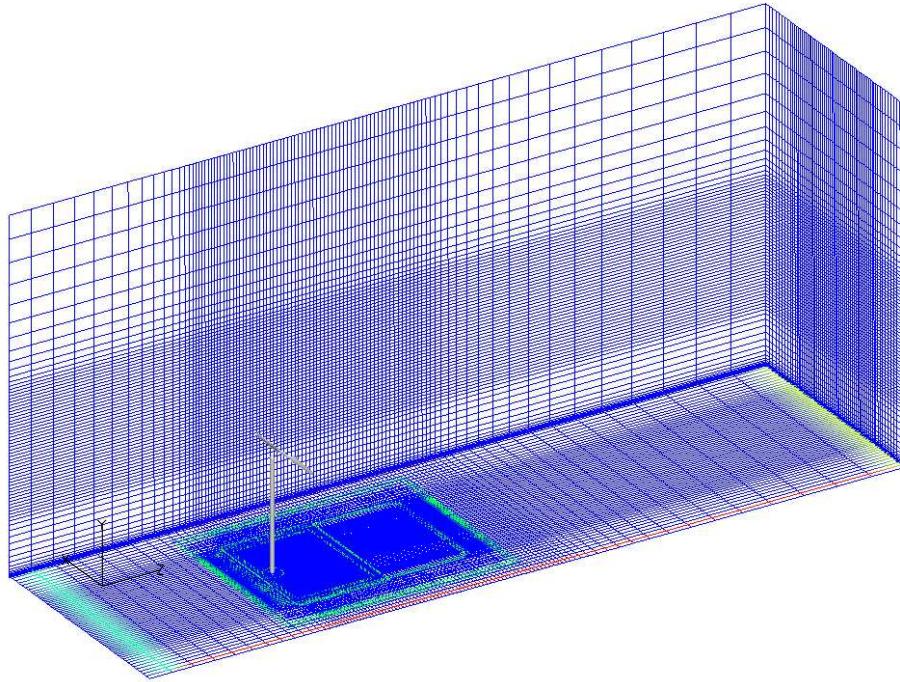


Figure 4.8: Mesh for the rotor-tower configuration.

The computational mesh for the rotor-tower configuration consisted of five individual block groups: two curvilinear meshes for the rotor and tower and three Cartesian meshes which were successively coarsened towards the farfield (See Figure 4.8). The rotor mesh was identical to the one used in grid N1. The tower mesh consisted of 128 cells around the cylinder and 384 cells in the vertical direction and one grid of 32×32 to cover the rounded free end. The mesh was grown out 1 m in the normal direction using 32 cells totalling 13 blocks of 32^3 cells. The height of the first cell on the surface was 1×10^{-4} m.

The cross-sectional dimesions of the domain were equal to those of the NASA Ames wind tunnel, $24.4\text{m} \times 36.6\text{m}$. The domain was 80 m long with the turbine placed 20 m downstream from the inlet. The bottom boundary

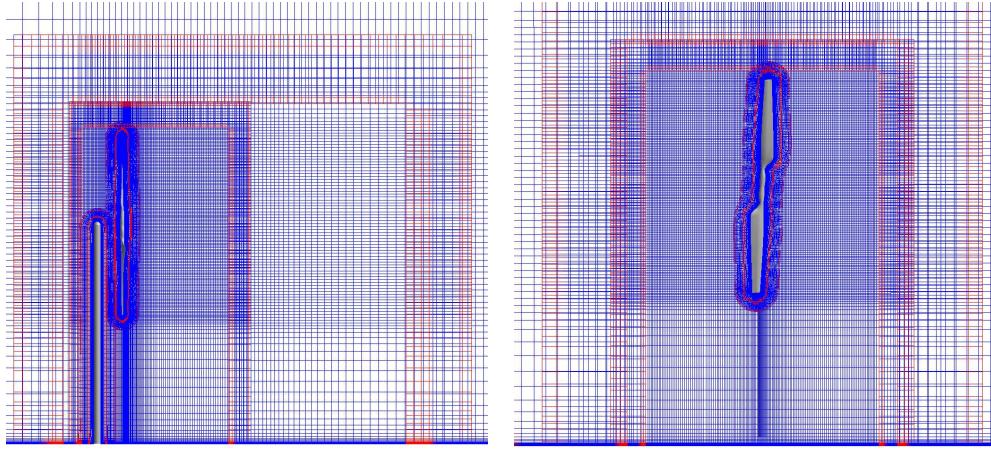


Figure 4.9: Side and back view of the mesh system for the rotor-tower configuration.

was prescribed as a no-slip wall, whereas the three other side walls of the tunnel were modelled using symmetry boundary conditions in order to avoid having to resolve the wall shear layers. A standard inlet condition with a given velocity was prescribed at the inlet, and a zero gradient condition was specified on the outlet. To capture the ground shear layer reasonably well, the three Cartesian groups and the tower grid had a non-dimensional first cell height at the wall of 1×10^{-4} m. The inner Cartesian block group had physical dimensions of $14\text{ m} \times 19\text{ m} \times 14\text{ m}$ and consisted of $160 \times 192 \times 128$ cells in the x , y and z directions, respectively. The mesh was made finer in the rotor plane to match the grid dimensions of the rotor grid to properly resolve the tip vortex from the rotor. The middle Cartesian block group had physical dimensions of $20\text{ m} \times 22.5\text{ m} \times 25\text{ m}$ and contained $96 \times 128 \times 96$ cells. The outer block group had the dimensions of the tunnel, $24.4\text{ m} \times 36.6\text{ m}$, and consisted of $64 \times 97 \times 97$ cells. The total number of cells in the grid was 9.6M cells. This grid will be referred to as grid N2. The grid assembly is shown in Figures 4.8, 4.9 and 4.10.

4.4 Computational Framework

The computations presented in this chapter were all carried out using the SUDS scheme to discretise the convective terms. The PISO algorithm was used to solve the coupled velocity/pressure equations. Due to the uncertainties associated with the limited implementation of the $k-\omega$ SST model as described in Section 2.3.4 the computations presented in this chapter were all computed using the original $k - \omega$ model by Wilcox [71].

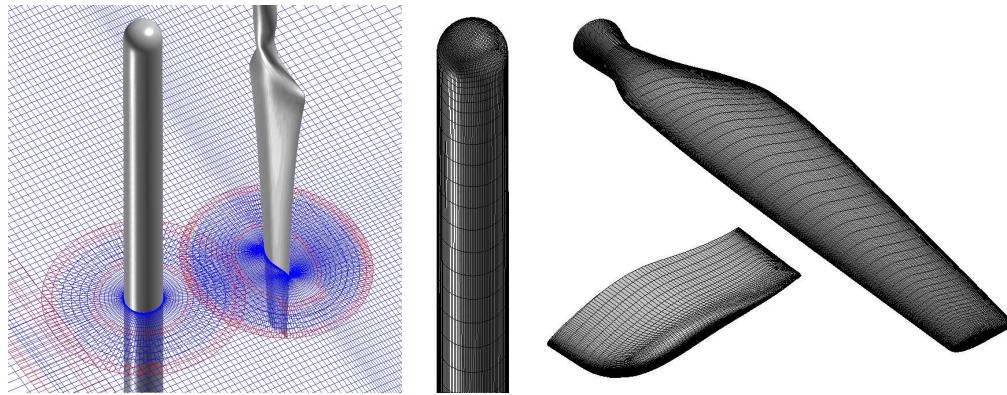


Figure 4.10: Detail of the rotor and tower grids.

From previous experience with the non-overset version of the code it was found that approximately 800 time steps per revolution were needed to obtain an accurate solution of the integral forces on the NREL blade. To speed up the convergence of the solution, the computations were started up at a coarse time step of $Deltat = 0.002$ and run until the flow was sufficiently developed. This was necessary particularly in the rotor-tower computations because of the long time required for the unsteady tower wake to develop fully. Once the wake had developed, the computations were restarted at a finer time step of $\Delta t = 0.001$. With a rotational frequency of 72 rpm, this corresponded to a time resolution of 0.43° per time step.

4.5 Isolated Rotor Computation

Computations of flow over the isolated NREL Phase VI rotor, as discussed in section 4.1, have previously been carried out using EllipSys3D by Sørensen et al. [61] using RANS. To validate the overset grid method, equivalent overset computations were carried out and compared to these results as well as the experiment. The upwind 7 m/s case with no teeter (Sequence H, see [21]) was chosen for the comparison.

The computations were carried out using grid N1 which was presented in detail in Section 4.3. The simulation was carried out as an unsteady moving grid simulation without any relative movement between the rotor grid and the background Cartesian grids, which is equivalent to what was done in [61]. This eliminated the need to update connectivity information between individual component grids, thus reducing the computational cost. The computations were otherwise carried out using the solver options described in Section 4.4.

Figure 4.11 shows the normal and tangential force coefficients along the blade computed at five spanwise locations corresponding to the pressure tap locations in the experiment. Although the results differ slightly from the experimental results the overall agreement is very good. There is also some discrepancy between the overset and non-overset computations. This, however, can most likely be explained by slight differences in the cell distribution in the normal direction of the rotor grid, as well as the fundamentally different mesh topology in the off-body grids, which will affect the development of the wake.

Figure 4.12 shows the pressure coefficient distributions at five spanwise locations on the blade for the overset computations compared to the non-overset computations and experimental results. As above, the agreement is very good between the two computations and the experiment, with only slight differences at the outermost spanwise location.

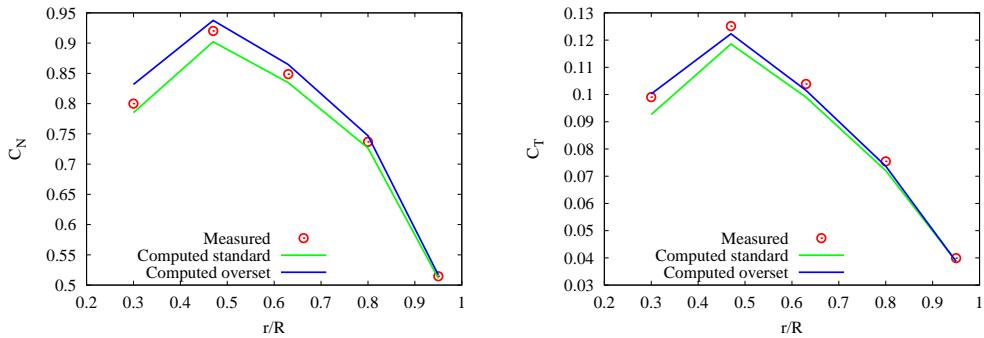


Figure 4.11: Normal and tangential force coefficients for a wind speed of 7 m/s for the isolated rotor.

The use of uniform Cartesian meshes makes it possible to preserve flow features such as the root and tip vortices downstream of the rotor. Although the grid used for this computation is coarsened downstream of the rotor it is evident in Figure 4.13 where the iso-vorticity and u -velocity contours are plotted that the tip vortex is preserved quite well up to approximately two rotor revolutions downstream.

The fact that this very complex flow field can be computed accurately using the overset grid method gives reason to believe that the solver is capable of computing flow over wind turbine blades with the same degree of accuracy as the original solver.

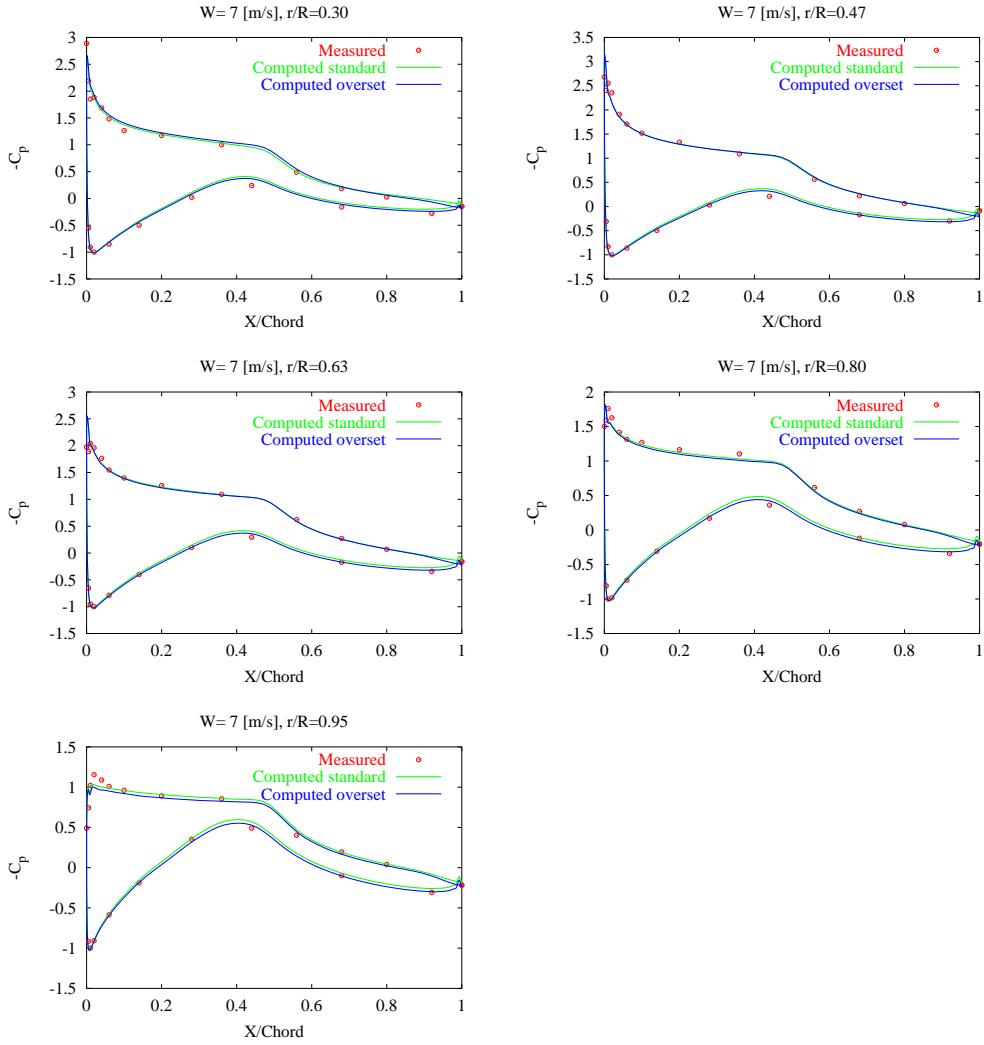


Figure 4.12: Pressure distribution coefficients for a wind speed of 7 m/s for the isolated rotor.

4.6 Isolated Tower Computation

The flow over a circular cylinder of finite height mounted on a ground plane forms a complex wake due to the combination of many effects. For high aspect ratio cylinders (typically above Height/Diameter=6) the flow will consist of three main flow regions: Along the ground plane, a horseshoe vortex will form as the ground boundary layer wraps around the cylinder base. At the free end, two vortices will be formed of opposite vorticity which are shed unsteadily from the top. A strong downward velocity at the tip causes these vortices to interact with the wake on the middle part of the cylinder. For sufficiently large aspect ratios, the flow away from the two ends will consist of alternately shed vortices forming the well-known von Kármán street. Due to the end

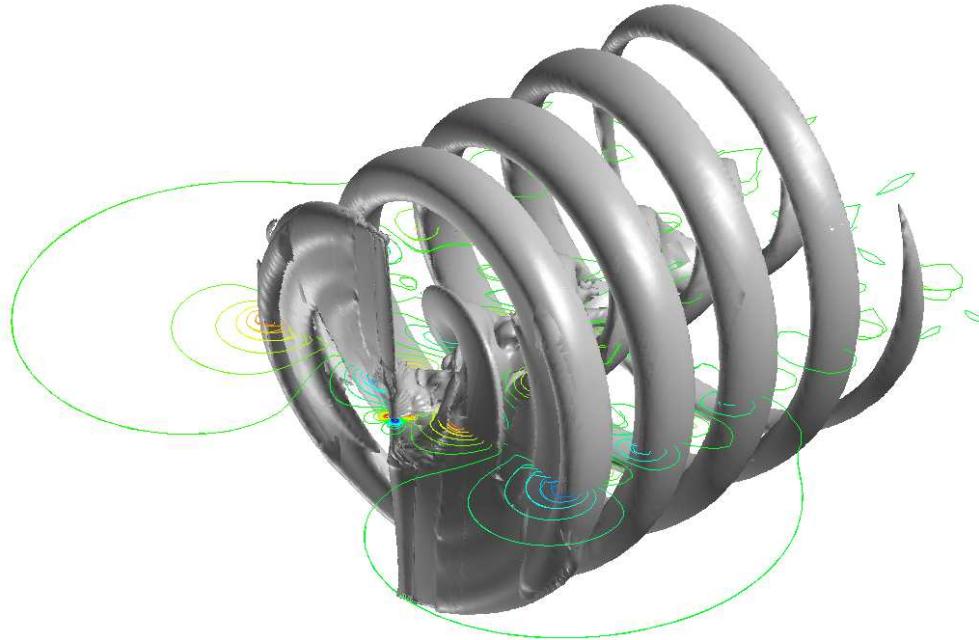


Figure 4.13: Iso-surface plot of vorticity magnitude and u -velocity contours of the flow over the isolated rotor for a wind speed of 7 m/s.

boundary conditions the vortex street can become stretched in the streamwise direction causing what is known as oblique vortex shedding, as well as causing dislocations of the vortex street splitting the wake into distinct shedding cells. The length of these shedding cells depends on a number of factors such as Reynolds number, surface roughness and end conditions. Typically for high Reynolds number flows the length of these cells will be in the order of three to four cylinder diameters. A thorough review of vortex shedding from circular cylinders can be found in the review article by Williamson [72].

As part of the preliminary studies of the wind turbine rotor-tower configuration, a computation of flow over an isolated tower with a ground boundary layer was performed to verify that the solver produced a realistic wake consisting of alternately shed vortices. The grid was identical to the one used for the rotor-tower computations (with the rotor grid omitted), eliminating any grid dependent differences in the solutions. The basic solver parameters were otherwise identical to the ones described in Section 4.4. The computation was run for a total time of 15 seconds started at a time step of $\Delta t = 0.005$ seconds to obtain a developed flow and subsequently at $\Delta t = 0.002$. The freestream

velocity was set to 6.7 m/s corresponding to the downstream Sequence B in the UAE experiment. The Reynolds number based on the cylinder diameter was $R_e = 1.84 \times 10^5$, which is in the sub-critical flow regime. However, since the computations are carried out assuming fully turbulent flow, the highly complex nature of the flow in this regime cannot be captured.

The pressure distribution and forces were extracted at eight different heights along the tower, five of which corresponded to the locations of the pressure taps on the rotor. Figure 4.14 shows the time history of the side force on the tower at five different heights. There is some variation in the shedding frequency along the span of the tower with a lower shedding frequency on the upper part where the flow is strongly influenced by the tip. The shedding frequency was approximately 4.1Hz on the middle part of the tower, giving a Strouhal number of 0.24 based on the tower diameter. Additionally, it was found that the amplitudes of the responses at each section along the height of the tower varied with a low frequency change.

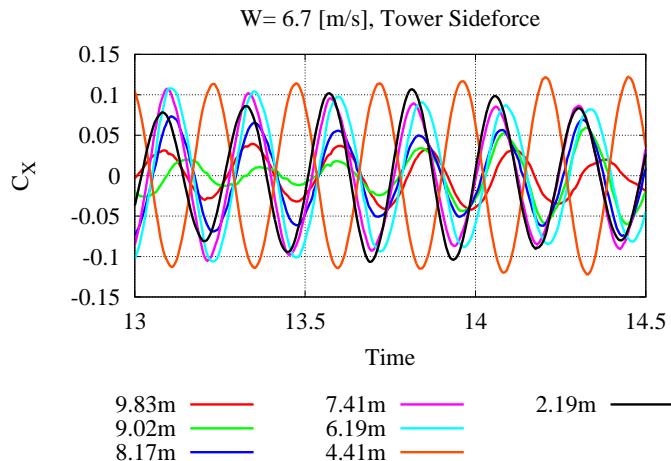


Figure 4.14: Time history of the side force coefficient at various heights on the tower for a wind speed of 6.7 m/s.

The mean pressure and skin friction coefficients on the tower at a height of 8.17 m are shown in Figure 4.15. As it can be seen, separation takes place at an angle of approximately $\theta = 100^\circ$ measured from the stagnation point.

Figure 4.16 shows the axial velocity deficit extracted on a horizontal line 1.37 m downstream of the tower and at a height of 7.97 m above the ground over one vortex shedding period. This horizontal line corresponds approximately to the line swept by the pressure probes mounted on the blade in the UAE experiment. The lack of smoothness of the data is due to the fairly coarse mesh in the tower wake and that the velocities were extracted directly

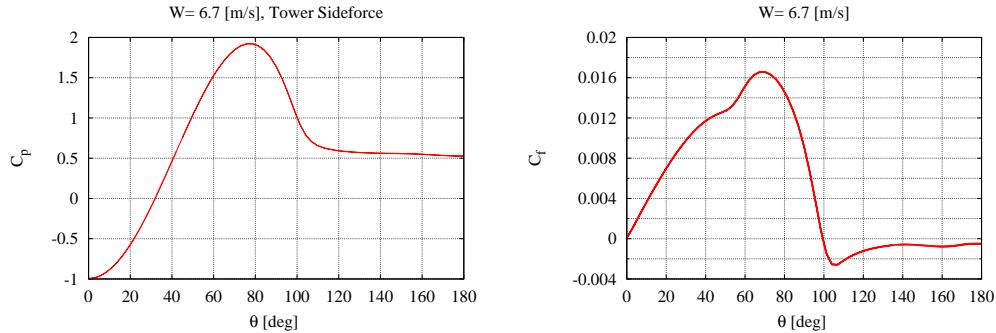


Figure 4.15: Mean pressure and skin friction coefficient distributions on the tower at a height of 8.17 m for a wind speed of 6.7 m/s.

from the cells in the grid without interpolation. The axial velocity was found to be reduced by 28% (relative to the freestream) on average with a deficit width of two cylinder diameters. The instantaneous axial velocity was found to vary within a 27% increase and 49% decrease (relative to the freestream) as a result of the vortex shedding. The tangential velocity, shown in Figure 4.17, was found to be highest along the centreline of the tower with a maximum velocity change of ± 3.5 m/s.

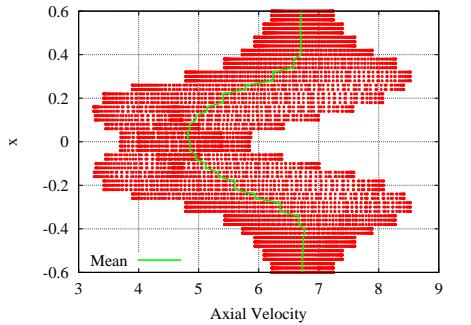


Figure 4.16: Axial velocity profile 1.37 m downstream of the tower at a height of 7.97 m for a wind speed of 6.7 m/s.

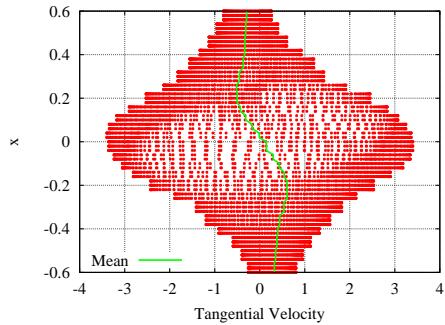


Figure 4.17: Tangential velocity profile 1.37 m downstream of the tower at a height of 7.97 m for a wind speed of 6.7 m/s.

Figures 4.18 and 4.19 show an iso-surface plot of the vorticity magnitude of the flow over the isolated tower and vorticity magnitude contours at various distances downstream of the tower. The three-dimensionality of the wake is clearly visible. The tip effect and at the wall boundary causes the flow to be highly unsteady and the vortex tubes to become stretched in the streamwise direction causing oblique vortex shedding. Additionally, the finite tip causes a number of dislocations of the vortex street. It was found that the position of the dislocations shifted along the span of the tower at a very low rate, possibly

due to the changes in amplitude and phasing of the side force on the tower as seen in Figure 4.14.

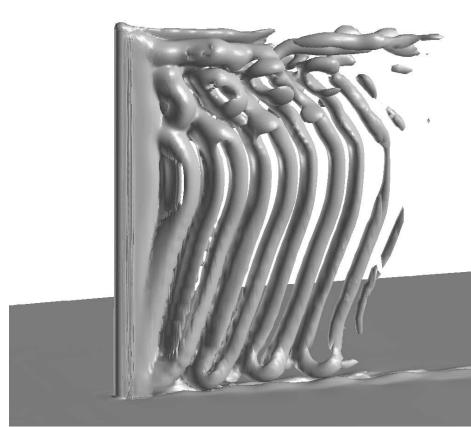


Figure 4.18: Iso-vorticity plot ($|\omega| = 2\text{s}^{-1}$) of the flow over the isolated tower for a wind speed of 6.7 m/s.

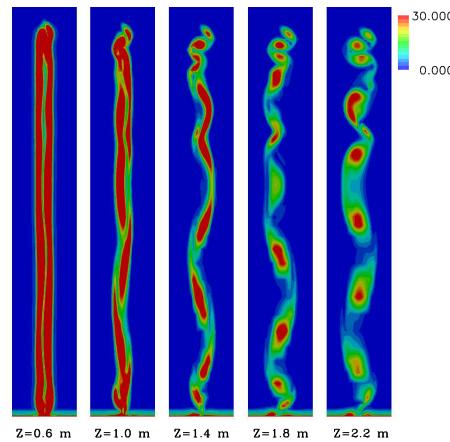


Figure 4.19: Vorticity contour plot of the flow at various distances downstream of the isolated tower for a wind speed of 6.7 m/s.

The relatively large degree of coherence of the wake and distinct lack of small scale structures is likely to be an artifact of the Reynolds averaging and turbulence modelling rather than being a physically correct representation. Additionally, the grid used to resolve the wake was reasonably coarse, and it is therefore possible that some improvements could be achieved with a finer grid. Several studies of vortex shedding on infinite circular cylinders have demonstrated that with the use of LES or hybrid LES, the unsteady wake characteristics can be captured more accurately with a substantial reduction in the coherence of the wake, see for example Travin et al. [67]. An obvious extension of the present work would therefore be to investigate the wake characteristics of a finite cylinder using this type of modelling of the flow.

As seen in Figure 4.19, at $z=0.6$ m the free shear layer shed from the tower is still largely two-dimensional. Further downstream the stretching of the vortex street results in a larger variation in the vertical direction. The alternately shed tip vortices are clearly visible, and the root vortex which wraps around the base of the tower along the ground plane can also be identified. Experimental studies have shown that the spanwise correlation length is considerably lower in this Reynolds number range. Humphreys [26] found that the correlation lengths at a Reynolds number in the transitional range was approximately three to four diameters, again suggesting that the RANS computations have a tendency to represent the wake as a more two-dimensional structure.

4.7 Rotor-Tower Computation

In this section the computations carried out on the complete rotor-tower configuration will be presented. One case from the Unsteady Aerodynamics Experiment was chosen where the nominal inflow velocity was equal to 7.0 m/s. However, due to wind tunnel anomalies the actual inflow velocity was 6.7 m/s, which will therefore be used in this computation. At this wind speed the flow over the blades will be attached.

The computational results will be analysed by firstly presenting the extracted flow field data, where the key features of the flow will be identified; secondly, the response of the blades to the wake interaction will be presented; thirdly, the computational results are compared to the experiment. Finally, the response of the tower to the interaction will be analysed.

Surface pressures and skin friction were extracted on both blades in the same radial positions as in the experiment (see Figure 4.3). Additionally, surface pressures were extracted on the tower at eight heights, five within the rotor radius and five below. Force coefficients were computed from the integrated pressure and skin friction coefficients and projected according to the conventions shown in Figure 4.4. Local flow velocities were extracted at the probe positions shown in Figure 4.3 which were used to compute the local flow angles. The entire flow field was exported every 30° azimuth for visualisation purposes.

4.7.1 Computational Framework

The mesh used was described in Section 4.3 and it contained five block groups with a total of 9.6 million grid cells. The computational parameters used in the simulations were described in Section 4.4. The computation was run for a total of 12 revolutions or approximately 10 seconds. Six revolutions were run at a coarse time step of $\Delta t = 2 \times 10^{-3}$ and the last six were run at the finer time step of $\Delta t = 1 \times 10^{-3}$. The coarse time step was needed since the tower wake took several revolutions to develop fully, particularly below the rotor disc.

4.7.2 Flow Field

Figure 4.20 shows the flow field surrounding the turbine using an iso-surface plot vorticity magnitude. The mesh is also shown in the plot with grid interfaces highlighted green. The unsteady vortex shedding from the tower is clearly

visible, and it can be seen that the vortex street is quite coherent and well-preserved approximately 10 m downstream. The root and tip vortices from the rotor are clearly visible and likewise preserved well for approximately 10 m downstream. The dissipation of the vortices does not appear to be linked to the presence of overset interfaces, but rather to the stretching of the grid. The rotor has a visibly strong influence on the tower shedding, causing a difference in shedding frequency on the upper and lower parts of the tower. This results in a dislocation with two distinct shedding cells separated by a disorganised region approximately 1 m below the rotor disc.

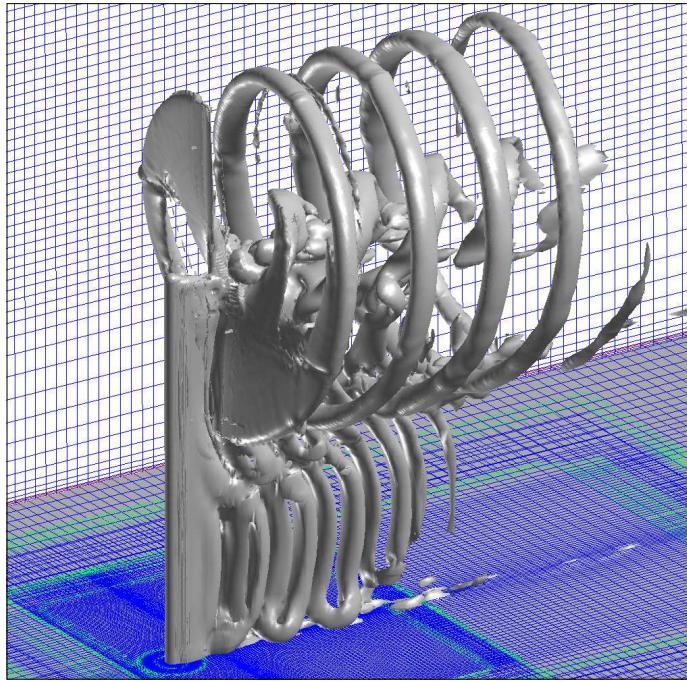


Figure 4.20: Iso-vorticity plot of the flow over the turbine for a wind speed of 6.7 m/s.

The flow is further illustrated in Figure 4.21 which shows the flow field at a vertical cut plane of $x=0$ bisecting the turbine, at an instant where the blades are aligned vertically with the tower. It is again evident from this plot that the vortex street is convected downstream at a different velocity on the upper and lower parts of the tower. It also appears as if the blade tip vortices cause the tower vortices to wrap up around the blade vortices enhancing the stretching of the tower vortices. Figure 4.22 shows a contour plot of the axial velocity (w -velocity) in the same plane ($x=0$) which highlights the velocity deficit in the wake of the tower and rotor. There are considerable 3D effects at the top of the tower, which result in a downward directed flow interfering

with the inner parts of the rotor. Again it is evident that the flow features are transferred smoothly across overset interfaces with no visible discontinuities in flow gradients.

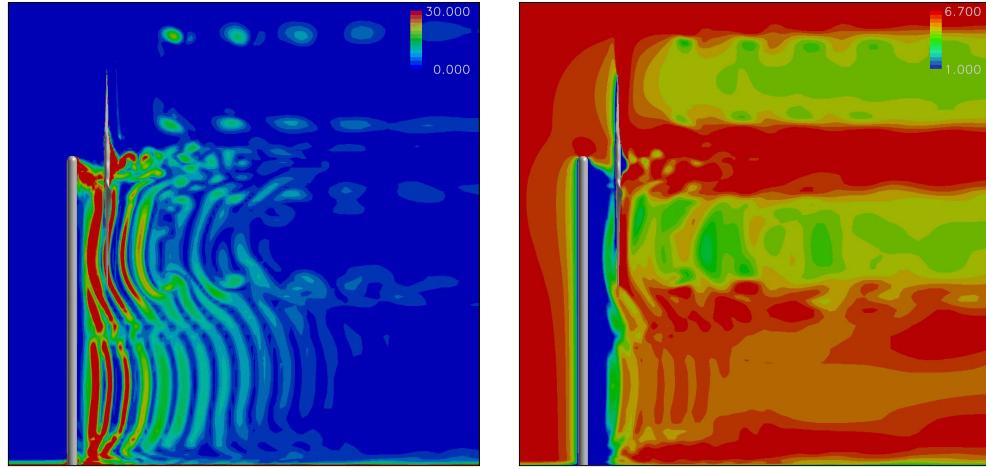


Figure 4.21: Vorticity magnitude contour plot of the flow over the turbine for a wind speed of 6.7 m/s.

Figure 4.22: Axial velocity contour plot of the flow over the turbine for a wind speed of 6.7 m/s.

Figures 4.23 and 4.24 show a sequence of snapshots of the flow field for two blade vortex encounters. In Figure 4.23 the wake consists of a vortex of negative vorticity that the blade encounters virtually head-on. The direct encounter between the blade and the vortex causes the vortex to split into two fragments, one small fragment upstream of the blade wake of and another one downstream which has been stretched considerably by the passing blade. This BVI event will in the remainder of this chapter be referred to as a *type 1* passage.

In Figure 4.24 the blade encounters a vortex of positive vorticity which is located slightly more upstream than in the previous event. Again the vortex is divided in two fragments, one small fragment downstream of the blade wake and a more stretched fragment upstream. This situation will be referred to as a *type 2* passage. These plots also highlight the large differences between the blade velocity and the vortex convection velocity which characterises the interaction taking place on the outer parts of the blade. On the inner parts of the blade, however, the blade velocity is much slower, therefore decreasing the relative velocity difference between the blade and vortex.

A constant y surface of the mesh is also shown on the plots to highlight the location of the overset interfaces. Again it is evident that the flow gradients are continuous across the overset interfaces despite the relatively coarse mesh

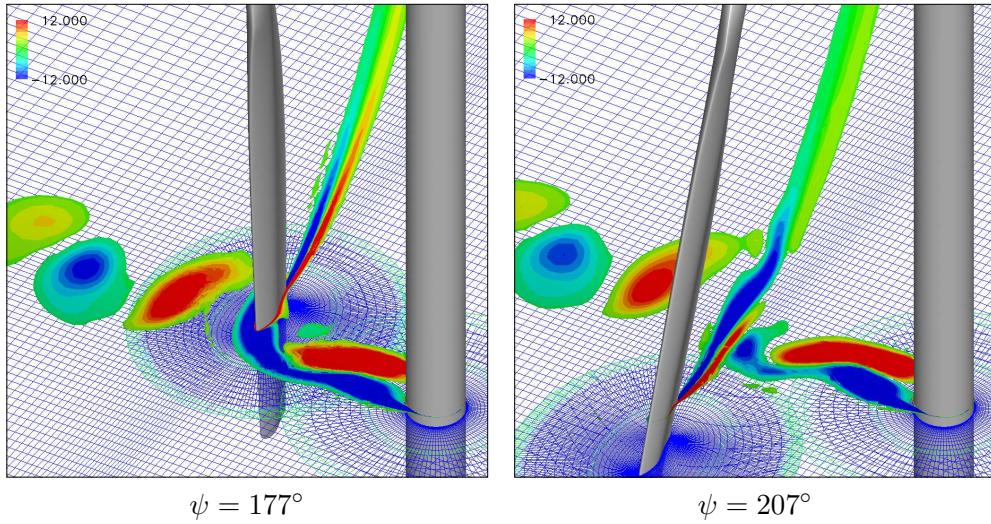


Figure 4.23: Vorticity contour plot of a *type 1* passage showing an encounter with a vortex of negative vorticity (clockwise rotation). The vorticity contours are plotted on a plane of constant radius of 4.02 m corresponding to 80 % span.

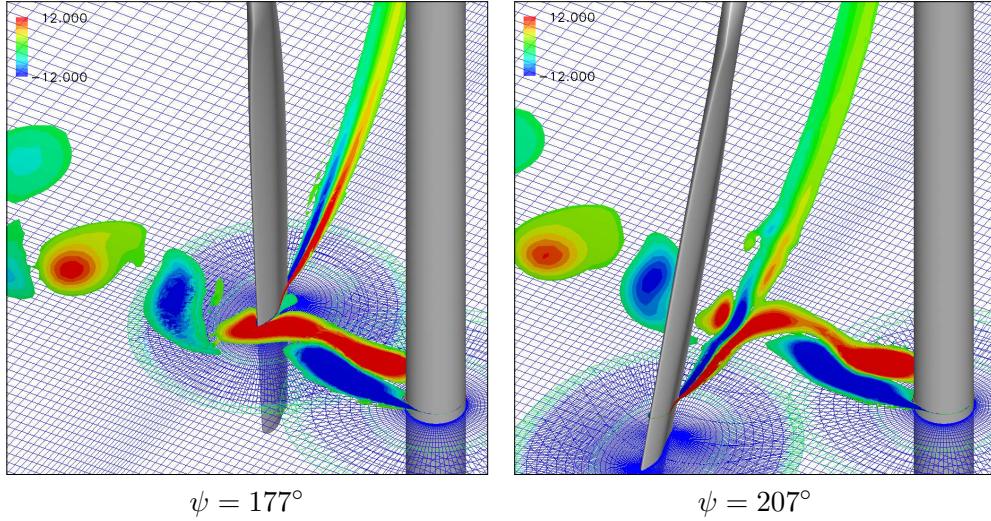


Figure 4.24: Vorticity contour plot of a *type 2* passage - showing an encounter with a vortex of positive vorticity (counter-clockwise rotation). The vorticity contours are plotted on a plane of constant radius of 4.02 m corresponding to 80 % span.

used to resolve the tower wake.

The aim of the present study has not been to capture the flow in the far wake of the turbine, and as such the grid was coarsened considerable about one rotor diameter downstream of the turbine causing distinct flow structures to dissipate rapidly. However, some interesting features of the far wake were captured. Figure 4.25 shows a series of contour plots of the wake development viewed directly from above. First of all, it is evident that the distinct tip

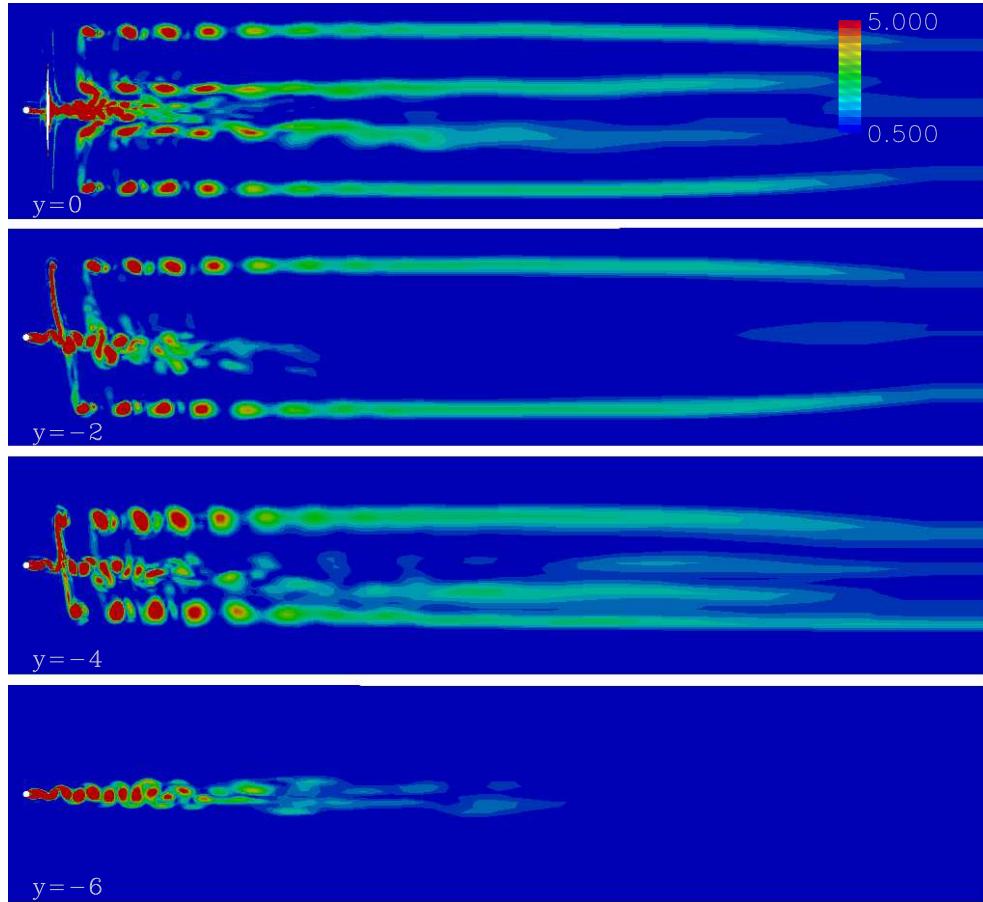


Figure 4.25: Vorticity magnitude contour plot of the flow over the turbine for a wind speed of 6.7 m/s, with the flow approaching the turbine from the left.

and root vortices become smeared into a continuous sheet approximately one and a half rotor diameters downstream. The rotor and tower wakes seem to interact in a way which causes a strong asymmetry in the far wake. The rotor root vortex is clearly not symmetric in that one side seems to have been destabilised. Likewise, the tower wake is due to the rotor wake rotation skewing to one side. It could thus appear as if the skewing of the tower wake causes the asymmetry of the root vortex. The rotation is strongest in the proximity of the tip vortex which is evident at $y=-4$ where the tower wake at two rotor diameters downstream has shifted approximate 1.5 meters to one side. This causes the tower wake to interact with the tip vortices in the far wake causing the rotor wake to shift to one side. Clearly, the mesh in the far wake is far too coarse to make any conclusive arguments with respect to the interaction between the rotor and tower wakes, but the above observations are interesting and should be investigated further.

4.7.3 Blade Response

In this section the effects of the rotor-tower interaction on the blade responses will be analysed. This is done by firstly presenting the integral normal and tangential force coefficients at the five spanwise sections on the two blades, followed by a detailed presentation of the responses at the 80% span section, where the local flow angles, moment coefficients and pressure distributions are presented. Finally, the computed results are compared with the experiment.

Normal and Tangential Force Coefficients

It was found in the computations that the responses of the two blades as they pass through the tower wake enter a periodic state where one blade experiences an interaction characterised by the flow field shown in Figure 4.23 - a *type 1* interaction - and the other blade experiences an interaction shown in Figure 4.23 - a *type 2* interaction. In the following analysis the blade responses to these two types of events will be analysed. The *type 1* response will be associated with blade 1, and the *type 2* response will be associated with blade 2. The somewhat surprising periodicity of the blade responses will be analysed further in a later discussion.

Figures 4.26, 4.27, 4.28 and 4.29 show the blade normal and tangential force coefficients as a function of blade azimuth for the five spanwise sections on the two blades. In each plot the forces for three consecutive passages are superimposed on each other. These figures clearly illustrate the periodicity of the blade responses, showing that there is only minor variations between consecutive passages.

Looking firstly at the *type 1* response shown in Figures 4.26 and 4.28, the normal force response of the blade (blade 1) as it enters the wake of the tower varies strongly along the span. The response at the two innermost stations differ considerably from the three outer stations in that the recovery from the wake covers an azimuthal interval of approximately 70° , whereas on the outer parts of the blade the recovery is much faster. The blade responses on the outer three stations are very similar and characterised by a sudden decrease in the normal force with a minimum reached before the blade reaches 180° azimuth. This is followed by an even faster increase in normal force with a maximum which is higher than the undisturbed level of the normal force. As the blade exits the wake, the normal force initially decreases slightly; this is followed by an increase restoring it to the undisturbed level. The tangential

forces shown in Figure 4.28 behave largely in the same manner as the normal forces.

Blade 2 encounters a different flow field in the wake of the tower shown in Figures 4.27 and 4.29, therefore responding quite differently from blade 1. Firstly, it can be seen that there is very little variation of the response along the span as was also seen for the three outermost stations on blade 1. As the blade approaches the tower, the normal force increases rapidly followed by a large decrease which reaches a minimum at 180° azimuth. The subsequent recovery is fairly quick, although the normal force continues to increase slightly until well past the tower wake. As for blade 1, the tangential forces behave similarly to the normal forces.

The two types of blade-vortex encounters thus produce very different normal and tangential force responses on the blades. Since the flow is attached on the blades, the velocity deficit causes a decrease in the loading due to the incidence reduction. A blade-vortex encounter causes different responses depending on the sense of rotation of the vortex. In the *type 1* encounter the blade encounters a vortex of negative vorticity as it exits the wake which appears not to generate as significant a fluctuation in the response as a *type 2* encounter where the blade encounters a vortex as it enters the wake.

Common to both blades is that the dynamic change in loading decreases with increasing radial position. On the inner sections the maximum fluctuation in the load is as much as 75%, whereas on the outer sections of the blades the loading fluctuates by approximately 40% during the tower passages. It is also evident that the wake interaction occupies an increasing proportion of the azimuthal cycle towards the root of the blades which is mainly due to the relative difference in azimuthal angle subtended by the tower wake between the inner and outer sections of the blade.

Local Flow Angles

The local flow angles were computed from velocity signals extracted at the positions of the five-hole pitot pressure probes shown in Figure 4.3. Figures 4.30 and 4.31 show the time traces of the velocity signals from the probes placed at 34% and 84% span on blades 1 and 2 with corresponding local flow angles (LFA) shown in Figures 4.32 and 4.33 for the last passage shown in Figures 4.26 and 4.27. The velocities are defined relative to the local blade coordinate system shown in Figure 4.6. Note that the azimuth angle corresponds to the probe position and not to the actual blade position, which will be slightly

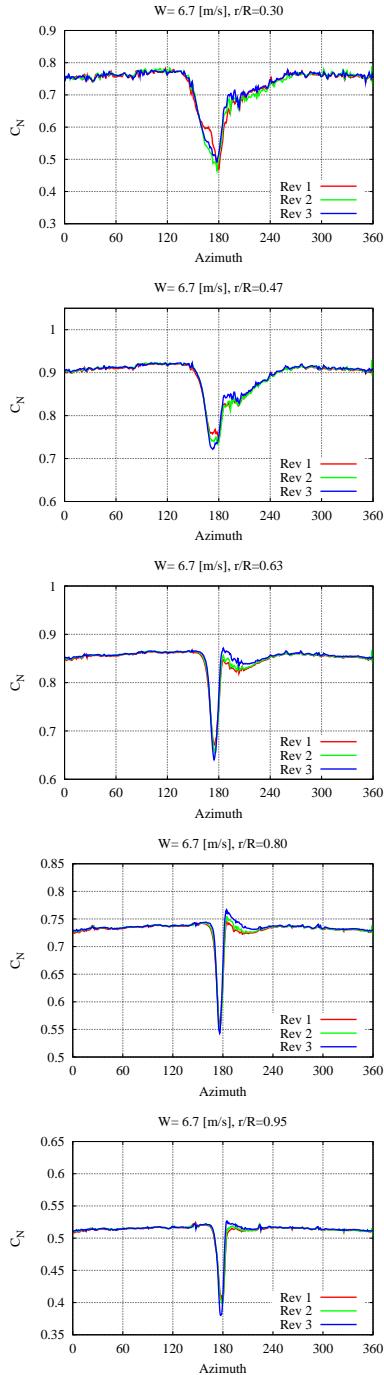


Figure 4.26: Normal force coefficient on blade 1 for three consecutive revolutions.

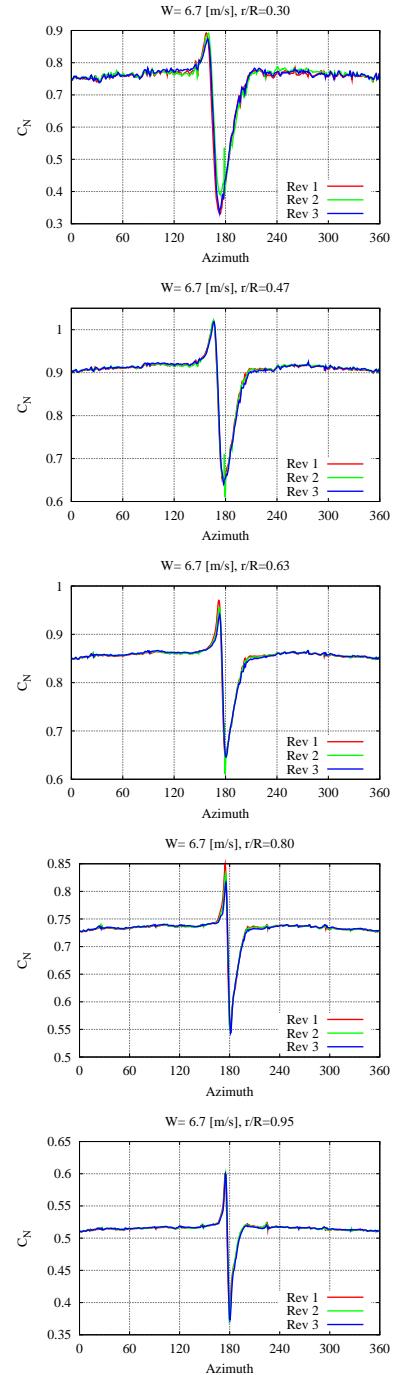


Figure 4.27: Normal force coefficient on blade 2 for three consecutive revolutions.

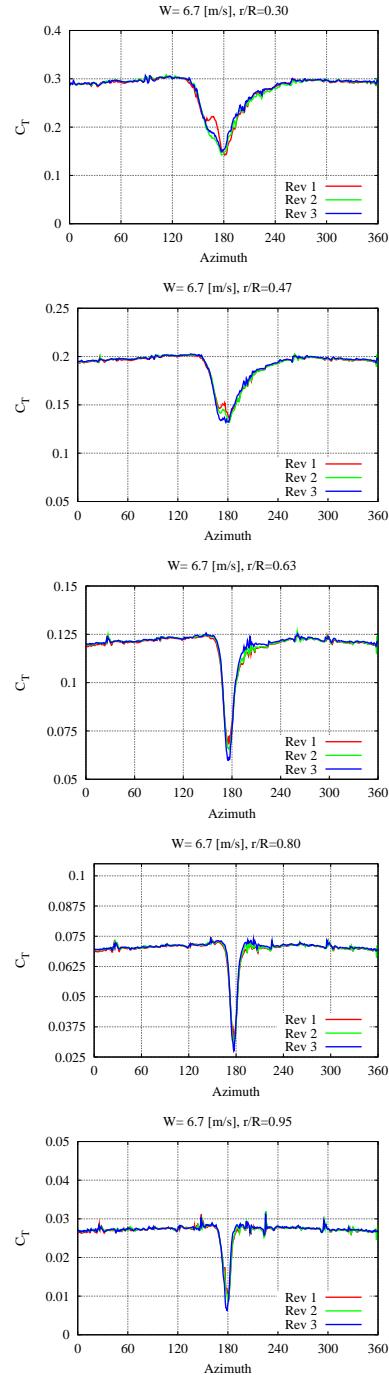


Figure 4.28: Tangential force coefficient on blade 1 for three consecutive revolutions.

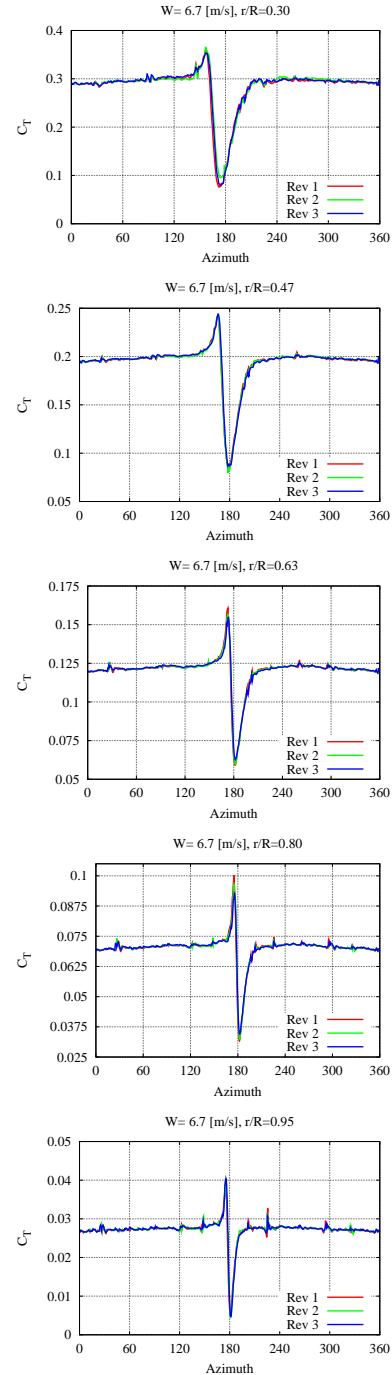


Figure 4.29: Tangential force coefficient on blade 2 for three consecutive revolutions.

lagged depending on the radial position of the probe.

The tower passages are characterised by large variations in axial and tangential velocities, whereas the change in spanwise velocity is smaller. The local velocity field is strongly dependent on the exact state of the wake and is very different during the two passages shown. The time series of the velocity signals for the outer sections correspond to the flow field snapshots shown in Figures 4.23 and 4.24 for blades 1 and 2, respectively.

On the inner section, blade 1 (Figure 4.30) experiences a large reduction in axial velocity accompanied by an equally large increase in tangential velocity upon entering the wake which both reach their respective minimum and maximum at $\psi = 180^\circ$. This is followed by a steep increase in axial velocity and decrease in tangential velocity. On the outer section, the blade experiences a strong reduction in axial velocity as it enters the deficit. As the blade approaches the vortex of negative vorticity, the tangential velocity starts to decrease sharply and the axial velocity decreases further and reaches a minimum at $\psi = 176^\circ$. As the axial velocity increases, the tangential velocity decreases further due to the strong rotation of the vortex and reaches a minimum at $\psi = 180^\circ$. The axial velocity reaches a maximum at $\psi = 182^\circ$. The recovery is fairly fast with freestream level reached at $\psi = 190^\circ$.

The situation is quite different for blade 2 (Figure 4.31) where the probes on both the inner and outer sections experience an upwash from the shed vortex before entering the deficit, with a maximum axial velocity at $\psi = 164^\circ$ and $\psi = 176^\circ$ for the 34% and 84% span sections, respectively. Opposite to blade 1 the recovery to freestream level is quite slow. At the inner probe the tangential velocity starts to decrease at $\psi = 160^\circ$ reaching a minimum at $\psi = 176^\circ$. This is followed by a steady increase towards the freestream value. For the outer section the situation is reversed: the counter-clockwise rotation of the vortex (Figure 4.24) causes an increase in tangential velocity which reaches a maximum at $\psi = 177^\circ$. The subsequent reduction in axial velocity reaches a minimum at $\psi = 181^\circ$. For both blades, it can be seen that when the tangential velocity increases the spanwise velocity decreases and vice versa.

The changes in local flow angle are proportional to the changes in tangential and axial velocity relative to the blade velocity at the given span section. On the inner parts of the blade the LFA will therefore be very sensitive to changes in both tangential and axial velocity, whereas on the outer sections where the blade velocity is far greater than the change in tangential velocity,

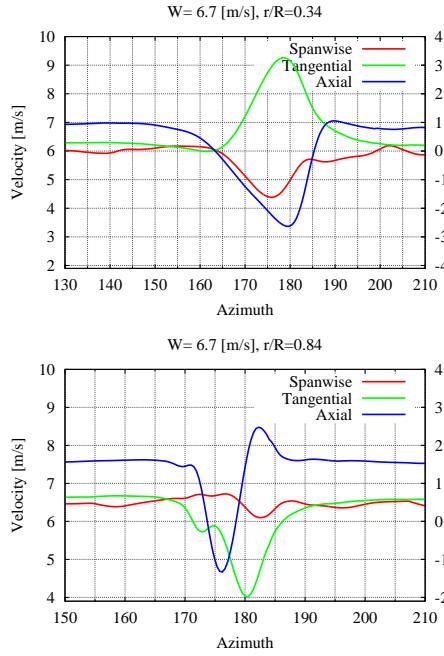


Figure 4.30: Probe velocity traces for blade 1 at 34% and 84% span. Axial velocity is shown on the left axis and spanwise and tangential velocity is shown on the right axis.

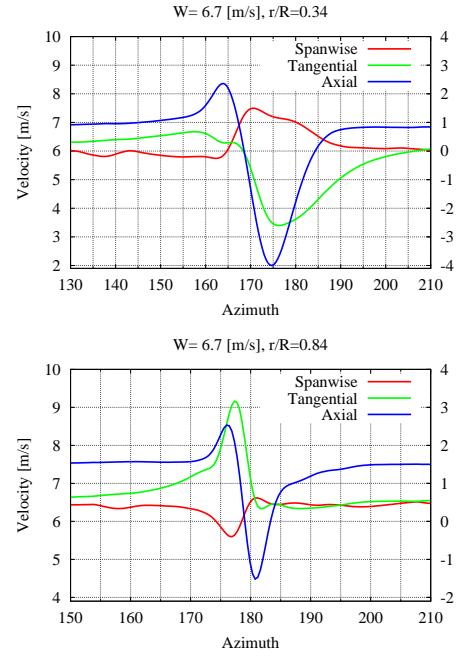


Figure 4.31: Probe velocity traces for blade 2 at 34% and 84% span. Axial velocity is shown on the left axis and spanwise and tangential velocity is shown on the right axis.

the local flow angle will mainly be dependent on the change in axial velocity.

Figures 4.32 and 4.33 show the local flow angles for the two blades at 34% and 84% span. The LFAs generally follow the behaviour of the axial velocity quite closely for both blade sections. On the inner section the LFA is 9.3° relative to the local blade section chord line and a bit higher on the outer section with 9.4° . From an azimuth angle of approximately 145° the inner section experiences a gradual decrease in flow angle which amounts to an almost 10° reduction at $\psi = 179^\circ$. This is followed by a sharp increase which reaches a maximum at $\psi = 187^\circ$. On the outer section of blade 1 the wake interaction does not start until $\psi = 170^\circ$. The probe experiences a decrease in LFA of 5.3° as it enters the deficit, followed by a sharp increase to 0.9° above freestream level as it exits the wake.

As shown in Figure 4.33, the inner probe on blade 2 experiences a gradual increase in LFA as it approaches the wake. This is followed by a steeper upwash which reaches a maximum at $\psi = 163^\circ$. The subsequent dramatic decrease in incidence is caused by a combination of an increase in relative axial velocity and large decrease in axial velocity. The outer probe on blade 2 experiences an upwash before entering the deficit with an increase in flow angle of 2.7° . The subsequent reduction in flow angle is similar to that of blade 1 with a

minimum of 4.1° . As blade 2 exits the wake the recovery on both sections is fairly smooth.

As can be seen in Figures 4.32 and 4.33, the changes in LFA are, as expected, considerably larger on the inner sections of the blade. This is also reflected in the normal and tangential forces as discussed previously.

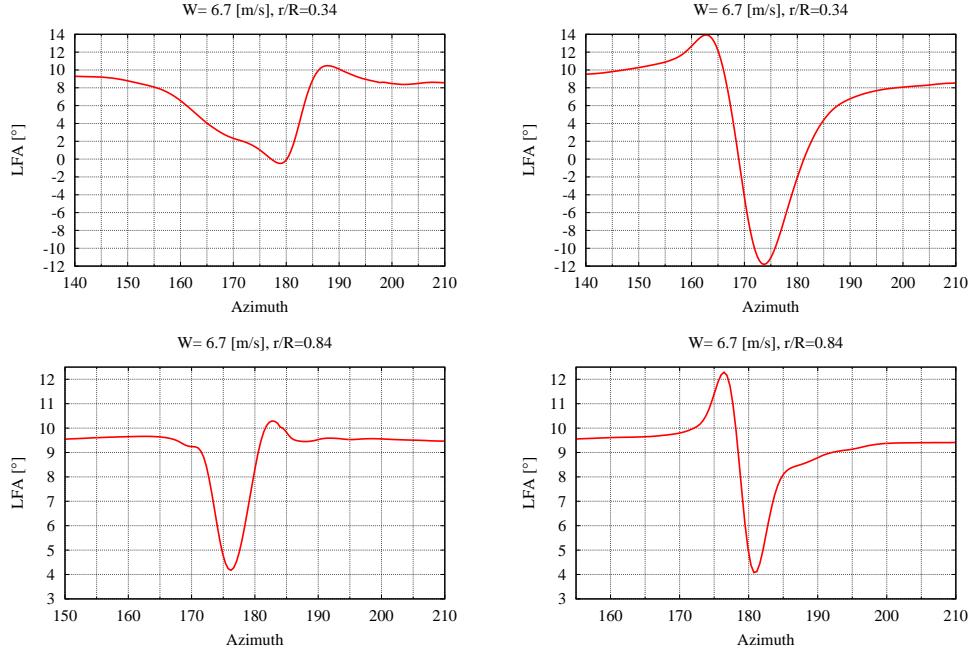


Figure 4.32: Probe local flow angles relative to the blade section chord line for blade 1 at 34% and 84% span.

Figure 4.33: Probe local flow angles relative to the blade section chord line for blade 2 at 34% and 84% span.

By assuming that the blade responses are quasi-static, thus neglecting unsteady 3D effects and hysteresis effects, the local blade incidence can be estimated as it travels through the wake by comparing the loadings on the blade in Figures 4.26 and 4.27 with the measured 2D blade force coefficients shown in Figure 4.34. The freestream incidence of the 80% span section of blade 1, corresponding to $C_N = 0.74$, should thus be approximately $\alpha = 5.8^\circ$. As the blade enters the wake the normal force reduces to $C_N = 0.55$ which translates into a reduction in incidence with a minimum of $\alpha = 4.2^\circ$. As the blade exits the wake the incidence increases to slightly above freestream level to $\alpha = 6^\circ$ before recovering completely. Blade 2 initially experiences an increase in incidence to $\alpha = 6.2^\circ$ before reducing to $\alpha = 4.1^\circ$ followed by a fairly slow increase in incidence.

The derived blade incidences differ significantly from the LFAs extracted upstream of the blade both in freestream level and during the blade wake

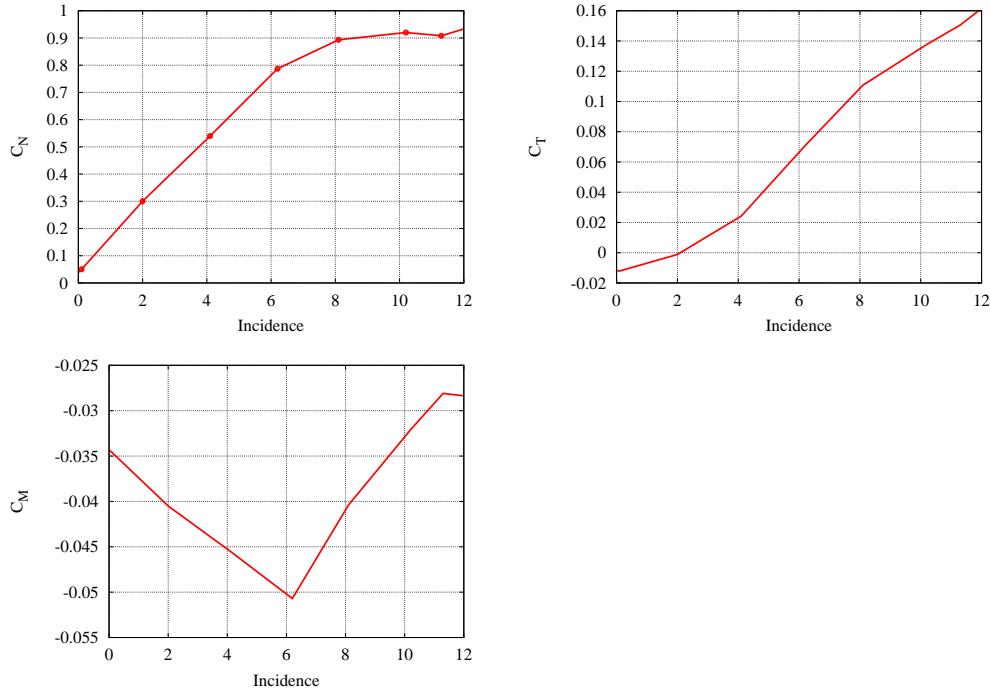


Figure 4.34: Aerodynamic force coefficients for the S809 aerofoil measured at Delft University at $R_e = 1 \times 10^6$ [57].

interaction. The most likely explanation for this is the fundamental difference between a local flow angle and the actual angle of attack on the blade. To relate the two quantities some correction for the lift induced upwash upstream of the blade has to be applied. The same argument can be used to explain why the absolute changes in LFA are not reflected directly in the C_N and C_T responses. As the blade travels through the wake the change in bound circulation will trigger a change in the self-induced upwash upstream of the blade, thus further obscuring the correlation between changes in local flow angle and changes in angle of attack.

Another important point is that the blade wake interaction must be regarded as a phenomenon which is characterised by a combination of a number of unsteady effects associated with the interaction of the blade with the unsteady tower wake and cannot be regarded as an event associated with changes in angle of attack.

Moment and Surface Pressure Coefficients

Figures 4.35 and 4.36 show detailed plots of the normal force and moment coefficients for the tower passages discussed above. These are accompanied by

plots of the pressure distributions at the six instances marked on the moment curves. The crosses on the normal and moment coefficient curves correspond to the flow field snapshots at $\psi = 177^\circ$ shown in Figures 4.23 and 4.23.

Following the argument presented in the previous section, the behaviour of the moment coefficient of the blade should correspond to the changes in incidence derived from the C_N curve if the behaviour is assumed to be quasi static. Figure 4.34 shows that the moment coefficient reduces with increasing incidence up to $\alpha = 6.2^\circ$ and it increases afterwards. At an incidence of $\alpha = 5.8^\circ$ the moment coefficient should thus be $C_M = -0.050$. However, as can be seen, the freestream value of the moment coefficient is considerably higher at $C_M = -0.042$. Assuming that the qualitative behaviour of the computed moment coefficient is similar to that from the wind tunnel measurements the reduction in the incidence of blade 1 should produce a nose up moment. Indeed this is the case as can be seen in point B of Figure 4.35. From the normal force plot the moment coefficient should continue to increase until an azimuth angle of $\psi = 177^\circ$. However, this is clearly not the case as marked by point C. The blade experiences a sharp nose down pitching moment which reaches a minimum at $\psi = 177^\circ$ azimuth which is followed by an increase with two distinct peaks marked by points D and E until reaching freestream level at point F.

Blade 2 appears to follow the static C_M curve slightly more where the decrease in incidence results in a large nose up moment. However, the decrease in incidence of 2.5° from point B to point C does not account for the large increase in the nose up moment.

This strongly suggests that the interaction of the blade with the tower wake introduces local changes in the pressure distribution beyond those associated with changes in incidence, which alters the moment acting on the blade. Figure 4.37 shows a time history of the surface pressures on the blade during the tower passage shown in Figure 4.23. In addition to the reduction in incidence associated with the velocity deficit, the interaction with the vortex seems to cause a complete collapse of the suction peak, which results in a strong nose down moment. As the blade exits the wake the suction peak recovers which increases the nose up moment. The events marked D and E could be associated with the slight changes in the trailing edge suction which is visible in Figure 4.37. Event D is thought to be caused by the trailing edge interacting with the vortex shed from the downwind side of the tower. The likely cause of event E is the shedding of bound vorticity from the leading edge as a result of the

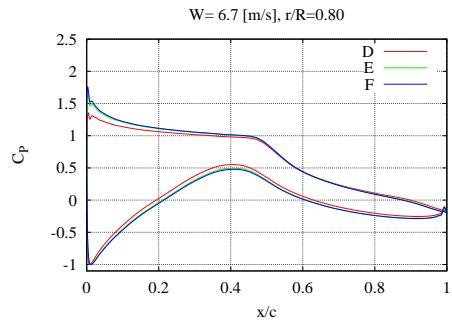
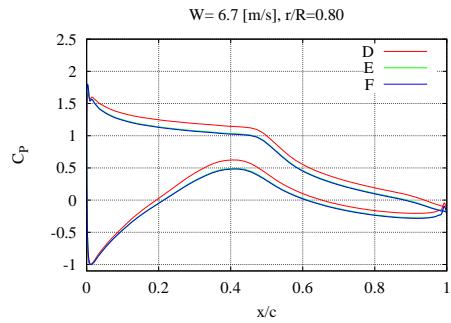
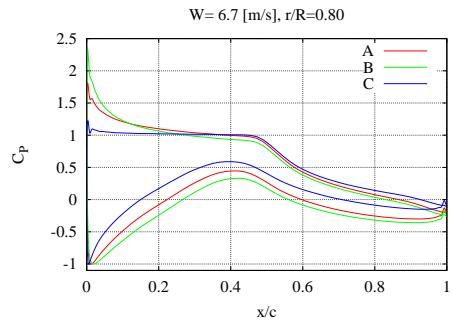
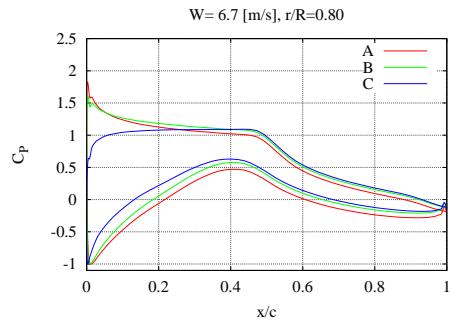
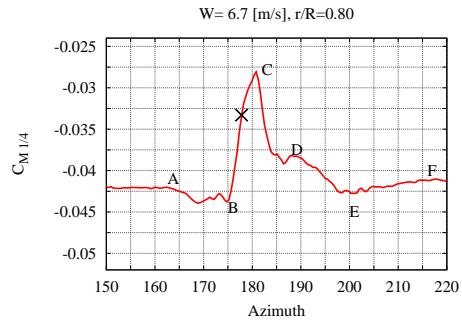
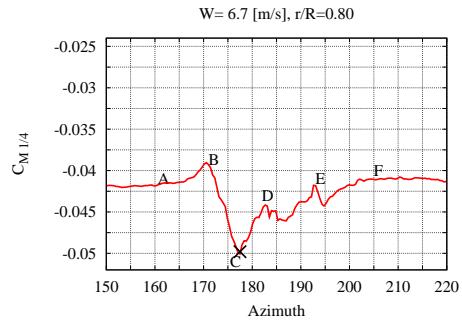
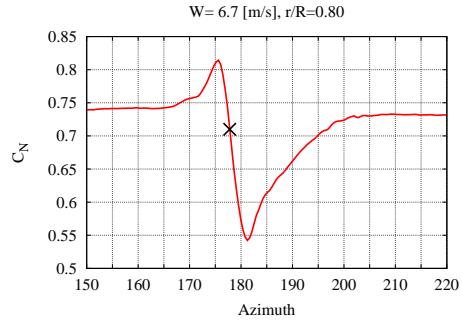
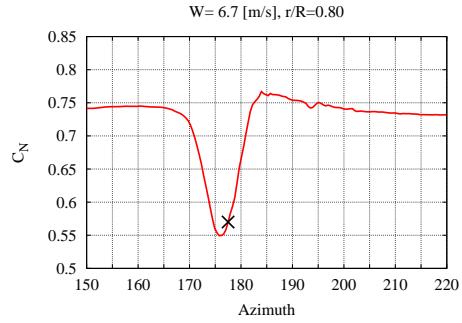


Figure 4.35: Detail of normal force and moment and pressure coefficients on blade 1 at 80% span corresponding to tower passage 3.

Figure 4.36: Detail of normal force and moment and pressure coefficients on blade 2 at 80% span corresponding to tower passage 3.

change in loading on the blade. As the vorticity convects over the surface and passes the trailing edge it causes a change in the moment acting on the blade.

A similar analysis can be performed on the moment response of blade 2 as shown in Figure 4.36. As shown in Figure 4.24 the blade does in this case

encounter a vortex of positive vorticity (type 2). This instant is marked by a cross on the curve in Figure 4.36 and corresponds to an azimuth angle of $\psi = 177^\circ$. The increase in incidence causes an increase in loading on the blade, most visible on the leading edge. The local minima in C_M marked by point B in Figure 4.35 could be associated with the increase in C_N which corresponds to an increase in incidence from $\alpha = 5.8^\circ$ to $\alpha = 6.2^\circ$. Referring to Figure 4.34 this would be in the range where the slope of the C_M curve changes sign. The encounter with the vortex does in this case not cause a collapse of the suction peak as was seen on blade 1. As the blade passes the vortex and experiences a decrease in incidence the moment decreases momentarily before increasing drastically reaching a maximum at point C. A change in the moment coefficient of 0.014 would from the quasi static consideration correspond to a decrease in incidence of approximately 5° . However, this does not correspond to the magnitudes of the normal and tangential force responses. It is likely that this large increase in the moment is caused by an increase in suction on the pressure side associated with the interaction with the vortex shed from the tower. This is followed by a rapid decrease in the moment as the vortex convects past the trailing edge, which at point D causes a small wiggle in the moment curve as is visible in Figure 4.37. As the blade incidence increases, the moment recovers steadily to the freestream level. Event E could as for blade 1 be associated with the shedding of bound vorticity.

Comparison With Experiment

As noted previously, the variation between consecutive passages in the computed blade responses was very limited, which indicates that the tower vortex shedding frequency coincided with the blade passage frequency. However, if one turns to the experimental results, this phenomenon cannot be observed. In fact, it is evident that the response of the blade varies greatly over the course of the measurement campaign. Figure 4.39 shows the blade normal force coefficients as a function of azimuth at the five stations along the blade compared to the experimental results. The experimental data set consists of 35 consecutive revolutions that are all plotted together with the azimuthal average. Due to the lack of variation of the response in the computations it was not meaningful to compute the averaged forces of the simulations. Instead, the responses characteristic of the two blades are plotted individually.

Firstly, it is evident that the freestream level of the normal force is in reasonably good agreement with the experiment. The agreement in the C_n

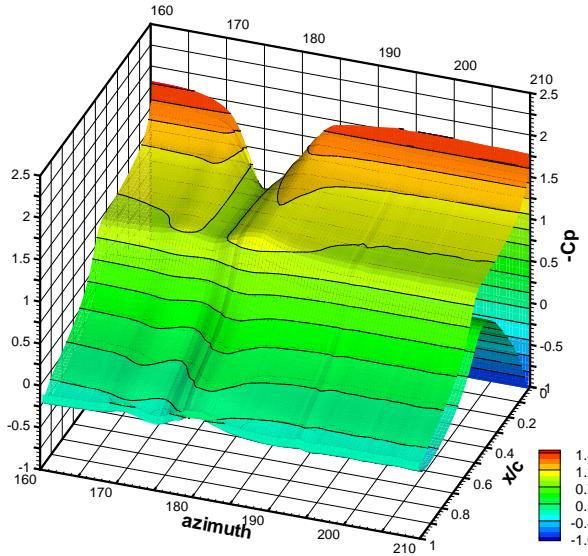


Figure 4.37: Surface pressure coefficients on blade 1 at 80% span corresponding to tower passage 3.

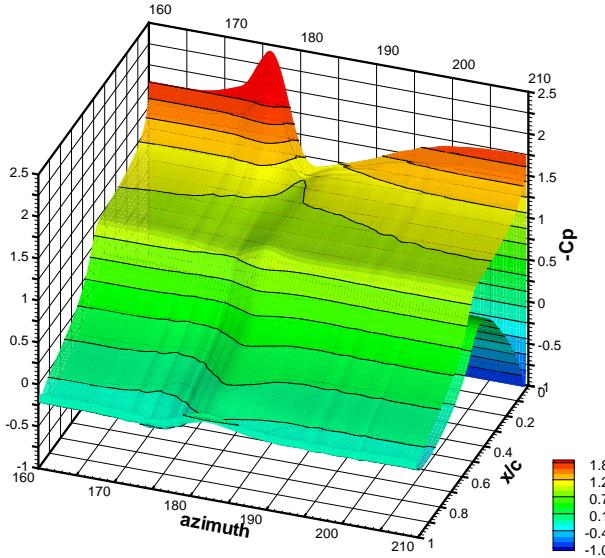


Figure 4.38: Surface pressure coefficients on blade 2 at 80% span corresponding to tower passage 3.

level is best on the middle sections of the blade where the flow is mostly two-dimensional, whereas on the inner and outer parts of the blade where three-dimensional effects are dominant, the agreement is slightly poorer. Looking at the three middle sections of the blade, the responses in the tower shadow region all fall within the experimental spread, although the large influence of

the tower top observed in the computations is not present in the experiment.

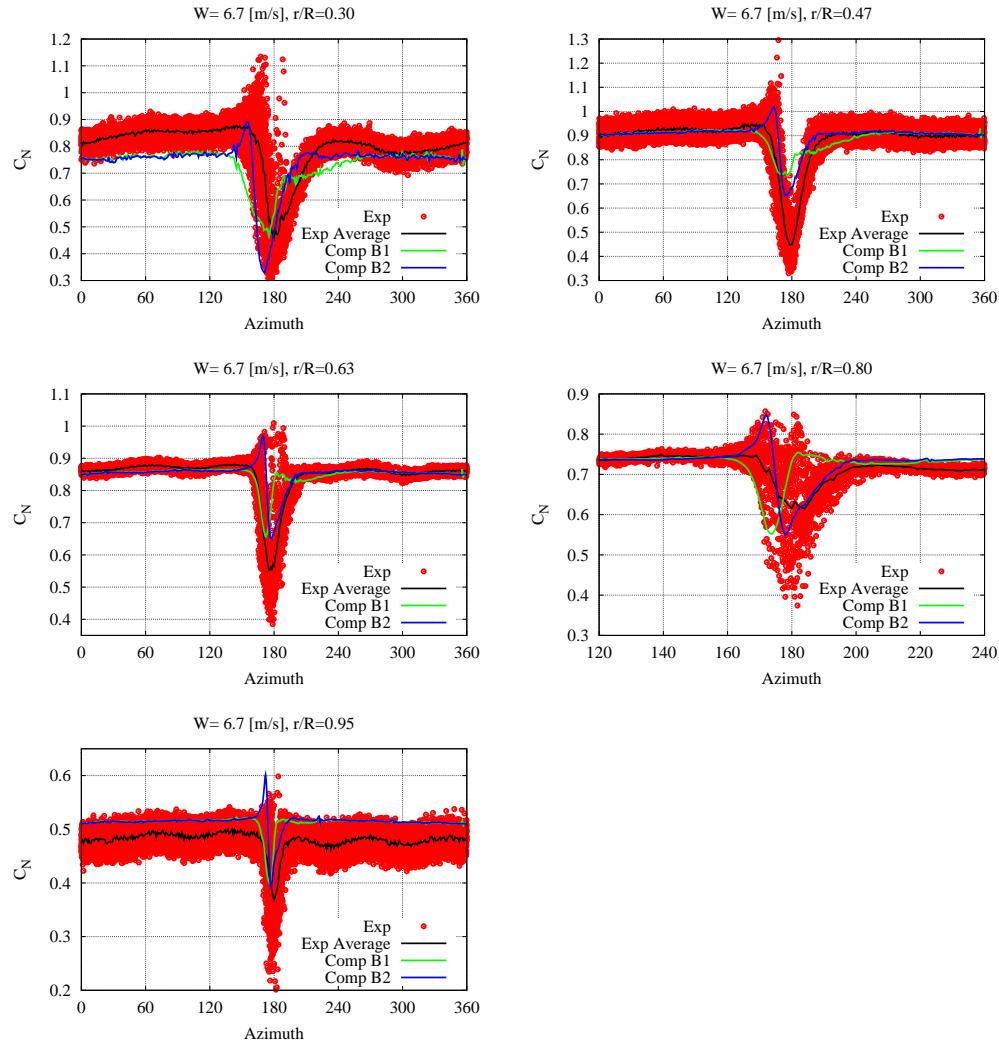


Figure 4.39: Comparison of computed and experimental normal force coefficients for the downwind configuration for a wind speed of 6.7 m/s.

As it was shown in the previous section, the exact response of the blade as it travels through the tower wake is strongly dependent on the specific state of the wake. To analyse the accuracy of the computations further and to investigate whether the computations capture the same type of effects as it was seen in the experiment, interaction events similar to the ones observed in the computations were therefore identified from the experimental results. Generally, it was found that a greater proportion of the blade responses in the experiment were of the same nature as the blade 2 type response, whereas the blade 1 type response was more rare. Figure 4.40 and 4.41 show a comparison of the C_N , C_T and $C_{M1/4}$ curves for two such events for the 80% section; one for blade 1 and another for blade 2. As it can be seen the comparison for the blade 2 type response is the most favourable of the two, where the agreement is very good in all three curves. The BWI event characterised by a sharp increase in C_N followed by a decrease and a slow recovery, does in the experiment also produce a strong nose up moment which has a maximum at the point where the C_N curve reaches its minimum. It also seems that the subsequent recovery as the blade exits the wake is in good agreement. As it was concluded in the previous section, the moment response is very sensitive to local changes in pressure on the blade section. The fact that isolated passages from the experiment are in such good agreement with the computations clearly suggests that the simulations capture the detailed physics underlying the interaction between the rotor and the tower wake very accurately.

Although the agreement in the blade 1 event is not as striking, it is however evident that the nature of the responses are similar. A BWI event characterised by a decrease in C_N followed a fairly smooth increase with no sharp peaks does in the experiment also produce a nose down moment which has a minimum at the point where the C_N curve reaches its minimum. Again this supports the argument that the numerical solution is indeed capable of capturing the key features of the BWI of a downwind turbine.

Lastly, the local flow angles at the 84% span section extracted from the computations were compared to the experimental results. Figure 4.42 shows the entire experimental data spread and the azimuth average compared to the computed values. As it can be seen the computed values lie approximately 1° above the experiment in freestream conditions. It is likely that this is caused by the fact that the method used to extract flow velocities in the computations uses the cell center value of the cell enclosing the probe point. Since the flow gradients are very high in the rotor plane, it is likely that a small difference in

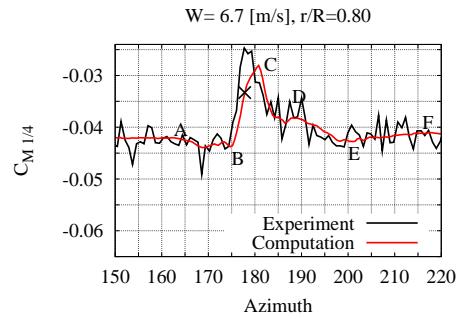
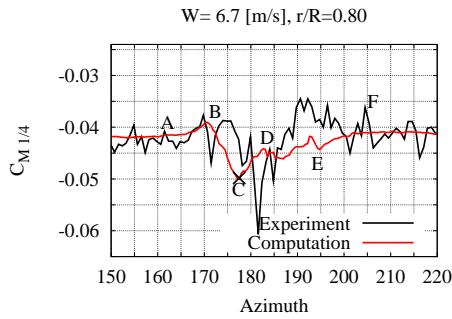
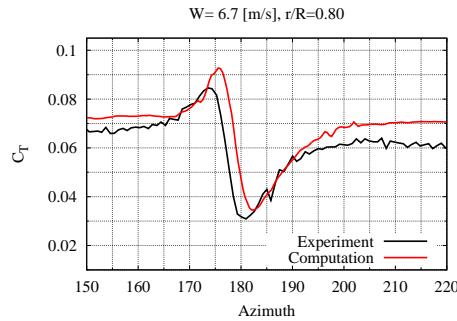
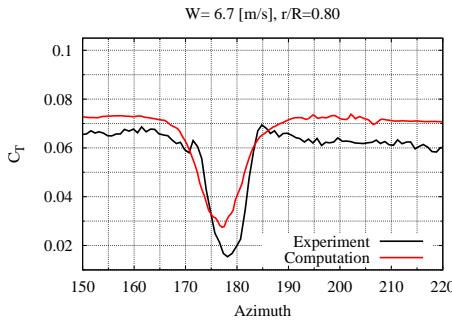
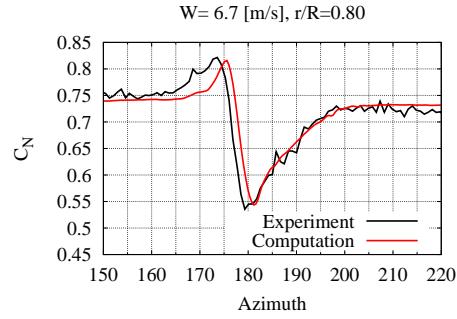
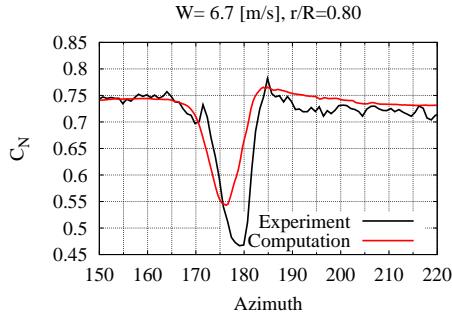


Figure 4.40: Normal force, tangential force and moment coefficients on blade 1 at 80% span. Comparison of computations and experiment for a single tower passage.

Figure 4.41: Normal force, tangential force and moment coefficients on blade 2 at 80% span. Comparison of computations and experiment for a single tower passage.

the location of the probe could be the root of the difference in the freestream flow angles.

Figures 4.43 and 4.44 show a comparison of the LFAs for a single tower passage corresponding to the passages shown in Figures 4.40 and 4.41. Although the exact behaviour of the LFA is not the same in experiment and computation, the LFA for blade 1 in both experiment and computation reaches a minimum at the point where the normal force has its minimum. Additionally, the reduction in flow angle is comparable of about 5° . The LFA for the blade 2 event is not in as good agreement, although the qualitative behaviour is the same. As it was found in the computations, the experimental results show that an increase in C_N is associated with an increase in LFA. The subsequent

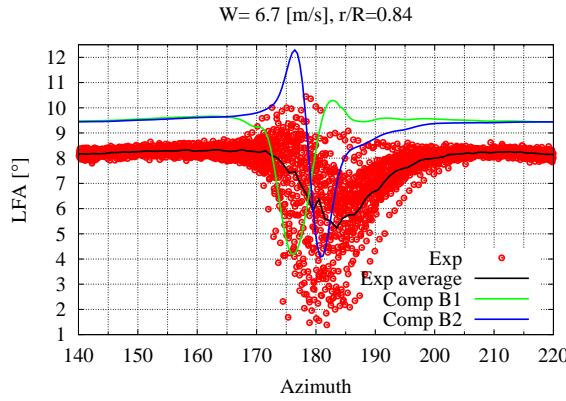


Figure 4.42: Local flow angles at 84% span for the downwind configuration for a wind speed of 6.7 m/s.

reduction of the LFA is, however, much greater in the computations.

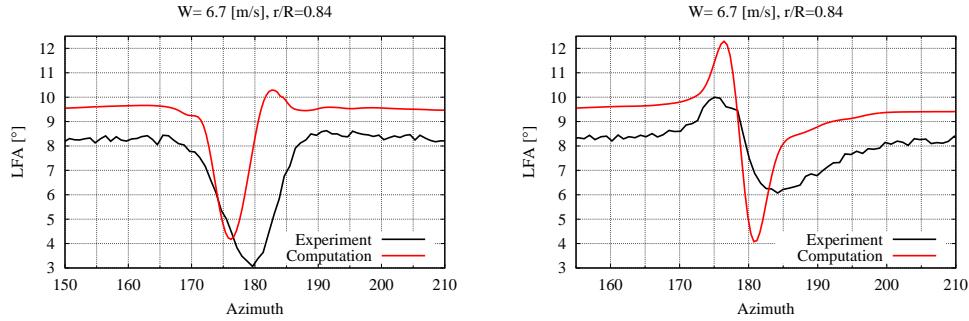


Figure 4.43: Local flow angles on blade 1 at 84% span. Comparison of computations and experiment for a single tower passage.

Figure 4.44: Local flow angles on blade 2 at 84% span. Comparison of computations and experiment for a single tower passage.

4.7.4 Tower Response

As it has already been pointed out, it seemed that the responses of both blades to the BWI entered a periodic state. Secondly, there was very little variation of the blade response along the span of the rotor. This suggests that the shedding frequency of the tower was in phase with the blade passage frequency and that the shedding was correlated along the span of the tower. However, in the computation of an isolated tower presented in Section 4.6, the shedding frequency was found to be 4.1Hz, which is not equal to or a multiple of the rotational frequency of 1.2 Hz (1P). Secondly, the shedding from the isolated tower was highly oblique, causing a phase difference in the shedding

along the span.

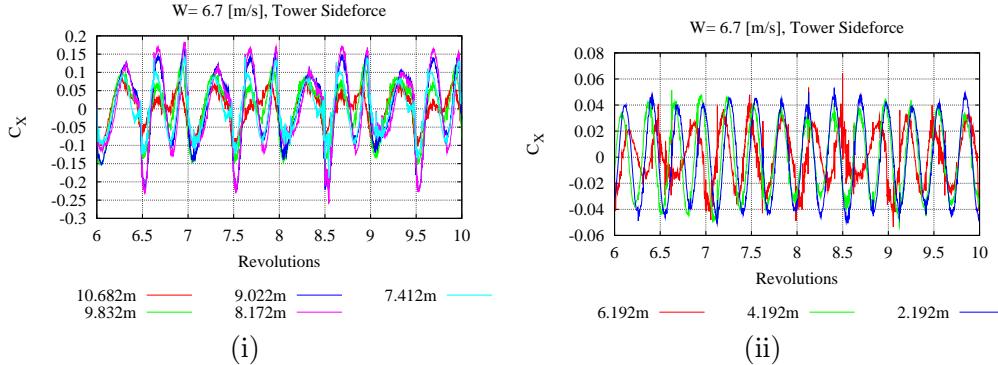


Figure 4.45: Time history of the side force coefficient for a wind speed of 6.7 m/s at (i) the equivalent heights of the five rotor stations and (ii) at various heights on the tower below the rotor disc. Blade passages take place every half revolution.

This indicates that the tower shedding was modified strongly by of the presence of the rotor. Indeed, this is illustrated in Figure 4.45 where the side force on the tower is plotted at the same vertical positions as the blade pressure tap positions as well as at three lower stations on the tower. The shedding pattern within the rotor radius is strongly affected by the blade passages which cause the shedding to become irregular and asymmetric. Additionally, the vortex shedding is clearly periodic with three shedding periods per revolution. As it can be seen, the shedding is strongly correlated along the span of the tower. The shedding strength is amplified significantly which appear to be strongest on one side of the tower with peak amplitudes of up to 0.24. The blade passage also has a strong effect outside the rotor disc as it can be seen in Figure 4.45 (ii) where the side force at $y=6.192$ m (1 m below the rotor) is still driven by the blade passage frequency. On the lower part of the tower, however, the shedding frequency is higher at 4.1 Hz, equal to that found on the isolated tower. The rotor thus causes a 12% shift of the shedding frequency on the part of the tower within the rotor disc compared to the shedding frequency on the lower part. As was previously pointed out, this change gives rise to a dislocation of the vortex street as shown in Figures 4.20 and 4.21.

Figure 4.46 shows the frequency content of the side force on the tower, with peaks at approximately 2.4 Hz (2P relative to the rotational frequency) and 4.8 Hz (4P) as well as the two vortex shedding frequencies at approximately 3.6 Hz (3P) and 4.1 Hz. The accuracy of the power spectrum, however, is quite low since the time series is too short to resolve the frequencies any finer than 0.244 Hz.

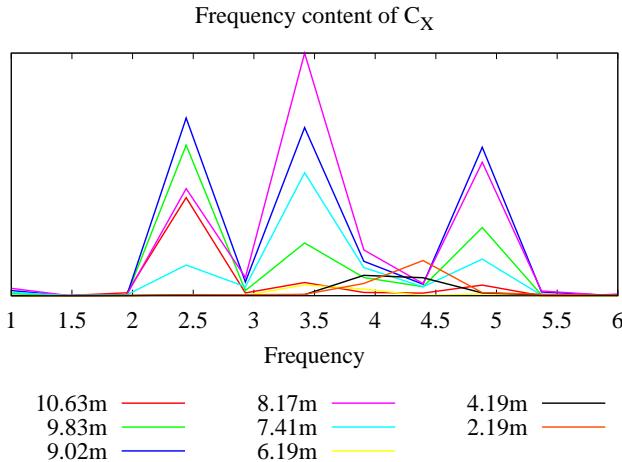


Figure 4.46: Frequency content of the side force on the tower at all vertical stations.

4.7.5 Discussion

The results of the current investigation show that not only does the tower wake have a strong effect on the local aerodynamic coefficients on the blade, the rotor also influences the vortex shedding from the tower. The complex interaction between the rotor and the tower appears to be governed by a number of factors, which are inherent to the exact state of the flow.

The aerodynamic response of the blade as it travels through the tower wake could be considered as a superposition of two separate types of effects, one being the dynamic response of the blade to changes in the axial velocity associated with the mean wake deficit, and the other being the blade response to the unsteady interaction with the shed vortices from the tower which induce considerable local changes in both axial and tangential velocity. The net effect of this cannot, however, be viewed as an effective change in incidence because the incidence will not be constant along the chord of the blade as it travels through the wake. The three-dimensionality of the flow over the blade adds further complexity to the interaction of the blade and the wake which is particularly significant in the root sections of the blade. The complexity of the problem of blade wake interaction is therefore considerable, due to the complex interaction between these effects. The fact that the nature of BWI is highly complex was clearly evident in the quarter chord pitching moment response which was shown to relate critically to the interaction of the blade with the shed vortices from the tower and did not follow the trend of the local flow angles.

On blade 1 the recovery of the normal and tangential forces was very slow on the inner parts of the blade (Figure 4.26) compared to the recovery on the outer sections. This differed from the behaviour of the blade 2 responses which were similar along the span of the blade. Part of the explanation for this could be that the inner part of the blade was affected by the highly three-dimensional flow produced at the top of the tower. The downward velocities and streamwise vorticity downstream of the tower top could be the main contributors to this response. The neglect of the nacelle in the computations makes it difficult to speculate whether this result is purely an artifact of the tip shape of the tower, or whether similar effects would exist had the nacelle been modelled.

The dynamic response of a blade in an unsteady flow is commonly associated with the so-called reduced frequency $k = \frac{c\omega}{2U_{rel}}$, where c is the local chord length, ω is some characteristic frequency, and U_{rel} is the relative velocity seen by the blade. In the case of blade wake interaction, the characteristic frequency will be proportional to the time spent in the unsteady wake of a given span section. Using the local flow angles presented in Section 4.7.3 to determine the time spent in the wake the following was derived (note that this also has to take into account that the local blade section has a finite length, which means that the root sections of the blade will subtend a larger azimuthal range than the outer sections, increasing the effective time spent in wake conditions): at the 34% section the time spent in the wake was 0.22 s assuming that the wake effect subtended 95° azimuth, resulting in a characteristic frequency $\omega = 4.6 s^{-1}$. At the 84% section the wake effect subtended 32° azimuth such that the time spent in the wake was 0.073 s, yielding a characteristic frequency $\omega = 17.29 s^{-1}$. This results in a reduced frequency $k = 0.123$ on the inner part of the blade and $k = 0.100$ on the outer part. Although the relative blade velocities vary significantly along the span, the effective reduced frequency is fairly constant since the velocities are balanced by the time scales associated with the BWI, which increase towards the root. This would suggest that - for the present wind turbine - unsteady effects such as phase-shifts and time delays in the blade response associated with BWI should not vary significantly along the span of the blade. Indeed, it was found in Section 4.7.3 that the normal and tangential forces on the blade generally followed the behaviour of the local flow angle quite closely without any significant phase-lag.

Common to both blade responses was that the normal and tangential force increased steadily from 0° azimuth to 180° azimuth. Equivalently, the loading decreased slightly throughout the second half of the revolution and reached a

minimum at 0° azimuth. This could be caused by the bottom wall boundary which was likely to give rise to a blockage of the flow resulting in a slight increase in axial velocity.

The tower wake development is in turn equally dependent on the rotor loading and blade passage frequency. The loading of the rotor dictates the axial induction which induces periodic changes in the axial and tangential velocities upstream of the rotor and hence modifies the shedding frequency, phasing and pattern from the tower. It can be seen in Figure 4.21 that the vortex tubes convect at different velocities along the span of the tower. The vortex shedding is strongly oblique in the region near the edge of the rotor disc. Two mechanisms appear to be at play: one being the difference in axial velocity caused by the rotor induction, and the other the interaction between the tower vortex street with the blade tip vortex, which seems to cause a roll-up and stretching of the tower vortex street. A second mechanism which further influences the phasing and pattern of the shedding is the blade vortex interaction due to the partial disintegration of the interacting vortices. The combination of these effects does, in the present case, lead to a phenomenon which has so far not been associated with wind turbine flows. This will be discussed shortly. Downstream of the turbine, the wake of the rotor and the tower also interact strongly. As was shown in Figure 4.21 the rotor tip vortices appear to cause a roll-up of the tower vortex street around the tip vortices. Additionally, it was, as shown in Figure 4.25, observed that the rotation of the rotor wake gives rise to a skewing of the tower wake to one side causing the rotor and tower wakes to interact in the far wake.

The distinct lack of variation in the blade response and the strong correlation of the tower wake naturally leads one to ask what caused this behaviour. From other aerodynamic applications, most notably the extensive body of work on forced vibration and vortex induced vibration of circular cylinders, it is well-known that vortex shedding from a cylinder can be altered significantly if the structure is subjected to periodic vibration with a forcing frequency that is close to the shedding frequency. The forcing can cause the spanwise correlation of the wake to increase and the frequency and phase of the shedding to be altered. Secondly, it can cause an increase of the shedding strength and mean drag on the cylinder. When the natural frequency of the system and the shedding frequency are sufficiently close to each other, the shedding frequency can become synchronised with the forcing frequency. This phenomenon is known as *lock-in*. Lock-in can also occur if the two frequencies are multiples

of each other, although the strength of the above mentioned effects become less pronounced. Depending on the amplitude of the forcing, the shift of the shedding frequency can be as much as 40%. A detailed review of this subject is given by Blevins [9].

The effect of the downstream rotor on the tower response clearly bears strong resemblances to the characteristics of *vortex lock-in* as described above. The lock-in of the shedding frequency on the blade passage frequency explains the periodicity and spanwise correlation of the blade responses. As discussed above, lock-in can occur when the shedding frequency is close to or a multiple of the forcing frequency, which in the present case was 3P. It is likely that lock-in could take place at a range of different wind speeds (or tip speeds), where the natural shedding frequency is close to a multiple of the rotational frequency. It seems likely that the rotor can modify the wake structure also when the turbine operates outside the range of lock-in although the effect will not be as strong as in the lock-in regime. To substantiate the above arguments, however, further investigations are necessary since at this stage, computations have only been carried out at an inflow velocity of 6.7 m/s.

As shown in Figures 4.26 and 4.27 it does appear as if the blade responses slowly change from passage to passage; a peak starts to appear as blade 1 exits the tower wake, and similarly, the peak of blade 2 becomes smaller. This suggests that the tower shedding frequency is slightly different from the blade frequency, causing a very low frequency change in the blade response, which could cause the flow to break up and settle into a new lock-in equilibrium. This is again also a characteristic of lock-in phenomena, where very low frequency oscillations have been observed. However, due to the high computational cost, it has not been possible to run the present simulation long enough to establish whether this also occurs in the rotor-tower flow.

Since there is no data available on the tower forces in the NASA Ames experiment, it is not possible to verify whether something similar to this phenomenon also occurred in the experiment. However, from an examination of the blade force responses it was not possible to identify strong periodicity at any of the wind speeds. Likewise, correlation of the blade responses in the spanwise direction were not as strong as in the computations, indicating that the cellular shedding structures were smaller in the experiment than it was found in the computations. One factor which could partly explain the lack of wake correlation and that lock-in was not observed could be that the tower operates in the critical Reynolds number regime where the wake of the tower

is highly disorganised due to the transition of the flow in the separation region on the cylinder. As discussed in Section 4.6, the present computation does not take transitional effects into account, which means that the highly unsteady flow characteristics in this regime are not modelled. In future experiments it would be advisable to either trip the flow or increase the tower diameter to make the flow more deterministic.

This clearly raises the question whether these findings are purely an artifact of the numerical model or whether indeed this could occur in real-life conditions. As it has already been discussed in Section 4.6, it is well-known that the use of Reynolds averaging has a tendency to cause structures in the flow to be artificially two-dimensional with large correlation lengths. All structures of small scales in the flow are modelled using an appropriate turbulence model, which particularly for massively separated flows is not always very accurate. LES or hybrid LES models, on the other hand, are capable of reproducing a flow field with three-dimensionally resolved turbulent quantities, which have been shown to predict separated flows more realistically. It is therefore worth investigating whether a more accurate modelling of the small length and time scales of the flow could have an effect on the lock-in phenomenon identified in this work.

The present computations were conducted under very controlled conditions with no background turbulence and a very thin ground shear layer which does not represent realistic atmospheric conditions. An obvious question would be what effect shear and background turbulence have on the vortex shedding on the tower. One might speculate that these factors could cause the vortex lock-in phenomenon to occur less frequently, since they are likely to reduce the spanwise correlation lengths of the vortex shedding.

Another important parameter, which has not been discussed so far, is the blade tower clearance. Its importance is clearly visible in Figures 4.23 and 4.24, where it can be seen that the blade passes through the wake in the region where the shed tower wake vortices are detaching alternately. As such the blade only encounters a single vortex. Had the clearance been slightly smaller, the blade would have passed through the wake in the vortex formation region, which would most likely have resulted in a significantly different characteristics of the interaction, where it would feel the influence of both the detaching and the developing vortex. A smaller clearance would also be likely to increase the likelihood of lock-in phenomena to occur, since the disruption of the vortex formation would be much greater. A further investigation of this parameter is

clearly necessary both with respect to its influence on the blade response and the tower response.

One of the major concerns related to downwind turbines is the high levels of low frequency noise (typically in the range of 20-100 Hz) observed to occur occasionally, which cause considerable nuisance to nearby inhabitants. It is generally agreed that these so-called thumping noises are related to the unsteady interaction between the blades and the unsteady wake behind the tower ([30], [35], and [56]). The large variation in the field measurements indicate that the phenomenon is critically dependent on the exact nature of the interaction.

As is evident from the present results as well as the UAE experiment, the highest fluctuations in the blade loading occurred when the blade experienced a direct encounter with a vortex of positive vorticity (type 2). As it is well-known, large fluctuations in the pressure distribution on the blade will cause high sound pressure levels, which leads to the conclusion that BVI events are the main contributors to the occasional high levels of low frequent noise.

In field tests of downwind turbines it was found that under certain flow conditions, and more generally during night time, the noise would increase. It seems likely that these events took place in conditions where the shedding frequency coincided with the blade passage frequency making BVI events more frequent. The present results indicate that the rotor causes a spanwise correlation of the vortex shedding to take place. This fact would certainly worsen the effect of a BVI event, because the entire blade would be affected causing the gross pressure fluctuations to be stronger. The additional finding that lock-in could take place on a downwind turbine would cause an increase in the likelihood of BVI events to occur, as well as increasing the severity of the interaction due to the increase of the vortex intensity. It is thus hypothesised that the present new finding could play a significant role in the understanding of the acoustic signature of a downwind turbine and should be investigated further both by computational and experimental means.

4.7.6 Computational Cost

In Section 3.6 the parallel performance of the overset solver was evaluated for a multibody moving grid problem on a fairly coarse grid. For that particular problem, it was found that the connectivity routines only occupied approximately 3% of the total processor time, whereas the exchange and interpolation of flow field information occupied approximately 25%.

The present computation was carried out on a much finer grid, which considerably increased the work load for the connectivity routine. Table 4.1 shows the computational time per iteration for the two main overset routines and the total computational time. As it can be seen the overset routines occupy approximately 50% of the total computational time, which is far in excess of what was seen in the spheroid problem. As it can be seen, this is mainly due to the high cost associated with the connectivity routine. The solver is seen to scale very well when increasing the number of processors from 45 to 67. The fact that the Copy3DOG routine appears to exhibit super scaling is purely related to the distribution of blocks on the processors, which cause large imbalances between the nodes, which will vary according to the number of processors used.

Procs	Copy3DOG	% time	ConnectOG	% time	Total
45	30.5	23.5	41.6	32.0	129.8
67	11.5	13.9	23.6	28.6	82.5

Table 4.1: Computational cost (wall-clock time) per iteration of the rotor-tower problem.

Figure 4.47 shows the computational cost of the two overset routines for each processor. The blocking nature of the communication used in the ConnectOG routine is clearly visible, since the computational time does not reflect the number of fringe cells on each node as it is seen for the Copy3DOG routine. In the present problem, all body fitted grids were distributed on the first processors, which are the nodes with the lowest load. The processors holding the Cartesian background block groups are subjected to a much higher work load. The large load imbalance introduced by using overset grids is clearly the main issue that needs to be addressed to make the overset solver more efficient.

The strategy of using Cartesian background grids to resolve the farfield hold many advantages such as being less dissipative so that flow features such as tip and root vortices are maintained for longer. Additionally, changes made to the curvilinear grids can be carried out without affecting the background grids in any way. Parametric studies of different rotor configurations can thus easily be carried out using the same basic mesh layout and only changing the rotor mesh. However, as discussed in Section 3.6, for such a solution strategy to be viable a more efficient solution method dedicated to Cartesian grids must be implemented to reduce the overall computational cost associated with the overset grid method.

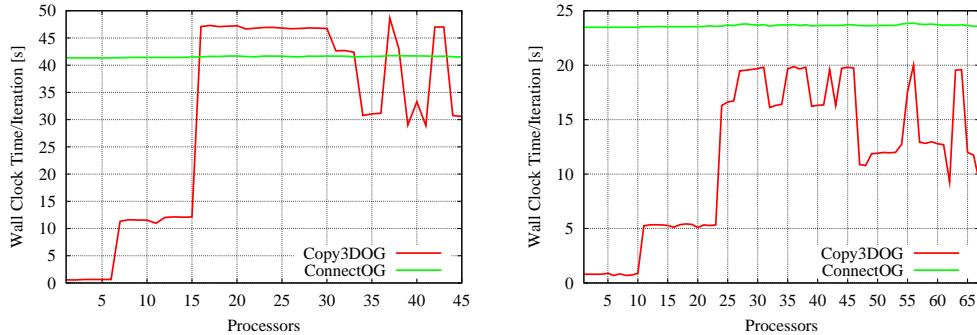


Figure 4.47: Individual processor time per iteration for the ConnectOG and Copy3D routines for the 45 and 67 processors, respectively.

4.8 Conclusions

In this chapter simulations of flow over firstly an isolated rotor, secondly over an isolated tower, and finally over a combined rotor and tower configuration have been presented. For all computations the overset grid version of the EllipSys3D code was used, and it was shown in Section 4.5 to produce results that were in very good agreement with both the non-overset version of the code and experimental results. With the computations on the isolated tower it was demonstrated that the flow downstream of the tower developed into the well-known street of alternately shed vortices and was highly three-dimensional with considerable stretching of the vortex street due to the effects of the ground boundary and free end.

Finally, the computations carried out on the combined rotor and tower configuration showed that the EllipSys3D solver combined with the newly implemented overset grid method is fully capable of capturing the unsteady interaction between the rotor and tower on a downwind turbine. It was shown that the computations were in excellent agreement with the experimental results capturing the key features of the flow that characterise the unsteady interaction. The blade-wake interaction on a downwind turbine is characterised by large fluctuations in the normal and tangential forces of up to approximately 40% of the freestream level. The normal and tangential forces are largely dependent on the local flow angle, and it was found that unsteady and three-dimensional effects play a significant role in the response of the blades. The pressure distribution was during a BWI event characterised by a large reduction in (and in some cases, a complete collapse of) the suction peak on the blade. Interaction with the shed vortices caused significant changes in the

pressure distribution on both pressure and suction side. Consequently, it was found that the quarter chord pitching moment varied significantly in nature according to the type of BVI event with up to 30% changes in the moment acting on the blade.

The rotor exerted a significant influence on the tower shedding, affecting shedding frequency, phasing and spanwise correlation of the wake. It was found that the tower shedding settled into a state of lock-in, which was driven by the blade passage frequency. Since only one wind speed was investigated, it was not possible to confirm whether this interesting phenomenon persisted at other wind speeds. However, these findings do, in the opinion of the author, warrant further investigation of the nature of the tower response on downwind turbines to gain further understanding of the complexities of rotor-tower interaction.

The large degree of correlation of the wake suggests that a different modelling of the turbulence could be necessary. The Reynolds averaged formulation used in the solver is believed to be the root of this lack of break up of the wake. Techniques such as Detached Eddy Simulation or hybrid RANS/LES are likely to be more appropriate for the current type of flow and would produce a flow field with larger variation in the scales of flow structures.

5

CONCLUSIONS

In this thesis the aerodynamics of wind turbines has been studied using the incompressible Reynolds-averaged Navier-Stokes flow solver EllipSys3D [42, 43, 59]. Previously, it was only possible to model the isolated rotor, thus neglecting the effect of the tower. The aim of the present study has therefore been to extend the capabilities of the solver to allow for the modelling of the entire wind turbine, and secondly to show that the solver is capable of simulating the unsteady interaction between the rotor and the tower on a downwind turbine. This required the implementation of a method to handle the relative movement between the rotor and tower. It was found that the overset grid method had the largest flexibility for the problem at hand, since it makes it possible to decompose the solution domain into a number of simpler grids that are allowed to overlap arbitrarily and move independently relative to each other.

Implementation of the overset grid method in an incompressible pressure-based flow solver requires a number of additions to be made, which were described thoroughly in Chapter 2. Although the overset grid method makes it possible to use existing solution algorithms, the fundamental difference between an overset grid approach and a standard multiblock approach lies in the fact that the overset grid approach is not conservative. The coupling of the individual overset grid groups requires the specification of appropriate boundary conditions on the internal overset boundaries. This is achieved through non-conservative interpolation of the velocity field resulting in a Dirichlet condition on the velocities. It must be emphasised that no direct interpolation of the pressure is possible because this would result in an ill-posed problem. As such, the pressure is driven by the fluxes (which are based on the interpolated velocities) effectively resulting in a Neumann condition on the pressure. To satisfy the integral continuity constraint inherent to the incompressible Navier-Stokes equations, a correction of the overset grid boundary fluxes is necessary.

This correction ensures that the solution converges to machine accuracy in all interior cells, with a finite conservation error along the overset boundaries.

An overset grid system typically consists of curvilinear grids fitted around solid bodies, combined with several Cartesian background grids to cover the rest of the solution domain. To increase the flexibility of the method, cells in the background grids are blanked to accomodate the solid bodies and allow these to move freely in the domain. The identification of unwanted cells is carried out using the boundary ray method [50]. For each fringe cell along the overset boundaries, appropriate donor cells in neighbouring grids must be identified. This process is referred to as the grid connectivity process. In EllipSys3D donor cells are located using a combination of an octree based search algorithm and the so-called *stencil jumping* technique [5]. This entire process is carried out in a fully parallelised manner to enable the simulation of large scale moving body problems where the connectivity information must be updated at every time step. To speed up this process the so-called n-th level restart approach is used [3]. All communication is carried out using the Message Passing Interface (MPI) in a non-blocking manner to minimise the latency associated with the communication.

Through a number of validation problems it was in Chapter 3 shown that the solution on overset grids was in good agreement with equivalent non-overset solutions and experimental results. The formal spatial and temporal order of accuracy of the solver was found not to be degraded when using overset grids. The size of the necessary overlap between overset grids was investigated and it was shown that in principle this overlap can be as small as one cell, although a larger overlap will decrease the conservation error associated with the non-conservative specification of boundary conditions.

An evaluation of the parallel performance of the code was also carried out, which showed that the code scales reasonably well on multiple processors. The nature of the overset grid method makes it difficult to achieve good load balancing on multiple processors; indeed it was found that this problem was the main cause for the lack of optimal scaling of the solver. The use of a combination of curvilinear and Cartesian block groups resulted in the processors holding the Cartesian groups being loaded most, while the body-fitted curvilinear grids were less loaded. A more optimal distribution of blocks on the parallel nodes could improve the performance of the overset routine although this could result in a poorer performance of the core solution routines.

Another method for improving the efficiency of the overset solver would be to have a dedicated solver for the Cartesian far-field grids which could take advantage of their simple topology.

It was in Chapter 4 shown that the overset grid version of EllipSys3D is fully capable of simulating the flow over wind turbines. This was demonstrated by firstly computing the flow over the isolated NREL Phase VI rotor. These results were compared to previously published results carried out using the EllipSys3D solver and experimental results from the NREL Phase VI Unsteady Aerodynamics Experiment. Secondly, the flow over an isolated tower corresponding to that used in the NREL experiment was simulated where the nature of the unsteady tower wake was investigated. Finally the flow over the combined rotor-tower configuration was computed, where the rotor, tower (without nacelle) and ground boundary were modelled. The solver was demonstrated to be fully capable of capturing the unsteady interaction between the rotor and the tower and was shown to be in excellent agreement with experimental results. The nature of the tower response is critically dependent on the state of the wake, strongly affected by the blade interaction with the shed vortices from the tower. The normal and tangential forces were largely dependent on the local flow angle which fluctuated the most in the root region during the tower passage. This caused significant fluctuations in the normal and tangential forces of up to 70% and 40% on the inner and outer parts of the blade, respectively, relative to freestream level. On the present turbine geometry it was found that the reduced frequency characterising the blade-wake interaction did not vary along the span of the blade, reducing the differences in phasing and hysteresis effects along the span of the blade. Indeed, the blade responses were not found to vary on account of phasing and hysteresis effects, but rather due to unsteady effects associated with the tower wake. Interaction with the shed vortices causes significant changes in the pressure distribution on both pressure and suction side. Most significantly, the pressure distribution was during a blade-wake interaction event characterised by a large reduction in (and in some cases, a complete collapse of) the suction peak on the blade. Consequently, it was found that the quarter chord pitching moment varied significantly in nature according to the type of blade-vortex interaction event with up to 30% changes in the moment acting on the blade.

The tower response was modified significantly by the presence of the rotor. Two factors were identified as the most important: firstly, the induction of the

rotor caused changes mainly in the axial and tangential velocities upstream of the rotor giving rise to changes in the shedding frequency and phasing inside the rotor disc; secondly, the effect of the blades passing through the tower wake caused an additional significant alteration of the shedding frequency and phasing due to the desintegration of the interacting vortices during blade-vortex encounters. In the present simulation this resulted in the shedding frequency locking in on the blade passage frequency, which resulted in the blade response becoming periodic. Although it was not possible to identify this behaviour from the experimental results, it is believed that this phenomenon should be investigated further.

An further implication of this finding could be a step towards understanding the aero-acoustic phenomena associated with downwind turbines. Indeed, occurrences of lock-in could explain the increase in noise persisting when the turbine operated under certain flow conditions. This argument is supported by the observations that the noise associated with the blade-wake interaction increased during night time, where it is known that the turbulence intensity is lower and that the boundary is stable with less shear - which are all factors that increase the likelihood of lock-in occurring. It is therefore the opinion of the author that future acoustic measurements on downwind turbines should include measurements of the tower forces to estimate the shedding frequencies.

5.1 Outlook

The development of the overset capabilities in EllipSys2D/3D has been ongoing throughout the entire thesis work. Many of the improvements have been sparked by specific obstacles associated with a certain type of problem. As such the code is still in its developmental phase, and not fully suited for release to the EllipSys user group. Although a significant effort has been made to render the overset solver as user-friendly as possible, there are still a number of issues particularly related to grid assembly that require some prior knowledge of the structure and methods in the code which should therefore be improved. At the present stage it is not possible to overlap surface grids since this requires implementation of a method to handle discrete surface representation mismatches on curved surfaces. This addition would be a very useful tool for computations of wind turbine aerodynamics because it would, for example, allow for an easier way of modelling aerodynamic devices on wings such as vortex generators and trailing edge devices. Additionally, as it was also dis-

cussed in the thesis, there are many areas of the code which can be optimised to reduce the computational cost, in particular the connectivity routines and the load balancing strategy. In addition to this, possible solvers dedicated to the Cartesian background grids should also be investigated. The immediate aim within the overset version of the code is, however, to improve its stability and user friendliness, such that it can be used widely for the solution of general problems involving complex geometries and relative grid movement.

The present study of the unsteady interaction between the rotor and the tower was limited to one case of the NREL unsteady aerodynamics experiment. Clearly, it is necessary to investigate the rotor-tower interaction at a wide range of wind speeds both in relation to the characteristics of the blade response and to the dynamics of the tower wake to obtain a more thorough understanding of the interaction.

The present computations were made using a rotor-tower configuration where the nacelle was omitted. A logical step forward would therefore be to include the nacelle in the simulation. A better understanding of the unsteady flow around the nacelle could also help to improve the accuracy of the measurements of wind direction and magnitude made during operation of the turbine.

The relatively small scale of the NREL Phase VI turbine causes in particular the tower Reynolds number to be fairly low. As it was clearly evident from the results, the unsteady response of the blades depends critically on the flow characteristics in the wake of the tower. Here, parameters such as wake formation length, vortex intensity, shedding frequency and wake coherence are important parameters which are likely to differ significantly for higher Reynolds number flows typical for modern wind turbines. Additionally, it is also necessary to investigate the influence of the blade-tower clearance. It is likely that the response of the blades would differ quite significantly for different clearance lengths, and it could be that some optimum distance relating to the vortex formation length exists where the fluctuating forces on the blades reach a minimum.

The effects of the atmospheric boundary layer were not included because the simulations were set up to mimic the flow conditions in the NASA Ames wind tunnel where the ground boundary layer was very thin. The characteristics of both the rotor dynamics and the tower wake are likely to be significantly altered by a velocity shear, which makes inclusion of this effect an important step towards understanding the aerodynamics of wind turbines operating in

field conditions.

The findings relating to the tower response and the possibility of lock-in occurring should in conjunction with the above future tasks also be investigated further. Additionally, the computations made using CFD can be used as input for engineering aero-acoustic models. It would be interesting to investigate whether the lock-in phenomena could indeed be part of the explanation for the high levels of low frequency noise emitted from downwind turbines.

BIBLIOGRAPHY

- [1] A. E. Abbott and I. H. von Doenhoff. *Theory of Wing Sections*. Dover Publications, Inc., New York, 1959.
- [2] E. Atta. Component-adaptive grid interfacing. In *19th Aerospace Sciences Meeting*, St. Louis, MO, January 1981. *AIAA paper 81-0382*.
- [3] E. Barszcz, W. S., and M. R. Dynamic overset grid communication on distributed memory parallel processors. *AIAA paper 93-3311*, 1993.
- [4] A. Bechmann. Large-eddy simulation of atmospheric flow over complex terrain. Technical Report Risø-R-0000, Risø National Laboratory, 2006.
- [5] J. A. Benek, F. C. Dougherty, and P. G. Buning. Chimera: A grid-embedding technique. *AEDC-TR-85-64*, December 1885.
- [6] J. A. Benek, J. L. Steger, and F. C. Dougherty. A flexible grid embedding technique with application to the euler equations. *AIAA paper 83-1944*, 1983.
- [7] F. Bertagnolio. 2D and 3D hole-cutting using the boundary ray method. Personal Communication, 2006.
- [8] F. Bertagnolio, N. N. Sørensen, and J. Johansen. Status for the two-dimensional Navier-Stokes solver EllipSys. Technical Report Risø-R-1282(EN), Risø National Laboratory, 2001.
- [9] R. D. Blevins. *Flow-Induced Vibration*. Van Nostrand Reinhold Co., New York, 1990.
- [10] D. L. Brown, W. D. Henshaw, and D. J. Quinlan. Overture: Object-oriented tools for overset grid applications. In *AIAA Conference on Applied Aerodynamics*, 1999.
- [11] T. M. Burton and J. K. Eaton. Analysis of a fractional-step method on overset grids. *Journal of Computational Physics*, 137:336–364, 2002.
- [12] G. Chesshire and W. D. Henshaw. Composite overlapping meshes for the solution of partial differential equations. *Journal of Computational Physics*, 90:1–64, 1990.
- [13] M. Drela and M. B. Giles. Viscous-inviscid analysis of transonic and low Reynolds number airfoils. *AIAA Journal*, 25(10):1347–1355, 1987.

- [14] E. P. N. Duque, M. D. Burkland, and W. Johnson. Navier-Stokes and comprehensive analysis performance predictions of the NREL Phase VI experiment. *Solar Energy Engineering*, 125:457–467, 2003.
- [15] E. P. N. Duque, C. P. van Dam, and S. C. Hughes. Navier-Stokes simulations of the NREL combined experiment Phase II rotor. *AIAA paper 99-0037*, 1999.
- [16] J. A. Ekaterinaris, N. N. Sørensen, and F. Rasmussen. Numerical investigation of airfoil dynamic stall in simultaneous harmonic oscillatory and translatory motion. In *35th Aerospace Sciences Meeting and Exhibit*, Reno, NV, USA, 1997. *AIAA paper 97-0615*.
- [17] J. H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer, 2002.
- [18] D. Fingersh, D. Simms, M. Hand, D. Jager, J. Cortrell, M. Robinson, S. Schreck, and S. Larwood. Wind tunnel testing of NREL’s unsteady aerodynamics experiment. *AIAA paper 2001-0035*, 2001.
- [19] R. A. M. Galbraith, M. W. Gracey, and E. Leith. Summary of pressure data for thirteen aerofoils on the university of Glasgow’s aerofoil database. Technical Report 9221, University of Glasgow, 1992.
- [20] P. Giguere and M. Selig. Design of a tapered and twisted blade for the nrel combined experiment rotor. Technical Report NREL/SR-500-26173, 1999.
- [21] M. M. Hand, D. A. Simms, L. J. Finersh, D. W. Jager, J. R. Cotrell, S. Schreck, and S. M. Larwood. Unsteady aerodynamics experiment Phase VI: Wind tunnel test configurations and available data campaigns. Technical Report NREL/TP-500-29955, NREL, December 2001.
- [22] W. D. Henshaw. A fourth-order accurate method for the incompressible Navier-Stokes equations on overlapping grids. *Journal of Computational Physics*, 113(1):13–25, 1994.
- [23] W. D. Henshaw. On multigrid for overlapping grids. *SIAM J. Sci. Comput.*, 26(5):1547–1572, 2005. ISSN 1064-8275.
- [24] M. Hinatzu and J. H. Ferziger. Numerical computation of unsteady incompressible flow in complex geometry using a composite multigrid technique. *International Journal for Numerical Methods in Fluids*, 13:971–997, 1991.
- [25] B. Hubbard and H.-C. Chen. A chimera scheme for incompressible viscous flows with application to submarine hydrodynamics. In *25th AIAA Fluid Dynamics Conference*, number AIAA 94-2210, Colorado Springs, CO, June 20-23 1994.
- [26] J. S. Humphreys. On a circular cylinder in a steady wind at transitional Reynolds numbers. *Journal of Fluid Fluid Mechanics*, 9:603–612, 1960.

- [27] R. I. Issa. Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics*, 62:40–65, 1985.
- [28] J. Johansen. Unsteady airfoil flows with application to aeroelastic stability. Ph.D. Thesis Risø-R-1116(EN), Risø National Laboratory, Roskilde, Denmark, 1999.
- [29] J. Johansen, N. N. Sørensen, J. Michelsen, and S. Schreck. Detached-eddy simulation of flow around the NREL Phase-VI rotor. *Wind Energy*, 5(2-3):185–197, 16-19 June 2003.
- [30] N. D. Kelly. Acoustic noise generation by the DOE/NASA MOD-1 wind turbine. In *Proceedings of Workshop on Wind Turbine Dynamics*, Cleveland State University, Cleveland, Ohio, 24-26 February 1981.
- [31] A. M. Larsen. *Risø Central Computer Facility, Mary*. URL http://www.risoe.dk/ita/linuxcluster/mary_uk.htm.
- [32] B. E. Launder and D. B. Spalding. The numerical computation of turbulent flows. *Computational Methods in Applied Mechanics and Engineering*, 3:269–289, 1974.
- [33] A. Le Pape and J. Lecanu. 3D Navier-Stokes computations of a stall-regulated wind turbine. *Wind Energy*, 7:309–324, Oct. 2004.
- [34] B. Leonard. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computational Methods for Applied Engineering*, 19:59–98, 1979.
- [35] S. Ljungren. A preliminary assessment of environmental noise from large wecs, based on experiences from swedish prototypes. Technical Report FFA TN 1884-48, FFA, 1984.
- [36] H. A. Madsen, N. N. Sørensen, and S. Schreck. Yaw aerodynamics analyzed with three codes in comparison with experiment. In *Proceedings of 2003 ASME Wind Energy Symposium*, number AIAA-2003-0518, 2003.
- [37] R. C. Maple and D. M. Belk. Automated set up of blocked, patched, and embedded grids in the beggar flow solver. In N. P. Weatherill, P. R. Eiseleman, J. Häuser, and J. F. Thompson, editors, *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, pages 305–314, Swansea, Wales, 1994. Pineridge Press.
- [38] R. Meakin. On adaptive refinement and overset structured grids. In *Proceedings of the 13th AIAA Computational Fluid Dynamics Conference*, Snowmass, CO, 1997. AIAA Paper 97-1858.
- [39] R. Meakin and A. M. Wissink. Unsteady aerodynamic simulation of static and moving bodies using scalable computers. In *Proceedings of the 14th AIAA Computational Fluid Dynamics Conference*, Norfolk, VA, USA, 1999. AIAA Paper 99-3302.

- [40] R. L. Meakin. A new method for establishing intergrid communication among systems of overset grids. *AIAA paper 91-1586*, June 1991.
- [41] F. R. Menter. Zonal two-equation $k - \omega$ models for aerodynamic flows. *AIAA paper 93-2906*, 1993.
- [42] J. A. Michelsen. Basis3D—a platform for development of multiblock PDE solvers. Technical Report AFM 92-05, Technical University of Denmark, 1992.
- [43] J. A. Michelsen. Block structured multigrid solution of 2D and 3D elliptic PDEs. Technical Report AFM 94-06, Technical University of Denmark, 1994.
- [44] M. S. Milgram. Does a point lie inside a polygon? *J. Comput. Phys.*, 84(1):134–144, 1989.
- [45] R. W. Noack. DiRTlib: A library to add an overset capability to your flow solver. In *17th AIAA Computational Fluid Dynamics Conference*, Toronto, Ontario, Canada, June 6-9 2006. *AIAA paper 2005-5116*.
- [46] R. W. Noack. SUGGAR: A general capability for moving body overset grid assembly. In *17th AIAA Computational Fluid Dynamics Conference*, Toronto, Ontario, Canada, June 6-9 2006. *AIAA paper 2005-5117*.
- [47] S. V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Hemisphere Publishing Corporation, 1980.
- [48] S. V. Patankar and D. B. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer*, 15:1787, 1972.
- [49] N. A. Petersson. An algorithm for assembling overlapping grid systems. *SIAM Journal of Scientific Computing*, 20(6):1995–2022, 1999.
- [50] N. A. Petersson. Hole-cutting for three-dimensional overlapping grids. *SIAM Journal of Scientific Computing*, 21(2):646–665, 1999.
- [51] N. C. Prewitt, D. M. Belk, and W. Shyy. Parallel computing of overset grids for aerodynamic problems with moving grids. *Progress in Aerospace Sciences*, 36:117–172, 2000.
- [52] C. M. Rhie and W. L. Chow. Numerical study of the turbulent flow past an aerofoil with trailing edge separation. *AIAA journal*, 21:1525–1532, 1983.
- [53] S. E. Rogers, D. Kwak, and C. Kiris. Numerical solution of the incompressible Navier-Stokes equations for steady-state and time-dependent problems. *AIAA Journal*, 29(4):603–610, 1991.
- [54] J. G. Schepers et al. Final report of IEA Annex XVIII enhanced field rotor aerodynamic database. Technical Report ECN-C-02-016, ECN, ECN, Petten, The Netherlands, February 2002.

- [55] S. Schreck. The NREL full-scale wind tunnel experiment introduction to the special issue. *Wind Energy*, 5(2-3):77–84, 2002.
- [56] K. P. Sheppard and H. H. Hubbard. Physical characteristics and perception of low frequency noise from wind turbines. *Journal of Noise Control Engineering*, 1991.
- [57] D. M. Somers. Design and experimental results for the S809 airfoil. Technical Report NREL/SR-440-6918, NREL, 1997.
- [58] J. N. Sørensen. VISCWIND: Viscous effects on wind turbine blades. Technical report ET-AFM 9902, Department of Fluid Mechanics, The Technical University of Denmark, 1999.
- [59] N. N. Sørensen. General purpose flow solver applied to flow over hills. Technical Report Risø-R-827(EN), Risoe National Laboratory, 1995.
- [60] N. N. Sørensen. HypGrid2D—a 2-D mesh generator. Technical report, Risø-R-1035(EN), Risoe National Laboratory, 1998.
- [61] N. N. Sørensen, J. A. Michelsen, and S. Schreck. Navier-Stokes predictions of the NREL Phase VI rotor in the NASA Ames 80ft × 120 ft wind tunnel. *Wind Energy*, 5:151–169, 2002.
- [62] P. R. Spalart, W. H. Jou, M. Strelets, and S. R. Almaras. Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach. *Proceedings 1st AFOSR International Conference on DNS/LES*, pages 137–147, August 1997.
- [63] J. L. Steger, F. C. Dougherty, and J. A. Benek. A chimera grid scheme. In K. N. Ghia and U. Ghia, editors, *Advances in Grid Generation*, volume 5, pages 59–69. ASME FED, 1983.
- [64] N. E. Suhs, S. E. Rogers, and W. E. Dietz. Pegasus 5: An automated pre-processor for overset-grid CFD. In *32nd AIAA Fluid Dynamics Conference*, St. Louis, Missouri, USA, 24-26 June 2002.
- [65] H. S. Tang. Study on a grid interface algorithm for solutions of incompressible Navier-Stokes equations. *Computers and Fluids*, 35(10):1372–1383, December 2006.
- [66] H. S. Tang, S. C. Jones, and F. Sotiropoulos. An overset-grid method for 3d unsteady incompressible flows. *Journal Of Computational Physics*, 191(2):567–600, 2003.
- [67] A. Travin, M. Shur, M. Strelets, and P. Spalart. Detached-eddy simulation past a circular cylinder. *Flow, Turbulence and Combustion*, 63:293–313, 1999.
- [68] J. Y. Tu and L. Fuchs. Calculation of flows using three-dimensional overlapping grids and multigrid methods. *International Journal for Numerical Methods in Engineering*, 38:259–282, 1995.

- [69] L. J. Vermeer, J. N. Sørensen, and A. Crespo. Wind turbine wake aerodynamics. *Progress in Aerospace Sciences*, 39:467–510, 2003.
- [70] T. G. Wetzel. *Unsteady flow over a 6:1 prolate spheroid*. PhD thesis, Virginia Polytechnic Institute and State University, Aerospace and Ocean Engineering, 1996.
- [71] D. C. Wilcox. *Turbulence Modeling for CFD*. DCW Industries, Inc., 2 edition, 1994.
- [72] C. H. K. Williamson. Vortex dynamics in the cylinder wake. *Annual Review of Fluid Mechanics*, 28:477–539, 1996.
- [73] A. M. Wissink and R. L. Meakin. On parallel implementation of dynamic overset grid methods. In *SC97: High Performance Networking & Computing*, San Jose, USA, November 15-21 1997.
- [74] J. A. Wright and W. Shyy. A pressure-based composite grid method for the Navier-Stokes equations. *Journal of Computational Physics*, 107: 225–238, 1993.
- [75] G. Xu. *Computational Studies of Horizontal Axis Wind Turbines*. PhD thesis, Georgia Institute of Technology, May 2001.
- [76] G. Xu and L. N. Sankar. Computational study of horizontal axis wind turbines. *AIAA Paper 99-0042*, 1999.
- [77] G. Xu and L. N. Sankar. Effects of transition, turbulence and yaw on the performance of horizontal axis wind turbines. *AIAA Paper 2000-0048*, 2000.
- [78] R. W. Yeo, P. Wood, and A. N. Hrymak. A numerical study of laminar 90-degree bend duct flow with different discretization schemes. *Journal of Fluids Engineering*, 112:563–568, 1991.
- [79] Y. Zang and R. L. Street. A composite multigrid method for calculating unsteady incompressible flows in geometrically complex domains. *International Journal for Numerical Methods in Fluids*, 20:341–361, 1995.
- [80] Y. Zheng and M.-S. Liou. A novel approach of three-dimensional hybrid grid methodology: Part 1. grid generation. *Comput. Methods Appl. Mech. Engrg.*, 192:4147–4171, 2003.