A high order Discontinuous Galerkin - Fourier incompressible 3D Navier-Stokes solver with rotating sliding meshes for simulating cross-flow turbines

by

Esteban Ferrer

Brasenose College

Department of Engineering Science

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity Term 2012

Acknowledgements

First of all, I would like thank my supervisors Dr. Richard H.J. Willden and Prof. Guy T. Houlsby for providing me with freedom and support to pursue my ideas in this thesis. A special acknowledgement to Richard for giving me the opportunity to develop this solver and providing useful guidance in my first steps. I am very grateful to the John Fell Oxford University Press fund for the financial support.

I would like to thank my colleagues in the Engineering department for their comments and the fruitful discussions that we shared throughout these years: Dr. Scott Draper, Dr. Ross McAdam, Dr. Claudio Consul, Dr. Takafumi Nishino, Clarissa Belloni, Sena Serhadlioglu, Christopher Vogel and William Hunter. Particularly, I would like to thank Dr. Scott Draper for introducing me to the Discontinuous Galerkin method and for our very interesting discussions on numerical techniques. I also acknowledge the help of Dr. Ross McAdam on the analysis of cross-flow turbine physical phenomena.
During these years, I had the opportunity to share ideas with the following academics, whose help and time is kindly appreciated: Prof. Endre Süli, Dr. Andrew Wathen, Prof. Ramon Codina, Prof. Spencer Sherwin and Prof. Hugh Blackburn.

In addition, I would like to acknowledge Prof. Spencer Sherwin for making available the h/p Spectral code *Nektar* and Prof. Hugh Blackburn for allowing me to use the h/p Spectral solver *Semtex* and the Spectral Vanishing Viscosity extension. Both solvers have proven to be very useful for verification of the developed DG-Fourier solver and have allowed some of the comparisons reported in this thesis.

Finally, I would like to thank my parents for all their moral support and my friends in Oxford for being there for me, through good and not so good times.

Abstract

A high order Discontinuous Galerkin - Fourier incompressible 3D Navier-Stokes solver with rotating sliding meshes for simulating cross-flow turbines

by Esteban Ferrer - Brasenose College, Oxford

A thesis submitted for the degree of Doctor of Philosophy

Trinity Term 2012

This thesis details the development, verification and validation of an unsteady unstructured high order (≥ 3) h/p Discontinuous Galerkin - Fourier solver for the incompressible Navier-Stokes equations on static and rotating meshes in two and three dimensions. This general purpose solver is used to provide insight into cross-flow (wind or tidal) turbine physical phenomena.

Simulation of this type of turbine for renewable energy generation needs to account for the rotational motion of the blades with respect to the fixed environment. This rotational motion implies azimuthal changes in blade aero/hydro-dynamics that result in complex flow phenomena such as stalled flows, vortex shedding and blade-vortex interactions. Simulation of these flow features necessitates the use of a high order code exhibiting low numerical errors. This thesis presents the development of such a high order solver, which has been conceived and implemented from scratch by the author during his doctoral work.

To account for the relative mesh motion, the incompressible Navier-Stokes equations are written in arbitrary Lagrangian-Eulerian form and a non-conformal Discontinuous Galerkin (DG) formulation (i.e. Symmetric Interior Penalty Galerkin) is used for spatial discretisation. The DG method, together with a novel sliding mesh technique, allows direct linking of rotating and static meshes through the numerical fluxes. This technique shows spectral accuracy and no degradation of temporal convergence rates if rotational motion is applied to a region of the mesh. In addition, analytical mappings are introduced to account for curved external boundaries representing circular shapes and NACA foils.

To simulate 3D flows, the 2D DG solver is parallelised and extended using Fourier series. This extension allows for laminar and turbulent regimes to be simulated through Direct Numerical Simulation and Large Eddy Simulation (LES) type approaches. Two LES methodologies are proposed.

Various 2D and 3D cases are presented for laminar and turbulent regimes. Among others, solutions for: Stokes flows, the Taylor vortex problem, flows around square and circular cylinders, flows around static and rotating NACA foils and flows through rotating cross-flow turbines, are presented.

Contents

Chapter 1

Introduction

1.1 Motivation and scope

Problems where the forces on rotating or oscillating bodies in a fluid are to be predicted are common in engineering applications and result in fluid-structure interaction situations. Examples are flows around isolated rotating bodies and foils, turbomachinery applications, insect flight aerodynamics, unmanned air vehicles and, more recently, flows through renewable energy devices, e.g. wind and tidal turbines.

These flows are characterised by long wake structures, vortex shedding and stalled regimes associated with flow unsteadiness. To simulate accurately wakes and vortex structures and their evolution, high order ($order \geq 3$) numerical methods (i.e. h/p conformal Spectral and h/p non-conformal Discontinuous Galerkin methods) are preferred, since dissipation and dispersion errors are reduced [101] when compared to low order methods (typically ≤ 3). Furthermore, for smooth problems, the exponential decay of the error with polynomial enrichment (p-refinement) in high order methods as opposed to the fixed decay rate characteristic of low order methods (i.e. h-refinement only) make high order methods particularly attractive to obtain accurate solutions for flows where viscosity limits discontinuities (i.e. elliptic type equations) as in the incompressible Navier-Stokes (NS) equations.

A particularly challenging problem is presented by cross-flow wind and tidal turbines for power generation (also known as vertical-axis, H-rotors or Darrieus turbines). These types of turbine consist of foil shaped blades that generate lift forces so as to rotate a shaft to which the blades are connected. Therefore azimuthal changes in blade aerodynamics (or hydrodynamics)[1] are common, resulting in complex flow phenomena such as stalled flows, vortex shedding and blade-vortex interactions. Such flow features, together with the requirement to preserve wake structures, necessitate the use of a high order code exhibiting low numerical errors.

The aim of this thesis is to present the development, verification and validation of a novel, highly accurate, numerical tool to simulate flows around rotating geometries. Particularly, the technique described can be applied to simulate wind and tidal cross-flow turbines and provide an understanding of their flow physics.

1.2 Thesis overview

This thesis presents the development of a general purpose high order 3D Discontinuous Galerkin- Fourier (DG-Fourier) solver for the incompressible Navier-Stokes equations with the following capabilities:

- Flow is described using primitive variables (velocity and pressure),
- A second order time discretisation,
- High order (≥ 3) spatial solutions are obtained on triangular and quadrilateral elements,
- 2D high order curved boundary conditions,
- A sliding mesh capability that enables high order solutions of rotating geometries,
- 2D and 3D versions, the latter being an extension of the 2D-DG using Fourier series,

[1] Airfoil aerodynamics and hydrofoil hydrodynamics are equivalent nomenclatures for foils operating in air or water environments. Since this thesis encompasses both wind and tidal turbine applications, from this point onwards, "foils" will denote either "airfoils" or "hydrofoils". In addition, the term "aerodynamic" can always be replaced by "hydrodynamic" in this thesis.

- Laminar and turbulent regimes can be simulated by means of Direct Numerical Simulation (DNS) or Large Eddy Simulation (LES) techniques,
- Two LES type techniques to account for turbulent flows,
- Parallelised for distributed memory clusters using a Message Passing Interface (MPI) paradigm.

Verification and validation cases with particular emphasis on bluff body flows, foil aerodynamics and cross-flow turbine physics are included under static and rotating conditions.

The structure of this thesis follows the developments of the various features with increasing complexity.

The present introductory chapter details preliminary work undertaken, in order to select the most appropriate method to simulate cross-flow turbines. In particular the main physical flow phenomena occurring in cross-flow turbines are introduced, followed by a brief review on the available numerical methods to solve partial differential equations. By comparing requirements and method capabilities, it was possible to select the most appropriate numerical technique: the Discontinuous Galerkin method.

Chapter 2 presents the solver essentials; temporal and spatial discretisations are introduced for straight edged triangular elements and non-moving meshes. Various verification cases allow the exploration of convergence rates and stability bounds of the method. The chapter finishes with the validation study of the flow around a square cylinder.

In chapter 3, curved edge elements and a novel sliding mesh technique are introduced to account for rotating geometries. In addition, a novel implementation of the no-slip boundary condition is included for curved edges; circular arcs and NACA 4-digit airfoils. Verification and error estimates are given for problems with exact solutions. Additionally, verification cases include a rotating square cylinders and static and rotating NACA foils.

Chapter 4 introduces the Fourier extension of the DG solver described in chapters 2 and 3. This extension allows the resolution of three dimensional flows when homogeneous geometries are considered in the Fourier dimension. To alleviate the increased computa-

tional cost associated with computing three dimensional solutions of the incompressible Navier-Stokes equations, a parallelisation strategy based on the Message Passing Interface (MPI) paradigm is described and implemented. In addition, 3D simulations of square and circular cylinders are included for validation purposes. The chapter concludes with the 3D simulation of the flow over a circular cylinder, which is shadowed by a pitching NACA0012, to illustrate the capability for simulating 3D flows where static and rotating geometries coexist in one single domain.

Chapter 5 describes two Large Eddy Simulation strategies (i.e. a Smagorinsky-DG and an Implicit Large Eddy Simulation - DG / Spectral Vanishing Viscosity - Fourier: ILES-DG SVV-Fourier) to account for turbulent flows at a reduced computational cost. Validation cases and comparisons with other numerical solvers are included for a circular cylinder at $\mathrm{Re} = 3900$ and NACA foils at $\mathrm{Re} = 10000$.

In chapter 6, the developed solver is used to simulate 2D and 3D cross-flow turbine flows under laminar flow regimes and analysis of various physical phenomena are provided. Namely, blade-wake interaction, flow blockage (i.e. ducted devices), Reynolds number and tip speed ratio effects are explored.

Finally, in chapter 7, conclusions and future research lines are outlined.

1.3 Novelty

The Discontinuous Galerkin Finite Element Method (DG-FEM) can be seen as an extension of h/p Spectral methods where the continuity requirement across element boundaries is relaxed or alternatively as a high order Finite Volume method with compact stencil. Contrary to conformal finite elements or h/p Spectral methods, DG methods are locally conservative by construction. As in spectral methods, high-order polynomials can be used within each element enabling exponential convergence, whilst discontinuities in the solution are permitted at element interfaces. It is this last property that allows for the implementation of the sliding mesh technique developed by the author, which is described

and detailed in chapter 3.

To the author's knowledge, this solver is the first high order ($order \geq 3$) DG solver with sliding meshes for the solution of the incompressible Navier-Stokes equations which does not require *mortar type* techniques [101]. The developed solver provides high order solutions, no degradation of temporal convergence rates and spectral convergence when solving the incompressible Navier-Stokes equations on meshes where fixed and rotating elements coexist.

This general purpose unstructured solver has been extended using Fourier series to encompass three dimensional flow features. Laminar and turbulent regimes can be simulated by means of Direct Numerical Simulation (DNS) or Large Eddy Simulation (LES) closure models. Two novel approaches to obtain numerical solutions at moderate Reynolds numbers are proposed.

Overall and to the authors knowledge, this thesis presents the first three dimensional high order DG-Fourier solver that is able to compute laminar and turbulent flows around rotating geometries using sliding meshes, and in particular around cross-flow turbines.

1.4 List of publications

The following publications have been issued during the three years and seven months for the completion of this thesis:

1.4.1 Journal publications

- **E Ferrer** and RHJ Willden, "A high order Discontinuous Galerkin - Fourier incompressible 3D Navier-Stokes solver with rotating sliding meshes", Accepted for publication in the Journal of Computational Physics, submitted July 2011.

- **E Ferrer** and RHJ Willden, "A high order Discontinuous Galerkin Finite Element solver for the incompressible Navier-Stokes Equations", Computers & Fluids, Vol 46 Issue 1, p224-230, 2010.

1.4.2 Conference publications

- **E Ferrer** and RHJ Willden, "Development and validation of a high order numerical solver for cross-flow turbine hydrodynamics", In Proceedings 9th European Wave and Tidal Energy Conference (EWTEC), September 2011, Southampton, UK.

- **E Ferrer** and RHJ Willden, "Development of a high order Discontinuous Galerkin Finite Element Solver", In Proceedings 5th European Conference on Computational Fluid Dynamics (ECCOMAS), June 2010, Lisbon, Portugal.

- CA Consul, RHJ Willden, **E Ferrer** and MD McCulloch, "Influence of Solidity on the Performance of a cross-flow turbine", In Proceedings 8th European Wave and Tidal Energy Conference (EWTEC), September 2009, Uppsala, Sweden.

1.5 Chapter outline

This chapter presents the motivation and driving forces for the numerical developments detailed in this thesis. In particular sections 1.6 and 1.7, introduce the necessity for renewable energy and the potential of cross-flow turbines (CFT) for wind and tidal power generation. Section 1.8 details and characterises CFT flows in terms of time, length and velocity scales for the different phenomena encountered when simulating this type of device. Section 1.9 addresses the necessity of accuracy when computing CFT flows. Different techniques are reviewed in section 1.10, ranging from analytical models to full Navier-Stokes calculations, that have historically been used to predict CFT performance. This section is followed by a brief description of the different techniques to solve numerically Partial Differential Equations (PDE's) in section 1.11. Comparing the requirements to compute CFT flows to the capabilities of the numerical techniques, the Discontinuous Galerkin method is identified as the most appropriate to simulate these type of flows in section 1.12. The chapter finishes with an historical review of Discontinuous Galerkin methods in section 1.13, and with a detailed overview of Interior Penalty formulations. In addition, a brief summary of previous published work on the computation of turbulent flows using DG methods is included.

1.6 Renewable Energy

The effect of human industrialisation on climate change has been accepted as a reality in the last decade. The International Energy Agency (IEA) concludes in *The World Energy Outlook 2008* [92] that the current energy consumption and production is "*patently unsustainable environmentally, economically, and socially*". Consequently, social concern is driving major changes in local policies which aim to act on a global scale. The Kyoto Protocol, agreed in December 1997, is the first global agreement to limit greenhouse gas concentration in the atmosphere [140] and was ratified in January 2009 by 183 parties [133]. The Kyoto agreement expires in 2012 and, although there has not been yet commitment for a new protocol, recent negotiations (Copenhagen 2009 and Durban 2011) seem to have brought hope that a new compromise can be reached.

To achieve the agreed objectives in Kyoto, different policies have been put in place by different countries. The reduction of energy demand is one of the primary aims (e.g. house insulation). However, it is also necessary to produce cleaner energy. Worldwide economic inequalities and different levels of industrialisation and development are major factors in determining the different approaches that local governments are taking, to fulfill the requirements of the Kyoto protocol. However, most governments are relying on renewable energies, to a greater or lesser extent, to progressively replace old and polluting sources of energy such as petrol or coal.

The UK has committed to a 60% reduction in greenhouse gases by 2060 and to 15% renewable energy production by 2020 [46]. This 2020 target can only be reached by the expanded use and combination of different renewable technologies, since it is imperative for the UK to maintain reliability of energy supply [46]. It is thus expected that both wind and tidal turbines play an important role in the near future.

1.7 Cross-flow turbines

One of the main objective of this thesis is to develop the most appropriate numerical technique for the simulation of cross-flow turbine (CFT) flows. This section presents an introduction of cross-flow turbines and their characterisation in terms of fluid and flow regimes for wind and tidal applications. Cross-flow turbines are sometimes referred to as vertical-axis, H-rotors or Darrieus type turbines, however the term cross-flow turbine is preferred since the absolute turbine position is omitted and the relative flow-axis geometry is emphasised through this terminology.

Fig. 1.1 introduces typical cross-flow turbines. A vertical axis wind turbine [188] is shown in Fig. 1.1.a and the prototype of a cross-flow turbine for tidal application being developed at the University of Oxford [119] is shown in Fig. 1.1.b. To date, this type

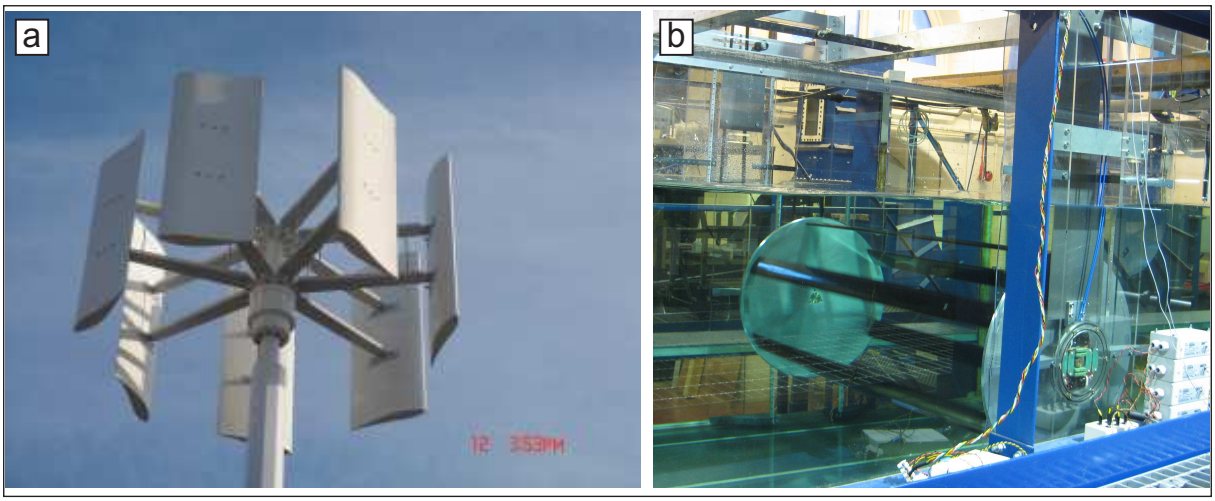

Figure 1.1: Cross-flow turbines (a) wind turbine [188] and (b) tidal turbine with permission of Kepler Energy [183], [119].

of turbine configuration has had limited use within the wind energy sector, where the three bladed axial flow turbine has been widely adopted. However, it is thought that CFT configurations can be advantageous for new emergent markets as offshore wind and tidal energy production and also for installation in urban environments. A brief review of some of the arguments in favour and against this type of configuration follows. A more

complete discussion can be found in [49].

On the one hand, the main drawback is that cross flow turbines are generally less efficient than axial flow turbines since the downstream half of the turbine produces less torque due to the shadowing from the upstream blades. Furthermore, since for part of the cycle the blade moves parallel to the flow, the lift force powering the blades, being proportional to the incident flow speed squared, is reduced.

On the other hand, this type of device has the advantage of not requiring a specific orientation relative to the flow, since it can work independently of the stream direction and hence some researchers argue that this technology is more suitable for the offshore environment (offshore wind and tidal) as it would minimise maintenance costs through reduced control systems (e.g. yaw mechanism) [65], [66]. Furthermore, due to its geometrical simplicity, the generator can be located far from the rotating blades, reducing installation complexity and maintenance cost.

Both technologies share geometrical similarities and flow properties as introduced in the next section.

1.8 Physical characterisation of cross-flow turbine flows

Both wind and tidal CFT technologies operate at relatively low flow speed, when compared to the speed of sound in their respective media, minimising the effects of fluid compressibility. Typical CFT flows can be characterised using the incompressible Navier-Stokes (NS) equations. This system of non-linear Partial Differential Equations (PDE's) can be written in its convective form as the set:

$$\text{Continuity :} \qquad \nabla.\mathbf{u} = 0, \tag{1.8.1}$$

$$\text{Momentum :} \qquad \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}.\nabla)\mathbf{u} = -\frac{1}{\varrho}\nabla p + \nu\nabla^2\mathbf{u}, \tag{1.8.2}$$

where $\mathbf{u} = (u, v, w)^T$ is the vector of the velocity components in three dimensions, p represents the static pressure, ϱ is the density and $\nu = \frac{\mu}{\varrho}$ is the kinematic viscosity, with μ the dynamic viscosity.

The previously defined NS equations can be written in non-dimensional form using the following non-dimensional variables:

$$\mathbf{u}^* = \frac{\mathbf{u}}{U}, \quad p^* = \frac{p}{\varrho U^2}, \quad t^* = t\frac{U}{L}, \quad \nabla^* = L\nabla, \tag{1.8.3}$$

where, in addition to the defined variables, the following scalars are introduced : U as the characteristic flow velocity (e.g. the freestream speed) and L as the characteristic flow length. Substitution of the non-dimensional quantities into Eq. 1.8.2, leads to a non-dimensional momentum equation:

$$\frac{\partial \mathbf{u}^*}{\partial t^*} + (\mathbf{u}^*.\nabla^*)\mathbf{u}^* = -\nabla^* p^* + \frac{1}{\mathrm{Re}}\nabla^{*2}\mathbf{u}^*, \tag{1.8.4}$$

where $\mathrm{Re} = LU/\nu$ is the Reynolds number.

The Reynolds number represents the ratio of inertial forces to viscous forces. For high Reynolds numbers, inertial forces dominate over viscous forces and the flow is called convection dominated. However, some caution is needed in this particular case since, even for high Reynolds numbers flows, viscous forces become important near walls due to the presence of a boundary layer (large normal gradients of streamwise velocity). For high Reynolds numbers, viscous damping is unable to suppress non-linear instabilities, leading to a turbulent regime (see section 1.8.2.2).

The main purpose of this section is to estimate the Reynolds numbers to characterise the flows for wind and tidal turbines. Table 1.1 shows characteristic fluid properties for air (at 15 degrees Celsius) and sea water (at zero degrees Celsius and moderate salinity). The table also includes typical flow speeds at wind and tidal sites. As can be seen, both fluids are similar in terms of kinematic viscosity. To estimate the Reynolds numbers, let

	Units	Air	Sea Water
Fluid property			
Density (ϱ)	kg/m^3	~ 1.225	~ 1027
Dynamic viscosity (μ)	$Pa.s$	$\sim 1.78 \times 10^{-5}$	$\sim 1.88 \times 10^{-3}$
Kinematic viscosity (ν)	m^2/s	$\sim 1.45 \times 10^{-5}$	$\sim 1.83 \times 10^{-6}$
Flow property			
Flow speed	m/s	~ 12	~ 1.5

Table 1.1: Air and sea water fluid and flow properties

us consider a small cross-flow turbine of radius $R = 5$ m (e.g. a wind turbine for urban environment and a typical tidal turbine), then the Reynolds number Re_D based on the turbine diameter D results in ${\mathrm{Re}_D}^{air} \approx {\mathrm{Re}_D}^{water} \approx 4 \times 10^6$, showing that from a flow dynamic point of view both technologies operate in similar regimes.

Remark: *When tidal turbine flows are considered, the Froude number* $\mathrm{Fr} = U/\sqrt{gL}$ *may be of importance. However for mildly blocked configurations (i.e. low ratio of turbine diameter to shallow water depth) this parameter can be shown to have a relatively low importance [45], [119] and will not be considered in this thesis.*

1.8.1 Disparity of flow scales and their simulation

This section describes the range of flow scales expected when simulating cross-flow turbine flows in relation to the resolution accuracy required for its simulation. Fig. 1.3 presents a schematic illustration of the three physical spatial scales expected on CFT flows. Namely, the foil scaling includes near wall scales and boundary layers, the vortex scaling considers vortex shedding and blade vortex interaction effects and finally the wake scaling includes shadowing effects due to the turbine blockage and the wake structure related to the rotating blades. In this section, estimates for the length, time and velocity scales for the different phenomena, are introduced.

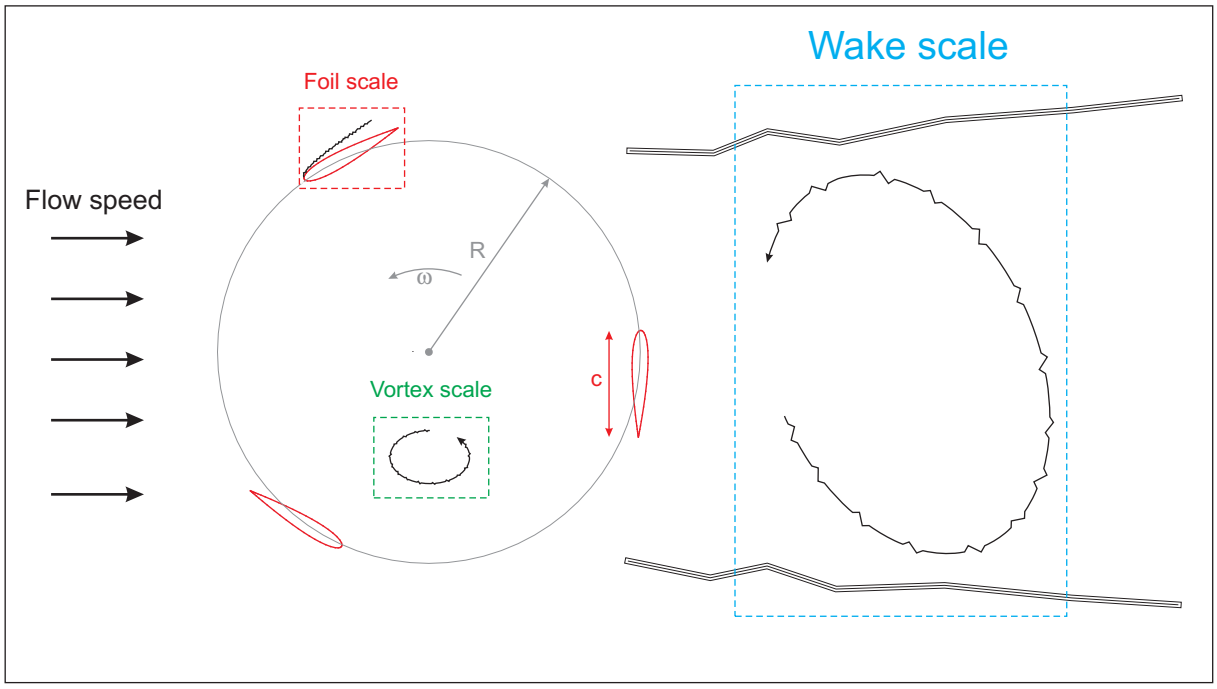

Figure 1.2: Flow scales in cross-flow turbine flows.

1.8.1.1 The foil scaling

In the presence of a non-slip condition at the wall surface, the flow is forced to slow down to have zero velocity at the wall. This creates the so called boundary layer (defined by Prandtl in 1904) that will develop on turbine blades. The logarithmic profile of tangential velocities developed at the wall (if attached flow is considered) has to be properly modelled in a numerical simulation, since it directly influences the aerodynamic forces on the blades. Typically, the foil boundary layer has a laminar portion which will undergo transition to turbulence some distance downstream (assuming a high enough Reynolds number); this distance being influenced by, among other factors, the pressure gradient on the surface, the wall surface roughness and the free stream turbulence outside the boundary layer. Turbulent boundary layers are thicker and carry more momentum and thus separate later than laminar boundary layers. Boundary layers are a vast topic of research and only the principal characteristics have been outlined here. For a detailed explanation the reader is referred to the monograph by Schlichting [151].

It is possible to estimate CFT's length scales for the boundary layer using the Blasius approximation [151] for the flow over a flat plate with zero pressure gradient. First, let us consider the rotational speed of the blades $U_{rot} = \omega R$ and the tip speed ratio $\lambda = U_{rot}/U$, where U represents the free stream velocity. For simplicity, the velocity that the foil experiences can be estimated as $U_{foil} = \sqrt{U_{rot}^2 + U^2} \approx U_{rot} = \lambda U$, which is a valid assumption for $\lambda >> 1$ (i.e. ignoring the free stream velocity U, a more precise definition and analysis is given in chapter 7). Then, Blasius' approximation provides a boundary layer of thickness: $\delta \approx x/(\frac{xU_{foil}}{\nu})^{\gamma} = x/(\frac{x\lambda U}{\nu})^{\gamma}$, where x is the the distance from the leading edge and the exponent γ relates to the boundary layer regime: $\gamma = 1/2$ for a laminar regime and $\gamma = 1/5$ for a turbulent boundary layer. By setting $x = c$, it is possible to obtain a characteristic length scale for the foil $\ell_{foil} = \frac{c}{(\mathrm{Re}_{foil})^{\gamma}}$, where $\mathrm{Re}_{foil} = \frac{cU_{foil}}{\nu}$ is the Reynolds number based on the foil chord. An estimate for the time scale can be readily found: $\tau_{foil} = \frac{\ell_{foil}}{U_{foil}} = \frac{c}{U\lambda(\mathrm{Re}_{foil})^{\gamma}}$. A summary of these scales is given in Table 1.2.

Length scale ℓ_{foil}	**Time scale** τ_{foil}	**Velocity scale** U_{foil}	**Reynolds number** Re_{foil}
$\frac{c}{(\mathrm{Re}_{foil})^{\gamma}}$	$\frac{c}{U\lambda(\mathrm{Re}_{foil})^{\gamma}}$	λU	$\frac{c\lambda U}{\nu}$

Table 1.2: Foil scales

1.8.1.2 The vortex scaling and blade-vortex interaction

For cross-flow turbines, a vortex created by the upstream half of the turbine may be convected downstream, interacting with the rear half of the turbine. Hence, vortex shedding and vortex impingement may play an important role in turbine performance. Numerical accuracy can significantly influence this phenomenon since, vortices may diffuse and lose their structure, changing the forces seen by the rear passing blade. In numerical simulation, high order (e.g. low numerical dispersion and dissipation) codes are preferred to capture accurately vortex evolution and blade vortex interaction phenomena.

An estimate for the scales corresponding to the vortex convection and the blade-vortex interaction phenomenon is provided. Firstly, let us assume a two dimensional vortex with area $A \approx \frac{\pi}{4}c^2$ (the circular area covered by the two dimensional vortex), where c is the characteristic length (the foil chord and the diameter of the vortex) and with characteristic turn over time $\tau_{vortex} \approx \frac{2\pi}{\omega_z}$, where ω_z represents the vorticity magnitude. By approximating the vorticity as a function of the foil circulation and thus of the lift generated by the foil (Kutta-Joukowski theorem [4]), it is possible to write $\omega_z \approx \frac{\Gamma}{A}$, with $\Gamma = \frac{L}{\varrho U_{foil}}$ representing the circulation and L the foil lift force. Since $L = \frac{1}{2}\varrho U_{foil}^2 c C_L$, with C_L the foil lift coefficient, the characteristic vortex time can be expressed as: $\tau_{vortex} \approx \frac{\pi^2 c}{(\lambda U C_L)}$. Finally the characteristic velocity of the vortex can be estimated as $U_{vortex} = \frac{\ell_{vortex}}{\tau_{vortex}} \approx \frac{\lambda U C_L}{\pi^2}$.

To estimate the lengths and times of the blade-vortex (BV) interaction phenomenon, let us assume a vortex of diameter c that convects through the turbine at a velocity $U_{axial} = \beta U$, where $\beta < 1$ represents the velocity deficit through the turbine[2] and U_{axial} is the mean streamwise velocity component within the turbine. The time for a vortex to convect from the front half to the rear half (i.e. to travel the distance $D = 2R$) can be estimated as $\tau_{BV} = \frac{\ell_{BV}}{U_{axial}} = \frac{D}{\beta U}$. Table 1.3 summarises these scales. A more detailed study of this phenomenon can be found in chapter 6.

Length scale ℓ_{vortex}	**Time scale** τ_{vortex}	**Velocity scale** U_{vortex}	**Reynolds number** Re_{vortex}
c	$\frac{\pi^2 c}{(\lambda U C_L)}$	$\frac{\lambda U C_L}{\pi^2}$	$\frac{c\lambda U C_L}{\pi^2 \nu}$
ℓ_{BV}	τ_{BV}	U_{BV}	Re_{BV}
D	$\frac{D}{\beta U}$	βU	$\frac{D\beta U}{\nu}$

Table 1.3: Vortex scales

[2] $\beta < 1$ for unblocked cases but may exceed 1 if blockage leads to accelerated flow through the turbine.

1.8.1.3 The wake scaling

Cross-flow turbine wake development and dissipation are of major importance since the performance of downstream turbines may be influenced by the structure and strength of upstream generated structures. Furthermore, the wake development modifies the incoming flow speed and angle of attack, varying the performance and efficiency of the turbine and thus its correct prediction is important. Unconfined wakes have been studied for a long time in the context of wind turbines. For example the paper by Vermeer et al. [171] is an extensive review of experimental and numerical results for axial wind turbine wakes. Although various empirical models exist to predict the extension of the wake behind an operating wind turbine, and many numerical codes have been used to describe the wake structure, the study concludes that more experimental data is needed and that wake characteristics are still not well understood. However, it seems clear that the wake can be subdivided into a near wake where coherent structures develop, and a far wake, where mixing generates an almost uniform velocity deficit.

Let us estimate the length and time scales for the near wake assuming a wake of diameter $D = 2R$, and a velocity in the wake region as $U_{wake} = \beta\beta' U$, where in addition to the induction factor β introduced in the previous section, $\beta' < 1$ is defined as the velocity deficit due to the rear half blade passage. The characteristic scales are detailed in Table 1.4.

Length scale ℓ_{wake}	**Time scale** τ_{wake}	**Velocity scale** U_{wake}	**Reynolds number** Re_{wake}
D	$\frac{D}{\beta\beta' U}$	$\beta\beta' U$	$\frac{D\beta\beta' U}{\nu}$

Table 1.4: Wake scales

1.8.1.4 Scale comparison and numerical resolution

It is possible to compare the different length and time scales obtained for the different phenomena (tables 1.2, 1.3 and 1.4) to estimate the zones that require more spatial and

temporal resolution for a numerical simulation. The most stringent length and time scales are clearly the foil scale since both decrease as the Reynolds number Re_{foil} increases. The vortex time scale is the smallest after the foil time scale but of the same order of magnitude as the blade-vortex interaction scale.

When comparing the time scales for the blade-vortex interaction and the near wake resolution, one can see that $\ell_{wake} \approx \ell_{BV}$ and $\tau_{wake} \approx \frac{\tau_{BV}}{\beta'}$, which implies that for $\beta' < 1$ the blade-vortex interaction phenomenon requires smaller temporal resolution than the wake resolution.

In addition, one may examine the factors influencing the separation of scales using the ratios of the smallest to largest scales. Let us define the ratio of length scales as $r_\ell = \frac{\ell_{foil}}{\ell_{wake}}$. Substitution of the appropriate length scales (from tables 1.2 and 1.4) lead to $r_\ell = \frac{c}{D(\mathrm{Re}_{foil})^\gamma}$. This expression can be further simplified by using the definition of the turbine solidity $\sigma/N = 2c/D$, where N is the number of turbine blades (see chapter 6 for further details). The final expression for the ratio of length scales reads: $r_\ell = \frac{\sigma}{2N(\mathrm{Re}_{foil})^\gamma}$. Examination of the last expression shows that the separation of these length scales becomes larger (i.e. smaller ratio) if the number of blades (N) and the Reynolds number (Re_{foil}) increase or the solidity (σ) decreases.
Similarly, a ratio for the time scales can be derived leading to $r_\tau = \frac{\tau_{foil}}{\tau_{wake}} = \frac{\sigma\beta\beta'}{2N\lambda(\mathrm{Re}_{foil})^\gamma}$. In this case, the time scale separation becomes larger if the number of blades (N), the Reynolds number (Re_{foil}) and the tip speed ratio (λ) increase, or the solidity (σ) and induction factors ($\beta\beta'$) decrease. It may be noted, from the derived expressions, that the time scale separation is more restrictive than the length scale separation since: $r_\tau = \frac{\beta\beta'}{\lambda} r_\ell$ with $\beta\beta' < 1$ and $\lambda >> 1$.

To conclude this section, a quantitative estimation of the various scales (tables 1.2, 1.3 and 1.4) is provided in table 1.5 for a typical CFT using the air and water environmental conditions introduced in section 1.8. Namely, let us consider a cross-flow turbine of radius $R = 5$ m consisting of airfoils of chord $c = 1$ m (i.e. solidity $\sigma/N = 0.2$). To evaluate the

length, time and velocity scales, it is necessary to approximate the following parameters: $\gamma = 1/5$ for a turbulent boundary layer, the tip speed ratio $\lambda \approx 3$, the lift coefficient $C_L \approx 1$ (e.g. maximum C_L for a NACA0015 at Re $= 1 \times 10^6$ [1]) and the induction factors $\beta \approx \beta' \approx 0.5$. In addition, calculations are performed for two flow speeds $U = 12$ and 1.5 m/s corresponding to air and water environments respectively (i.e. wind and tidal turbines). Note that the two media require different values for the kinematic viscosity: $\nu = 1.45 \times 10^{-5}$ and 1.83×10^{-6} m^2/s for air and salted water, respectively. Inspection of table 1.5 reveals that smaller time scales are present when considering wind turbines over tidal turbines. However, it can be noted that the ratios of time scales (i.e. r_τ) are similar for both types of turbines.

	Length scale $\ell\ [m]$	**Time scale** $\tau\ [s]$	**Velocity scale** $U\ [m/s]$	**Reynolds number** Re
Wind turbine				
Foil	0.053	0.001	36.000	2.5×10^6
Vortex	1.000	0.274	3.648	2.5×10^5
BV	10.000	1.667	6.000	4.1×10^6
Wake	10.000	3.333	3.000	2.1×10^6
Tidal turbine				
Foil	0.053	0.012	4.500	2.5×10^6
Vortex	1.000	2.193	0.456	2.5×10^5
BV	10.000	13.333	0.750	4.1×10^6
Wake	10.000	26.667	0.375	2.0×10^6

Table 1.5: Evaluation of CFT scales.

The results from this simplified analysis confirm the intuitive reasoning on scale separation. However, it appears useful as it outlines the wide range of length and time scales that need to be accurately simulated to capture correctly the complex flow physics involved in cross-flow turbine simulations. In addition, the analysis allows for quantification of the relative importance of the various flow phenomena.

1.8.2 Additional scales

1.8.2.1 Three dimensional scales

Truly two dimensional flows are rare in nature. Very low Reynolds number flows or flows where geometric confinement limits three dimensionality (e.g. shallow water flows) are examples of two dimensional flows. However, when no geometric confinement is considered and the Reynolds number increases such as to trigger three dimensional modes (see chapter 4 for details), flows show generally a three dimensional character, even for laminar regimes. The three dimensional mode character is influenced, among other factors, by the Reynolds number and the geometry of the body. It is, however, difficult to predict a priori these three dimensional flow patterns, which may influence the forces on the geometry and the wake dynamics. It is, therefore, suggested that three dimensional computations should be undertaken whenever possible. For even larger Reynolds numbers, the flow becomes turbulent and three dimensional computations become essential.

1.8.2.2 Turbulence

When inertia forces become significantly greater than viscous forces (i.e. high Reynolds numbers) viscosity is unable to suppress flow instabilities and turbulence develops. Turbulence is difficult to define but certain characteristics can be established. Turbulent flows are three dimensional, unsteady, have high vorticity, behave chaotically and are hence unpredictable (although statistical properties may be comparable). Moreover, turbulent flows are characterised by a broad spectrum where a wide range of length and time scales coexist [53]. Richardson in 1922 and Kolmogorov in 1941, introduced the idea of *energy cascade* in turbulent flows, where energy from larger structures (i.e. eddies) is handed down to smaller structures, which are eventually dissipated by laminar viscosity. It is this broad spectrum character and the *energy cascade* that renders turbulent flows very difficult to measure and simulate. However, the increase of computational power has made the simulation of basic flows possible (at low Reynolds numbers) using Direct Numerical

Simulation (DNS) where all of the flow scales are simulated without the necessity of any turbulence modelling. DNS of turbulence is computationally prohibitive for high Reynolds numbers, since the number of degrees of freedom necessary to capture the flow complexity scales as $Re^{9/4}$ (based on the magnitude of the velocity fluctuation and the integral scale in three dimensional flows [54]). Hence, when simulating high Reynolds number flows, it is necessary to use a *turbulence model* that mimics the effects of the turbulent scales. This thesis considers the Large Eddy Simulation (LES) approach. Within this technique, large energy-containing vortices are resolved whilst small quasi-isotropic eddies are modelled, reducing modelling requirements when compared to more traditional turbulent models as in the Reynolds Averaged Navier-Stokes (RANS) approach (see [179] for a review). A deeper explanation of turbulence and its modelling can be found in chapter 5.

1.8.2.3 Dynamic stall

The phenomenon of "dynamic stall" occurs when the foil sees a "rapid" variation in the angle of attack. This leads to a hysteresis cycle in the lift and drag forces [112]. This dynamic phenomenon is apparent in both wind and tidal cross-flow turbines and particularly for low to medium tip speed ratios ($\lambda = \omega R/U$). Dynamic stall requires accurate spatial and temporal flow predictions since the rapid variation of the (increased) forces may cause structural damage and reduce fatigue life. Cross-flow turbines are likely to encounter dynamic stall since the angle of attack is continuously changing during the blade cycle [134].

1.9 Engineering accuracy vs high accuracy

Before introducing the different techniques to solve the NS equations in order to find the most appropriate method to compute CFT flows, let us address the issue of accuracy.

It is often argued that high accuracy is not compulsory for most engineering applications since an error of 5 to 10% may be acceptable. To exemplify why high accuracy may

be of interest, the example depicted in Fig. 1.3 is considered. The figure depicts an array of two turbines where the second operates in the wake of the first.

Let us assume that the flow field around the first turbine is accurately resolved up

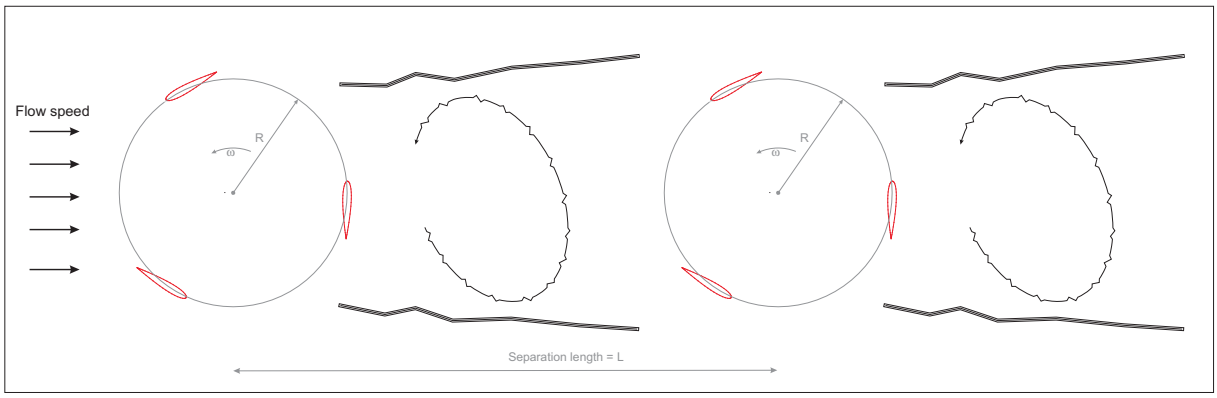

Figure 1.3: Array of turbines schematic.

to an error of 5%: $U = U^{exact}(1 \pm 0.05)$. Assuming also a 5% error when calculating U_{axial}, the error in the induction factor $\frac{U_{axial}}{U} = \beta$ (see section 1.8.1) can be bounded as $\beta \approx \beta^{exact}(1 \pm 0.07)$. If the induction factor for the rear half $\beta' = \frac{U_{wake}}{U_{axial}}$ has the same errors associated than β (which is clearly a conservative estimate), one obtains $\beta' \approx \beta'^{exact}(1 \pm 0.07)$. The wake velocity can now be estimated following: $U_{wake} = \beta\beta' U$ and has bounds $U_{wake} = \beta^{exact}\beta'^{exact}U^{exact}(1\pm0.07)(1\pm0.07)(1\pm0.05) \approx U^{exact}_{wake}(1\pm0.11)$. This corresponds to a 11% error in the incoming velocity for the second turbine. The examples shows how an initially "engineering acceptable" error of 5% may result in unacceptable levels after only one row of turbines. If a similar analysis is performed for the second row, it can be shown that the error increases to 15% for the wake (assuming the same errors in the induction factors).

Error bounds can also be estimated for blade forces and power output for the turbines. Let us consider an error when calculating the lift force of 7%: $L = L^{exact}(1\pm0.07)$, which relates quadratically to the 5% error in the velocity U. The power generated by the device is $Power = \omega T$, where T is the torque (i.e. the foil force in the tangential direction to the circular path) and ω is assumed to be an error free rotational speed. For simplicity,

let us state that $T = f(L)$, where f is a linear function depending mainly on the angle of attack. One can thus express the power as $Power = \omega f(L^{exact}(1 \pm 0.07))$. It is common practice, in the turbine community, to calculate the non-dimensional coefficient of power $Cp = \frac{Power}{1/2\varrho AU^3}$, where $A = D = 2R$ represents the frontal area for a CFT, which results in $Cp = Cp^{exact}\frac{(1\pm0.07)}{(1\pm0.05)^3} \approx Cp^{exact}(1 \pm 0.11)$. Once more a 5% error evolves into 11% error in the variable of interest. As for the downstream turbine that sees, as its incoming flow, the wake velocity of the first turbine, the error in the power coefficient increases to 25% (i.e. $Cp = Cp^{exact}\frac{(1\pm0.15)}{(1\pm0.11)^3}$).

These two simple examples illustrate why high accuracy may be of interest. Because the physical phenomena and interactions (e.g. stalled flows, blade-vortex interaction) are much more involved than presented by this simplified approach, unacceptable error levels can be expected if low accuracy is used to compute complex physics related to cross-flow turbines.

1.10 Predicting cross-flow turbine physics: state of the art

Experimental tests (wind tunnel, flume, cavitation tanks or towing tanks) are expensive and difficult to undertake. The necessity of building a realistic physical model and the requirement of accurate instrumentation to obtain detailed data can render experimental tests expensive. Furthermore, when carrying out experimental tests some issues need to be considered that might lead to inaccuracies in the data; a few to consider are: similitude effects (non-dimensional numbers (e.g. Reynolds, Froude) need to be conserved between full and model scales), free stream turbulence, wind tunnel or flume blockage effects and discrepancies between the idealised model and tested model (e.g. manufacturing tolerance levels or wall roughness). The alternative, analytical or numerical approaches, can in general complement experimental data. In particular solving the NS equations may

prove useful since information about primitive variables (i.e. velocity and pressure) and derived variables (e.g. vorticity) can be accessed at any required spatial coordinate at any time, providing in depth information about the flow behaviour. Numerical simulations are becoming more cost effective than experiments due to the proliferation of cheap computational resources. Furthermore, simulations can be performed for full scale cases avoiding scaling errors.

A brief literature review on the available techniques for predicting CFT flows is given herein. These include analytical models, Blade Element Momentum techniques, vortex based methods and full NS calculations.

1.10.1 Analytical approaches

Analytical models are based on the pioneering work by Rankine [143] in 1865 and Froude [58] 1889, who developed the Linear Momentum Actuator Disc Theory (LMADT) in the context of ship propellers. Lanchester [110] in 1915 and Betz [18] in 1920 adopted the same approach to model a horizontal axis wind turbine in uniform flow. This theory reduces the turbine to an actuator disc (rigid plate with defined porosity) and is based on inviscid flow and the control volume technique. This model has proven very useful to predict upper limits on the extractable energy by axial flow turbines.

This method was extended by Newman [123] in 1983, to compute cross-flow turbines by considering two parallel actuators where the first represent the effect of the front part of the blade circumference and the second the rear half (as considered in the simplified analysis presented in previous sections).

Within tidal turbine research, most efforts have been focussed on extending LMADT models to account for blockage effects due to turbine confinement (between free surface and sea ground). Recent advances in theoretical models have allowed for the estimation of tidal turbine performance with mean free surfaces effects [178]. It was shown that the tidal turbine performance increases significantly due to the free surface deformation. This

analytical model has been recently extended to account for mixing in the far wake [83], [45].

These models have proven to be very useful to acquire physical understanding on the relative importance of the parameters at play and also to provide bounds on the extractable energy by devices. However, these models remain simple, and cannot take the entire complexity of cross-flow turbine flows into account. Furthermore, it has been stated in [94], that they may become invalid for large tip speed ratios ($\lambda = \omega R/U$) and also for high rotor solidities ($\sigma = Nc/R$, with N the number of blades).

1.10.2 Blade Element Momentum approaches

In 1947, Glauert [61] extended the LMADT model in the propeller context (also applied to axial flow turbines) by applying a series of stream tubes on an actuator disc discretised along the radial direction. This method relies on pre-existing aerodynamic data, that may be obtained from experiments or other numerical calculations, to account for the local resistance to the flow. Glauert's technique was extended to cross-flow turbines by Templin [165] in 1974 (see Fig. 1.4.a) using a unique streamtube that encompasses the cross-flow turbine, with the method achieving limited success for high flow velocities. Multiple streamtube techniques (see example in Fig. 1.4.b) have been proposed (Wilson and Lissaman [181] in 1974 or Strickland [159] in 1975). In this model the actuator disc is separated into a series of adjacent streamtubes and thus allows for more accurate power predictions.

Variations of the presented models and other analytical techniques have been developed and reviews can be found in [94] or [134], but all share the limitation of considering a steady wake and relying on pre-existing foil data (typically lift, drag and moment coefficients). They are widely used within the wind turbine community as they provide fast and reliable calculations for problems where stall, dynamic effects, flow curvature or blade vortex interactions can be ignored.

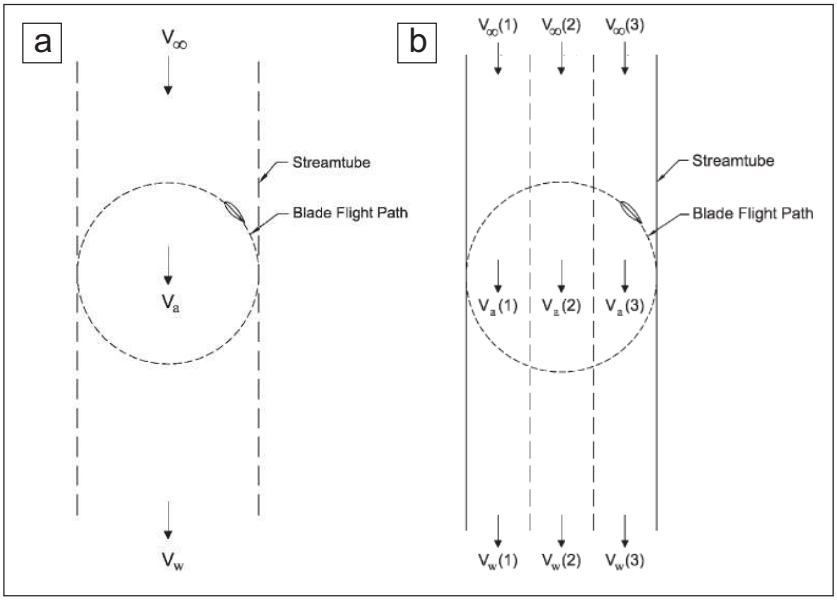

Figure 1.4: Streamtube models (a) single streamtube by Templin [165] and (b) multiple streamtube by Wilson and Lissaman [181]. Figures taken from [94].

1.10.3 Lifting line and vortex models

To capture wake structures and their evolution, lifting line methods were developed. Instead of using LMADT type approximations (based on Bernoulli's principles), they assume a continuous shedding of vortices from the blades. These methods also rely on pre-existing lift and drag curves to model the blade aerodynamics, but allow for free development of wake structures and their evolution. The Biot-Savart law can be used to calculate the induced velocities by the shed vortices allowing for back-effects from the wake development on the aerodynamic forces and induced angles of attack. These models are potential flow models and thus viscous effects are generally disregarded in the wake. They rely on pre-determined lift and drag curves and hence share most of the limitations as Blade Element Momentum models (e.g. neglecting dynamic effects, flow curvature and blade vortex interactions). A review of the historical developments of two and three dimensional vortex models for cross-flow turbines can be found in [94].

1.10.4 Full Navier-Stokes solvers

Two techniques are typically considered in the context of "full NS solvers". The first involves applying the lifting line modelling principle, which relies on pre-existing foil data, to model blade forces, which are in turn embedded into a full NS solver. This approach is similar to the vortex model technique, but allows viscous effects to be taken into account when predicting the wake structure and its development. Models of this type includes the work by Brahimi et al. [20] in 1995 and Antheaume et al. [5] in 2008. Although increased accuracy is obtained with respect to vortex models, the underlying limitations persist and are mainly related to the necessity of pre-existing foil data.

Full NS simulations have also been considered, but to the author's knowledge only with low order numerical techniques and mainly with finite volume (FV) commercial packages. Table 1.6 summarises, in chronological order, some of these simulations, giving details of the numerical techniques used, the geometries considered, and the flow conditions when the information was available. It is possible to conclude from the table that, to date and to the author's knowledge, all CFT turbine simulations have been performed using low order numerical techniques (mainly commercial Finite Volume codes) in two dimensions and with RANS turbulence models.

This section can be summarised by noting that a recent study [95] compared four different methods (none of them high order): a BEM model, a free vortex method code, a boundary element/finite volume vortex model and a commercial FV NS solver, to predict cross-flow turbine performance. This study showed disparity in the results for the various methods which were mainly attributed to the treatment of blade stall by the different techniques. It thus appears that analytical and numerical techniques are not mature enough to provide consistent predictions for cross-flow turbines and more sophisticated solvers are required.

Reference	**Num.** (a)	**Solver**	**Use** (b)	**Blades**	**Turbine diameter**	**Foil profile**	**Reynolds number**(c)	**tip speed ratio**	**Turbulence model**
McLaren et al. (2011) [120]	FV	CFX	W	3	2.5	NACA 0015	3.6×10^4	various	$SST - k - \omega$
Howell et al. (2010) [84]	FV(d)	FLUENT	W	3	?	NACA 0022	3×10^4	2	$RNG - k - \varepsilon$
Dai-Lam (2009) [39]	FV	CFX	T	3	0.45,5	NACA 0018, 0025	?	various	$SST - k - \omega$
Gretton et al. (2009) (∗) [67]	FV	CFX	T	3	2.5	NACA 0024	1×10^6	2-3.6	$SST - k - \omega$
Consul et al. (2009) (∗) [37]	FV	FLUENT	T	2-3	2.5	NACA 0015	1×10^6	1-7	$SST - k - \omega$
Klaptocz et al. (2007) [109]	FV	FLUENT	T	3	0.91	NACA 63_4021	0.3×10^6	1.75-3.5	S-A (e)
Ferreira et al. (2007) [51]	FV	FLUENT	W	1	0.4	NACA 0015	5×10^4	2	(f)
Hwang et al. (2006) [90]	FV	STAR-CD	W	4	2	(g)	0.45×10^6	2	?
Horiuchi et al. (2005) [82]	FV	STAR-CD	W	3	2.5	TWT11251	0.24×10^6	1.8-4.8	2D-DES
Vassberg et al. (2005) [170]	(h)	in-house	W	1	?	NACA 0015 WARP0015-RC8	1×10^6	5	B-L (i)
QinetiQ (2004) [141]	FV	FLUENT	T	3 and 6	10	NACA 0012	2×10^6	2	$k - \varepsilon$
Ishimatsu et al. (2002) [93]	FV	?	?	3	1	NACA 0018, 0030	1×10^6	?	laminar
Paraschivoiu (2002)[134]	FE(j)	in-house	W	1	1.22	NACA 0015	6.7×10^4	2.5	laminar, C-S, J-K (k)
Hansen-Sorensen (2001) [73]	FV	STAR-CD	W	1	1.22	NACA 0015	6.7×10^4	2.5	$k - \varepsilon$, non-linear $k - \varepsilon$

Table 1.6: Full NS simulations of cross-flow turbines for wind and tidal applications

(∗) More details and other conditions can be found in the first author PhD thesis Refs. [36] and [64].
(a) Numerical technique for spatial discretisation (see section 1.11 for descriptions).
(b) Cross-flow turbine application: Wind (W) or Tidal (T).
(c) This is the Reynolds number based of the foil chord Re=$\frac{U_{air}c}{\nu}$.
(d) FV: Low order Finite Volume method.
(e) S-A: Spallart and Allmaras.
(f) S-A, $k - \varepsilon$, 2D-Detached Eddy Simulation (DES), 2D-Large Eddy Simulation (LES).
(g) NACA 0018, 2418, 4418, 6418, 63_3018, 63_3218, 63_3418, 63_3618.
(h) Time spectral method combined with a FV formulation for spatial discretisation.
(i) B-L: Baldwin-Lomax.
(j) FE: Low order Finite Element method.
(k) C-S: Cebeci-Smith and J-K: Johnson-King.

1.11 Numerical techniques for the solution of PDE's

Computational Fluid Dynamics (CFD) methods have more than half a century of history. The first reported numerical results for the flow past a circular cylinder were performed in 1933 by Thom [166] and repeated by Kawaguti [102] in 1953 using a mechanical desktop calculator. Since then, major developments in numerical methods and computer science have allowed the field to grow to reach its current maturity. There has been an impressive development of numerical techniques for the solution of partial differential equations in the last thirty years mainly because of the increase of computer capabilities to handle big matrices efficiently and solve large systems of equations. Furthermore, the continuous increase in computational power [184] and the prospect of further increase, following Moore's law and through new hardware developments, such as multiprocessor chips or Graphical Processor Units (GPUs), is allowing more complex flow physics to be simulated, rendering CFD a promising tool for fluid flow studies.

The Navier-Stokes equations define the dynamic behaviour of fluids and are a system of partial differential equations that, generally and in the absence of simplifications, can only be solved using numerical techniques. To solve these equations, a mesh is used to discretise the flow field, enabling the numerical method to approximate the flow variables and their derivatives to a system of algebraic equations. Various numerical methods exist to solve systems of partial differential equations and the some of the most popular are reviewed here.

Generally these methods adopt the method of lines approach, which decouples the time integration and spatial discretisation (although other alternatives have been considered: see for example [108] for a space-time DG formulation for the compressible NS equations). Temporal discretisation is often performed using the finite difference technique whilst various approaches exist for the spatial approximation. In the following, the characteristics of the various spatial discretisations are described.

1.11.1 Low order and high order methods

Let us distinguish between traditional low order ($k \leq 3$), where k is the formal order of the scheme) methods: Finite Differences, Finite Volumes and Finite Elements, and high order ($k \geq 3$) techniques: spectral methods (e.g. purely spectral using Fourier series or h/p Spectral) and Discontinuous Galerkin (DG).

On the one hand, low order methods can only rely on mesh refinement (i.e. h-refinement) to obtain more accurate solutions. This strategy involves increasing the number of mesh elements for a fixed polynomial order k[3]. On the other hand, high order methods allow for mesh and polynomial refinement (i.e. p-refinement) strategies. This last path to obtain better accuracy considers a mesh with a fixed number of elements where the polynomial order within each element is increased, leading to a better approximation of the exact solution for smooth problems.

To illustrate the advantage of high order methods over low order methods, Fig. 1.5 introduces the reduction of the error (relative error in the L^2 norm) against the total number of Degrees of Freedom (DOF, see chapter 2 for definitions) for a two dimensional Poisson equation (i.e. $-\triangle u = 2\pi^2 cos(\pi x)cos(\pi y)$, see chapter 2 section 2.5.1 and appendix C for details) using the Discontinuous Galerkin solver developed by the author. The figure shows the errors for four meshes with 32, 50, 98 and 128 triangular elements and polynomial orders k ranging from 2 to 7 (shown in boxes). It can be seen that for an equal number of DOF, the p-refinement strategy on a coarse mesh leads to a more accurate solution than the h-refinement strategy. Alternatively, it can also be seen that to achieve a fixed level of accuracy, fewer number of DOF are required when a high order polynomial in a coarse mesh is set than when h-refinement is performed using a fixed polynomial order.

To illustrate this further, Fig. 1.6 shows solution snapshots of the 2D laminar vortex shedding (chord Reynolds number Re=800) aft a NACA0012 using the developed DG

[3]Note that throughout this thesis the order of the polynomial is denoted k but the action of enriching the polynomial space is called p-refinement

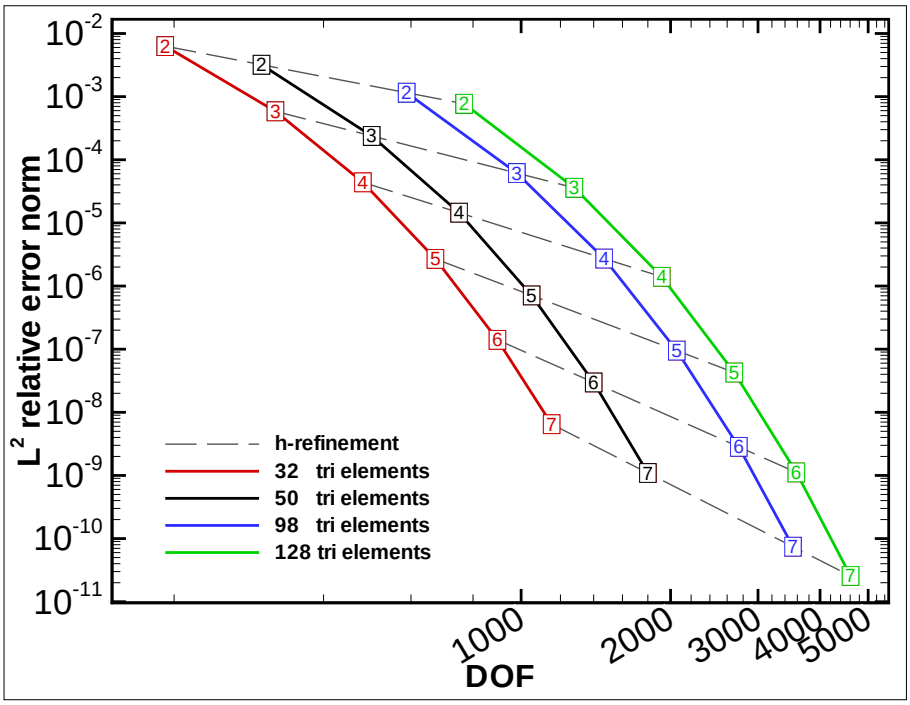

Figure 1.5: h-refinement vs p-refinement for a Poisson problem using the developed DG solver.

solver and the low order finite volume code Fluent [56]. The DG solution has been obtained using a mesh constituted of 798 tri-quad elements and polynomial order $k = 5$ leading to a total number of degrees of freedom of 17448. The finite volume solution required second order discretisation in space and time and a mesh constituted of 29516 DOF (1.7 times more DOF than the DG solution). It is noticeable the low numerical diffusivity on the shed vortices (vortex structures are preserved) for the high order DG code, when compared to the low order solution. It is this feature of high order h/p spectral and DG methods that has led to their increased popularity in the flow community.

As it has been introduced in previous sections, a numerical method to simulate CFT flows should be able to cope with the relative motion of an inner rotating mesh with respect to an outer static mesh zone. The sliding mesh technique couples two mesh regions and allows for relative mesh motion. This relative motion creates "hanging nodes" at the interface between static and rotating elements as shown in Fig. 1.7. In Fig. 1.7.a.1 and 1.7.a.2 no special treatment is considered at the sliding interface, showing overlaps

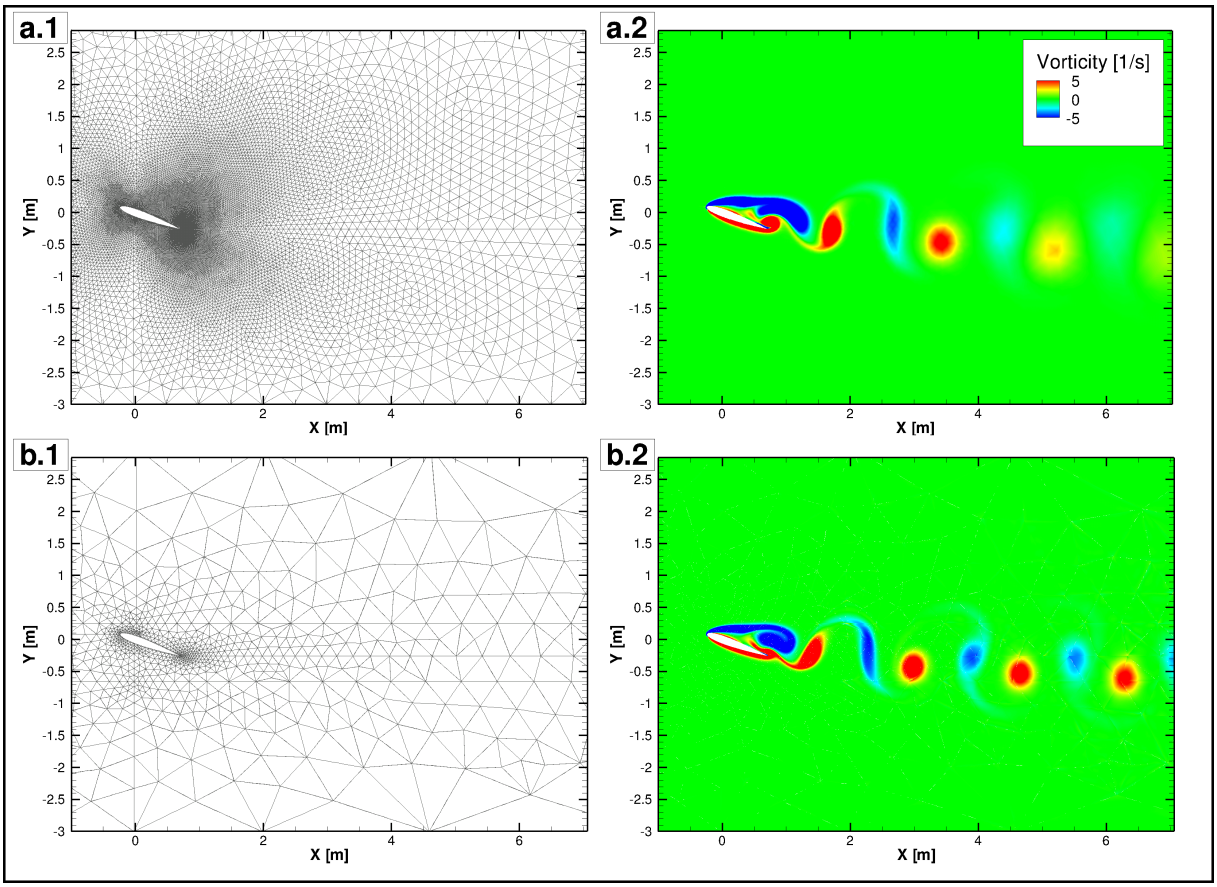

Figure 1.6: Snapshots of z-vorticity for the vortex shedding aft a NACA0012 at Re=800: (a) Finite volume code (b) DG solver.

between elements which renders the approach not acceptable, and remeshing should be considered. This problem can be overcome by considering curved edges as shown in Fig. 1.7.b.1 and 1.7.b.2, where the interface between static and rotating meshes is matched. This latter option is the essential idea for the development of the sliding mesh technique detailed in chapter 3.

The first feature, the ability to perform h and p refinement, is highly desirable feature of any numerical method where high accuracy is required. The second, the sliding mesh capability, is essential to simulate cross-flow turbine flows and other turbomachinery-like applications involving rotational motion. Both features should be considered when selecting a numerical technique to simulate complex flows with rotating geometries.

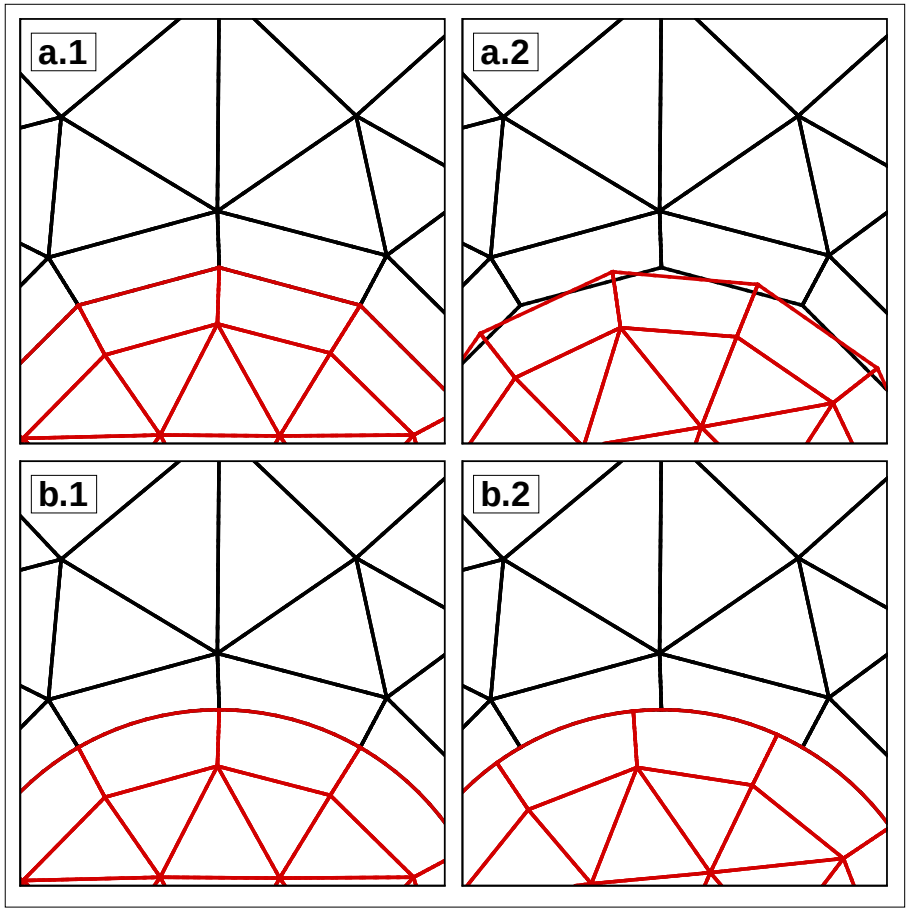

Figure 1.7: Hanging nodes and curved edges. Sliding mesh interface defined through: (a) straight edges and (b) circular curved edges.

1.11.2 Low order techniques

In this section Finite Differences (FD), Finite Volumes (FV) and Finite Elements (FE) are briefly introduced and the key advantages and disadvantages with regards to simulating CFT are outlined. It is interesting to note that the distinction between techniques is often historical and that it is possible, in some cases, to express one technique using another one. For example, the FV method can be shown to be a simplified version of the FE method [91].

The following references have been used: for the Finite Difference method [80], [81], [137]; for the Finite Volume method [54], [172], [137]; for the Continuous Finite Element method [190], [191], [192], [137].

1.11.2.1 Finite Differences (FD)

This method was first introduced by Euler in the 18th century for the solution of partial differential equations and requires the differential form of the NS equations. The space is discretised by a set of grid points at which the variable values are unknown. Taylor series expansions are used to approximate the derivatives of the unknowns using neighbouring grid points. This method has been widely used and validated and is very efficient for simple geometries. Moreover, its implementation is the simplest of all, but has limitations. The method can be extended to high order by increasing the number of grid points (i.e. stencil size) used to compute the required variables. However, this increase renders the imposition of boundary conditions difficult for arbitrary geometries and is thus generally applied to relatively simple grids (e.g. regular structured grids) and simple geometries.

1.11.2.2 Finite Volumes (FV)

The FV method uses the integral form of the NS equations and Green's theorem to reformulate the convective and diffusive terms from volume to surface integrals (i.e. flux contributions). In this case, the spatial discretisation does not consist of grid points, but of small volumes (areas in 2D) and the integral form of the equation can be directly calculated in the discretised physical space without further transformation. The method ensures conservation of mass and momentum locally and should not provide unphysical results (e.g. negative densities for compressible flows), which is regarded as an important advantage by its developers and users. This feature cannot be ensured by either FD or continuous FE methods (as explained in the next section). This method is widely used by the fluid dynamics community, probably due to its simplicity and intuitive implementation. The flow variables are normally approximated in a piecewise constant, volume by volume, fashion and the surface integrals (or fluxes) are calculated using values from neighbouring cells. In order to render the method high order, a larger stencil of cells needs to be considered, which becomes difficult for complex geometries. Generally, this

method is limited (in commercial and academic packages) to a formal order of convergence of $k \leq 3$ and mesh refinement (h-refinement either globally or locally) is the choice to achieve more accurate solutions. Due to its construction based on element to element formulation, this method can incorporate meshes with hanging nodes and sliding mesh interfaces to account for relative motion of mesh zones. Boundary conditions are easy to handle for the low order version of the method since they can be imposed through the fluxes. However, for high order spatial approximations a larger stencil is required and the implementation of boundary conditions becomes more difficult.

1.11.2.3 Finite Elements (FE)

Historically, the Finite Element method was developed for applications in structural and solid mechanics and it was not until the 1970's that the method was extended to fluid dynamics [81]. As in the FV method, the FE method uses a mesh constituted of 3D volumes (or 2D areas in two dimensions). After multiplying the PDE set by a test function, the equations are integrated by parts (which relaxes the smoothness required) leading to the so called *weak formulation*. Formally, the residual of the equation (difference between the exact solution and the approximated solution) is required to be orthogonal to all test functions in the polynomial space spanned by the test function, hence minimising the residual. The solution approximation is constructed using basis functions (or ansatz functions), which are typically piecewise linear functions. Both test and basis functions are required to be continuous across elements and this is the reason for these type of methods to be named *conformal discretisations*. The choice of the test and basis functions defines various submethods. The most common, in both structural and fluid dynamics, is the Galerkin method where the test functions are chosen to be the same as the basis functions. An alternative formulation for convection dominated problems is called Streamline Upwind Petrov-Galerkin (SUPG). If standard linear shape functions are used to describe basis and test functions (i.e. Galerkin method), it can be shown that

the resulting scheme gives oscillatory results for convection dominated flows (as in FD or FV if central differences are used without upwinding). To avoid this problem, the SUPG method, that uses different basis and test functions, was developed. In this method, the test functions are upwinded in the streamwise direction. Since upwinding is performed only in the streamwise direction, a discontinuity is introduced in the normal flow direction. Since discontinuities are not allowed in continuous Galerkin methods, some oscillations may develop. To damp these oscillations a smoothing diffusion term may be added in the direction normal to the streamline [191], which complicates the implementation of the method for general complex meshes.

Boundary conditions are generally implemented "strongly" by setting the appropriate functions to fulfil the boundary condition. Due to its global setting, only global conservation is ensured (and not locally at an element level), which has led to criticism. However, the robust mathematical theory behind this method (e.g. *a-priori* and *a-posteriori* error estimates) has established a robust framework which does not exist in other low order techniques. Instead of using piecewise linear polynomials, higher order polynomials can be used. These family of methods are often referred to as h/p Spectral methods (see following section).

1.11.3 High order techniques

Two high order methods are introduced in this section. The following references have been used to summarise the essential of the method's capabilities and their limitations: for the h/p Spectral method [101] and for the Discontinuous Galerkin Finite Element methods [113], [77], [101], [137].

1.11.3.1 h/p Spectral

Numerical implementation in computational fluid dynamics of purely spectral methods (no mesh is required) were first developed by Gottlieb and Orzag in the 1970's [63]. In

most classical spectral methods, Fourier expansions are used to approximate the solution and Fast Fourier Transformation (FFT) to transform from the computational space (or Fourier modal space) to the physical space, which renders the method very efficient (see chapter 4 for details). However, spectral methods with Fourier series are limited to periodic boundary conditions and hence expansions based on polynomials (e.g. Chebyshev, Jacobi polynomials) have been used to handle more general boundary conditions.

In the h/p Spectral method, a mesh constructed of volumes or areas is required as in FV and FE methods. This method, introduced to fluid flows in 1984 by Patera [136] (see also the monograph of Szabó and Babuška [161] for the introduction of the method in the context of structural mechanics), is a modification of the classical FE method where the basis functions (typically piecewise linear in FE) are replaced by polynomials of arbitrary degree that maintain continuity across elements. The capability to use high order polynomials allows for high order spatial calculations on a coarse meshes (p-refinement). The overwhelming advantage of spectral type methods over low order methods is that *exponential* or *spectral convergence* can be achieved. This means that, when the number of degrees of freedom is doubled, the error in the solution decreases by at least a factor of two and not with a fixed factor as in low-order methods (as shown previously in Fig. 1.5). High order methods provide fast convergence and low diffusion and dispersion errors (see [101] and references herein) and are very suitable for applications where numerical errors have to be minimised. h/p Spectral methods are suitable for problems where high accuracy is required, but their flexibility is limited by the requirement of continuity between elements. If different polynomial orders are used in neighbouring elements, continuity is no longer easily ensured and *functional incompatibility* arises. On the other hand if hanging nodes (nodes that are not shared by two edges of neighbouring elements) are used, a *geometric incompatibility* is created. Various methods have been developed to handle these incompatibilities in continuous h/p Spectral methods: *iterative patching*, *constrained approximation* or *mortar patching* are the most popular [101], but

these methods increase the computational cost and do not always maintain the accuracy of the scheme. A natural alternative is to relax the continuity condition across elements, which leads to the Discontinuous Galerkin approach.

1.11.3.2 Discontinuous Galerkin (DG)

This method can be defined as an extension of h/p Spectral methods, where the continuity requirement across element boundaries is relaxed or as a high order FV method with compact stencil. As in spectral methods, high-order polynomials can be used within each element allowing for *exponential convergence*. However, in DG methods, continuity is not required and therefore different polynomial orders can be easily used in different elements. Moreover, if hanging nodes are present in the mesh, as required when the sliding mesh technique is used to model CFT turbines, the DG method provides a natural advantage as the discontinuities can be naturally handled. Another advantage of DG methods, over continuous techniques (e.g. conformal discretisations using FE or h/p Spectral), is that, as for the FV method, the conservation of flow properties (i.e. mass and momentum) are explicitly ensured in an element fashion by the numerical fluxes. This property is directly inherited by the local element nature of the method.

The DG method offers high flexibility, since unstructured meshes, formed of triangular and/or quadrilateral elements, can be used without compromising the order of the scheme. Furthermore, boundary conditions can be easily implemented and do not require special treatment to maintain high order convergence as is required by FD and FV methods. As in spectral methods, highly accurate solutions can be obtained by performing h/p refinement. The main drawback of the method is that it requires more degrees of freedom than its continuous counterpart. The additional degrees of freedom arise from the discontinuities between elements since different edge functions coexist in neighbouring elements. The resulting equation system is consequently larger than when considering continuous elements and requires more computational effort than classical continuous methods. This

statement is true if low order polynomials are used. However, the relative cost of the additional boundary degrees of freedoms to the internal degrees of freedom, decreases rapidly for high polynomials. To exemplify this, let us consider a two dimensional problem and a mesh with N^{el} triangular elements with polynomial spaces k. On the one hand, if a conformal h/p Spectral discretisation is considered, the number of global number of degrees of freedom (DOF) required is $DOF_{Spectral} \approx \frac{N^{el}}{2}k^2$ (see [173]). On the other hand, a non-conformal DG discretisation leads to a global number of degrees of freedom of $DOF_{DG} \approx \frac{N^{el}}{2}(k+1)(k+2) = \frac{N^{el}}{2}k^2(1+\frac{3}{k}+\frac{2}{k^2})$ (see chapter 2 for details). From this last expression, it can be seen that for high polynomial orders k, the number of degrees of freedom for the DG discretisation tends to $DOF_{DG} \approx \frac{N^{el}}{2}k^2$. This simple analysis shows that for high polynomial orders $DOF_{DG} \approx DOF_{Spectral}$ and the additional cost of DG methods is far from overwhelming.

Furthermore, to some extent this additional cost may be alleviated through parallelisation (i.e. distribution of the problem over various computational nodes) since most DG methods, and in particular the Interior Penalty formulation used throughout this thesis, have compact stencils [156].

Recently, researchers have focussed on minimising the additional cost associated with DG methods by reducing the degrees of freedom arising from the DG discretisation, which is giving rise to new methods (e.g. Hybridizable methods [124]).

A more detailed description of the particularities of the incompressible NS equations and the use of the DG method for elliptic problems is introduced in section 1.13.

1.12 Cross-flow turbine physics and numerical methods

One of the aims of this project is to create an accurate CFD tool to simulate the major physical phenomena present in cross-flow turbine flows. From the previous discussion, it

is possible to determine the numerical requirements of each of the physical phenomena outlined. These are summarised in table 1.7, where the flow physics are matched to the numerical requirements. Bearing in mind the physical phenomena, the main strengths

Physical phenomenon	**Numerical requirement**
Geometry	2D or 3D
Complex geometry	Unstructured mesh
Incompressible flows	Incompressible solver
Moving geometries	Allow mesh discontinuities and sliding meshes
Boundary layer resolution	High order method and high order boundary conditions
Blade vortex interaction	High order method and low numerical dissipation
Turbine wake and conservation of wake shape structure	High order method and low numerical dissipation
High Reynolds number	Turbulence modelling
Flow complexity and calculation speed	Parallelisation

Table 1.7: Numerical requirements for cross-flow turbine flows

and weaknesses of each method are summarised in table 1.8. The table shows three levels (*difficult*, *fair* and *good*) describing the difficulty in implementing the required capability. When the word *difficult* is used it does not mean that the capability is not feasible in the method, but that it requires additional effort since the method does not allow for a natural implementation of the technique (e.g. handling discontinuities if continuous Finite Element or h/p Spectral methods are used). At the other extreme, *good* means that the implementation is direct.

Inspection of tables 1.7 and 1.8, reveal the DG method as the most appropriate for the computation of cross-flow turbine physics, since this technique matches all of the requirements imposed by the flow physics. In particular, the flexibility of the scheme to perform h/p refinement should prove to be an advantage when simulating CFT turbine flows that have very different characteristic sizes throughout the domain (e.g. foil versus wake scales). In addition, the capability to handle meshes with discontinuities (e.g. with

	FD	FV	FE	h/p Spectral	DG
Numerical requirement					
2D	good	good	good	good	good
3D	good	good	good	good	good
Unstructured mesh	difficult	good	good	good	good
Incompressible solver	good	good	good	good	fair(a)
High order (low dissipation)	good	difficult	difficult	good	good
Discontinuities between elements and hanging nodes	good	good	difficult	difficult	good
Parallelisation	good	good	good	fair	fair(b)
Local conservation	good	good	difficult	difficult	good
h-refinement	good	good	good	good	good
p-refinement	difficult	difficult	difficult	good	good
Turbulence modelling	good	good	good	fair(a)	fair(a)
Efficiency requirements					
CPU/memory requirement	good	good	good	difficult	difficult
Compactness of the scheme when high order (stencil size)	difficult	difficult	fair	good	good
Programming complexity	good	good	good	fair	fair

Table 1.8: Comparison of numerical techniques

(a) This relates to the method being relatively new and hence less literature is available. (b) It is often argued that DG methods are easy to parallelise due to the local (element based) nature of the method. This is generally true for compressible solvers (i.e. hyperbolic type equations) where the flux contributions are treated explicitly. However, in incompressible formulations (i.e. elliptic type equations) parallelisation is not trivial due to the coupling enforced by the implicit treatment of the numerical fluxes.

hanging nodes) is a desirable characteristic to simulate rotating blades. A drawback of this method is that it is relatively new and thus less literature exist. Furthermore, many submethods exist and are still under development, which renders the selection and implementation of the method complicated.

1.13 High order Discontinuous Galerkin methods: state of the art

Discontinuous Galerkin methods were developed by the pioneering work of Reed and Hill [144] in 1973 in the framework of hyperbolic partial differential equations for the neutron transport equation, but it was only in the late 1990's that the method was generalised to elliptic and convection-diffusion problems: Bassi and Rebay in 1997 [15], Baumann in 1997 [16] or Cockburn and Shu in 1998 [32]. Since, DG methods have proven useful in solving the compressible (e.g. [15], [111] or [125]) and the incompressible NS equations: [13], [146], [30], [156], [122], [124], including the author's [52].

Various DG type submethods for elliptic type problems exist (see [9], [60], [146] for reviews), which differ in their treatment of the numerical fluxes used to communicate the information through discontinuous element interfaces. A general framework for derivation and analysis of DG methods for elliptic equations (e.g. Interior Penalty, Local Discontinuous Galerkin, Bassi-Rebay) was derived by Arnold et al. [9] where it was shown that all known methods can be compared through the so called *primal formulation*. In this work, only the Interior Penalty (IP) method is considered. This technique is a subfamily of the DG method that uses a penalisation technique to enforce a certain degree of smoothness in the numerical solution. A detailed derivation of the Interior Penalty method following this approach can be found in appendix B and the following references [9], [22] and [74].

Conceptually, IP methods were first introduced in the late 1960s and 1970s as a mechanism to impose weakly Dirichlet boundary conditions under conformal discretisations ([114], [127] or [11]). In 1978, Wheeler [176] generalised the concept for inter-element continuity for second order elliptic problems. The analysis of the method was later provided by Arnold in 1979 [7]. During the 1980s and 1990s, and mainly due to the computational cost associated with DG methods for solving elliptic problems, interest declined and focussed on using DG for hyperbolic type problems (see [31] for a review). During

this time computational power has risen spectacularly and in 1997, Baumann [16] reactivated interest in using IP methods to solve elliptic and parabolic problems. At the same time, other DG methods for elliptic type problems were developed (e.g. the Bassi-Rebay method in 1997 [15] or the local discontinuous Galerkin method of Cockburn and Shu in 1998) showing the potential of these new techniques. Subsequently DG methods and IP variants have become a rich area of research.

The DG method is particularly new when applied to general high Reynolds number turbulent flows related to airfoil aerodynamics and a brief overview is given here. Bassi et al. [15] presented in 2005, the first results applying a RANS modelling approach in conjunction with the DG method for the compressible NS equations. The $k-\omega$ model was modified to ensure stability in which $log(\omega)$ instead of ω is used as the unknown variable. In the 2007 AIAA conference, various papers (see [125], [132]) presented DG formulations in conjunction with RANS modelling to compute airfoil aerodynamics. In particular some modifications of the Spalart-Allmaras turbulence model under the DG framework were presented in [125]. Landmann et al. [111] developed, in 2008, a compressible parallel DG code using two turbulence models (Bassi modification of the $k-\omega$ model and standard Spallart-Allmaras). They showed results for flows around cylinders, flat plates and a NACA-0012 airfoil. Landmann's code can cope with curved boundaries and is limited to 2D flows in the context of helicopter aerodynamics. Since then, other RANS models for DG methods for compressible flows have been developed (e.g. [75]) but to the author's knowledge none for the incompressible NS equations.

Regarding the Large Eddy Simulation (LES) approach to compute turbulent flows using DG methods, limited published work exists and has been mainly restricted to simple geometries: fully developed channel flows [34], [142], two-dimensional channel flows [152] or the flow over a back-facing step [185]. More recently and in the context of compressible flows, Uranga et al. [168] presented the first DG method for three-dimensional flows and aerodynamic applications (flow over airfoils) using an implicit LES (ILES) model.

To the author's knowledge, there have not been calculations of incompressible Navier-Stokes using LES models in conjunction with DG method for aerodynamic applications until this thesis (see chapter 5).

Chapter 2

A high order Discontinuous Galerkin solver for incompressible flows

This chapter reports the verification and validation of a high order Discontinuous Galerkin (DG) method for solving the incompressible Navier-Stokes (NS) equations in two dimensions on triangular meshes with straight edges. Verification is performed with simplified problems where the exact solution is known and hence the error between the exact solution and the numerical solution can be quantified. The validation of the solver demonstrates its ability to provide accurate results for complex fluid flow problems.

The numerical solution of the incompressible NS equations requires discretisation in space (spatial derivatives) and time (temporal derivatives). Spatial and temporal derivatives can generally be decoupled leading to the *method of lines*. In this work, finite differences are selected to discretise the equations in time (here a fractional-step method) whilst a high order DG method is retained for spatial discretisation.

To perform the temporal discretisation of the incompressible NS equations, a fractional-step method (i.e. dual splitting technique) is chosen. Reviews of temporal discretisations to solve the incompressible NS equations can be found in [43], [68] or [101]. In particular, the code developed here uses the splitting projection scheme detailed in Karniadakis et al. [100]. Although this temporal scheme has been previously used in conjunction with high

order continuous modal h/p Spectral [100] and discontinuous high order nodal [77] spatial discretisations, it has not, to the author's knowledge, been used with the DG Interior Penalty (DG-IP) formulation in combination with modal basis functions. The selected combination of spatial and temporal discretisation schemes presents a novel approach for the solution of the incompressible NS equations. As in the present implementation, the temporal discretisation scheme, when implemented with the continuous modal and discontinuous nodal formulations, yields stable solutions when the polynomial order for the velocity and pressure approximations are equal, provided stability conditions are fulfilled. In this chapter, stability conditions are analysed for the combination of the splitting scheme selected and the DG spatial discretisation and equal order pressure-velocity pairs. Various test cases are presented for verification and validation purposes.

2.1 Chapter outline

Firstly, section 2.2 describes the methodology used which requires temporal (section 2.2.1) and spatial discretisation (section 2.2.2). The temporal discretisation leads to a decoupled system (i.e. minor coupling through boundary conditions) for the velocity and pressure, and requires the solution of elliptic problems that encompass the Poisson and Helmholtz equations. The solution of these equations using the DG formulation is described in section 2.2.2.3. This section also includes other aspects such as numerical evaluation of volume and surface integrals (i.e. quadrature). Relevant implementation details are given in section 2.3. A brief outline of the norms and functionals (e.g. lift and drag forces) used throughout the thesis, to quantify results, can be found in section 2.4.

Section 2.5 is devoted to the verification of the method. Verification of the method is performed for elliptic problems, followed by an unsteady Stokes problem (without non-linear terms) where an analysis of the stability of the scheme is presented together with stability conditions for equal order pressure and velocity spaces. Section 2.6, explores the solver's properties for the full NS equations (i.e. including non-linear terms) using the

Taylor Vortex problem with known analytical solution. Finally, the method is validated with a square cylinder problem for two laminar flow cases: Reynolds number (Re) of 10, that leads to a steady wake and $Re = 100$ that results in unsteady periodic vortex shedding. Comparison to the continuous h/p Spectral code Nektar [101] are also included.

2.2 Methodology

Let Ω be a domain in $\mathbb{R}^2$ with boundaries $\partial\Omega$ of Dirichlet ($\partial\Omega_D$) or Neumann ($\partial\Omega_N$) type, where $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ has associated the outward unit normal vector $\mathbf{n}$ and $\partial\Omega_D \cap \partial\Omega_N = \emptyset$. The unsteady non-dimensional incompressible NS equations can be written using the convective formulation for the non-linear terms as:

$$\begin{aligned}
\tfrac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}.\nabla)\mathbf{u} &= -\nabla p + \tfrac{1}{\mathrm{Re}}\nabla^2\mathbf{u} && \text{in} && \Omega \times [0,T], \\
\nabla.\mathbf{u} &= 0 && \text{in} && \Omega \times [0,T], \\
\mathbf{u}(t=0) &= \mathbf{u_0} && \text{in} && \Omega, \\
\mathbf{u} &= L_D && \text{on} && \partial\Omega_D, \\
\tfrac{1}{\mathrm{Re}}\tfrac{\partial \mathbf{u}}{\partial n} + p\mathbf{n} &= 0 && \text{on} && \partial\Omega_N,
\end{aligned}$$

where t represents the the dimensionless time, $\mathbf{u} = (u, v)^T$ and p are the non-dimensionalised velocity vector and static pressure (normalised using upstream dynamic pressure $q = \varrho U^2$, where ϱ is the fluid density) respectively and Re is the Reynolds number (i.e. $\mathrm{Re} = UL/\nu$ where U and L are the characteristic velocity and length scale and ν is the kinematic viscosity). In addition, $\mathbf{u_0}$ represents an initial condition for the velocity field and L_D defines the Dirichlet boundary condition for the velocity.

2.2.1 Temporal discretisation of the incompressible NS equations

There is vast literature on temporal procedures for the solution of the incompressible NS equations. Reviews can be found in [101] or [43]. Among these options, a second order dual stiffly stable method, developed by Karniadakis et al. [100], is selected to discretise the equations in time. This choice in mainly driven by the success that this splitting scheme has shown when combined with high order conformal discretisations (h/p Spectral type).

Within this scheme the non-linear terms are treated explicitly whilst viscous and pressure terms are handled implicitly. The resulting temporally discretised momentum equation is given by:

$$\begin{aligned} \frac{\gamma_0 \mathbf{u}^{n+1} - \alpha_0 \mathbf{u}^n - \alpha_1 \mathbf{u}^{n-1}}{\Delta t} = & - \nabla p^{n+1} - \beta_0 \mathbf{N}(\mathbf{u}^n) - \beta_1 \mathbf{N}(\mathbf{u}^{n-1}) \\ & + \frac{1}{\mathrm{Re}} \nabla^2 \mathbf{u}^{n+1}, \end{aligned} \tag{2.2.1}$$

where γ_0, α_0, α_1, β_0 and β_1 are constants chosen to achieve the required level of temporal accuracy (see details below) and the indices $n-1$, n and $n+1$ refer to time steps separated in time by an increment Δt. In [100], first, second and third order schemes were derived. However within this thesis only the second order version is considered.
In addition, $\mathbf{N}(\mathbf{u})$ represents the non-linear contributions which can be, for instance, expressed in their convective form $\mathbf{N}(\mathbf{u}) = (\mathbf{u}.\nabla)\mathbf{u}$ (further details are given in section 2.2.2.2).

The present method allows, by the use of intermediate variables, the incompressible NS equations to be split into three distinct equations that can be solved successively at each time-step; an explicit non-linear convection equation (step 1), a Poisson equation for the pressure (step 2) and a Helmholtz equation to account for viscous diffusion (step 3).

STEP 1 - Non-linear term:

The first step accounts for the explicit computation of the non-linear convective term using the Adams-Bashforth second order scheme. Introducing the intermediate velocity $\tilde{\mathbf{u}}$, the first step can be written as:

$$\frac{\gamma_0\tilde{\mathbf{u}} - \alpha_0\mathbf{u}^n - \alpha_1\mathbf{u}^{n-1}}{\Delta t} = -\beta_0\mathbf{N}(\mathbf{u}^n) - \beta_1\mathbf{N}(\mathbf{u}^{n-1}). \tag{2.2.2}$$

By choosing $\gamma_0 = 3/2$, $\alpha_0 = 2$, $\alpha_1 = -1/2$, $\beta_0 = 2$ and $\beta_1 = -1$, the method is second order accurate in time for velocity. However, this second order method is not self-starting. To start the scheme one can use: $\gamma_0 = \alpha_0 = \beta_0 = 1$ and $\alpha_1 = \beta_1 = 0$ which reduces to the first order explicit forward Euler method.

STEP 2 - Pressure term:

The second step treats the pressure term implicitly whilst imposing the divergence-free constraint on the intermediate velocity $\tilde{\tilde{\mathbf{u}}}$, resulting in a Poisson equation for the pressure. Introducing a second intermediate solenoidal velocity $\tilde{\tilde{\mathbf{u}}}$, one can write:

$$-\nabla p^{n+1} = \gamma_0\frac{\tilde{\tilde{\mathbf{u}}} - \tilde{\mathbf{u}}}{\Delta t}. \tag{2.2.3}$$

Taking the divergence of Eq. 2.2.3, and imposing $\nabla.\tilde{\tilde{\mathbf{u}}} = 0$ the following equation is obtained:

$$-\nabla^2 p^{n+1} = -\frac{\gamma_0}{\Delta t}\nabla.\tilde{\mathbf{u}}. \tag{2.2.4}$$

This is a Poisson problem that can be closed using a Neumann boundary condition at inflow and wall boundaries that may be named "Neumann-Vorticity" which is derived from taking the dot product of the outward normal vector at the boundaries $\mathbf{n}$ with the

momentum equation:

$$\frac{\partial p}{\partial n}^{n+1} = \mathbf{n}.\frac{\partial \mathbf{u}}{\partial t}^{n+1} - \beta_0 \mathbf{n}.\left(\mathbf{N}(\mathbf{u}^n) + \tfrac{1}{\mathrm{Re}}\nabla \times \boldsymbol{\omega}^n\right) - \beta_1 \mathbf{n}.\left(\mathbf{N}(\mathbf{u}^{n-1}) + \tfrac{1}{\mathrm{Re}}\nabla \times \boldsymbol{\omega}^{n-1}\right), \tag{2.2.5}$$

where the vorticity definition ($\boldsymbol{\omega} = \nabla \times \mathbf{u}$) and the identity for an incompressible flow: $\nabla^2\mathbf{u} = -\nabla \times \nabla \times \mathbf{u} = -\nabla \times \boldsymbol{\omega}$, have been used. This expression indirectly enforces the incompressibility constraint and gives a consistent method [101]. Furthermore, it was shown in [100] that the *boundary divergence flux* is controlled by the time step if this vorticity formulation is used.

Remark: *As noted in [68], this definition, for the boundary condition, requires the velocity to have computable second derivatives. For this reason, all computations are performed using polynomial orders $k \geq 2$.*

Remark: *The term $\frac{\partial \mathbf{u}}{\partial t}^{n+1}$ may be set to zero for static walls. However, an appropriate expression is derived in the chapter 3 for rotating walls.*

Following the solution of the Poisson equation, p^{n+1} can be used to update the intermediate velocity field $\tilde{\tilde{\mathbf{u}}}$ by rearranging Eq. 2.2.3.

STEP 3 - Viscous term :

The final step accounts for the implicit treatment of the viscous term:

$$\gamma_0 \left(\frac{\mathbf{u}^{n+1} - \tilde{\tilde{\mathbf{u}}}}{\Delta t}\right) = \frac{1}{\mathrm{Re}}\nabla^2\mathbf{u}^{n+1}, \tag{2.2.6}$$

which can be written as a Helmholtz equation:

$$-\nabla^2\mathbf{u}^{n+1} + \frac{\mathrm{Re}\gamma_0}{\Delta t}\mathbf{u}^{n+1} = \frac{\mathrm{Re}\gamma_0}{\Delta t}\tilde{\tilde{\mathbf{u}}}. \tag{2.2.7}$$

Eq. 2.2.7 is solved subject to the non-slip condition at wall boundaries to develop the velocity field at the end of the time step $\mathbf{u}^{n+1}$. Using Eq. 2.2.3, $\tilde{\tilde{\mathbf{u}}}$ can be eliminated from

Eq. 2.2.7 leading to:

$$-\nabla^2\mathbf{u}^{n+1} + \frac{\mathrm{Re}\gamma_0}{\Delta t}\mathbf{u}^{n+1} = \mathrm{Re}(\frac{\gamma_0}{\Delta t}\tilde{\mathbf{u}} - \nabla p^{n+1}). \quad (2.2.8)$$

It can be seen that adding Eq. 2.2.2, Eq. 2.2.3 and Eq. 2.2.7 (or Eq. 2.2.2 and Eq. 2.2.8) one recovers the original momentum equation discretised in time, Eq. 2.2.1.

Remark: *By taking the divergence of Eq. 2.2.7, it can be seen that the conditions* $\nabla.\tilde{\tilde{\boldsymbol{u}}} = 0$ *and* $\nabla.\boldsymbol{u}^{n+1} = 0$ *are equivalent.*

2.2.1.1 Boundary conditions

Throughout this thesis, external aerodynamic problems are considered and hence three types of boundary condition need to be defined: *inflow*, *outflow* and non-slip *walls*. The associated boundary conditions for the pressure (STEP 2: Poisson equation) and velocity (STEP 3: Helmholtz equation) steps when using the defined scheme are summarised in table 2.1.

Boundary condition	Pressure	Velocity
Inflow	Neumann-Vorticity	Dirichlet
Outflow	Dirichlet	Neumann
Walls	Neumann-Vorticity	Dirichlet

Table 2.1: Boundary condition for the splitting scheme.

2.2.2 High order DG spatial discretisation

The previous temporal scheme requires the spatial solution of non-linear terms (step 1) and of elliptic equations (steps 2 and 3) at each time step. This section summarises the DG spatial discretisation of these terms, including the notation and discontinuous polynomial spaces required to define the Discontinous Galerkin method (omitted details can be found in appendix B). This chapter concentrates on the basic properties of the solver and hence only triangular elements with straight edges are considered. In chapter 3,

other element types: quadrilateral elements with straight and curved edges, are introduced to resolve curved external boundary conditions. In addition, curved elements are necessary to develop the sliding mesh capability presented in the following chapter.

2.2.2.1 Preliminaries: Spaces and traces

Let Ω be a domain in $\mathbb{R}^2$. Let us introduce a triangular tessellation of N^{el} elements: $\Omega_h = \{el^{tri}\}$ of Ω with interior boundaries Γ_h (i.e. edge shared two elements) and external boundaries $\partial\Omega_h$ composed of Dirichlet (Γ_D) and Neumann (Γ_N) type, which do not overlap: $\Gamma_D \cap \Gamma_N = \emptyset$ Each element $el \in \Omega_h$ with boundaries ∂el, has Γ as the positively oriented, i.e. anticlockwise, single edge of the element boundaries ∂el and the normal vector $\mathbf{n}_\Gamma$ as its unit outward pointing normal.

Let us introduce the discontinuous finite element space $D_k(\Omega_h)$ as:

$$D_k(\Omega_h) \quad = \quad \{\upsilon_h \in L^2(\Omega) : \forall el^{tri} \in \Omega_h, \upsilon_h|_{el^{tri}} \in \mathbb{P}_k(el^{tri})\}, \tag{2.2.9}$$

and its vector version $[D_k(\Omega_h)]^2$ where $\mathbb{P}_k(el^{tri})$ denotes the space of polynomials of order less than or equal to k.

Remark: *Only equal order element pairs for velocity and pressure* $[D_k(\Omega_h)]^2_{vel} - [D_k(\Omega_h)]_{press}$ *are considered in this thesis.*

Discontinuous discretisations require definitions for the jump $[[\bullet]]$ and the average $\{\{\bullet\}\}$ across the interface between neighbouring elements el_1 and el_2 (see further details in appendix B):

$$[[\bullet]] = (\bullet|_{el_1}) - (\bullet|_{el_2}),\ \{\{\bullet\}\} = \frac{1}{2}(\bullet|_{el_1}) + \frac{1}{2}(\bullet|_{el_2}),\ \forall\Gamma = \partial el_1 \cap \partial el_2, \tag{2.2.10}$$

and at boundary edges as: $[[\bullet]] = \{\{\bullet\}\} = (\bullet|_{el_1}),\ \forall\Gamma = \partial el_1 \cap \partial\Omega_h$.

2.2.2.2 Spatial discretisation of the non-linear terms

The non-linear terms in the Navier-Stokes equations (i.e. momentum equation Eq. 2.2) may be expressed in various equivalent forms, by imposing the divergence free condition on the velocity $\nabla.\mathbf{u} = 0$ and using the equality:

$$\nabla.(\mathbf{u} \otimes \mathbf{u}) = (\mathbf{u}.\nabla)\mathbf{u} + \mathbf{u}(\nabla.\mathbf{u}) = (\mathbf{u}.\nabla)\mathbf{u}, \tag{2.2.11}$$

where $\mathbf{u} \otimes \mathbf{v} := u_i v_j, \ i, j = 1, .., d$ and d indicates the spatial dimension. The most common formulations are summarised here:

- **Convective form**: $\mathbf{N}(\mathbf{u}) = (\mathbf{u}.\nabla)\mathbf{u}$
- **Conservative or divergence form**: $\mathbf{N}(\mathbf{u}) = \nabla.(\mathbf{u} \otimes \mathbf{u})$
- **Classic skew symmetric form**: $\mathbf{N}(\mathbf{u}) = 1/2[(\mathbf{u}.\nabla)\mathbf{u} + \nabla.(\mathbf{u} \otimes \mathbf{u})]$

An alternative skew symmetric form is proposed and can be derived by combining the classic skew symmetric with Eq. 2.2.11:

- **Proposed skew symmetric form**: $\mathbf{N}(\mathbf{u}) = (\mathbf{u}.\nabla)\mathbf{u} + 1/2\,(\nabla.\mathbf{u})\,\mathbf{u}$

This last form was first introduced by Temam [164] and has the advantage over the classical skew symmetric form is that fewer derivatives need to be evaluated.

In the continuous case, all forms are equivalent, however in the discrete case this is not necessarily true (see [101] for details). The classical skew-symmetric form has shown the advantage of minimising aliasing errors [100], [101]. It is expected that the proposed skew symmetric form also minimises aliasing errors to some extent. However, it was shown in [106] and [105] for conformal h/p Spectral discretisations that over-integration minimises aliasing errors. Since over-integration is the approach retained in this work (see section 2.2.4), the particular formulation selected to express the non-linear terms is expected to have a minimal effect. In addition, let us note that the proposed skew symmetric form [164] does not influence the energy balance when non divergence free velocities (but only

weakly divergence free solutions) are considered. Further details can also be found in the recent monograph on DG methods by Di Pietro and Ern [139].

The following formulation (under its weak setting) is hence proposed: find $\mathbf{u}_h \in [D_k(\Omega_h)]^2$ such that:

$$\begin{aligned}\int_{\Omega_h} (\mathbf{N}(\mathbf{u}_h)) .\boldsymbol{v}_h d\mathbf{x} &= \sum_{el\in\Omega_h} \int_{el} ((\mathbf{u}_h.\nabla)\mathbf{u}_h) .\boldsymbol{v}_h d\mathbf{x} \\ &\quad +1/2 \sum_{el\in\Omega_h} \int_{el} ((\nabla.\mathbf{u}_h)\mathbf{u}_h) .\boldsymbol{v}_h d\mathbf{x}, \;\; \forall \boldsymbol{v}_h \in [D_k(\Omega_h)]^2.\end{aligned} \tag{2.2.12}$$

Previous developments of DG methods to solve the incompressible NS equations have required the use of numerical fluxes for the discretisation of non-linear terms . Examples can be found in [156] and [77] where Lax-Friederich DG variants were used or in [146] where a DG modification of the Lesaint-Raviart approximation was incorporated.

The Lesaint-Raviart formulation modified for DG methods [60], [146], where upwinding is performed to account for convection dominated flows reads: find $\mathbf{u}_h \in D_k^2(\Omega_h)$ such that:

$$\begin{aligned}\int_{\Omega_h} (\mathbf{N}(\mathbf{u}_h)) .\boldsymbol{v}_h d\mathbf{x} &= \sum_{el\in\Omega_h} \int_{el} (\mathbf{u}_h.\nabla\mathbf{u}_h).\boldsymbol{v}_h d\mathbf{x} + 1/2 \sum_{el\in\Omega_h} \int_{el} ((\nabla.\mathbf{u}_h)\mathbf{u}_h) .\boldsymbol{v}_h d\mathbf{x} \\ &\quad + \sum_{el\in\Omega_h} \int_{\partial el^-} \left(|\{\{\mathbf{u}_h\}\}.\mathbf{n}_\Gamma|(\mathbf{u}_h^{int} - \mathbf{u}_h^{ext})\right) .\boldsymbol{v}_h^{int} ds \\ &\quad -1/2 \sum_{\Gamma\in\Gamma_h\cup\Gamma_D} \int_\Gamma [[\mathbf{u}_h]].\mathbf{n}_\Gamma\{\{\mathbf{u}_h.\boldsymbol{v}_h\}\} ds, \;\; \forall \boldsymbol{v}_h \in [D_k(\Omega_h)]^2.\end{aligned} \tag{2.2.13}$$

The third term on the right hand side is the upwinding term, required for convection dominated flows. The quantity $\mathbf{u}_h^{int} - \mathbf{u}_h^{ext}$ represents the jump in the direction of the flow and $\boldsymbol{v}_h^{int}$ is the test function on the side of *el* coming from the interior [146]. ∂el^- defines the portion of the boundary where the flow enters the element (i.e. $\{\{\mathbf{u}_h\}\}.\mathbf{n}_\Gamma < 0$). It can be seen that for vanishing jumps, i.e. continuous approximation, Eq. 2.2.13 reduces to the proposed skew-symmetric form Eq. 2.2.12.

This DG Lesaint-Raviart formulation has been implemented and tested by the author and results were reported in [52]. However, further examination and testing showed that almost identical results could be obtained without the use of flux enforcement and with the formulation detailed in Eq. 2.2.12, when high order polynomials were used.

Both formulations differ in the flux enforcement terms (surface integral in Eq. 2.2.13). In the formulation defined by Eq. 2.2.12, no upwinding or flux enforcement is present and only volume terms are retained, which reduces the computational cost of the evaluation of the non-linear terms. The selection of the simpler expression is closely related to the selected time advancement procedure and has a threefold justification:

1. Only continuous solutions, where viscosity dominates and there is lack of hyperbolicity (e.g. convection dominated flows, shocks), are of interest in this work.
2. Within the splitting scheme used, the non-linear terms act as forcing terms in the Poisson and Helmholtz equations (Eq. 2.2.4 and 2.2.8). If surface integrals (e.g. upwinding) with jumps and averages are added to the non-linear terms (as in Eq. 2.2.13), these should have a small contribution when compared to the volume terms. Furthermore, as the polynomial order increases the numerical solution tends to be continuous (for a viscosity dominant flow) hence reducing the contribution of the jumps (assuming a consistent and conservative formulation for the numerical flux, see appendix B for definitions). Overall, the forcing terms will effectively be unchanged and so will the solution.
3. From the stability point of view of the temporal scheme, positivity is not required since the non-linear terms are treated explicitly.

Remark: *As a result of the previous discussion, the non-linear terms are computed using Eq. 2.2.12 for all laminar cases included in this thesis. However, it has proven useful to replace this expression by the DG Lesaint-Raviart formulation (Eq. 2.2.13) to enhance stability when computing turbulent flows. This formulation leads to the ILES-DG formulation detailed in chapter 5 section 5.2.2.*

Finally, let us note that the explicit treatment of the non-linear terms introduces a Courant-Friedrichs-Lewy (CFL) type restriction on the time step. The CFL estimate for high order spatial methods leads to $\triangle t < \triangle t_{CFL} = \mathcal{O}(h/U_{max}k^2)$ as shown in [101] for an advection model problem, where h is the mesh element size, U_{max} the maximum velocity and k represents the polynomial order.

2.2.2.3 The DG-SIPG spatial discretisation for elliptic equations

The described temporal splitting method requires the solution of purely elliptic equations (Poisson and Helmholtz). Among the various existing IP methods (see appendix B and C), the Symmetric Interior Penalty Galerkin (SIPG) method is retained to discretise these equations spatially. The details of the derivation are omitted here, but can be found in appendix B. Comparison of various IP methods to solve elliptic equations are included in appendix C.

The general formulation for the continuous elliptic problem (Laplace, Poisson and Helmholtz) reads:

$$\begin{aligned} -\triangle u + \alpha u &= g \quad \text{in } \Omega, \\ u &= L_D \quad \text{on } \Gamma_D, \\ \nabla u.\mathbf{n} &= L_N \quad \text{on } \Gamma_N, \end{aligned} \tag{2.2.14}$$

where $u \in H^1(\Omega)$ is a scalar solution (but extension to the vector formulation is direct), $\sqrt{\alpha}$ is real and represents the wave number for the Helmholtz equation, $\mathbf{n}$ is the outward pointing unit normal vector on $\partial\Omega$, $g \in L^2(\Omega)$ is the forcing term, and $L_D \in H^{1/2}(\Gamma_D)$ and $L_N \in L^2(\Gamma_N)$ represent Dirichlet and Neumann boundary conditions respectively. Eq. 2.2.14 defines the Laplace, Poisson and Helmholtz equations as follows:

- **Laplace**: forcing term $g = 0$ and convective term $\alpha = 0$,
- **Poisson**: forcing term $g \neq 0$ and convective term $\alpha = 0$,

- **Helmholtz**: forcing term $g \neq 0$ and convective term $\alpha \neq 0$.

To find the discrete weak solution associated with the continuous problem represented by Eq. 2.2.14 using the DG-SIPG method, the bilinear $a(\bullet, \bullet)$ and linear $\ell(\bullet)$ forms, associated with the finite element discrete variational formulation arising from the weak form of the continuous problem, are introduced (see appendix B for derivation). The bilinear form is defined as:

$$\begin{aligned} a_{(\alpha,\sigma)}(u_h, v_h) &= \sum_{el\in\Omega_h} \int_{el} (\nabla u_h . \nabla v_h) d\mathbf{x} + \sum_{el\in\Omega_h} \int_{el} (\alpha u_h v_h) d\mathbf{x} \\ &\quad - \sum_{\Gamma\in\Gamma_h\cup\Gamma_D} \int_{\Gamma} \{\{\nabla u_h\}\}.\mathbf{n}_\Gamma [[v_h]] ds - \sum_{\Gamma\in\Gamma_h\cup\Gamma_D} \int_{\Gamma} \{\{\nabla v_h\}\}.\mathbf{n}_\Gamma [[u_h]] ds \\ &\quad + \sum_{\Gamma\in\Gamma_h\cup\Gamma_D} \frac{\sigma}{|trace|^\beta} \int_{\Gamma} [[u_h]][[v_h]] ds, \quad \forall v_h \in D_k(\Omega_h), \end{aligned} \tag{2.2.15}$$

with the corresponding linear form:

$$\begin{aligned} \ell_{(g,\sigma)}(v_h) &= \sum_{el\in\Omega_h} \int_{el} g v_h d\mathbf{x} - \sum_{\Gamma\in\Gamma_D} \int_{\Gamma} (\nabla v_h . \mathbf{n}_\Gamma - \frac{\sigma}{|trace|^\beta} v_h) L_D ds \\ &\quad + \sum_{\Gamma\in\Gamma_N} \int_{\Gamma} v_h L_N ds, \quad \forall v_h \in D_k(\Omega_h), \end{aligned} \tag{2.2.16}$$

where σ and β are positive real parameters and $|trace|$ is defined as the length of the element edge in 2D. Then, the DG-SIPG approximate solution reduces to finding $u_h \in D_k(\Omega_h)$ provided by:

$$a_{(\alpha,\sigma)}(u_h, v_h) = \ell_{(g,\sigma)}(v_h), \quad \forall v_h \in D_k(\Omega_h). \tag{2.2.17}$$

The bilinear form associated with the DG-SIPG method is symmetric leading to a positive definite matrix, which presents advantages in the choice of the iterative solver that may be used. The combination of σ and β in Eq. 2.2.15 and Eq. 2.2.16, defines the penalty parameter and has to be chosen to be large enough to enforce coercivity of the bilinear

form [146]. The DG-SIPG method provides optimal convergence in the L^2 norm, i.e. $\mathcal{O}(h^{k+1})$, for h-refinement independently of the penalisation used as long as this is large enough. A detailed comparison of the effect of the penalisation parameter on the L^2 norm of the error for elliptic problems using the DG-SIPG and other IP variants can be found in appendix C.

The determination of the analytical expression for the lower bound (LB) of the penalty parameter for the DG-SIPG method to be stable and convergent was obtained in [48] for triangular elements and is revisited here. Let us consider el_1 and el_2, two triangular elements sharing the edge Γ, then the lower bound for the penalisation for triangular elements reads:

$$\sigma^{LB} = \frac{3}{2}|trace|^{\beta-1}\left(k^{el_1}(k^{el_1}+1)\cot\theta^{el_1} + k^{el_2}(k^{el_2}+1)\cot\theta^{el_2}\right), \quad (2.2.18)$$

where for the element el, k^{el} represents the polynomial order and θ^{el} is the smallest angle in the element. If one assumes equal polynomial orders in all elements $k^{el_1} = k^{el_2} = k$ and regularity in the mesh (i.e. non-distorted elements) $\cot\theta^{el_1} \approx \cot\theta^{el_2} \approx 1$ (e.g. $\cot 45°$ for right angle triangles), the previous expression reduces to:

$$\sigma^{LB} = 3k\,(k+1)\,|trace|^{\beta-1}. \quad (2.2.19)$$

Replacing $\sigma = \sigma^{LB}$ one obtains the following expression for the penalty term: $\frac{\sigma}{|trace|^{\beta}} = \frac{\sigma^{LB}}{|trace|^{\beta}} = \frac{3k(k+1)}{|trace|}$, which reduces to setting $\sigma = 3k(k+1)$ and $\beta = 1$ in the original scheme. It can also be shown that at Dirichlet boundaries $\sigma^{LB}_{\Gamma_D} = 2\sigma^{LB}$. These approximations for the penalty parameter will be used hereafter.

Remark: *In chapter 3, quadrilateral elements are introduced. The penalty parameter for this type of element is also defined as $\sigma = 3k(k+1)$, which is a conservative estimate for quadrilaterals since generally the expression $\sigma \approx k^2$ is used.*

To conclude this section and to underline some characteristics of the presented DG-SIPG method (when compared to continuous conformal discretisations); let us examine the effect of the penalisation σ on the numerical solution and the effect of the weak imposition of boundary conditions for a Poisson problem (see definition in appendix C section C.2.1). Fig. 2.1 shows a discontinuous DG-SIPG approximation and the effect of increasing the penalty parameter on the solution continuity. It can be seen that a low penalty enables discontinuities in the solution. An increase in the penalty parameter tends to enforce, to some extent, continuity in the numerical solution. It is interesting to note that this effect does not relate to the accuracy of the solution (in the L^2 norm). In particular, the DG-SIPG method shows some degradation in the L^2 accuracy as the penalty parameter increases (see appendix C). This shows that a discontinuous solution may approximate a continuous problem more accurately than a continuous solution.

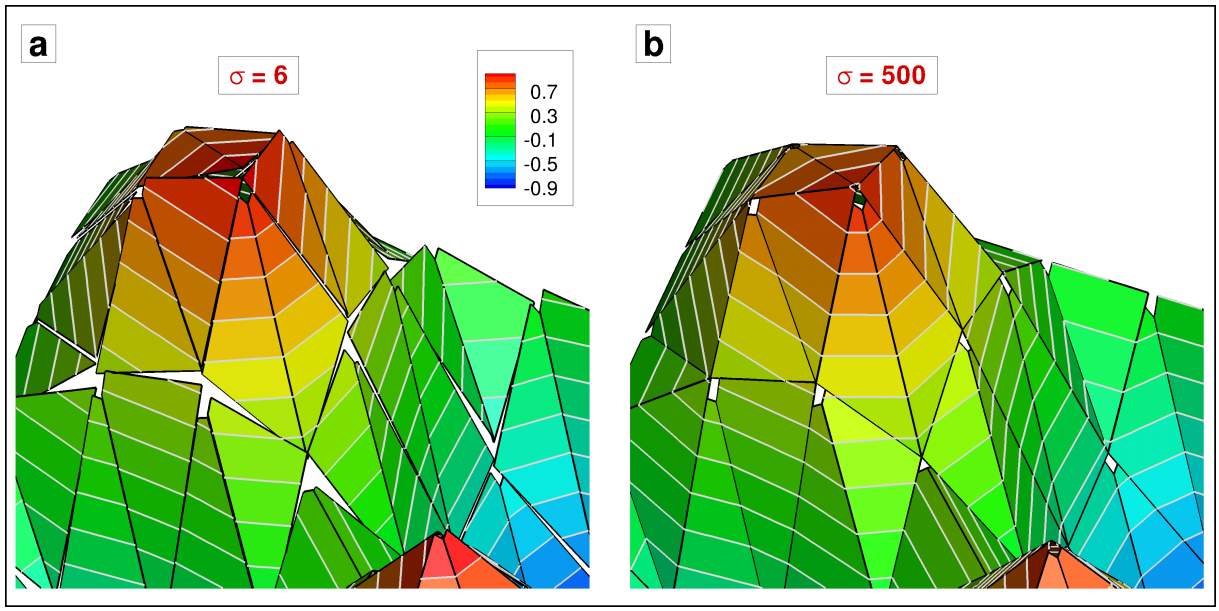

Figure 2.1: Effect of the penalty parameter on the discontinuous solution for a Poisson problem solved with a polynomial order $k = 1$ and penalty parameters (a) $\sigma = 6$ and (b) $\sigma = 500$.

With the penalty fixed, exploration of the weak imposition of boundary conditions is performed on a mesh consisting of two triangular elements and various polynomial orders. Fig. 2.2 shows that weak imposition does not satisfy the boundary condition

in a pointwise manner but only in a weak or integral sense. It can be seen that as the polynomial order k is increased, the weak imposition resembles the strong imposition, satisfying the boundary condition pointwise.

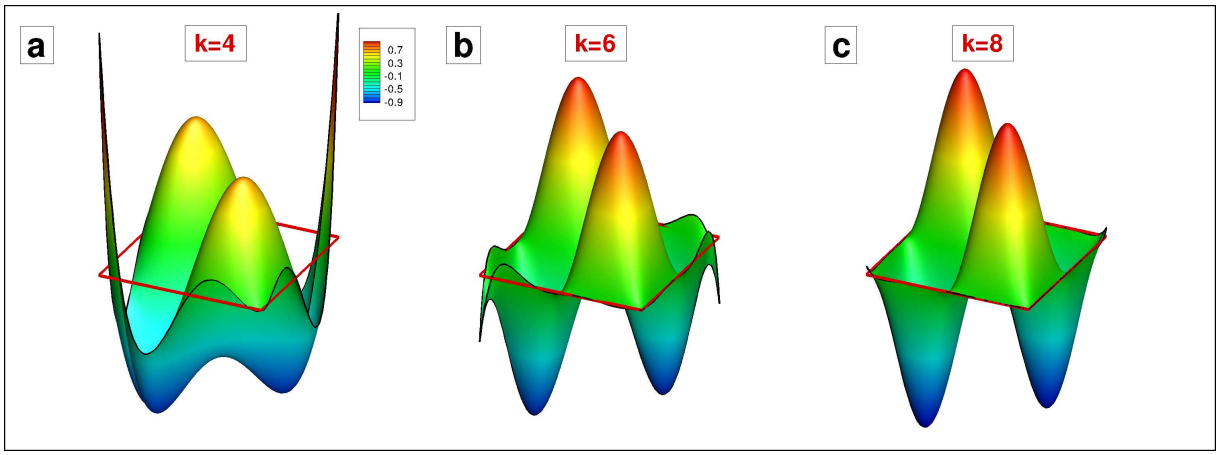

Figure 2.2: DG solution and weak imposition of boundary conditions for a Poisson problem solved using two elements and polynomial order (a) $k = 4$, (b) $k = 6$ and (c) $k = 8$.

2.2.2.4 Mapping from computational to physical space: triangular elements

Three types of elements are considered in this thesis: triangular and quadrilateral elements with straight edges and quadrilateral elements with one curved edge; the first being treated in this section and last two in chapter 3. However, a brief discussion is provided here concerning the rationale behind the use of different types of elements.

On the one hand, and as will be subsequently detailed, the use of triangular elements has the major advantage of having a constant determinant of the Jacobian, which reduces notably the computational cost of evaluating volume integrals. Furthermore, through the appropriate selection of orthonormal basis functions, the mass matrix reduces to the identity matrix, which significantly reduces the cost of all operations involving this matrix. However, if curved edges are considered, then this appealing property of the determinant of the Jacobian is lost.

On the other hand, quadrilateral elements do not generally have a constant determinant of the Jacobian (see appendix A for a special case) and hence curving an edge does not

increase the costs of evaluating volume integrals (e.g. the mass matrix). Curving an edge of a quadrilateral element is preferred, over curving an edge of a triangular element, since its Jacobian is less likely to become ill conditioned (i.e. degenerated element) as a result of the larger angles between consecutive edges. In addition, quadrilateral elements may be stretched near boundaries (i.e. high aspect ratio elements), which proves useful to reduce the overall cell count in the computational mesh. For these reasons, chapter 3 introduces quadrilateral elements with straight and curved edges to account for curved boundary conditions and sliding mesh interfaces. In this section, only straight sided triangular elements are considered.

This section introduces the necessary mapping to relate computational to physical space for triangular elements. Mappings for straight and curved sided quadrilateral elements can be found in chapter 3 section 3.2.2.3 and further details in appendix A.

The non-linear terms (Eq. 2.2.12) and the DG-SIPG weak form (Eq. 2.2.15 and Eq. 2.2.16) require computation of volume and surface integrals. To perform these integrations, it is advantageous to consider mappings Ψ from the reference element I_{el} in computational space $\boldsymbol{\xi} = (\xi_1, \xi_2)$ to the element el in physical space $\mathbf{x} = (x, y) \in el$ as $I_{el} := \Psi^{-1}(el)$ with the associated determinant of the Jacobian $|\mathbf{J}_\Psi|$ for the transformation. In addition, it is necessary to introduce the transformation of the edges Γ of element el in physical space to the mapped edge in the computational space: $\Gamma = \boldsymbol{C}([-1, 1])$ with the norm of the differential parametrisation (i.e. differential length) of $\boldsymbol{C}(t)$ defined as $|\mathbf{J}_{\boldsymbol{C}}(t)|$. Using these definitions, one can equate the volume and surface integrals for a general polynomial function $f(\mathbf{x})$ defined in physical space to their equivalent form in computational space:

$$\int_{el} f(\mathbf{x}) d\mathbf{x} = \int_{I_{el}} f(\boldsymbol{\xi}) |\mathbf{J}_\Psi| d\boldsymbol{\xi}, \tag{2.2.20}$$

$$\int_{\Gamma} f(\mathbf{x}) d\mathbf{s} = \int_{-1}^{1} f(\boldsymbol{C}(t)) |\mathbf{J}_{\boldsymbol{C}}(t)| dt. \tag{2.2.21}$$

Fig. 2.3 defines the mapping Ψ for straight edged triangular elements (further details of this mapping can be found in appendix A and in [101]) as a combination of two transformations:

- **1a and 1b - Triangular map:** (1a) standard square (ξ_1, ξ_2) to standard triangle (η_1, η_2) and (1b) standard triangle to arbitrary triangle (x, y).

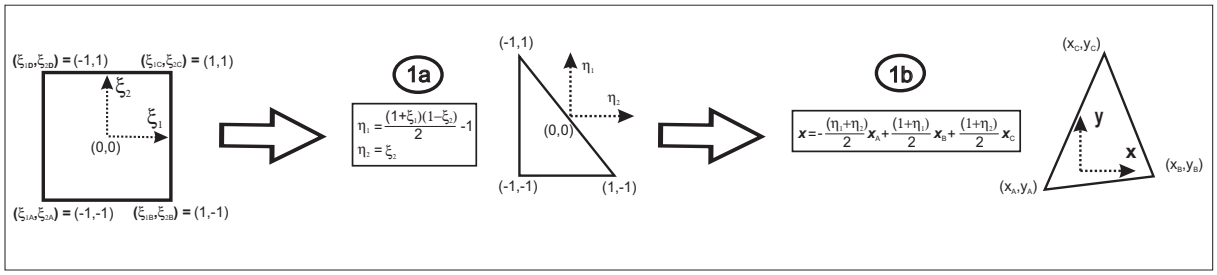

Figure 2.3: Mapping from computational space to physical space for straight sided triangular elements.

In addition to the determinant of the Jacobian $|\mathbf{J}_\Psi|$ and the norm of the differential length $|\mathbf{J}_{\boldsymbol{C}}(t)|$ required to evaluate volume and surface integrals (Eq. 2.2.20 and Eq. 2.2.21), partial derivatives from physical to computational space (i.e. gradients) and edge normals are also required to compute the non-linear terms Eq. 2.2.13 and the DG-SIPG weak form Eq. 2.2.15 and Eq. 2.2.16. The mapping defined in Fig. 2.3 allows for the computation of these metrics (see appendix A for details). In particular, it can be seen that this mapping leads to a constant determinant of the Jacobian $|\mathbf{J}_\Psi|$.

The differential length for triangular elements with straight edges can be conveniently computed using the chain rule for differentiation as:

$$|\mathbf{J}_{\boldsymbol{C}}(t)| = \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}\, dt = \frac{|trace|}{2}, \tag{2.2.22}$$

where $|trace|$ represents the length of the element edge.

2.2.2.5 Modal basis functions for triangular elements

Within each element el, the approximate solution u_h is expanded using modal basis functions $\phi_i(\mathbf{x})$ as:

$$u_h(\mathbf{x}) = \sum_{i=1}^{LDOF} \widehat{u}_i \phi_i(\mathbf{x}), \tag{2.2.23}$$

where $\mathbf{x} = (x, y) \in el$, $LDOF$ is the Local number of Degrees Of Freedom and $\widehat{u}_i$ represent the modal amplitudes associated to the basis functions $\phi_i(\mathbf{x})$. Hierarchical tensorial and orthonormal (with respect to the L^2-inner product) expansion bases (see [101] for details) are selected. Following the notation introduced in Fig. 2.3, these bases can be defined in both physical and computational space as:

$$\phi_i^{tri}(\mathbf{x}) = \phi_{pq}^{tri}(\mathbf{x}) = \phi_{pq}^{tri}(\eta_1, \eta_2) = \varphi_p^{tri\,A}(\xi_1)\varphi_{pq}^{tri\,B}(\xi_2), \tag{2.2.24}$$

where p, q denote the different components of the tensorial expansion. Let us note that the basis functions $\phi(\mathbf{x})$ defined in physical space can be expressed in terms of principal basis functions $\varphi(\xi_1)$ and $\varphi(\xi_2)$ defined in the unit square:

$$\begin{aligned} \varphi_p^{tri\,A}(\xi_1) &= \sqrt{\frac{2p+1}{2}} P_p^{(0,0)}(\xi_1), \\ \varphi_{pq}^{tri\,B}(\xi_2) &= \sqrt{p+q+1} \left(\frac{1-\xi_2}{2}\right)^p P_q^{(2p+1,0)}(\xi_2), \end{aligned} \tag{2.2.25}$$

where $P_p^{(\alpha,\beta)}(z)$ is the pth order Jacobi polynomial (and $\alpha = \beta = 0$ corresponds to the Legendre polynomials). For triangular elements $0 \leq p, q$ and $p + q \leq k$ and $LDOF_{tri} = (k+1)(k+2)/2$. These hierarchical expansion bases are depicted on the standard triangle in Fig. 2.4.

Details on the derivation of well conditioned triangular basis functions can be found in [101], where it is detailed that the term $(\frac{1-\xi_2}{2})^p$ is necessary to maintain the expansion as polynomials in terms of cartesian coordinates and that the use of $\alpha = 2p + 1$ helps

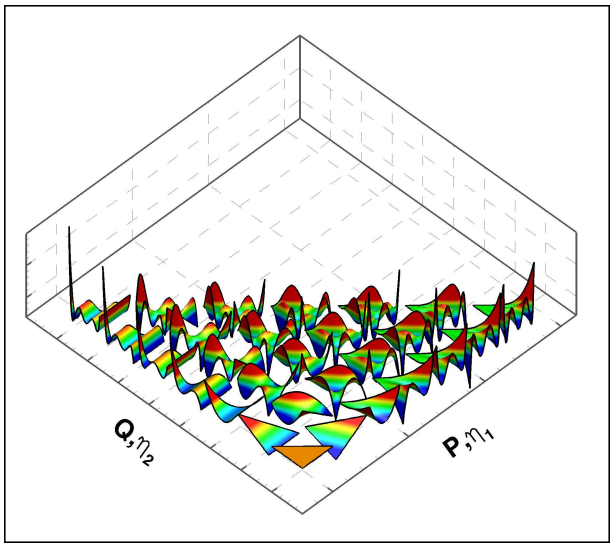

Figure 2.4: Hierarchical modal basis functions for various polynomial $k = 0$ to 6 for triangular elements.

to decluster zeros away from $z = +1$ (i.e. the collapsed vertex). Declustering helps in maintaining the linear independence of the expansion leading to a well conditioned numerical system [101].

The reader may have noted that these bases have been normalised providing a mass matrix equal to the identity matrix (i.e. $\mathbf{M} = |\mathbf{J}_\Psi|\mathbf{Id}$, see section 2.2.3 for definitions) since these elements have a constant determinant of the Jacobian. There is a clear advantage in obtaining a diagonal mass matrix, when the determinant of the Jacobian is constant, since all matrix operations involving this matrix become trivial (e.g. $\mathbf{Mx} = |\mathbf{J}_\Psi|\mathbf{xId}$).

The derivatives of the bases can be calculated using the generalised expression for the derivatives of the Jacobi polynomials: $\frac{dP_n^{\alpha,\beta}}{dz}(z) = \frac{1}{2}(n+\alpha+\beta+1)P_{n-1}^{\alpha+1,\beta+1}(z)$. Using this relationship, the derivatives of the principal basis functions can be obtained:

$$\frac{d\varphi_p^{triA}}{d\xi_1}(\xi_1) = \frac{1}{2}(p+1)P_{p-1}^{(1,1)}(\xi_1), \tag{2.2.26}$$

$$\frac{d\varphi_{pq}^{triB}}{d\xi_2}(\xi_2) = \left[\frac{1-\xi_2}{2}\right]^{(p-1)}\left[-\frac{1}{2}pP_q^{(2p+1,0)}(\xi_2) + \left(\frac{1}{4}(q+2p+2)(1-\xi_2)P_{q-1}^{(2p+2,1)}(\xi_2)\right)\right].$$

2.2.3 Summary of the method and matrix formulation

This section details the time advancement algorithm introduced in section 2.2.1 in matrix form, since this should prove useful for further analysis (see section 2.5.3.2). To define the algebraic time advancement, let us define the necessary local (element based) and global matrices.

Firstly, the local or element based matrices (i.e. corresponding to element el) of size $LDOF \times LDOF$ are introduced. Setting $i, j = 1, ..., LDOF$ as the indexes corresponding to the test function v_h and the basis function index of u_h respectively, the local Mass, Derivative and Laplacian matrices can be defined.

- **Local Mass matrix**: $\mathbf{M^{el}} := \{M^{el}{}_{ij} = \int_{el} \phi_i(\mathbf{x})\phi_j(\mathbf{x})d\mathbf{x}\}$,
- **Local Derivative matrix**: $\mathbf{K_r^{el}} := \{K_r^{el}{}_{ij} = \int_{el} \phi_i(\mathbf{x})\partial_r\phi_j(\mathbf{x})d\mathbf{x}\}$, where $r = x$ or y,
- **Local partial Laplacian or partial Stiffness matrix**: $\mathbf{L_r^{el}} := \{L_r^{el}{}_{ij} = \int_{el} \partial_r\phi_i(\mathbf{x})\partial_r\phi_j(\mathbf{x})d\mathbf{x}\}$. To obtain the **Local Laplacian** it suffices to perform summation over $r = x, y$: $\mathbf{L^{el}} = \mathbf{L_x^{el}} + \mathbf{L_y^{el}}$.

Secondly, global matrices of size $N^{el}LDOF \times N^{el}LDOF$, where N^{el} is the number of elements in Ω_h, are defined as the direct sum of the local matrices previously defined:

- **Global Mass matrix**: $\mathbf{M} = \sum_{el\in\Omega_h} \mathbf{M^{el}}$,
- **Global Derivative matrix**: $\mathbf{K_r} = \sum_{el\in\Omega_h} \mathbf{K_r^{el}}$, where $r = x$ or y,
- **Global partial Laplacian or partial Stiffness matrix**: $\mathbf{L_r} = \sum_{el\in\Omega_h} \mathbf{L_r^{el}}$, and performing summation over $r = x, y$, one obtains the **Global Laplacian L** $= \mathbf{L_x} + \mathbf{L_y} = \sum_{el\in\Omega_h}(\mathbf{L_x^{el}} + \mathbf{L_y^{el}})$.

In addition, the **Global Flux matrix** is defined as:
$\mathbf{T} := \{\sum_{\Gamma\in\Gamma_h\cup\Gamma_D} \int_\Gamma t_{ij}^\Gamma ds = \sum_{\Gamma\in\Gamma_h\cup\Gamma_D} \int_\Gamma t_{ij}^{el_\ell el_\chi} ds\}$, where ℓ, χ denote the elements sharing the edge Γ, resulting from the last three terms in the bilinear form Eq. 2.2.15 that involve surface integrals, which are obtained by expanding the average and jump terms

(i.e. $\{\{\bullet\}\}$, $[[\bullet]]$) and replacing the approximate solution by its expansion Eq. 2.2.23. These contributions can be recasted as:

$$\int_\Gamma t_{ij}^{el_\ell el_\chi} ds = \gamma \frac{1}{2} \int_\Gamma \nabla \phi_j^{el_\chi} . \mathbf{n}_\Gamma \phi_i^{el_\ell} ds + \delta \frac{1}{2} \int_\Gamma \phi_j^{el_\chi} \nabla \phi_i^{el_\ell} . \mathbf{n}_\Gamma ds + \rho \frac{\sigma}{|trace|} \int_\Gamma \phi_j^{el_\chi} \phi_i^{el_\ell} ds, \tag{2.2.27}$$

where γ, δ and ρ are defined as follows:

$$\gamma = \begin{cases} 1; & \ell \neq 1 \\ -1; & \ell = 1 \end{cases}, \quad \delta = \begin{cases} 1; & \chi \neq 1 \\ -1; & \chi = 1 \end{cases}, \quad \rho = \begin{cases} 1; & \ell = \chi \\ -1; & \ell \neq \chi \end{cases}. \tag{2.2.28}$$

If only two unique elements el_1 and el_2 share the face Γ, then $\ell, \chi = 1, 2$ and four flux contributions are retrieved: $\int_\Gamma t_{ij}^\Gamma ds = \int_\Gamma (t_{ij}^{11} + t_{ij}^{22} + t_{ij}^{12} + t_{ij}^{21}) ds$.

Remark: *Note that in chapter 3, this definition is revisited to account for three elements sharing the face* Γ *(i.e. one hanging node) which is required for the implementation of the sliding mesh technique.*

The bilinear and linear forms for general elliptic problems Eq. 2.2.15 and Eq. 2.2.16 produce the global matrix $\mathbf{A}_{(\alpha,\sigma)}$ and the right hand side vector $\mathbf{RHS}_{(g,\sigma)}$ respectively:

- **Global DG Bilinear matrix**: $\mathbf{A}_{(\alpha,\sigma)} := a_{(\alpha,\sigma)}(\phi_j, \phi_i)$,
- **Global DG Linear Right Hand Side vector**: $\mathbf{RHS}_{(g,\sigma)} := \ell(\phi_i)$, where the explicit dependency of the bilinear form on α and g, the square of the wave number and the the forcing term respectively, for a general Helmholtz equation and on σ, the penalty parameter, have been outlined.

In addition, the DG bilinear matrix can be defined in terms of global Mass, Laplacian and Flux matrices as:

$$\mathbf{A}_{(\alpha,\sigma)} = \mathbf{L} + \alpha \mathbf{M} + \mathbf{T}. \tag{2.2.29}$$

The algebraic system: $\mathbf{A}_{(\alpha,\sigma)}\mathbf{X} = \mathbf{RHS}_{(g,\sigma)}$, where $\mathbf{X}$ represents the vector of global

unknowns (i.e. $N^{el}LDOF \times 1$) is the algebraic equivalent of Eq. 2.2.17.

Remark: *The Global Mass (**M**), Derivative (**K**) and Laplacian (**L**) matrices are block diagonal matrices. The Global Flux matrix (**T**) is responsible for coupling the degrees of freedom from neighbour elements. Weak imposition of Dirichlet boundary conditions translates into a block diagonal contribution from the element on Γ_D through the Global Flux matrix and a contribution to the right hand side.*

Finally, the explicitly treated **Non-linear terms** give rise to: $\mathbf{N_r}(U) := \sum_{el \in \Omega_h} \int_{el} \mathbf{N_r}^{el}(\mathbf{u_h}) \boldsymbol{\phi}_i d\mathbf{x}$, where $r = x$ or y.

Using these definitions for the global matrices, it is possible to rewrite the temporal discretisation scheme in matrix form with $\tilde{U}_r$, U_r and P vectors of size $N^{el}LDOF \times 1$ with $r = x, y$ (no-summation in Eq. 2.2.30 and Eq. 2.2.32, but summation over r in Eq. 2.2.31):

$$\mathbf{M}\frac{\gamma_0 \tilde{U}_r - \alpha_0 {U_r}^n - \alpha_1 {U_r}^{n-1}}{\Delta t} = -\beta_0 \mathbf{N_r}(U^n) - \beta_1 \mathbf{N_r}(U^{n-1}), \tag{2.2.30}$$

$$\mathbf{A}^{Pois}_{(\alpha=0,\sigma)} P^{n+1} = -\frac{\gamma_0}{\Delta t}\mathbf{K_r}\tilde{U}_r + \mathbf{RHS}^{Pois}_{(g=0,\sigma)}, \tag{2.2.31}$$

$$\mathbf{A}^{Helm}_{(\alpha=\frac{\mathrm{Re}\gamma_0}{\Delta t},\sigma)} {U_r}^{n+1} = \mathrm{Re}\left(\frac{\gamma_0}{\Delta t}\mathbf{M}\tilde{U}_r - {\mathbf{K_r}}^T P^{n+1}\right) + \mathbf{RHS}^{Helm}_{(g=0,\sigma)}. \tag{2.2.32}$$

It is easy to see that:

$$\mathbf{A}^{Helm}_{(\alpha=\frac{\mathrm{Re}\gamma_0}{\Delta t},\sigma)} = \mathbf{A}^{Helm}_{(\alpha=0,\sigma)} + \frac{\mathrm{Re}\gamma_0}{\Delta t}\mathbf{M}. \tag{2.2.33}$$

To conclude this section, let us show that to recover the weak form of the momentum equation in matrix form, it suffices to combine Eq. 2.2.30, Eq. 2.2.32 and Eq. 2.2.33:

$$\begin{aligned}\mathbf{M}\frac{(\gamma_0 {U_r}^{n+1} - \alpha_0 {U_r}^n - \alpha_1 {U_r}^{n-1})}{\Delta t} = &- {\mathbf{K_r}}^T P^{n+1} - \beta_0 \mathbf{N_r}(U^n) - \beta_1 \mathbf{N_r}(U^{n-1}) \\ &- \mathbf{A}^{Helm}_{(\alpha=0,\sigma)} + \mathbf{RHS}^{Helm}_{(g=0,\sigma)}.\end{aligned} \tag{2.2.34}$$

2.2.4 Numerical quadrature

To numerically evaluate the volume and surface integrals in computational space (i.e. Eq. 2.2.20 and Eq. 2.2.21) and hence the matrices defined in section 2.2.3, it is necessary to use quadrature.

One-dimensional quadrature is needed to compute surface integrals of a generic polynomial function $f(\xi_1)$:

$$\int_{-1}^{1} f(\xi_1)d\xi_1 \approx \sum_{s=0}^{Q1-1} w_s f(\xi_{1s}), \tag{2.2.35}$$

where $Q1$ is the number of quadrature points in the ξ_1 direction and ξ_{1s}, w_s represent the quadrature points and weights respectively.

Two-dimensional quadrature is used to compute volume integrals of a generic polynomial function $f(\xi_1, \xi_2)$ in the reference element I_{el} (i.e. the unit square, see Fig. 2.3):

$$\int_{-1}^{1}\int_{-1}^{1} f(\xi_1, \xi_2)d\xi_1 d\xi_2 \approx \sum_{s=0}^{Q1-1} w_s \left\{ \sum_{ss=0}^{Q2-1} w_{ss} f(\xi_{1s}, \xi_{2ss}) \right\}, \tag{2.2.36}$$

where $Q1$ and $Q2$ are the number of quadrature points in the ξ_1 and ξ_2 directions and ξ_{1s},w_s and ξ_{2ss},w_{ss} the respective quadrature points and weights.

If the generic polynomial function $f(\xi_1, \xi_2)$ can be written as a product of tensorial functions $f(\xi_1, \xi_2) = g(\xi_1)h(\xi_2)$ (e.g. principal basis functions), then Eq. 2.2.36 simplifies to:

$$\int_{-1}^{1}\int_{-1}^{1} f(\xi_1, \xi_2)d\xi_1 d\xi_2 \approx \left\{ \sum_{s=1}^{Q1} w_s g(\xi_{1s}) \right\} \left\{ \sum_{ss=1}^{Q2} w_{ss} h(\xi_{2ss}) \right\}. \tag{2.2.37}$$

This technique is known as the sum-factorisation technique [101] and has proven to be very efficient for the evaluation of volume integrals.

Remark: *This technique can only be applied for elements with constant determinant of the Jacobian as defined for the straight sided triangular elements.*

Gauss-Legendre (GL) quadrature (i.e. interval ends are not included) is preferred and retained throughout this thesis since fewer quadrature points are required when compared

to other quadrature rules (e.g. Gauss-Radau which includes a point at -1, Gauss-Lobatto includes both end points). However, a combination of Gauss-Legendre-Gauss-Lobatto quadrature is used for post processing.

Remark: *To compute the flux terms (surface integrals) it is required to evaluate the basis functions at the intervals ends ($\xi_1, \xi_2 = -1$ or 1), which are not directly accessible when using GL quadrature. However, it is possible to evaluate these functions at the pre-processing stage to use them when necessary with minimum extra computational cost.*

A $\mathcal{Q}$-point GL quadrature integrates exactly (i.e. up to machine precision) a polynomial f of order $\mathcal{O}(f) = \mathcal{F} = 2\mathcal{Q} - 1$. Therefore, the minimum number of quadrature points to integrate exactly the polynomial f of order $\mathcal{O}(f) = \mathcal{F}$ is $\mathcal{Q}_{min} = (\mathcal{F} + 1)/2$.

Remark: *It may be noted that if the function to be evaluated is not a polynomial (e.g. trigonometric functions), the required points may need to be increased to obtain an approximate solution, which may not be exact. However, a recent study [153] showed that Gauss quadrature is more efficient and accurate than other quadrature rules (e.g. trapezoidal, Romberg's rule) when dealing with non-polynomial functions.*

Table 2.2 summarises the maximum polynomial order and the required number of quadrature points necessary to integrate exactly the various terms defining the bilinear form Eq. 2.2.15 when triangular elements with straight edges are used for discretisation. The various terms include volume integrals (e.g. Local Mass matrix $\mathbf{M^{el}}$, Local Gradient matrix $\mathbf{K_r^{el}}$ and Local Laplacian matrix $\mathbf{L_r^{el}}$) and surface integrals (e.g. Penalty terms and flux terms that contribute to the Flux matrix $\mathbf{T}$). The table shows the maximum polynomial order $\mathcal{F}$ and associated minimum number of quadrature points $\mathcal{Q}_{min}$ required per direction for the considered matrices. For example, the Local Mass matrix entries can be expanded as:

$$
\begin{aligned}
{M^{el}}_{ij} &= \int_{el} \phi_i(\mathbf{x})\phi_j(\mathbf{x})d\mathbf{x} = \int_{el} \phi_{pq}(\mathbf{x})\phi_{rs}(\mathbf{x})d\mathbf{x} \\
&= \frac{\mathcal{A}}{2}\int_{-1}^{1} \varphi_p^{tri^A}(\xi_1)\varphi_r^{tri^A}(\xi_1)d\xi_1 \int_{-1}^{1} \varphi_{pq}^{tri^B}(\xi_2)\varphi_{rs}^{tri^B}(\xi_2)\frac{1-\xi_2}{2}d\xi_2,
\end{aligned}
\tag{2.2.38}
$$

where $\mathcal{A}$ represents the triangle area of the element in physical space and relates to the determinant of the Jacobian from an arbitrary triangle to the unit triangle. In addition, the last term results from the determinant of the Jacobian from the unit triangle to the unit square (see details in appendix A section A.1). Inspection of Eq. 2.2.38 shows that the polynomial order to be integrated in the ξ_2 direction has order $\mathcal{F} = 2k+1$ and therefore requires $\mathcal{Q}_{min} = k+1$ quadrature points.

To calculate the required number of quadrature points to compute the non-linear terms, the argument introduced in [105] and [106] is considered. Namely, since the non-linear terms are computed in physical space and subsequently projected into modal space (through the L^2 inner product), it can be seen that it is necessary to perform the integration of polynomials: $\mathcal{O}(f) = \mathcal{F} = \mathcal{O}(\int_{el} \mathbf{N}^{el}(\mathbf{u_h})\phi_k(\mathbf{x})d\mathbf{x}) \approx \mathcal{O}(\int_{el} \phi_i(\mathbf{x})\phi_j(\mathbf{x})\phi_k(\mathbf{x})d\mathbf{x}) = 3k+1$. Finally, table 2.2 shows that integration of the non-linear terms requires more quadrature points, followed by the Flux matrix and the Mass matrix.

	$\mathcal{F}$	$\mathcal{Q}_{min}$
Mass matrix	$2k+1$	$k+1$
Gradient matrix	$2k$	$k+1/2$
Laplacian matrix	$2k-1$	k
Flux matrix	$2k+1$	$k+1$
Non linear terms	$3k+1$	$3/2k+1$

Table 2.2: Required number of Gauss-Legendre quadrature points ($\mathcal{Q}_{min}$) to evaluate volume and surface integrals consisting of polynomials of degree $\mathcal{F}$ for triangular elements; k denotes the order of the basis and test functions.

2.3 Implementation and solution of the linear system

The method described has been implemented using the object oriented language C++ with the sole use of external libraries to perform matrix-vector (BLAS-2), matrix-matrix (BLAS-3) multiplications and to solve the linear system resulting from the DG-SIPG discretisation. For most verification cases (small in size), direct solvers (e.g. Cholesky factorisation for positive definite systems or LU decomposition for non positive definite

systems (as needed for the IIPG and NIPG methods in appendix C) have been used to avoid inaccuracies introduced in the solution through the use of iterative solvers. For larger problems and the DG-SIPG system, the intel MKL-Pardiso solver [118] and its sparse version of the direct-iterative solver (i.e. Cholesky preconditioned conjugate gradient iterative solver with a stoping criterion of 10^{-9}) has been selected. The Intel MKL compressed sparse row (CSR) format has been used in its 3-array variation form [118].

For non-moving meshes, all matrices can be pre-computed and stored in memory. In particular, it is not necessary to recompute the Global Laplacian or the Flux matrix at each time step and only the right hand side vector needs to be computed on the fly. This results in a relatively low computational cost, which is dominated by the system solve (i.e. Poisson and Helmhotz steps). As it will be seen in the next chapter, this is more involved when rotating meshes are considered.

The grid generator used to construct the computational meshes is Gambit [55] using the *Neutral* format, and post-processing is performed using Tecplot [163]. A manual of the code describing functions and classes has been created using the Doxygen tool [182].

2.4 Post-processing: Norms and Forces

To quantify the error in the DG solution, when the exact solution is known, the following $L^p(\Omega)$ error norms are introduced. In addition, more evolved problems require definitions for the forces over wall bounded geometries.

2.4.1 Norms

When the exact solution $u^{exact}(\mathbf{x}, t)$ is known (i.e. verification cases), the $L^2(\Omega)$ (i.e. $p = 2$) norm and occasionally the $L^{inf}(\Omega)$ (i.e. $p = \infty$) norm in their absolute and relative versions have been considered:

- **Absolute Error**: $||u^{exact} - u^{DG}||_p$,

- **Relative Error:** $\frac{||u_i^{exact}-u_i^{DG}||_p}{||u_i^{exact}||_p}$, with i=1,2 (i.e. $(u_1,u_2)=(u,v)$).

The $L^2(\Omega)$ and $L^{inf}(\Omega)$ norms, are defined by considering the broken space Ω_h:

$$||u^{exact}-u^{DG}||_2^{\Omega_h} = \sqrt{\sum_{el\in\Omega_h}||u^{exact}-u^{DG}||_2^{el}} = \sqrt{\sum_{el\in\Omega_h}\int_{el}(u^{exact}-u^{DG})^2 d\mathbf{x}}, \tag{2.4.1}$$

$$||u^{exact}-u^{DG}||_\infty^{\Omega_h} = MAX_{el\in\Omega_h}||u^{exact}-u^{DG}||_\infty^{el} = MAX_{el\in\Omega_h}\{MAX_{\mathbf{x}\in el}(u^{exact}-u^{DG})\}. \tag{2.4.2}$$

2.4.2 Forces

Throughout this thesis, the following definitions to compute the forces that the fluid exerts on the geometry, are defined. Pressure and viscous forces in terms of Cartesian coordinates [101], [77] are computed using the following expression $\mathbf{F} = \mathbf{F_p} + \mathbf{F_v} = (F^x, F^y)^T = (F_p^x + F_v^x, F_p^y + F_v^y)^T$:

$$\begin{aligned} \mathbf{F_p} &= \oint p\mathbf{n}ds, \\ \mathbf{F_v} &= -\oint \nu[\nabla\mathbf{u} + (\nabla\mathbf{u})^T].\mathbf{n}ds, \end{aligned} \tag{2.4.3}$$

where $\mathbf{n}$ is the outward pointing normal at walls.

2.5 Verification and numerical properties

This section details verification problems used to investigate the method's convergence properties and accuracy. First, a purely elliptic problem is considered, followed by an unsteady Stokes problem. Other verifications cases can be found in appendix C where various Interior Penalty Galerkin methods are compared. This section presents the more relevant cases for verification which are used to provide insight into the numerical properties of the scheme (i.e. error behaviour in the L^2 norm for spatial and temporal refinement)

2.5.1 Elliptic equation: Poisson-Helmholtz problem

The inhomogeneous Helmholtz equation, i.e. $-\triangle u + \alpha u = g$, is solved in $\Omega = [0,1]^2$. The wave number $\sqrt{\alpha}$ can be chosen arbitrarily (e.g. $\alpha = 0$ corresponds to the Poisson equation) provided that the forcing term is of the form $g(x,y) = (\alpha + 2\pi^2)\cos(\pi x)\cos(\pi y)$. The problem is closed using Dirichlet boundary conditions derived from the exact solution $u^{exact}(x,y) = \cos(\pi x)\cos(\pi y)$. The penalty parameters are fixed to $\beta = 1$ and $\sigma = 3k(k+1)$ (as derived in section 2.2.2.3) and the L^2 relative error norm is used to quantify the solver accuracy.

Fig. 2.5.a shows $log(L^2) - k$ results for two semi-structured triangular meshes with $N^{el} =$ 50 and 200 elements (similar to the mesh depicted in Fig. C.2, appendix C) and two distinct wave numbers $\sqrt{\alpha}$. Exponential convergence is observed for all cases, when $\alpha = 0$ (i.e. Poisson equation) and $\alpha = 1 \times 10^5$. It is interesting to analyse the behaviour of

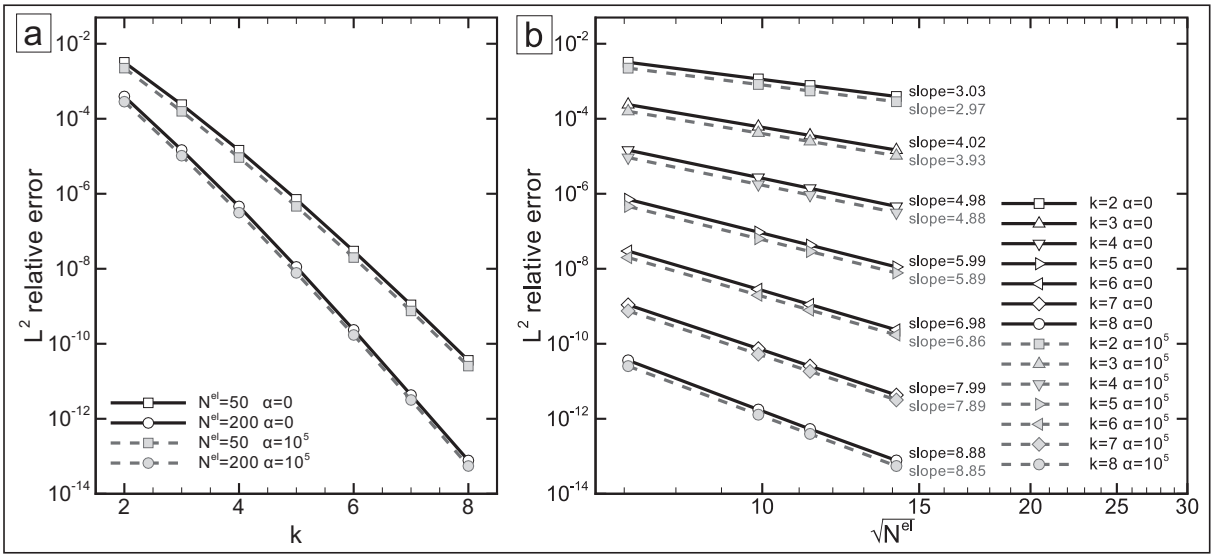

Figure 2.5: L^2 relative error norm for the Poisson-Helmholtz equation for (a) p-refinement (polynomial enrichment) and (b) h-refinement.

the method for large wave numbers because when considering the Helmholtz step in the splitting scheme, the wave number corresponds to $\alpha = Re\gamma_0/(\triangle t)$, which can become large for high Reynolds number flows or when a small time step is selected. As shown in

Fig. 2.5.a, the L^2 error decreases for higher wave numbers.

Fig. 2.5.b depicts the convergence rates for h-refinement using four meshes with $N^{el} = 50$, 98, 128 and 200 semi-structured triangular elements, again for two distinct wave numbers and polynomial orders ranging from $k = 2$ to 8. The DG-SIPG method provides optimal convergence rates (i.e. $\mathcal{O}(h^{k+1})$) as predicted by theoretical analysis [146]. Further details are provided in appendix C.

2.5.2 Unsteady Stokes equations

The time discretisation scheme defined in section 2.2.1 may be used without considering the non-linear terms to solve the unsteady Stokes equations. In this section, the following solution for the Stokes problem [156] is considered:

$$\begin{aligned} (u, v) &= (\sin(x)(a\sin(ay) - \cos(a)\sinh(y)), \cos(x)(\cos(ay) - \cos(a)\cosh(y)))\, e^{-\lambda t}, \\ p &= \lambda \cos(a)\cos(x)\sinh(y) e^{-\lambda t}, \end{aligned} \tag{2.5.1}$$

with $a = 2.883356$ and $\lambda = 9.313739$. This unsteady problem can be solved in $\Omega = [-1, 1]^2$ with Dirichlet boundary conditions and initial condition provided by the exact solution. Tests are conducted using two meshes constituted of 72 and 128 semi-structured triangular elements and the penalties $\beta = 1$ and $\sigma = 3k(k + 1)$. Let us note that only equal order polynomial spaces $[D_k(\Omega_h)]^2_{vel} - [D_k(\Omega_h)]_{press}$ for velocity and pressure are considered. Fig. 2.6 shows the L^2 relative error norm convergence for pressure and velocity obtained for time step refinement. For spatially well resolved simulations ($k \geq 7$), convergence rates for the L^2 norm for pressure of 1.82 and 2.21 for velocity are obtained. These slopes are in very good agreement with the theoretical temporal convergence rates in the L^2 norm obtained in [71] for conformal discretisations and the splitting scheme employed, which are $\mathcal{O}(\triangle t^{3/2})$ for pressure and $\mathcal{O}(\triangle t^2)$ for velocity. The figures also show that the overall L^2 errors for pressure and velocity behave as $\mathcal{O}(\triangle t^2 + h^{k+1}/\triangle t)$ with h representing the

element size.

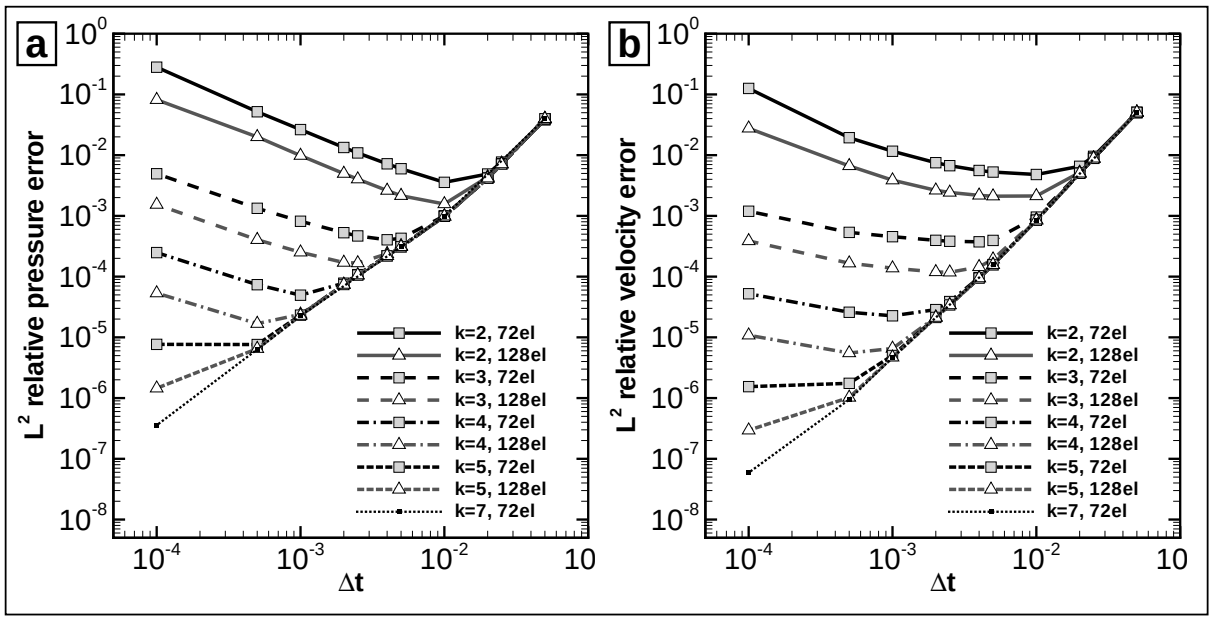

Figure 2.6: Time step refinement study for the unsteady Stokes equations at final time $T = 0.1$, for two meshes and various polynomial orders. L^2 relative error norm for (a) pressure and (b) velocity.

Remark: *A similar behaviour has been observed using the same splitting technique for a strong Semi-Lagrangian spectral element technique [101] (and references herein) and has been attributed to the Lagrangian treatment of the advective equation and the splitting error.*

It can be seen that for all polynomial orders the error decreases with $\triangle t$ provided that the time step remains larger that a certain $\triangle t_{min}(h,k)$, which itself reduces as the spatial discretisation increases (using either h or p-refinement). To provide an estimate for $\triangle t_{min}(h,k)$, it suffices to equate to zero the derivative of the expression $L^2(\triangle t) = C_1 \triangle t^2 + C_2 h^{k+1}/\triangle t$ to yield $\triangle t_{min}(h,k) = (\frac{C_2}{2C_1} h^{k+1})^{1/3}$, where C_1 and C_2 are problem dependent constants (e.g. dependent on the Reynolds number) but independent of h and k. The next section provides a possible explanation of this behaviour, which is related to the inf-sup condition of the discretised scheme.

Remark: *In practice, the lower limit for the time step does not represent a restric-*

tion, when the full Navier-Stokes equations are considered (i.e. CFL condition taken into account), since $\triangle t_{min} = \mathcal{O}(h^{(k+1)/3})$ *decreases faster than* $\triangle t_{CFL} = \mathcal{O}(h/Uk^2)$ *as the spatial discretisation is increased (using either h or p-refinement), which allows for flexibility in the selection of the time step. This can be seen by expressing the usability condition* $\triangle t_{CFL} > \triangle t_{min}$ *as* $\triangle t_{CFL} - \triangle t_{min} \approx \mathcal{O}(h/Uk^2 - h^{(k+1)/3}) > 0$. *Factoring h and assuming* $Uk^2 \neq 0$ *leads to the condition* $\mathcal{O}(Uk^2 h^{(k/3-2/3)}) < \mathcal{O}(1)$. *This condition shows that for low polynomials it is likely that the stability condition is violated. For example if* $k = 2$, *one obtains* $\mathcal{O}(4U) < \mathcal{O}(1)$. *However for* $k = 3$, *the condition becomes* $\mathcal{O}(9Uh^{1/3}) < \mathcal{O}(1)$, *which is likely to be fulfilled for high mesh resolutions (*$h << 1$*). This condition shows that as k increases the stability condition becomes less restrictive.*

Since for low order methods (e.g. $k = 2$*), it is not clear that* $\triangle t_{CFL} > \triangle t_{min}$, *the method is not useful with the present setting (i.e. equal order velocity/pressure pairs, see following sections). This could explain the popularity of this scheme in conjunction with high order spatial discretisation but the limited success when using low order Finite Element techniques.*

2.5.3 Temporal stability and the inf-sup condition

As showed in the previous section, instabilities (or unbounded L^2 errors) may arise if a small time step $\triangle t < \triangle t_{min}$ is selected for computation. It has also been shown that this limit decreases faster (when increasing the spatial resolution) than the time step issued from the CFL condition (i.e. $\triangle t_{CFL} > \triangle t_{min}$), showing the usability of the described scheme. In this section, the author would like to explore the causes of this behaviour seeking inspiration in the work performed in [68]. In this paper, Guermond et al. provide a summary of the stability conditions for projection schemes (i.e. pressure-correction methods, velocity-correction methods and consistent splitting methods). They review different temporal schemes including the one considered in this thesis and address the issue: Do projection schemes require an inf-sup condition to be satisfied?

For many years, it had been thought that this type of schemes do not require an inf-sup condition to be fulfilled since velocity and pressure are essentially "decoupled". However in recent years, it has been shown that these schemes correspond to stabilised like schemes (see also [33] or [12]).

In [68], the authors consider the Chorin-Temam [27], [164] algorithm and detail that the stability of projection schemes can be reduced to the analysis of the equivalent steady Stokes problem. A similar approach is used in the next section, to analyse the stability of the selected splitting scheme.

2.5.3.1 Continuous system

Using the continuous form of the NS equations (Eq. 2.2 and Eq. 2.2), and considering the equivalent steady state problem without non-linear terms, the following matrix form for the continuous steady Stokes problem (i.e. the saddle point system) can be derived:

$$\mathbf{S} = \begin{pmatrix} -\frac{1}{\mathrm{Re}}\nabla^2 & \nabla \\ \nabla. & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ 0 \end{pmatrix}. \tag{2.5.2}$$

The discretised form $\mathbf{S_h}$ of this Stokes system $\mathbf{S}$ must satisfy the inf-sup (Ladyzhenskaya-Babushka-Brezzi or LBB) condition [21], [47] or [68]. Typically the inf-sup condition states that an equal order pressure and velocity pair do not lead a stable system. To solve this problem one can augment the velocity space with respect to the pressure space or add some stabilisation whilst maintaining the same space dimensions for pressure and velocity.

Let us consider the semi-discrete (i.e. continuous in space and discretised in time) system resulting from the splitting scheme defined in section 2.2.1 without non-linear terms, and derive its steady state form ($\mathbf{u}^{n+1} = \mathbf{u}^n = \mathbf{u}$ and $p^{n+1} = p$). Under the steady

Stokes assumptions, the three required steps defined in section 2.2.1 reduce to:

$$\tilde{\mathbf{u}} = \mathbf{u}, \tag{2.5.3}$$

$$\gamma_0 \frac{\tilde{\tilde{\mathbf{u}}} - \tilde{\mathbf{u}}}{\Delta t^*} = -\nabla p, \tag{2.5.4}$$

$$\gamma_0 \frac{\mathbf{u} - \tilde{\tilde{\mathbf{u}}}}{\Delta t^*} = \frac{1}{\mathrm{Re}} \nabla^2 \mathbf{u}, \tag{2.5.5}$$

where Δt^* does not represent a physical time step, but is a factor that accounts for the difference between $\tilde{\tilde{\mathbf{u}}}$ and $\tilde{\mathbf{u}}$. It is possible to eliminate the intermediate velocities ($\tilde{\tilde{\mathbf{u}}}$ and $\tilde{\mathbf{u}}$): firstly, let us substitute Eq. 2.5.3 in Eq. 2.5.4, take the divergence of the resulting equation and make use of $\nabla . \tilde{\tilde{\mathbf{u}}} = 0$; secondly, one may add Eq. 2.5.4 and Eq. 2.5.5. Finally, let us multiply the first equation by Re and divide the second by the same term, to obtain the following matrix form:

$$\mathbf{S_{split}} = \begin{pmatrix} -\nabla^2 & \mathrm{Re}\nabla \\ \frac{1}{\mathrm{Re}}\nabla . & -\frac{\triangle t^*}{\mathrm{Re}\gamma_0}\nabla^2 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ 0 \end{pmatrix}. \tag{2.5.6}$$

By comparison of Eq. 2.5.6 to Eq. 2.5.2, it becomes clear that the new system $\mathbf{S_{split}}$ corresponds to a pseudo-compressible scheme, where $\frac{\triangle t^*}{\mathrm{Re}\gamma_0}$ may be interpreted as the pseudo-compressibility coefficient [157] or $\frac{\triangle t^*}{\mathrm{Re}\gamma_0}\nabla^2$ as a "pressure stabilisation terms", which relaxes the incompressibility constraint. As noted in [68] and [69] for the original splitting scheme of Chorin and Temam, if the spatial discretisation $\mathbf{S_{split}}_h$ of the continuous system $\mathbf{S_{split}}$ uses inf-sup unstable pressure-velocity pairs, then if $\triangle t^*$ becomes "too small", spurious modes manifest themselves, which may lead to instabilities. In [69], the authors proposed the use of $\triangle t^* = \triangle t + h_{max}^{k+1}$, where h_{max}, is the maximum size of the elements, to avoid too small times steps and stabilise the system. This correction has not been tested in this thesis.

2.5.3.2 Algebraic system

The same approach can be used to obtain a system corresponding to the discretised algebraic scheme defined in section 2.2.3. Once more, the steady Stokes problem is considered with equal order polynomial spaces for pressure and velocity. The system reads:

$$\mathbf{S}^{\mathbf{alg}}_{\mathbf{split_h}} = \begin{pmatrix} \mathbf{A}^{Helm}_{(\alpha=0,\sigma)} & \mathrm{Re}\mathbf{K}^T \\ \frac{1}{\mathrm{Re}}\mathbf{K} & \frac{\triangle t^*}{\mathrm{Re}\gamma_0}\mathbf{A}^{Pois}_{(\alpha=0,\sigma)} \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} \mathbf{RHS}^{Helm}_{(g=0,\sigma)} \\ \frac{1}{\mathrm{Re}}\mathbf{RHS}^{Pois}_{(g=0,\sigma)} \end{pmatrix}. \tag{2.5.7}$$

This matrix form shows that if $\frac{\triangle t^*}{\mathrm{Re}}$ becomes small, the discrete version of the saddle point Stokes system $\mathbf{S}$ (i.e. without stabilisation) is recovered, which would require an inf-sup stable pressure-velocity pair (e.g. smaller polynomial order for pressure than for velocity). It may be concluded that when equal order velocity/pressure pairs are used, the term $\frac{\triangle t^*}{\mathrm{Re}\gamma_0}\mathbf{A}^{Pois}_{(\alpha=0,\sigma)}$ dictates the temporal stability of the scheme. In particular, it may be intuitively argued that "enough" stabilisation is provided by the combination of:

- a large enough time step $\triangle t^*$,
- a low enough Reynolds number (Re) or equivalently high enough viscosity (ν),
- a large enough spectral radius for the matrix $\mathbf{A}^{Pois}_{(\alpha=0,\sigma)}$ when compared to the ratio $\frac{\triangle t^*}{\mathrm{Re}}$.

More precisely, one can eliminate the velocity from the system described by Eq. 2.5.7 to explore the solvability condition for the pressure. Assuming that $\mathbf{A}^{Helm}_{(\alpha=0,\sigma)}$ is invertible (i.e. the DG-SIPG method provides a positive definite matrix), one obtains:

$$\left(\frac{\triangle t^*}{\mathrm{Re}\gamma_0}\mathbf{A}^{Pois}_{(\alpha=0,\sigma)} - \mathbf{K}[\mathbf{A}^{Helm}_{(\alpha=0,\sigma)}]^{-1}\mathbf{K}^T\right) P = \frac{1}{\mathrm{Re}}\left(\mathbf{RHS}^{Pois}_{(g=0,\sigma)} - \mathbf{K}[\mathbf{A}^{Helm}_{(\alpha=0,\sigma)}]^{-1}\mathbf{RHS}^{Helm}_{(g=0,\sigma)}\right). \tag{2.5.8}$$

It can be seen that the solvability of Eq. 2.5.8 is determined by the invertibility of the pressure Schur complement:

$$Schur(\mathbf{S}^{\mathbf{alg}}_{\mathbf{split_h}}) = \frac{\triangle t^*}{\mathrm{Re}\gamma_0}\mathbf{A}^{Pois}_{(\alpha=0,\sigma)} - \mathbf{K}[\mathbf{A}^{Helm}_{(\alpha=0,\sigma)}]^{-1}\mathbf{K}^T, \tag{2.5.9}$$

where the dependency of the discrete stabilisation term $\frac{\triangle t^*}{\mathrm{Re}\gamma_0}\mathbf{A}^{Pois}_{(\alpha=0,\sigma)}$ is explicit.

It is possible to provide a necessary condition for solvability by considering the L^2 norm of the Schur complement defined by Eq. 2.5.9:

$$\begin{aligned} ||Schur(\mathbf{S}^{\mathbf{alg}}_{\mathbf{split_h}})||_2 &= ||\frac{\triangle t^*}{\mathrm{Re}\gamma_0}\mathbf{A}^{Pois}_{(\alpha=0,\sigma)} - \mathbf{K}[\mathbf{A}^{Helm}_{(\alpha=0,\sigma)}]^{-1}\mathbf{K}^T||_2 \\ &\geq Cte\left|\frac{\triangle t^*}{\mathrm{Re}\gamma_0}||\mathbf{A}^{Pois}_{(\alpha=0,\sigma)}||_2 - ||\mathbf{K}[\mathbf{A}^{Helm}_{(\alpha=0,\sigma)}]^{-1}\mathbf{K}^T||_2\right|, \end{aligned} \tag{2.5.10}$$

which follows from the reverse triangle inequality, where $||\bullet||_2$ denotes the L^2 norm and Cte is a constant independent of the discretisation. The L^2 norm of a matrix $\mathbf{A}$ corresponds to computing its maximum singular value (i.e. spectral radius or maximum eigenvalue for a positive definite matrix): $||\mathbf{A}||_2 = \sigma_{max}(\mathbf{A})$. Eq. 2.5.10 can hence be bounded as:

$$||Schur(\mathbf{S}^{\mathbf{alg}}_{\mathbf{split_h}})||_2 \geq Cte\left|\frac{\triangle t^*}{\mathrm{Re}\gamma_0}\sigma_{max}(\mathbf{A}^{Pois}_{(\alpha=0,\sigma)}) - \sigma_{min}(\mathbf{K}[\mathbf{A}^{Helm}_{(\alpha=0,\sigma)}]^{-1}\mathbf{K}^T)\right|. \tag{2.5.11}$$

To simplify the notation, let us set ${\sigma_{min}}^a = \sigma_{min}(\mathbf{K}[\mathbf{A}^{Helm}_{(\alpha=0,\sigma)}]^{-1}\mathbf{K}^T)$ and ${\sigma_{max}}^b = \sigma_{max}(\mathbf{A}^{Pois}_{(\alpha=0,\sigma)})$. The Stokes problem, studied in section 2.5.2, can be solved in two triangular meshes (with 18 and 72 elements) and polynomial orders ranging from $k = 2$ to 10, to estimate the spectral radius of these matrices.

On the one hand, the minimum singular value for $\mathbf{K}[\mathbf{A}^{Helm}_{(\alpha=0,\sigma)}]^{-1}\mathbf{K}^T$ is zero: ${\sigma_{min}}^a = 0$, independently of h and k (i.e. showing that inf-sup unstable pairs have been used). On the other hand, the slope of ${\sigma_{max}}^b$ for p-refinement is depicted in Fig. 2.7 and behaves as $\mathcal{O}(k^{3.2})$, which agrees very well with the theoretical estimate $\mathcal{O}(k^3)$ derived in [85]. Using

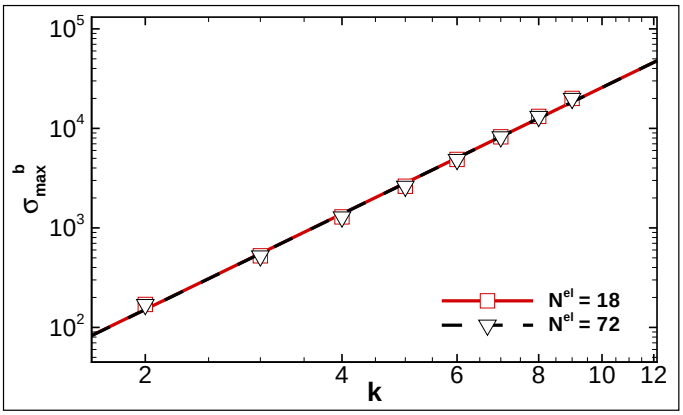

Figure 2.7: Singular values for DG-SIPG matrix (Stokes problem): two meshes and polynomial orders ranging from $k = 2$ to 10. N^{el} denotes the number of triangular elements.

these empirical correlations, the limit for solvability (resp. stability) Eq. 2.5.11 becomes:

$$\lim_{\frac{\triangle t^*}{\mathrm{Re}}\to 0} \frac{\triangle t^*}{\mathrm{Re}\gamma_0} {\sigma_{max}}^b - {\sigma_{min}}^a \longrightarrow 0,$$
$$\lim_{\frac{\triangle t^*}{\mathrm{Re}}\to 0} \frac{\triangle t^*}{\mathrm{Re}\gamma_0} \mathcal{O}(k^{3.2}) \longrightarrow 0. \tag{2.5.12}$$

From Eq. 2.5.12, it can be seen that if the term $\frac{\triangle t^*}{\mathrm{Re}}$ decreases faster than $\mathcal{O}(k^{-3.2})$, then it is likely that the system becomes ill conditioned (or equivalently requires inf-sup stable pressure-velocity pairs). Let us note that Guermond et al. [68] provide an estimate for the time step critic for stability (based on observations), for a similar time splitting scheme and conformal discretisations: $\triangle t \sim \mathcal{O}(k^{-3})$, which is consistent with the present analysis.

Remark: *This new limit for stability is different from the previously obtained in section 2.5.2 from observations of the L^2 error for time refinement using a Stokes problem. However, in the former the effect of the viscosity was not taken into account and hence the new limit is believed to be more general. Nonetheless, both analysis show that if high order polynomial spaces are selected, the method provides usable solutions.*

It may be concluded that equal pressure/velocity spaces can be used in conjunction with the selected splitting scheme as long as the ratio of time step to Reynolds number

does not becomes smaller than $\mathcal{O}(k^{3.2})$. Once more, it can be seen that a high order spatial discretisation in conjunction with the selected splitting scheme constitutes a useful method. Low order methods however, may require further stabilisation.

2.5.3.3 Stabilisation for low order spatial discretisations

Although it appears that high order spatial discretisations do not require of further stabilisation, a possible stabilisation strategy is presented here and may be found useful if low order polynomials are used.

The continuous Laplacian operator ∇^2 is unique in Eq. 2.5.6. However, in the discrete case, the bilinear form used for the velocity step (Helmholtz equation) and pressure step (Poisson equation) can be different as outlined in previous sections by setting two distinct matrices $\mathbf{A}^{Pois}_{(\alpha=0,\sigma)}$ and $\mathbf{A}^{Helm}_{(\alpha=0,\sigma)}$. The particularity of the DG-SIPG method is that in the discrete bilinear form, the penalty parameters (i.e. σ) for the pressure and the velocity are not required to be equal, leading to $\mathbf{A}^{Pois}_{(\alpha=0,\sigma=\sigma_{Poisson})}$ and $\mathbf{A}^{Helm}_{(\alpha=0,\sigma=\sigma_{Helmholtz})}$ with $\sigma_{Poisson} \neq \sigma_{Helmholtz}$.

This particularity can be used to stabilise the method by defining a large penalty parameter for the Poisson equation (the pressure step). For example one may set:

$$\sigma_{Poisson} = \frac{3k(k+1)}{|trace|}\frac{\mathrm{Re}}{\triangle t^*}, \tag{2.5.13}$$

where $\frac{3k(k+1)}{|trace|}$ is the previously derived necessary penalty parameter (see section 2.2.2.3). Recalling the definition of the DG-SIPG bilinear matrix in terms of the global Mass, Laplacian and Flux matrices (as introduced in section 2.2.3), one can expand the Poisson bilinear matrix as:

$$\mathbf{A}^{Pois}_{(\alpha,\sigma=\sigma_{Poisson})} = \mathbf{L} + \alpha\mathbf{M} + \mathbf{T1} + \sigma_{Poisson}\mathbf{T2}, \tag{2.5.14}$$

where $\mathbf{T} = \mathbf{T1} + \sigma_{Poisson}\mathbf{T2}$, $\mathbf{T}$ is the Flux matrix and:

$$\sigma_{Poisson}\mathbf{T2} := \left\{ \frac{\mathrm{Re}}{\triangle t^*} \sum_{\Gamma \in \Gamma_h \cup \Gamma_D} \frac{3k(k+1)}{|trace|} \int_\Gamma [[u_h]][[v_h]] ds \right\}. \tag{2.5.15}$$

Substituting Eq. 2.5.14 and Eq. 2.5.13, into the matrix form Eq. 2.5.7 leads:

$$\mathbf{S}^{\mathbf{alg}}_{\mathbf{split_h}} = \begin{pmatrix} \mathbf{A}^{Helm}_{(\alpha=0,\sigma)} & \mathrm{Re}\mathbf{K}^T \\ \frac{1}{\mathrm{Re}}\mathbf{K} & \frac{\triangle t^*}{\mathrm{Re}\gamma_0}(\mathbf{L} + \alpha\mathbf{M} + \mathbf{T1}) + \frac{3k(k+1)}{\gamma_0|trace|}\mathbf{T2} \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} \mathbf{RHS}^{Helm}_{(g=0,\sigma)} \\ \frac{1}{\mathrm{Re}}\mathbf{RHS}^{Pois}_{(g=0,\sigma)} \end{pmatrix}. \tag{2.5.16}$$

Inspection of this matrix system shows that for small time steps and high Reynolds numbers, the term $\frac{\triangle t^*}{\mathrm{Re}\gamma_0}(\mathbf{L} + \alpha\mathbf{M} + \mathbf{T1})$ tends to zero, but leaves the term $\frac{3k(k+1)}{\gamma_0|trace|}\mathbf{T2}$ to provide stabilisation.

To illustrate this discussion, let us reconsider the Stokes problem and the time refinement study performed in section 2.5.2. Fig. 2.8 compares the L^2 relative errors for the original scheme (gray lines shown previously in Fig. 2.6) and the new stabilised scheme (colored lines) for various polynomial orders. As predicted, the stabilising effect for small time steps becomes noticeable and is particularly important for low order spatial discretisations.

Effectively, this technique increases the spectral radius of the Poisson operator: $\mathbf{A}^{Pois}_{(\alpha,\sigma=\sigma_{Poisson})}$, which scales linearly with the penalty parameter: $\mathcal{O}(\sigma_{Poisson})$ for the Stokes problem (these results are not shown), rendering the scheme stable for smaller values of $\frac{\triangle t^*}{\mathrm{Re}}$. In fact, this modification provides a scheme that is very similar to the stabilised technique described by Cockburn et al. in [30]. As noted in this reference, this technique requires post-processing to obtain truly divergence free velocities since the stabilisation term (i.e. the penalty term integral) alters the divergence free condition. Furthermore, this technique increases the condition number of the pressure solve, which may pollute the solution if iterative solvers are used.

In the remaining of this thesis, this stabilisation strategy is not considered and high

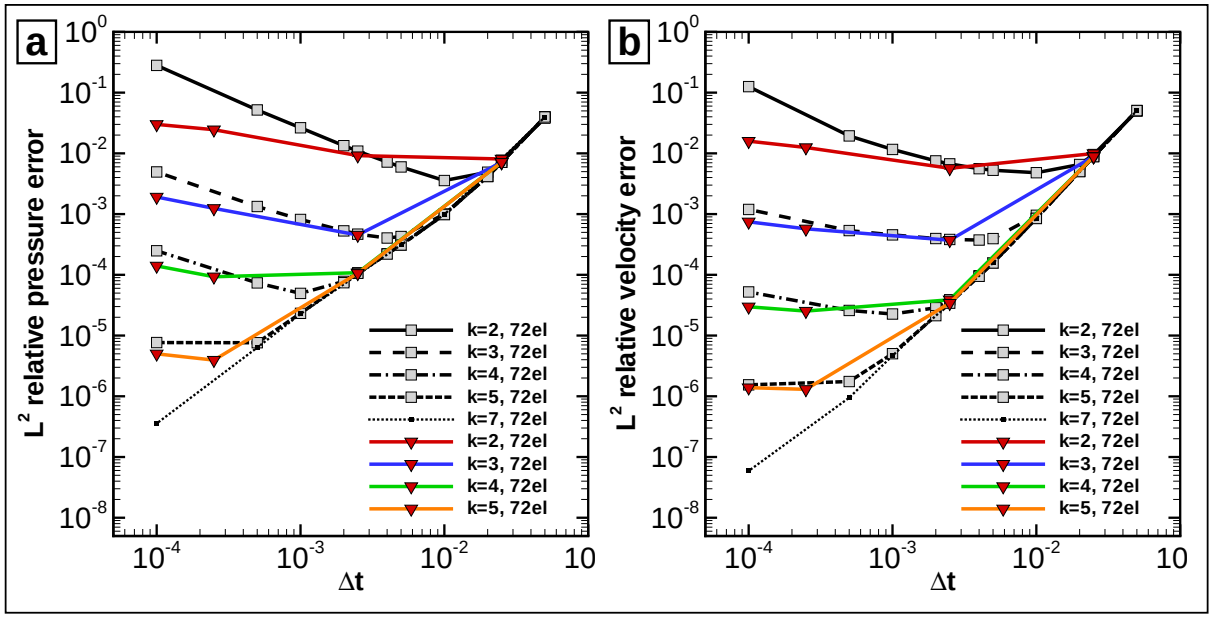

Figure 2.8: Time step refinement study for the unsteady Stokes equations at final time $T = 0.1$, for various polynomial orders. L^2 relative error norm for (a) pressure and (b) velocity. Gray lines show the original scheme and colored lines the stabilised scheme.

order polynomials are used to avoid instabilities if small time steps or small viscosities are considered.

2.6 Navier-Stokes equations

This section considers the full NS equations including the non-linear terms. Firstly, the Taylor vortex problem is used for further verification and evaluation of numerical properties. Secondly, the flow around a square cylinder is used for the validation of the solver.

2.6.1 Taylor Vortex problem

The selected problem to test the full incompressible NS equations is the so called Taylor vortex [101], [156]. The time dependent solution that is used to impose Dirichlet boundary

conditions and the initial condition is given by:

$$\begin{aligned}(u, v) &= (-\cos(\pi x)\sin(\pi y), \sin(\pi x)\cos(\pi y))\, e^{(-2/Re)\pi^2 t}, \\ p &= -\frac{1}{4}\left(\cos(2\pi x) + \cos(2\pi y)\right) e^{(-4/Re)\pi^2 t}. \end{aligned} \tag{2.6.1}$$

The problem, which represents standing vortices decaying in time, is solved in $\Omega = [-1, 1]^2$ discretised with 72 triangular elements and the following parameters: Re = 50, $\beta = 1$, $\sigma = 3k(k+1)$ and equal order elements for velocity and pressure. Fig. 2.9 shows the DG solution for k =4.

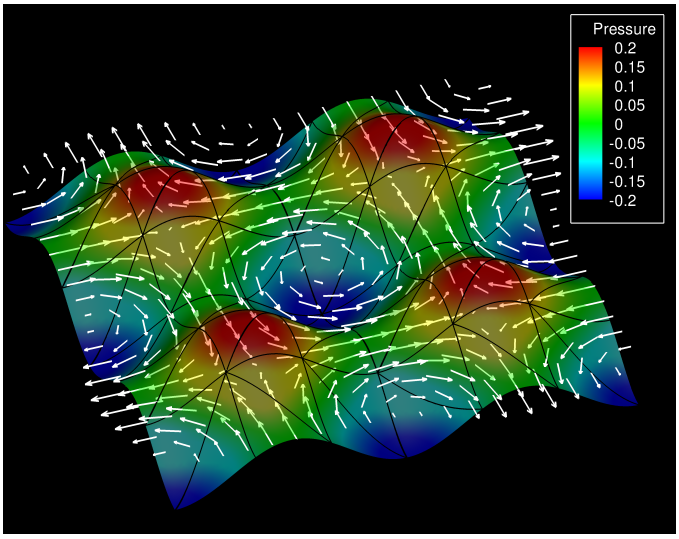

Figure 2.9: DG solution for Taylor vortex problem for a mesh with 72 triangular elements and polynomial order $k = 4$.

Firstly, Fig. 2.10.a shows results for p-refinement. The semi-log of the L^2 relative error norm and maximum pointwise error (i.e. L^{inf} norm) show exponential convergence for pressure and velocity.

Secondly, h-refinement results are depicted in Fig. 2.10.b, for various meshes ranging from N^{el} =98 to 338 triangular elements and polynomial orders $k = 2$ to 5. The figure shows convergence rates with their associated slopes. It can be seen that the method provides, at least, optimal convergence rates $\mathcal{O}(h^{k+1})$ for all cases.

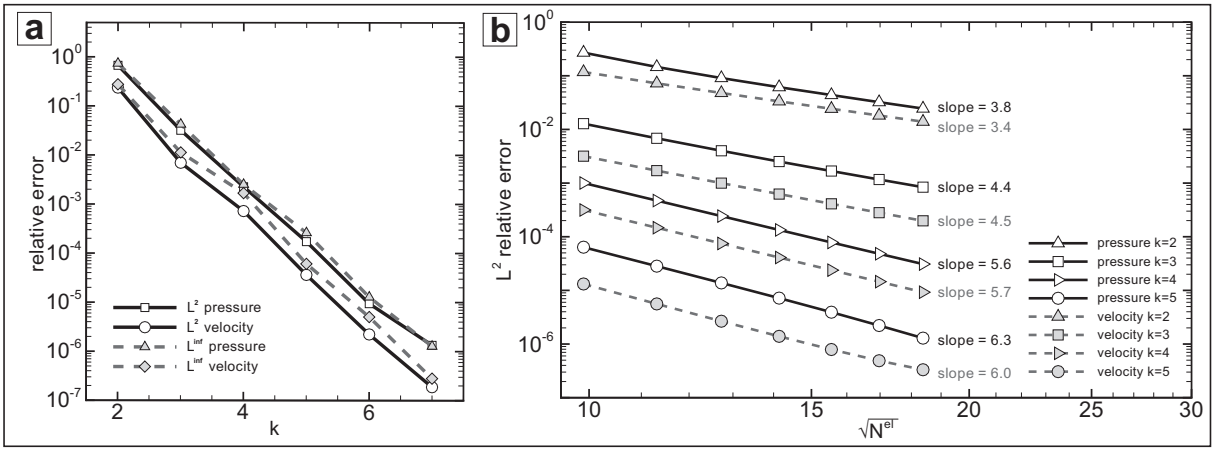

Figure 2.10: Relative error norms for the Taylor Vortex problem, $\triangle t = 0.001$ and final time $T = 0.1$ (a) p-refinement and (b) h-refinement

2.6.2 Square cylinder problem

The flow around a square cylinder has been selected as a test case to validate the solver in a complex flow. The selected Reynolds numbers are Re = 10 and 100, which are low enough for the flow to remain two dimensional and laminar. For low Reynolds numbers ($\mathrm{Re} < \mathrm{Re}_{crit} \approx 54$) [103], there is no vortex shedding and thus the solution is steady in time, enabling validation of the code in steady flow conditions. For Reynolds numbers larger than Re_{crit}, vortex shedding develops forming a Von Kármán vortex street with a characteristic vortex shedding frequency; the Strouhal number $\mathrm{St} = fb/U$, where f, $b = 1$ and $U = 1$ are the non-dimensional vortex shedding frequency, body dimension and free stream velocity. Both regimes are illustrated in Fig. 2.11 by means of DG solution snapshots (2.11.a Re = 10 and 2.11.b Re = 100).

The square cylinder case has been previously studied experimentally by Okajima [131], and numerically by Darekar and Sherwin [40] using the continuous h/p Spectral code Nektar [101]. The same domain boundaries and effective blockage ratio ($B = 2.3\%$) used in [40], are retained for the present simulations. This section presents, in addition, of published data ([131] and [40]), simulation results for the DG solver and Nektar using identical meshes and polynomial orders. The mesh consists of 836 triangular elements.

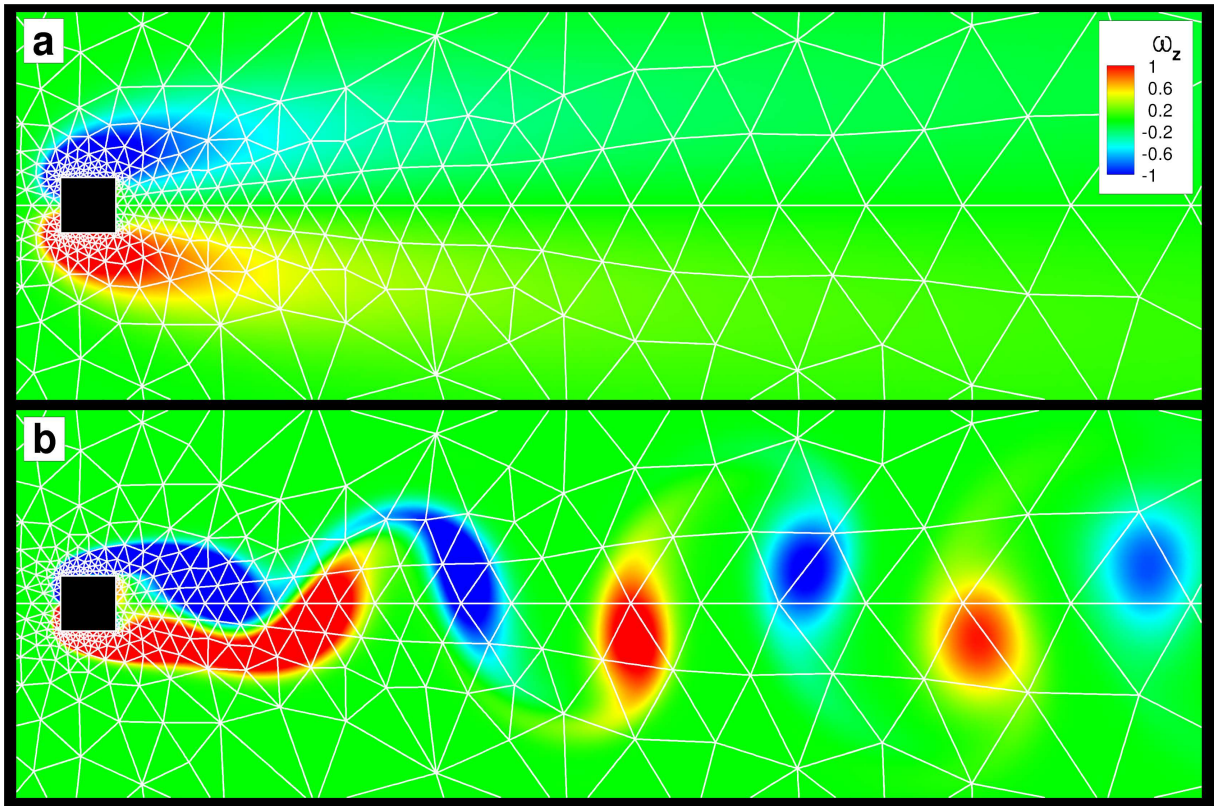

Figure 2.11: DG solution: Vorticity contours $-1 \leq \omega_z \leq 1$ for the square cylinder with the unstructured triangular mesh overlaid in the figure for (a) Re = 10 and $k = 5$ and (b) Re = 100 and $k = 7$.

First, DG and Nektar results, for the steady wake case at Re = 10, are compared for the pressure drag and viscous drag components in table 2.3. All cases are simulated using a non-dimensional time step $\triangle t = 0.01$. Results and discrepancies (i.e. $\%error = 100\frac{F_{Nektar}-F_{DG}}{F_{Nektar}}$) are summarised in the table, where it can be seen that both solvers provide very similar results. Fig. 2.11.a depicts the DG solution for k=5 at Re = 10.

Secondly, an increase in the Reynolds number to 100 enables comparison of vortex shedding frequencies. This case requires a non-dimensional time step of $\triangle t = 0.002$. A DG snapshot of the temporally evolving vorticity field is shown in Fig. 2.11.b, for Re = 100.

In order to present meaningful comparisons between the different formulations (i.e. non-conformal DG and conformal h/p Spectral Nektar), one may estimate the total num-

k	F_p^x Nektar	F_p^x DG	% $error_p$	F_v^x Nektar	F_v^x DG	% $error_v$
2	1.0523	1.0516	5.9E-02	0.5384	0.5385	-1.9E-02
3	1.0673	1.0671	1.8E-02	0.5206	0.5203	5.9E-02
5	1.0785	1.0784	5.9E-03	0.5245	0.5243	4.4E-02

Table 2.3: Results for DG and Nektar solvers for the pressure and viscous components of the drag force over a square cylinder at Re = 10.

ber of degrees of freedom (DOF) for each method ignoring the domain boundaries, which moderately reduce the number of DOF. Taking N^{el} as the number of triangular elements in the mesh, let us approximate the total number of DOF for the DG method as: $DOF_{DG} \approx \frac{N^{el}}{2}(k+1)(k+2)$. For the h/p Spectral method, since continuity is enforced across elements, the required number of DOF is reduced: $DOF_{Spectral} \approx \frac{N^{el}}{2}k^2$ (this estimate is taken from [173]). Fig. 2.12 depicts the computed St number as a function of the number of DOF, for the two solvers using the same mesh together with published results from Darekar and Sherwin [40].

In addition, table 2.4 compares the Strouhal number obtained with the present DG implementation together with data from physical experiments [131], showing an excellent level of agreement.

It is concluded that both solvers provide good agreement with the experimental data and that both display similar convergence trends.

Experimental data	Blockage (B)	St
Okajima [131]	0%	0.141-0.145
DG (k=5, DOF=17556)	2.3%	0.144

Table 2.4: Strouhal number St for the square section cylinder at Re = 100; experimental data and numerical results from the present DG implementation (for polynomial order $k = 5$).

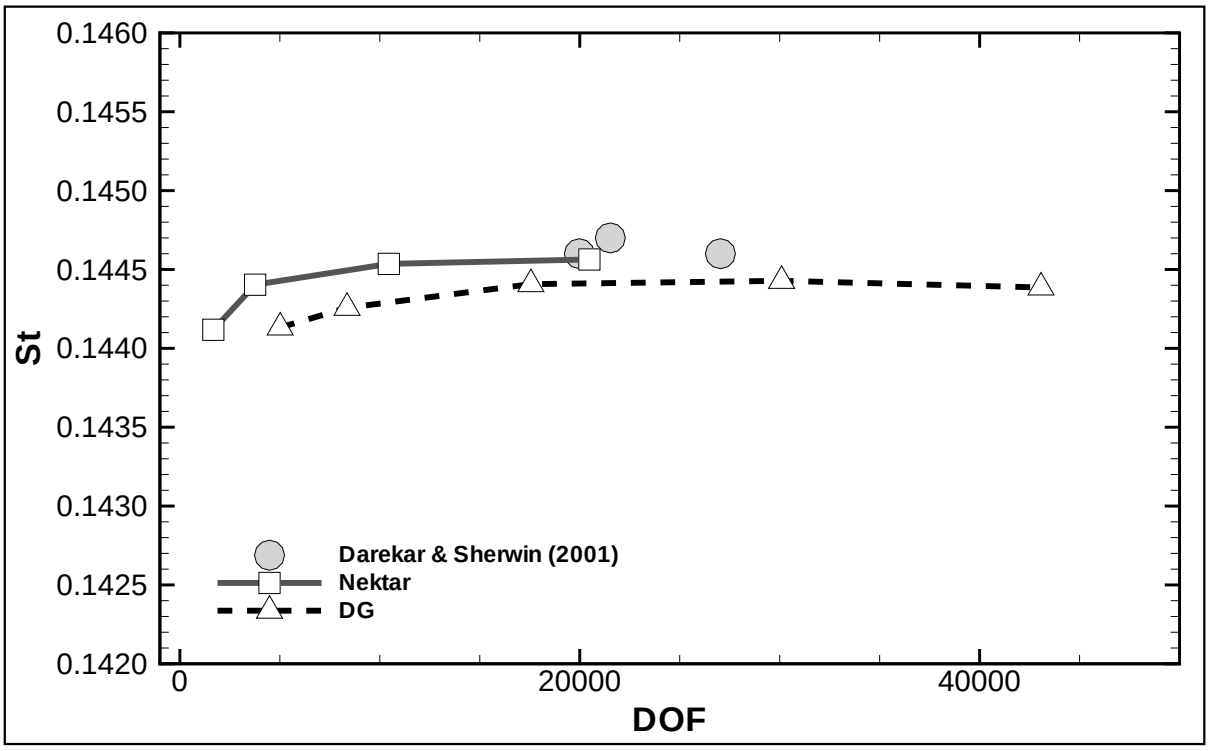

Figure 2.12: Square section cylinder at Re=100: Computed Strouhal number (St) as a function of the number of DOF. DG and continuous h/p Spectral code Nektar results are plotted together with published results for Nektar [40].

Chapter 3

High order sliding meshes and curved elements

The previous chapter presented the formulation for a high order DG solver for non moving geometries and meshes where discretisation was performed using straight edged triangular elements. The numerical properties of the solver were examined through various test cases, concluding with the analysis of vortex shedding aft a square cylinder at various Reynolds numbers. Within this chapter, the previous formulation is extended to account for rotationally moving geometries through a sliding mesh technique without the requirement to use *mortar* type techniques. To this end, the previous implementation is extended to enable high order solutions on mixed triangular-quadrilateral meshes and elements with curved edges. In addition, an exact implementation for no-slip boundary conditions is included for curved edges; circular arcs and NACA 4-digit foils, where analytic expressions for the geometry are used to compute the required metrics. To the author's knowledge, the methodologies presented in this chapter conform a novel approach for the simulation of fluid-structure interaction problems involving rotation.

The solver capabilities are tested for a number of problems governed by the incompressible Navier-Stokes equations on static and rotating meshes: the Taylor vortex problem, the Wannier flow problem for a circular cylinder, a rotating square cylinder and a static

and rotating symmetric NACA0015 foil.

The sliding mesh technique allows for mesh motion where an inner mesh section rotates with respect to an outer static mesh. This relative motion creates hanging nodes at the interface between static and rotating elements and necessitates the use of elements with internal curved edges (see chapter 1 section 1.11). In addition, to achieve high order solutions near external boundaries (i.e. walls), curved boundary elements are essential. To clarify these concepts before continuing, Fig. 3.1 depicts an example of a mesh, where the static and rotating subdomains, the curved edges for a symmetric foil and the curved sliding mesh with the associated hanging nodes have been highlighted.

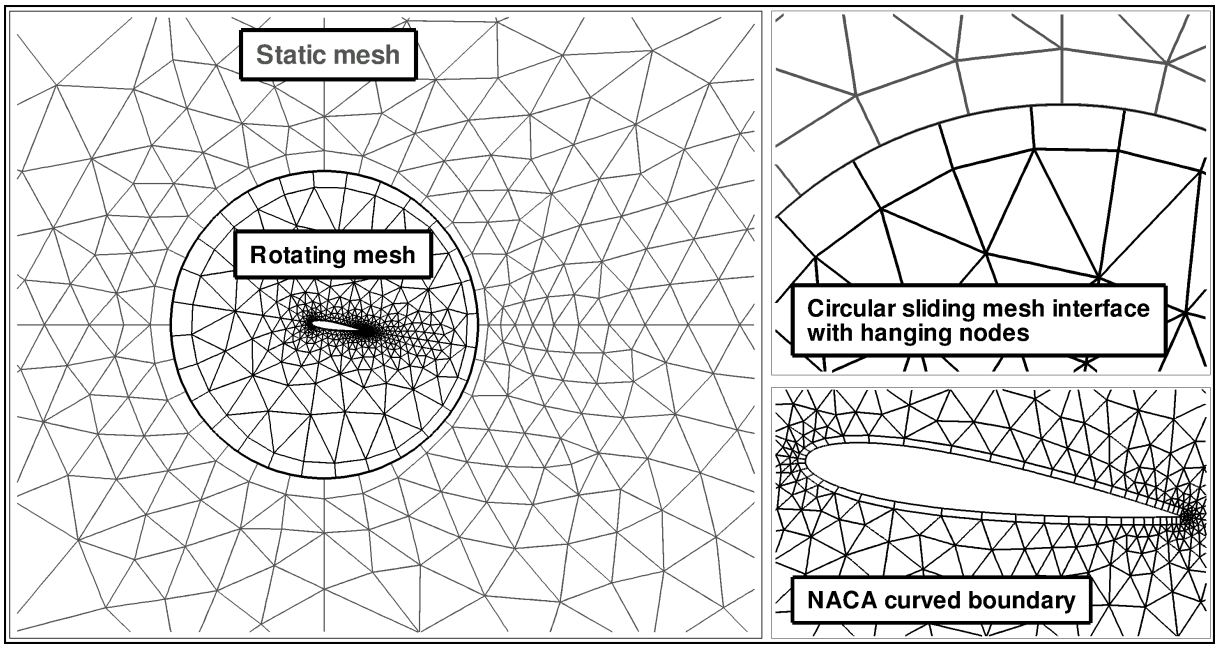

Figure 3.1: Mixed triangular-quadrilateral mesh for a symmetric foil with curved boundaries and a circular sliding mesh. Static and rotating subdomains are distinguished.

The present work shows that in the DG context, the *geometric incompatibility* (see chapter 1 section 1.11.3.1 for definitions) arising from the hanging nodes due to mesh rotation does not cause loss of exponential convergence properties. Further, it is shown that the *functional incompatibility* originating from the use of triangular and quadrilateral element types in combination with orthogonal modal basis functions, does not present a problem. Finally, the sliding mesh implementation shows high order spatial accuracy

and no degradation in temporal convergence rates when solving the incompressible NS equations.

To account for the relative mesh movement of the inner mesh with respect to the static outer mesh, it is advantageous to write the equations for fluid motion in their Arbitrary Lagrangian-Eulerian (ALE) form [44]. The ALE description was first introduced for finite difference methods and unstructured meshes for fluid simulation [78] and subsequently extended to finite elements [87]. A wide explanation of this method can be found in [44].

The ALE approach is generally used for dynamically deforming mesh elements (i.e. arbitrary node movement) and has been widely explored in combination with DG methods to solve hyperbolic type equations including the compressible NS equations (see for example [115], [138], [169] or [126]). As for the incompressible ALE form of the NS equations with arbitrary mesh movement, work was limited for a long time to conformal h/p Spectral discretisations [17]. However, very recent work combines this technique with a DG approach [145].

Deforming element techniques (e.g. ALE for deforming elements) require either generally small body motions or remeshing for large motions that would have otherwise lead to unacceptably element distortions. To avoid these limitations, an appealing approach is provided by combining the ALE approach with the sliding mesh technique. This method is particularly suitable to problems where the mesh movement is known a priori; e.g. rigid body rotation without mesh deformation. We chose to follow this approach, the ALE formulation with sliding meshes for non deforming elements, and summarise some of its advantages below:

1. Inertially fixed and rotating objects can be present in the same simulation (see examples in chapter 6).
2. No remeshing is required (with its associated computational cost) as mesh elements do not distort enabling unlimited rotation as opposed to non-sliding methods in which large distortions and subsequent remeshing need to be handled.

3. No projection of the solution into a new mesh is required to advance the solution in time, which is generally a non-conservative process.
4. No interpolation through the sliding interface is required (as in low order methods) which would introduce high numerical errors, destroying the high order properties of the method.

The ALE approach with sliding mesh interfaces, has been previously studied in the context of h/p conformal Spectral methods that require *mortar* techniques for subdomain linking [3]. Recently, a sliding mesh capability has been described for the incompressible NS equations [38], where the authors used *isogeometric analysis* and a low order conformal discretisation (using NURBS) coupled with a special treatment for the sliding curved interface exploiting ideas from DG methods. Purely DG discretisations in conjunction with the sliding mesh approach, have been used for the solution of electromagnetic problems (i.e. Maxwell equations) [2], [23]. In [23], it was shown that to solve the Maxwell eigensystem with sliding mesh interfaces, *mortar* techniques are beneficial.

To the author's knowledge, the work presented herein details the first high order ($order \geq 3$) DG solver with sliding meshes for the solution of the incompressible NS equations. The method presents a novel approach, where a non-conformal DG discretisation is used in all elements and the curved sliding mesh interface is approximated through an analytical mapping for the description of the circular interface. It is shown that the developed approach and implementation does not require of *mortar* type techniques to accurately resolve the incompressible NS equations. In addition, an analytical mapping to account for curved external boundary conditions (i.e. walls) is introduced, to represent NACA 4-digit foils as required by high order techniques to obtain smooth solutions.

3.1 Chapter outline

This chapter is organised as follows. Firstly, section 3.2 introduces the ALE methodology used to discretise the equations in time (section 3.2.1) and space (section 3.2.2). Emphasis is placed on the required mappings to account for the curved edges (section 3.2.2.3) and the discretisation and implementation of surface integrals for curved sliding interfaces (section 3.2.2.4). Section 3.4 includes various test cases to assess the accuracy, convergence properties and usability of the solver. Namely, the Taylor vortex problem and the Wannier flow problem, are used to examine spatial and temporal convergence rates. In addition, results are reported for a rotating square cylinder and a NACA0015 foil under static and rotating conditions. The foil cases enable the exploration of the numerical properties associated to the analytical NACA boundary conditions.

3.2 Methodology

The formulation, introduced in the previous chapter, is extended to account for relative movement between mesh regions in the computational domain. To solve the incompressible NS equations in primitive variable formulation in a unique domain composed of non overlapping static and moving regions, two distinct subdomains may be defined: $\Omega(t) = \Omega_{sta} \cup \Omega_{rot}(t)$ where $\Omega_{sta} \cap \Omega_{rot}(t) = \Gamma_{sta-rot}$ and $\Gamma_{sta-rot}$ defines the circular sliding mesh interface. The non-dimensional incompressible NS equations in ALE form [44] can be written (in the absence of body forces):

$$\frac{\partial \mathbf{u}}{\partial t} + ((\mathbf{u} - \mathbf{w}).\nabla)\,\mathbf{u} = -\nabla p + \frac{1}{\mathrm{Re}}\nabla^2 \mathbf{u}\ , \quad \nabla.\mathbf{u} = 0 \quad \text{in } \Omega(t), \tag{3.2.1}$$

with appropriate boundary conditions and initial condition (see chapter 2), where $\mathbf{u} = (u, v)^T$ and $\mathbf{w}$ are non-dimensionalised using the characteristic free stream flow velocity magnitude U and represent the flow and mesh velocities and Re is the Reynolds number.

Remark: *Let us note that the ALE form of the NS equations only requires modifica-*

tion of the non-linear terms, which are written in convective form $\mathbf{N}(\boldsymbol{u}) = ((\boldsymbol{u} - \boldsymbol{w}).\nabla)\,\boldsymbol{u}$ *(further details can be found in [44] and section 3.2.2.2).*

Let us distinguish between the static and rotating subdomain meshes (see Fig. 3.1) by setting:

$$\mathbf{w} = \begin{cases} \mathbf{0} & \forall el \in \Omega_{sta} \\ \\ \boldsymbol{\varpi} \times \mathbf{x} & \forall el \in \Omega_{rot}(t) \end{cases}, \tag{3.2.2}$$

where el are the mesh elements in $\Omega(t)$, $\boldsymbol{\varpi} = (0, 0, L\omega/U)$ is the non-dimensional mesh rotational velocity and ω represents the dimensional mesh rotational speed. As seen in chapter 2, to solve the described system, the author has chosen to discretise temporally using a splitting method and a high order DG-SIPG method for spatial discretisation. The following section 3.2.1 details the modifications required by the splitting scheme to account for rotating mesh elements using the ALE formulation. In addition, in section 3.2.2, the DG-SIPG formulation is extended to accommodate for curved elements and sliding mesh interfaces.

3.2.1 Arbitrary Lagrangian-Eulerian temporal discretisation

The ALE version of the algorithm described in chapter 2 has been previously used in the context of h/p Spectral discretisations for arbitrary mesh deformation in [17], and for rotating geometries in [3] using a *mortar* technique. The resulting ALE temporally discretised momentum equation is given by:

$$\begin{aligned} \frac{\gamma_0 \mathbf{u}^{n+1} - \alpha_0 \mathbf{u}^n - \alpha_1 \mathbf{u}^{n-1}}{\Delta t} = & - \nabla p^{n+1} + \frac{1}{\mathrm{Re}} \nabla^2 \mathbf{u}^{n+1} \\ & - \beta_0 [((\mathbf{u}^n - \mathbf{w}^n).\nabla)\,\mathbf{u}^n] - \beta_1 \left[\left((\mathbf{u}^{n-1} - \mathbf{w}^{n-1}).\nabla \right) \mathbf{u}^{n-1} \right]. \end{aligned} \tag{3.2.3}$$

Again, second order Adams-Bashforth values for the coefficients are used: $\gamma_0 = 3/2$, $\alpha_0 = 2$, $\alpha_1 = -1/2$, $\beta_0 = 2$ and $\beta_1 = -1$. The algorithm for the incompressible NS

equations in ALE form, that accounts for rotating mesh motion, follows through sequential solution of equations 3.2.4 to 3.2.7, where Eq. 3.2.6 and 3.2.7 are computed using the DG-SIPG variational framework (as described in chapter 2 section 2.2.2). For rigid mesh motion, the mesh movement is known a priori and thus the position of the new nodes can be analytically computed as shown by Eq. 3.2.5. The sequence of steps reads:

$$\begin{aligned}
\tilde{\mathbf{u}} = (\alpha_0/\gamma_0)\mathbf{u}^n + (\alpha_1/\gamma_0)\mathbf{u}^{n-1} &- (\beta_0\Delta t/\gamma_0)[((\mathbf{u}^n - \mathbf{w}^n).\nabla)\,\mathbf{u}^n] \\
&- (\beta_1\Delta t/\gamma_0)\left[\left((\mathbf{u}^{n-1} - \mathbf{w}^{n-1}).\nabla\right)\mathbf{u}^{n-1}\right], \quad (3.2.4)\\
\mathbf{x}^{n+1} &= \mathbf{R}(\Delta\theta)(\mathbf{x}^n - \mathbf{x}_0) + \mathbf{x}_0, \quad (3.2.5)\\
-\nabla^2 p^{n+1} &= -\frac{\gamma_0}{\Delta t}\nabla.\tilde{\mathbf{u}}, \quad (3.2.6)\\
-\nabla^2\mathbf{u}^{n+1} + \frac{\mathrm{Re}\gamma_0}{\Delta t}\mathbf{u}^{n+1} &= \mathrm{Re}(\frac{\gamma_0}{\Delta t}\tilde{\mathbf{u}} - \nabla p^{n+1}), \quad (3.2.7)
\end{aligned}$$

where $\Delta\theta = \Delta t L\omega/U$ is the non-dimensional discrete differential rotation angle, corresponding to the difference between angle at time n (i.e. θ^n) and $n+1$ (i.e. θ^{n+1}). The rotation center for the associated circle is represented by $\mathbf{x}_0$. Furthermore, the rotation matrix $\mathbf{R}(\Delta\theta)$ is defined as:

$$\mathbf{R}(\Delta\theta) = \begin{pmatrix} \cos\,\Delta\theta & -\sin\,\Delta\theta \\ \sin\,\Delta\theta & \cos\,\Delta\theta \end{pmatrix}. \quad (3.2.8)$$

Eq. 3.2.6 defines a Poisson equation for the pressure which is closed using the following boundary condition at walls and inflow:

$$\begin{aligned}
\frac{\partial p}{\partial n}^{n+1} = \mathbf{n}.\frac{\partial \mathbf{u}}{\partial t}^{n+1} &- \beta_0\mathbf{n}.\left(((\mathbf{u}^n - \mathbf{w}^n).\nabla)\,\mathbf{u}^n + \frac{1}{\mathrm{Re}}\nabla\times\boldsymbol{\omega}^n\right) \\
&-\beta_1\mathbf{n}.\left(\left((\mathbf{u}^{n-1} - \mathbf{w}^{n-1}).\nabla\right)\mathbf{u}^{n-1} + \frac{1}{\mathrm{Re}}\nabla\times\boldsymbol{\omega}^{n-1}\right), \quad (3.2.9)
\end{aligned}$$

where $\boldsymbol{\omega} = \nabla\times\mathbf{u}$ is the vorticity. Chapter 2 introduced the required boundary conditions to close the Poisson and Helmholtz equations resulting from the splitting scheme consid-

ered. This section particularises these boundaries for the no-slip boundary condition (i.e. walls) when present in the rotating subdomain $\Omega_{rot}(t)$.

At rotating walls, a Dirichlet boundary condition for the velocity (Helmholtz equation Eq. 3.2.7) and a Neumann condition for the pressure (Poisson equations Eq. 3.2.6) are required. To define the Dirichlet condition for the velocity, let us introduce the velocity relative to the moving zone: $\mathbf{u}_{rel} = \mathbf{u} - \mathbf{w}$ and note that for rotating no-slip boundary conditions, the relative velocity vanishes ($\mathbf{u}_{rel} = \mathbf{0}$) leading to $\mathbf{u} = \mathbf{w}$.

The Poisson boundary condition defined in chapter 2 Eq. 3.2.9 can be improved by expanding the time derivative of the velocity and considering $\mathbf{u}_{rel} = \mathbf{0}$ at no-slip moving boundaries:

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t}^{n+1} &= \frac{\partial (\mathbf{u}_{rel} + \mathbf{w})}{\partial t}^{n+1} = \frac{\partial \mathbf{w}}{\partial t}^{n+1} = \frac{\partial [\boldsymbol{\varpi} \times \mathbf{x}]}{\partial t}^{n+1} \\ &= \boldsymbol{\varpi} \times \frac{\partial \mathbf{x}}{\partial t}^{n+1} = \boldsymbol{\varpi} \times \mathbf{w} = \boldsymbol{\varpi} \times \boldsymbol{\varpi} \times \mathbf{x}^{n+1}, \end{aligned} \tag{3.2.10}$$

where the following equalities have been used: $\mathbf{w} = \boldsymbol{\varpi} \times \mathbf{x}$, $\frac{\partial \mathbf{x}}{\partial t}^{n+1} = \mathbf{w}^{n+1}$. In addition, the rotational speed has been considered constant: $\boldsymbol{\varpi}^{n+1} = \boldsymbol{\varpi}^n = \boldsymbol{\varpi}$ (i.e. zero angular acceleration $\frac{\partial \boldsymbol{\varpi}}{\partial t}$).

3.2.2 High order DG-SIPG spatial discretisation

Chapter 2 section 2.2.2 introduced the DG-SIPG spatial discretisation necessary to solve the NS equations using the temporal splitting scheme selected. This section expands the previous polynomial spaces to include meshes where quadrilateral elements with either straight or curved edges. These new curved elements can be used to define external curved boundaries or internal curved interfaces as necessary to account for the sliding mesh technique.

3.2.2.1 Preliminaries: extended spaces

Let $\Omega = \Omega_{sta} \cup \Omega_{rot}$ be a domain in $\mathbb{R}^2$ fixed at an instant in time, with boundaries $\partial\Omega$ (straight or curved) of Dirichlet (Γ_D) or Neumann (Γ_N) type, where $\Gamma_D \cap \Gamma_N = \emptyset$. Let us introduce a triangular-quadrilateral tessellation of N^{el} elements: $\Omega_h = \{el^{tri}, el^{quad}\}$ of Ω with external boundaries $\partial\Omega_h$ and interior boundaries Γ_h (straight or curved). In addition, let us define $\mathbf{n}_{\Gamma-local}$ as the local pointwise unit outward pointing normal (i.e. $\mathbf{n}_{\Gamma-local} = \mathbf{n}_\Gamma$ for straight edges), which is required when considering curved edges.

The extended discontinuous finite element space $D_k(\Omega_h) = D_k^{tri}(\Omega_h) \cup D_k^{quad}(\Omega_h)$ is composed by:

$$\begin{aligned} D_k^{tri}(\Omega_h) &= \{\upsilon_h \in L^2(\Omega) : \forall el^{tri} \in \Omega_h, \upsilon_h|_{el^{tri}} \in \mathbb{P}_k(el^{tri})\}, \\ D_k^{quad}(\Omega_h) &= \{\upsilon_h \in L^2(\Omega) : \forall el^{quad} \in \Omega_h, \upsilon_h|_{el^{quad}} \in \mathbb{Q}_k(el^{quad})\}, \end{aligned}$$

and their vector version $[D_k^{tri}(\Omega_h)]^2$, $[D_k^{quad}(\Omega_h)]^2$ and $[D_k(\Omega_h)]^2 = [D_k^{tri}(\Omega_h)]^2 \cup [D_k^{quad}(\Omega_h)]^2$ where $\mathbb{P}_k(el^{tri})$ and $\mathbb{Q}_k(el^{quad})$ denote the space of polynomials of order less than or equal to k for triangular and quadrilateral elements respectively. Note that equal polynomial orders are considered for triangles and quadrilaterals and equal order elements for velocity and pressure.

3.2.2.2 Non-linear terms under rotational motion

The formulation for the non-linear terms introduced in chapter 2 section 2.2.2.2 is re-considered in this section to account for the ALE form of the NS equations. A similar

formulation (under its weak setting) is proposed: find $\mathbf{u}_h \in [D_k(\Omega_h)]^2$ such that:

$$\begin{aligned}\int_{\Omega_h} (\mathbf{N}(\mathbf{u}_h)) \,.\boldsymbol{v}_h d\mathbf{x} &= \int_{\Omega_h} (((\mathbf{u}_h - \mathbf{w}).\nabla)\mathbf{u}_h) \,.\boldsymbol{v}_h d\mathbf{x} \\ &= \sum_{el\in\Omega_h} \int_{el} (((\mathbf{u}_h - \mathbf{w}).\nabla)\mathbf{u}_h) \,.\boldsymbol{v}_h d\mathbf{x} \\ &\quad +1/2 \sum_{el\in\Omega_h} \int_{el} ((\nabla.(\mathbf{u}_h - \mathbf{w}))\,\mathbf{u}_h) \,.\boldsymbol{v}_h d\mathbf{x}, \quad \forall \boldsymbol{v}_h \in [D_k(\Omega_h)]^2.\end{aligned} \tag{3.2.11}$$

Let us note that this formulation makes use of the property $\nabla.\mathbf{w} = 0$, which is valid for rigid body motion.

As introduce in the previous chapter, the explicit treatment of the non-linear terms introduces a Courant-Friedrichs-Lewy (CFL) type restriction on the time step. The CFL estimate for high order spatial methods leads to $\triangle t < \triangle t_{CFL} = \mathcal{O}(h/U_{max}k^2)$, where h is the mesh element size, U_{max} the maximum velocity and k represents the polynomial order (see chapter 2 section 2.2.2.2). When considering the ALE formulation, this estimate becomes: $\triangle t < \triangle t_{CFL} = \mathcal{O}(h/|\mathbf{u} - \mathbf{w}|_{max}k^2)$, where $\mathbf{u}$ and $\mathbf{w}$ are the flow and mesh velocities. Let us consider the limiting case $|\mathbf{w}|_{max} = R\omega$ (i.e. the sliding interface), where R defines the radius of the circle associated to the sliding interface and ω the rotational speed. It is possible to identify the most stringent case, which occurs when $\mathbf{w}$ and $\mathbf{u}$ (with magnitude $|\mathbf{u}| = U$) have opposite directions and leads to a modified estimate for the limiting time step: $\triangle t < \triangle t_{CFL} = \mathcal{O}(h/(U + R\omega)k^2)$.

3.2.2.3 Quadrilateral elements with straight and curved edges

This section introduces the necessary mappings to relate computational to physical space for quadrilateral elements with straight or one curved edge. Some advantages for the selection of quadrilateral elements with curved edges, over triangular elements, have been outlined previously in section 2.2.2.4 chapter 2. A brief reminder, of the notation introduced in section 2.2.2.4, where the mapping for triangular elements with straight edges was defined, is presented first.

The mapping Ψ defines the transformation from the reference element I_{el} in computational space $\boldsymbol{\xi} = (\xi_1, \xi_2)$ to the element el in physical space $\mathbf{x} = (x, y)$ as $I_{el} := \Psi^{-1}(el)$, with $|\mathbf{J}_\Psi|$ the determinant of the Jacobian associated with the mapping . The transformation of the edges Γ of element el in physical space to the mapped edge in the computational space is defined through $\Gamma = \boldsymbol{C}([-1, 1])$, with the associated norm of the differential parametrisation (i.e. differential length): $|\mathbf{J}_{\boldsymbol{C}}(t)|$. The volume and line integrals of a general polynomial function $f(\mathbf{x})$ can then be equated from physical space to their equivalent form in computational space:

$$\int_{el} f(\mathbf{x}) d\mathbf{x} = \int_{I_{el}} f(\boldsymbol{\xi}) |\mathbf{J}_\Psi| d\boldsymbol{\xi}, \tag{3.2.12}$$

$$\int_{\Gamma} f(\mathbf{x}) d\mathbf{s} = \int_{-1}^{1} f(\boldsymbol{C}(t)) |\mathbf{J}_{\boldsymbol{C}}(t)| dt. \tag{3.2.13}$$

In addition to the triangular element mapping (**1a and 1b - Triangular map**) defined previously, this section defines the following mappings Ψ, which are depicted in Fig. 3.2:

- **2 - Quadrilateral map:** standard square (ξ_1, ξ_2) to arbitrary quadrilateral (x, y) (see [62] for details).
- **3 - Curved-quadrilateral map:** standard square (ξ_1, ξ_2) to arbitrary quadrilateral with one curved edge (x, y) (see [62] or [101] for details).

The mappings defined in Fig. 3.2 lead to a constant determinant of the Jacobian $|\mathbf{J}_\Psi|$ for straight sided quadrilateral elements with parallel edges only but not for general quadrilaterals (see appendix A for details).

It is important to remember at his point, that in addition to the determinant of the Jacobian $|\mathbf{J}_\Psi|$ and the norm of the differential length $|\mathbf{J}_{\boldsymbol{C}}(t)|$ required to evaluate volume and surface integrals (Eq. 3.2.12 and Eq. 3.2.13), partial derivatives from physical to computational space (i.e. gradients) and edge normals are also required. These metrics can be analytically obtained through the defined maps. A detailed description of these metrics can be found in in appendix A.

The *Curved-quadrilateral map-3*, depicted in Fig. 3.2, was introduced in [62] and has previously been used in the context of conformal discretisations (i.e. h/p Spectral [101]). This map is necessary to account for external curved boundaries and the interior curved sliding interface.

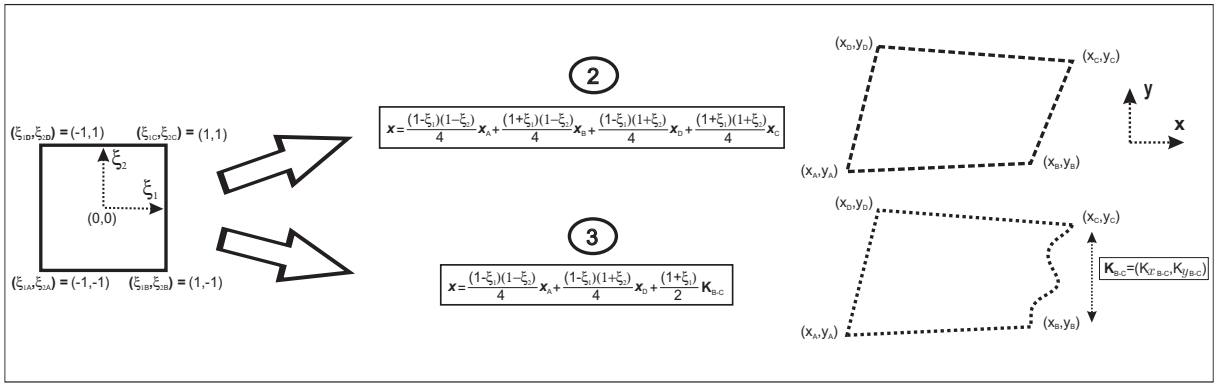

Figure 3.2: Mapping from computational space to physical space for quadrilateral elements with straight edges (map 2) or one curved edge (map 3).

The general approach used in high order methods, to account for curved elements, is to use *iso-parametric* mapping. This technique approximates the curved edge using polynomial functions (splines or NURBS) which have the same degree that the polynomials used within the element to approximate the flow solution [101]. Subsequently, the approximated curve (i.e. fitted polynomial) is used to compute the required metrics (i.e. $|\mathbf{J}_\Psi|$, $|\mathbf{J}_{\boldsymbol{C}}(t)|$, gradients and normals).

The present approach is somewhat different in the sense that it does not require to fit a polynomial to the non-polynomial curve, defining the curved edge, to compute the required metrics from the fitted polynomial. Instead, the analytical expression that defines the curved edges is used to computed the required metrics (at quadrature points). Conceptually, the selected approach is similar to the NURBS-Enhanced FEM [154] approach since both allow for decoupling of the geometry (and associated metrics) with respect to the polynomial space required to approximate the solution. The approach considered in this thesis can be seen as less general, but it may be argued that an analytical expression

of the geometry can always be obtained (e.g. fitting a polynomial of arbitrary order).

Returning to the *Curved-quadrilateral map-3* that accounts for one curved edge; only the edge B-C is considered curved which corresponds to the edge $\boldsymbol{\xi} = (\xi_1 = 1, \xi_2)$ in the computational space as shown in Fig. 3.2. This can be ensured by the appropriate node numbering when generating the mesh (but generalisation to other edges is direct). The function $\boldsymbol{K}_{B-C}(\xi_2) = (K_{x_{B-C}}(\xi_2), K_{y_{B-C}}(\xi_2))$ in Fig. 3.2 defines the shape of the curved edge. To calculate surface integrals on external curved edges, using the defined mappings, it suffices to replace $\mathbf{C}(t = \xi_2) = \mathbf{K}_{B-C}(\xi_2)$ in Eq. 3.2.13. The flux treatment for curved edges lying on the sliding mesh interface (with a hanging node) is presented in section 3.2.2.4.

Finally, the differential length for quadrilateral elements with straight edges can be computed as: $|\mathbf{J}_{\boldsymbol{C}}(t)| = \sqrt{[\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2]}dt = \frac{|trace|}{2}$, where $|trace|$ is the length of the element edge in 2D. However, when curved edges are considered, these metrics require the evaluation of the expression:

$$|\mathbf{J}_{\boldsymbol{C}}(t)| = \sqrt{\left(\frac{dK_{x_{B-C}}(\xi_2)}{d\xi_2}\right)^2 + \left(\frac{dK_{y_{B-C}}(\xi_2)}{d\xi_2}\right)^2}d\xi_2. \tag{3.2.14}$$

Further details can be found in appendix A.

This work considers three types of curved edges: circular arcs, symmetric NACA 4-digit edges and cambered NACA 4-digit edges.

Circular arced elements: The circular parametrisation, which was previously introduced in [101], is defined as:

$$\begin{aligned} K^{CIRC}_{x_{B-C}}(\xi_2) &= R\cos\left(\frac{1-\xi_2}{2}\theta_B + \frac{1+\xi_2}{2}\theta_C\right), \\ K^{CIRC}_{y_{B-C}}(\xi_2) &= R\sin\left(\frac{1-\xi_2}{2}\theta_B + \frac{1+\xi_2}{2}\theta_C\right), \end{aligned} \tag{3.2.15}$$

where $R = R_B = R_C$ and θ_B, θ_C are the polar-coordinates of nodes B and C with respect to an origin at the centre of curvature. This parametrisation is used hereafter to account for the circular sliding interface (see details in section 3.2.2.4) and can be utilised to define circular external boundaries (see example in section 3.4.2 for the Wannier flow problem). It can be shown that the circular parametrisation $\boldsymbol{K}_{B-C}^{CIRC}(\xi_2)$ provides a constant determinant of the Jacobian $|\mathbf{J}_\Psi|$ provided $\theta_A = \theta_B$, $\theta_C = \theta_D$ and $R_B = R_C$, $R_A = R_D$. In the present implementation, however, the general case, where the determinant of the Jacobian is not constant, is retained and enhances the flexibility in the mesh generation process.

Symmetric NACA 4-digit elements: For the NACA 4-digit (see for example Abbott and Von-Doenhoff [1] for definitions), the analytical expression for the NACA foil can be used to define the mapping required to approximate curved external boundaries:

$$\begin{aligned} K_{x_{B-C}}^{NACA}(\xi_2) &= \frac{1-\xi_2}{2}x_B + \frac{1+\xi_2}{2}x_C, \qquad (3.2.16)\\ K_{y_{B-C}}^{NACA}(\xi_2) &= \pm\frac{\zeta}{0.2}(\tau_0\sqrt{K_{x_{B-C}}^{NACA}(\xi_2)} - \tau_1[K_{x_{B-C}}^{NACA}(\xi_2)] - \tau_2[K_{x_{B-C}}^{NACA}(\xi_2)]^2 \\ &\quad -\tau_3[K_{x_{B-C}}^{NACA}(\xi_2)]^3 - \tau_4[K_{x_{B-C}}^{NACA}(\xi_2)]^4), \end{aligned}$$

where ζ is the foil maximum thickness (with respect to unit chord $c = 1$) and $\tau_0 = 0.29690$, $\tau_1 = 0.12600$, $\tau_2 = 0.35160$, $\tau_3 = 0.28430$ and $\tau_4 = 0.20250$ are the NACA constants required to define the profile geometry for symmetric NACA foils. The foil nose definition requires a circular mapping with radius $r_{nose} = 1.1019\zeta^2$ and Eq. 3.2.15 is used. The transition from the analytic nose and foil body formulations is not particularly smooth if the value suggested in [1] ($x/c = 0.005$) is used. By selecting this transition to occur at $x/c = 0.002$, a smooth enough geometric transition, is obtained for the foils tested in this thesis (i.e. $\zeta = 12\%$ and 15%). In all cases, the NACA blunt trailing edge is extended, with straight edges, to obtain a sharp trailing edge, which extends the chord to $c \simeq 1.01$. This mapping is defined for $0 < x < 1$ but can be generalised to arbitrary values when

combined with translation and rotation.

Cambered NACA 4-digit elements: The previous definition for the symmetric NACA can be extended to account for camber (see Abbott and Von-Doenhoff [1]). To this end, let us introduce the coordinates for the mean camber line y_{camb} and the coordinates for the upper and lower foil surfaces $(K_{x_{B-C}}^{CAMB-UPP}(\xi_2), K_{y_{B-C}}^{CAMB-UPP}(\xi_2))$ and $(K_{x_{B-C}}^{CAMB-LOW}(\xi_2), K_{y_{B-C}}^{CAMB-LOW}(\xi_2))$ respectively. The previous definition for the symmetric NACA $(K_{x_{B-C}}^{NACA}(\xi_2), K_{y_{B-C}}^{NACA}(\xi_2))$ can be used to define the coordinates for the cambered profile:

$$\begin{aligned}
K_{x_{B-C}}^{CAMB-UPP}(\xi_2) &= K_{x_{B-C}}^{NACA}(\xi_2) - \left(K_{y_{B-C}}^{NACA}(\xi_2)\sin\delta(\xi_2)\right), \\
K_{y_{B-C}}^{CAMB-UPP}(\xi_2) &= y_{camb}(\xi_2) + \left(K_{y_{B-C}}^{NACA}(\xi_2)\cos\delta(\xi_2)\right), \\
K_{x_{B-C}}^{CAMB-LOW}(\xi_2) &= K_{x_{B-C}}^{NACA}(\xi_2) + \left(K_{y_{B-C}}^{NACA}(\xi_2)\sin\delta(\xi_2)\right), \\
K_{y_{B-C}}^{CAMB-LOW}(\xi_2) &= y_{camb}(\xi_2) - \left(K_{y_{B-C}}^{NACA}(\xi_2)\cos\delta(\xi_2)\right),
\end{aligned} \tag{3.2.17}$$

where $y_{camb}(\xi_2)$ and $\delta(\xi_2)$ are defined as:

$$y_{camb}(\xi_2) = \begin{cases} \left(\frac{m}{p^2}\right) K_{x_{B-C}}^{NACA}(\xi_2)\left(2p - \frac{K_{x_{B-C}}^{NACA}(\xi_2)}{c}\right);\ 0 \le K_{x_{B-C}}^{NACA}(\xi_2) \le pc, \\ \left(\frac{m}{(1-p)^2}\right)\left(c - K_{x_{B-C}}^{NACA}(\xi_2)\right)\left(1 + \frac{K_{x_{B-C}}^{NACA}(\xi_2)}{c} - 2p\right);\ pc \le K_{x_{B-C}}^{NACA}(\xi_2) \le c, \end{cases} \tag{3.2.18}$$

and

$$\delta(\xi_2) = atan\left(\frac{dy_{camb}(\xi_2)}{dK_{x_{B-C}}^{NACA}(\xi_2)}\frac{dK_{x_{B-C}}^{NACA}(\xi_2)}{d\xi_2}\right). \tag{3.2.19}$$

To define the cambered NACA, the maximum camber (m) and the location of maximum camber (p), have been introduced. In this case, the expressions have been generalised to foil chords different than unity. Finally, it can seen that for $m = 0$ this definition reduces to the symmetric NACA map (Eq. 3.2.16).

3.2.2.4 High order sliding meshes

Due to the inner domain rotation relative to the outer static mesh, there is, for arbitrary rotation, the creation of hanging nodes and an associated *geometric incompatibility*. This section shows that this incompatibility does not represent a problem in the DG context.

When implementing the sliding mesh technique, it becomes necessary to revisit the bilinear form defined in section 2.2.2.3, and in particular the last three terms in Eq. 2.2.15, involving surface integrals. A general expression for this term was derived in chapter 2 section 2.2.3 and is repeated here for clarity.

Let ℓ, χ denote the elements sharing the edge Γ, then the contributions of the numerical DG-SIPG flux in Eq. 2.2.15 are $\int_\Gamma t_{ij}^\Gamma ds = \int_\Gamma t_{ij}^{el_\ell el_\chi} ds$, where $i = 1$ to $LDOF$ is the test function index of v_h and $j = 1$ to $LDOF$ is the basis function index of u_h. These flux contributions are defined as:

$$\int_\Gamma t_{ij}^{el_\ell el_\chi} ds = \gamma \frac{1}{2} \int_\Gamma \nabla \phi_j^{el_\chi} . \mathbf{n}_{\Gamma - local} \phi_i^{el_\ell} ds + \delta \frac{1}{2} \int_\Gamma \phi_j^{el_\chi} \nabla \phi_i^{el_\ell} . \mathbf{n}_{\Gamma - local} ds + \rho \frac{\sigma}{|trace|} \int_\Gamma \phi_j^{el_\chi} \phi_i^{el_\ell} ds, \tag{3.2.20}$$

where $\mathbf{n}_{\Gamma - local}$ defines the local normal vector (i.e. $\mathbf{n}_{\Gamma - local} = \mathbf{n}_\Gamma$ for straight edges) and γ, δ and ρ are defined as:

$$\gamma = \begin{cases} 1; & \ell \neq 1 \\ -1; & \ell = 1 \end{cases}, \quad \delta = \begin{cases} 1; & \chi \neq 1 \\ -1; & \chi = 1 \end{cases}, \quad \rho = \begin{cases} 1; & \ell = \chi \\ -1; & \ell \neq \chi \end{cases}. \tag{3.2.21}$$

The present implementation considers that the elements sharing the sliding mesh interface have one curved edge and a unique hanging node per element interface. Furthermore, the hanging node lies on the edge $\boldsymbol{\xi} = (\xi_1 = 1, \xi_2)$ in the computational space (i.e. edge B-C in physical space). A schematic of three elements with curved edges sharing the sliding mesh interface and the resulting hanging node is depicted in Fig. 3.3, where the connectivity between elements in both physical and computational space is detailed. The circular

edge mapping previously defined in Eq. 3.2.15 is used to describe the curved edges of the elements sharing the circular sliding interface. The flux contributions defined in Eq.

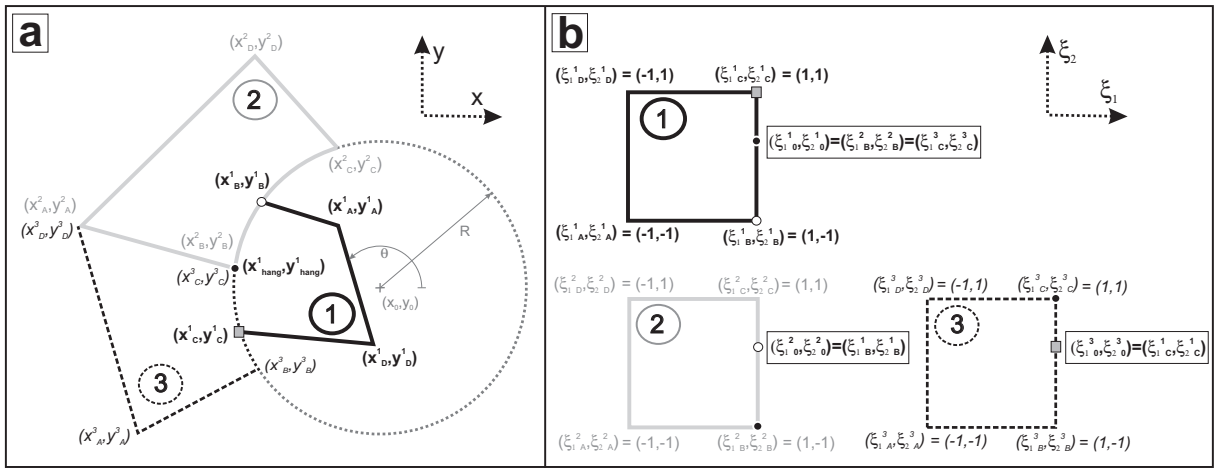

Figure 3.3: Sliding mesh connectivity scheme showing three elements and the shared interfaces with a hanging node in (a) the physical space and (b) the computational space.

3.2.20 over the curved interface (i.e. $\Gamma \in \Gamma_{sta-rot}$) have to take into account the presence of a hanging node and that the edge is split in two segments: Γ_{12} that connects element el_1 and el_2 and Γ_{13} connecting el_1 to el_3. It is thus necessary to extend these surface integrals, defined for a single edge (i.e. only shared by two elements), to account for the interface including a hanging node $\Gamma = \Gamma_{12} \cup \Gamma_{13}$. Before proceeding, it proves useful to redefine the norm of the differential parametrisation (or differential length) $|\mathbf{J}_{\boldsymbol{C}}(t)|$ introduced in Eq. 3.2.13 that defines the surface integral in computational space. Recalling the definition of the circular mapping introduced in Eq. 3.2.15 and setting $\boldsymbol{C}(t) = \mathbf{K}^{CIRC}_{B-C}(\xi_2 = h(t))$, where $h(t)$ represents a linear function to be subsequently detailed; one can define:

$$|\mathbf{J}_{\boldsymbol{K}^{CIRC}_{B-C}}(t)| = \sqrt{\left(\frac{\partial K^{CIRC}_{x_{B-C}}(\xi_2)}{\partial \xi_2}\frac{\partial \xi_2}{\partial t}\right)^2 + \left(\frac{\partial K^{CIRC}_{y_{B-C}}(\xi_2)}{\partial \xi_2}\frac{\partial \xi_2}{\partial t}\right)^2}\, dt. \tag{3.2.22}$$

For edges where no hanging nodes are present, the linear function $h(t)$ reduces to $h(t) = t$ (i.e. $\xi_2 = t$). Let us consider the internal element el_1 as the reference element and follow the notation introduced in Fig. 3.3. Combining Eq. 3.2.13, Eq. 3.2.20 and Eq. 3.2.22,

one obtains:

$$\begin{aligned}
\int_{\Gamma\in\Gamma_{sta-rot}} t_{ij}^{el_\ell el_\chi}(\mathbf{x})ds &= \int_{\Gamma_{12}} t_{ij}^{el_\ell el_\chi}(\mathbf{x})ds_{12} + \int_{\Gamma_{13}} t_{ij}^{el_\ell el_\chi}(\mathbf{x})ds_{13} \\
&= \int_{x_B^1,y_B^1}^{x_{hang}^1,y_{hang}^1} t_{ij}^{el_\ell el_\chi}(\mathbf{x})ds_{12} + \int_{x_{hang}^1,y_{hang}^1}^{x_C^1,y_C^1} t_{ij}^{el_\ell el_\chi}(\mathbf{x})ds_{13} \\
&= \int_{-1}^{1} t_{ij}^{el_\ell el_\chi}(\boldsymbol{K}_{B-C}^{CIRC}(t_1))|\mathbf{J}_{\boldsymbol{K}_{B-C}^{CIRC}}(t_1)|dt_1 \\
&\quad + \int_{-1}^{1} t_{ij}^{el_\ell el_\chi}(\boldsymbol{K}_{B-C}^{CIRC}(t_2))|\mathbf{J}_{\boldsymbol{K}_{B-C}^{CIRC}}(t_2)|dt_2, \qquad (3.2.23)
\end{aligned}$$

where $-1 \leq t_1, t_2 \leq 1$ are the parametrisations for Γ_{12} and Γ_{13} respectively and $\ell, \chi = 1, 2, 3$, but $\ell, \chi \neq 3$ in Γ_{12} and $\ell, \chi \neq 2$ in Γ_{13}. This parametrisation can be related to the computational space by introducing the following intermediate linear mappings $\xi_2 = h(t)$:

$$\begin{aligned}
\xi_2 &= h_{12}(t_1) = \frac{1-t_1}{2}(-1) + \frac{1+t_1}{2}\xi_{2_0}^{\ 1} \quad \text{on } \Gamma_{12}, \\
\xi_2 &= h_{13}(t_2) = \frac{1-t_2}{2}\xi_{2_0}^{\ 1} + \frac{1+t_2}{2}(1) \qquad \text{on } \Gamma_{13}, \qquad (3.2.24)
\end{aligned}$$

where the hanging node on the computational space is defined to map to the hanging node in physical space (see Fig. 3.3): $(\xi_{1_0}^{\ 1}, \xi_{2_0}^{\ 1}) \longleftrightarrow (x_{hang}^1, y_{hang}^1)$. It can also be seen that $\xi_{1_0}^{\ 1} = 1$ (since $\xi_1 = 1$) and that $\xi_{2_0}^{\ 1} = 2\frac{(\theta^1{}_{hang} - \theta^1{}_B)}{(\theta^1{}_C - \theta^1{}_B)} - 1$, with θ_{hang}^1, θ_B^1 and θ_C^1, the corresponding polar angles for the reference element in physical space. The map for the circular edge Eq. 3.2.15 and the intermediate mapping Eq. 3.2.24 can be differentiated to obtain the required definition of the norm of the differential parametrisation $|\mathbf{J}_{\boldsymbol{K}_{B-C}^{CIRC}}(t)|$, defined in Eq. 3.2.22. Substituting this expression into Eq. 3.2.23, one can derive the necessary definition for the fluxes across the curved interfaces with a hanging node:

$$\begin{aligned}
&\int_{\Gamma\in\Gamma_{sta-rot}} t_{ij}^{el_\ell el_\chi}(\mathbf{x})ds = \qquad (3.2.25) \\
&\frac{\theta_C - \theta_B}{4} R \left[(\xi_{2_0}^{\ 1} + 1) \int_{-1}^{1} t_{ij}^{el_\ell el_\chi}(\boldsymbol{K}_{B-C}^{CIRC}(t_1))dt_1 + (1 - \xi_{2_0}^{\ 1}) \int_{-1}^{1} t_{ij}^{el_\ell el_\chi}(\boldsymbol{K}_{B-C}^{CIRC}(t_2))dt_2 \right],
\end{aligned}$$

with $i,j = 1 : LDOF$, $\ell,\chi = 1,2,3$, but $\ell,\chi \neq 3$ in the first integral and $\ell,\chi \neq 2$ in the second. These integrals can now be evaluated using any quadrature rule (e.g. Gauss-Legendre).

The penalty term in Eq. 3.2.20 requires a definition for the $|trace|$, which is chosen to be: $|trace|_{12} = |trace|_{13} = |\theta_C - \theta_B|R$.

3.2.2.5 Modal basis functions for quadrilateral elements

Modal basis functions for triangular elements were introduced in chapter 2 section 2.2.2.5. The present section introduces the necessary modal bases for quadrilateral elements. In this case, also hierarchical tensorial and orthonormal (with respect to the L^2-inner product) expansion bases (see [101] for details) are defined in both physical and computational space following the notation introduced in Fig. 3.2:

$$\phi_i^{quad}(\mathbf{x}) = \phi_{pq}^{quad}(\mathbf{x}) = \varphi_p^{quad^A}(\xi_1)\varphi_q^{quad^B}(\xi_2), \tag{3.2.26}$$

where p,q denote the different components of the tensorial expansion. Here again, the basis functions $\phi(\mathbf{x})$ for quadrilateral can be expressed in terms of principal basis functions $\varphi(\xi_1,\xi_2)$, which are defined in the standard square:

$$\varphi_p^{quad^A}(\xi_1) = \sqrt{\frac{2p+1}{2}}P_p^{(0,0)}(\xi_1)\ , \quad \varphi_q^{quad^B}(\xi_2) = \sqrt{\frac{2q+1}{2}}P_q^{(0,0)}(\xi_2), \tag{3.2.27}$$

where $P_p^{(\alpha,\beta)}(z)$ is the pth order Jacobi polynomial. The tensorial indices are defined by $0 \leq p,q \leq k$ and the Local number of Degrees of Freedom (LDOF) for quadrilateral elements as: $LDOF_{quad} = (k+1)^2$, given a polynomial order k. The principal basis functions on the standard square are depicted in Fig. 3.4.

As for triangular elements, the bases have been normalised to provide a mass matrix that is the identity matrix (i.e. $\mathbf{M} = |\mathbf{J}_\Psi|\mathbf{Id}$) for quadrilaterals with parallel edges (i.e. constant determinant of the Jacobians).

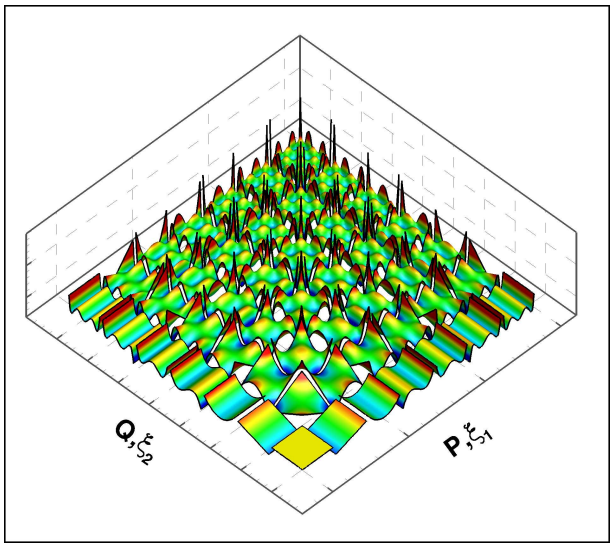

Figure 3.4: Hierarchical modal basis functions for various polynomial $k = 0$ to 6 for quadrilateral elements.

Remark: *When considering a non-constant determinant of the Jacobian together with the defined basis functions for quadrilateral elements, a symmetric mass matrix (i.e.* $\boldsymbol{M} = \boldsymbol{M}^{\mathrm{T}}$*) is obtained, which substantially reduces the cost of computing the mass matrix and subsequent operations involving this matrix, relative to the cost for a full matrix.*

It can be seen that since $LDOF_{quad} \neq LDOF_{tri}$ and no special treatment is performed at element edges, *functional incompatibilities* arise for adjacent triangular and quadrilateral elements, however, the results section 3.4.1.1 shows that these discontinuities do not present a problem in the DG context.

Finally, the derivatives of the principal basis functions can be obtained similarly as for triangular elements:

$$\frac{d\varphi_p^{quad^A}}{d\xi_1}(\xi_1) = \frac{1}{2}(p+1)P_{p-1}^{(1,1)}(\xi_1), \quad \frac{d\varphi_q^{quad^B}}{d\xi_2}(\xi_2) = \frac{1}{2}(q+1)P_{q-1}^{(1,1)}(\xi_2). \quad (3.2.28)$$

3.2.2.6 Numerical quadrature revisited

In this section, the table presented in chapter 2 section 2.2.4, for straight sided triangular elements, is extended to incorporate the minimum number of quadrature points (using

Gauss-Legendre quadrature) required to evaluate volume and surface integrals for quadrilateral elements with non-constant determinant of the Jacobian $|\mathbf{J}_\Psi|$ and with curved edges. Before proceeding, let us recall (see chapter 2 section 2.2.4) that the minimum number of quadrature points to integrate exactly a polynomial f of order $\mathcal{O}(f) = \mathcal{F}$ is $\mathcal{Q}_{min} = (\mathcal{F}+1)/2$.

Table 3.1 summarises the polynomial orders and the required number of quadrature points corresponding to the terms defining the bilinear form Eq. 2.2.15, which includes volume integrals (e.g. Local Mass matrix $\mathbf{M^{el}}$, Local Gradient matrix $\mathbf{K_r^{el}}$ and Local Laplacian matrix $\mathbf{L_r^{el}}$) and surface integrals (e.g. Penalty terms and flux terms contributing to the Flux matrix $\mathbf{T}$) for quadrilateral elements. Since only quadrilateral elements with a non-constant determinant of the Jacobian (i.e. quadrilateral elements with non parallel edges or with one curved edge) are considered, the general formulation to evaluate volume integrals Eq. 2.2.36, introduced in chapter 2 section 2.2.4, is retained. To calculate the number of quadrature points for this type of elements, it is advantageous to define the order of the determinant of the Jacobian $\mathcal{O}(|\mathbf{J}_\Psi|) = \mathcal{J}$ (e.g. $\mathcal{J} = 0$ if $|\mathbf{J}_\Psi|$ is constant over the element). In addition, let us introduce the order of the differential length $\mathcal{O}(|\mathbf{J}_{\boldsymbol{C}}(t)|) = \mathcal{D}$ and the order of the non-constant normal (i.e. curved edges) $\mathcal{O}(n_\Gamma) = \mathcal{N}$. As seen for triangular elements, the non-linear terms require the integration of polynomials of order: $\mathcal{O}(f) = \mathcal{F} = \mathcal{O}(\int \phi_i(\mathbf{x})\phi_j(\mathbf{x})\phi_k(\mathbf{x})d\mathbf{x}) \approx 3k + \mathcal{J}$. Integration of the non-linear terms require more quadrature points that all the other integrals.

Remark: *In practice, one can set* ${\mathcal{Q}^{quad}}_{min} = 3k$ *as a conservative estimate. Inspection of the required number of quadrature points to integrate the Flux matrix terms, shows that exact integration is provided when* $\mathcal{D} + \mathcal{N} < 4k - 1$. *In addition, inspection of the non-linear terms, leads to* $\mathcal{J} < 3k - 1$. *Both requirements are largely fulfilled if polynomials of the order* k *are used to define the geometry and leave margin for accurate integration of non-polynomial functions (see demonstration below).*

To verify the last remark, let us analyse the necessary number of quadrature points to

	$\mathcal{F}^{quad}$	${\mathcal{Q}^{quad}}_{min}$
Mass matrix	$2k + \mathcal{J}$	$k + (\mathcal{J} + 1)/2$
Gradient matrix	$2k - 1 + \mathcal{J}$	$k + \mathcal{J}/2$
Laplacian matrix	$2k - 2 + \mathcal{J}$	$k + (\mathcal{J} - 1)/2$
Flux matrix	$2k + \mathcal{D} + \mathcal{N}$	$k + (\mathcal{D} + \mathcal{N} + 1)/2$
Non linear terms	$3k + \mathcal{J}$	$3/2k + (\mathcal{J} + 1)/2$

Table 3.1: Required number of Gauss-Legendre quadrature points (${\mathcal{Q}^{quad}}_{min}$) to evaluate volume and surface integrals consisting of polynomials of degree $\mathcal{F}^{quad}$ for quadrilateral elements; k denotes the order of the basis and test functions. $\mathcal{J} = \mathcal{O}(|\mathbf{J}_\Psi|)$, $\mathcal{D} = \mathcal{O}(|\mathbf{J}_{\boldsymbol{C}}(t)|)$ and $\mathcal{N} = \mathcal{O}(n_\Gamma)$ define the orders of the non-constant determinant of the Jacobian, the edge differential length and edge normal, respectively.

accurately solve a linear Poisson problem (detailed in appendix C section C.2.1) using a mesh constituted of triangular and quadrilateral elements with one curved edge. Fig. 3.5 shows the mesh and the corresponding L^2 relative error for various numbers of quadrature points (Q) and polynomial orders (k). It can be seen, in Fig. 3.5.b and Fig. 3.5.c, that

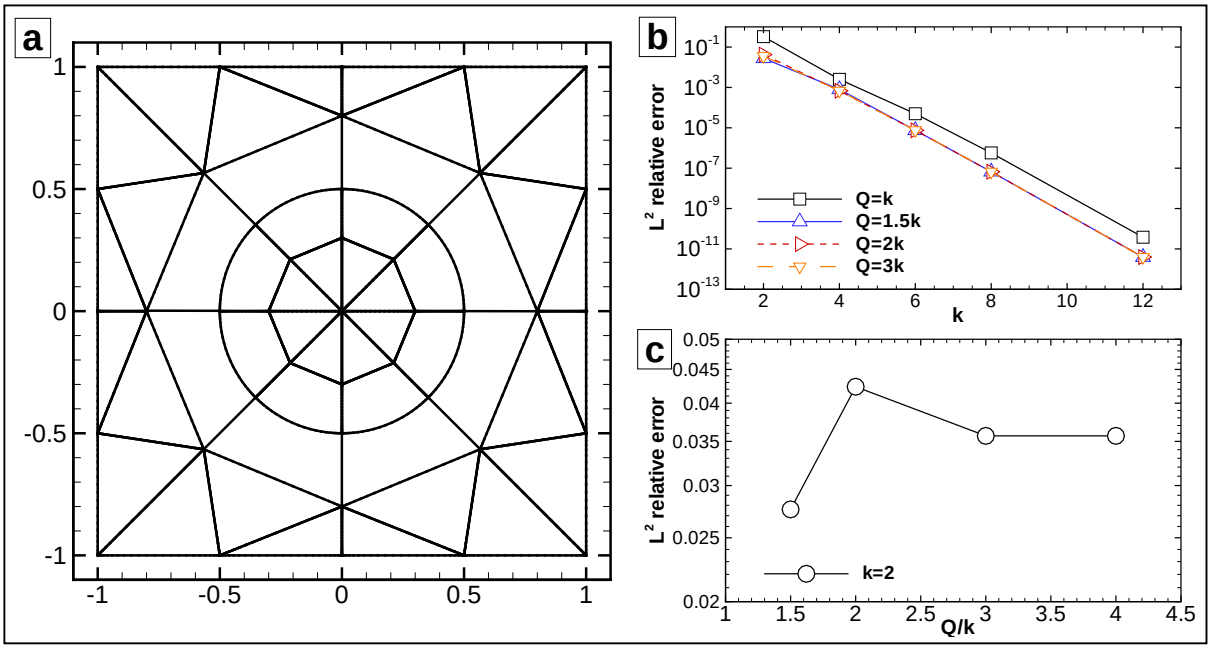

Figure 3.5: Necessary number of quadrature points to integrate circular curved edges for a linear Poisson problem: (a) tri-quad mesh with internal curved edges, (b) L^2 relative error for p-refinement using various number of quadrature points (Q) and (c) L^2 relative error for polynomial order $k = 2$ (i.e. the most stringent case) using increasing ratios of number of quadrature points to polynomial order (Q/k).

for this linear problem (i.e. without non-linear terms), $\mathcal{Q} = 3k$ is sufficient to accurately integrate the curved elements with the current implementation that includes an analytical description of the curved edges.

3.3 Implementation

To account for the sliding mesh connectivity, C++ *list classes* are used. This list includes edge connectivity and hanging node coordinates to facilitate the track of the relative mesh rotation and changes in element connectivity.

In the case of static meshes, matrices could be pre-computed and kept in memory since connectivity and metrics were unchanged throughout the simulation. In the case of moving meshes, metrics are changing after each iteration (in the rotating mesh zone) and thus pre-computation is not direct. In the current implementation, the matrices associated to the static zone are pre-compute, but the matrices required for the rotating mesh are recomputed after each iteration, once the mesh geometry has been rotated. To alleviate the computational cost of these computations, a quadrature-free formulation for triangular elements is used. However, this approach is not optimal and further improvements should help alleviate the cost associated with these calculations but are outside the scope of this thesis.

3.4 Verification and examples

This section is devoted to the verification and determination of the convergence properties of the described method. Firstly, the previously described Taylor Vortex problem is selected. The analytical solution being known, one can explore the numerical characteristics of the triangular-quadrilateral element mixing, internal curved edges and sliding mesh interfaces. Secondly, solutions are computed for a rotating square cylinder showing reasonable flow physics when using the sliding mesh capability. Thirdly, a static (non-

rotating) NACA0015 at zero angle of attack is simulated to explore the solver convergence properties with relation to the NACA 4-digit boundary mapping. The final section explores the solver's ability to compute a pitching NACA0015 foil.

3.4.1 Taylor Vortex problem

The Taylor vortex problem has been described in chapter 2 section 2.6.1, where the time-dependent exact solution was introduced. The problem is solved in $\Omega = [-1, 1]^2$ using various meshes. The selected Reynolds number is $\mathrm{Re} = 50$ and all simulations are computed up to a non-dimensional time $t = 0.1$.

3.4.1.1 Non-rotating cases

The first verification case involves only non-rotating meshes (i.e. Eulerian approach) to assess the solver accuracy on mixed triangular-quadrilateral element meshes with internal curved edges and hanging nodes. The time step is fixed to $\triangle t = 0.001$.

Fig. 3.6 depicts six different static meshes (i.e. (a) to (f)) and the corresponding L^2 relative error norm for p-refinement. The various computational domains consist of:

- (a): a semi-structured triangular element mesh with 72 elements,
- (b): a structured quadrilateral element mesh with 36 elements,
- (c): an unstructured triangular element mesh with 86 elements,
- (d): an unstructured triangular-quadrilateral element mesh with 100 elements,
- (e): an unstructured triangular-quadrilateral element mesh with internal curved edges and 100 elements,
- (f): an unstructured triangular-quadrilateral element mesh with internal curved edges and hanging nodes with 100 elements.

All cases show exponential convergence for pressure and velocity.

Fig. 3.6.d shows that the *functional incompatibility* arising from the use of triangular-quadrilateral elements does not damage the exponential convergence. Similarly in Fig.

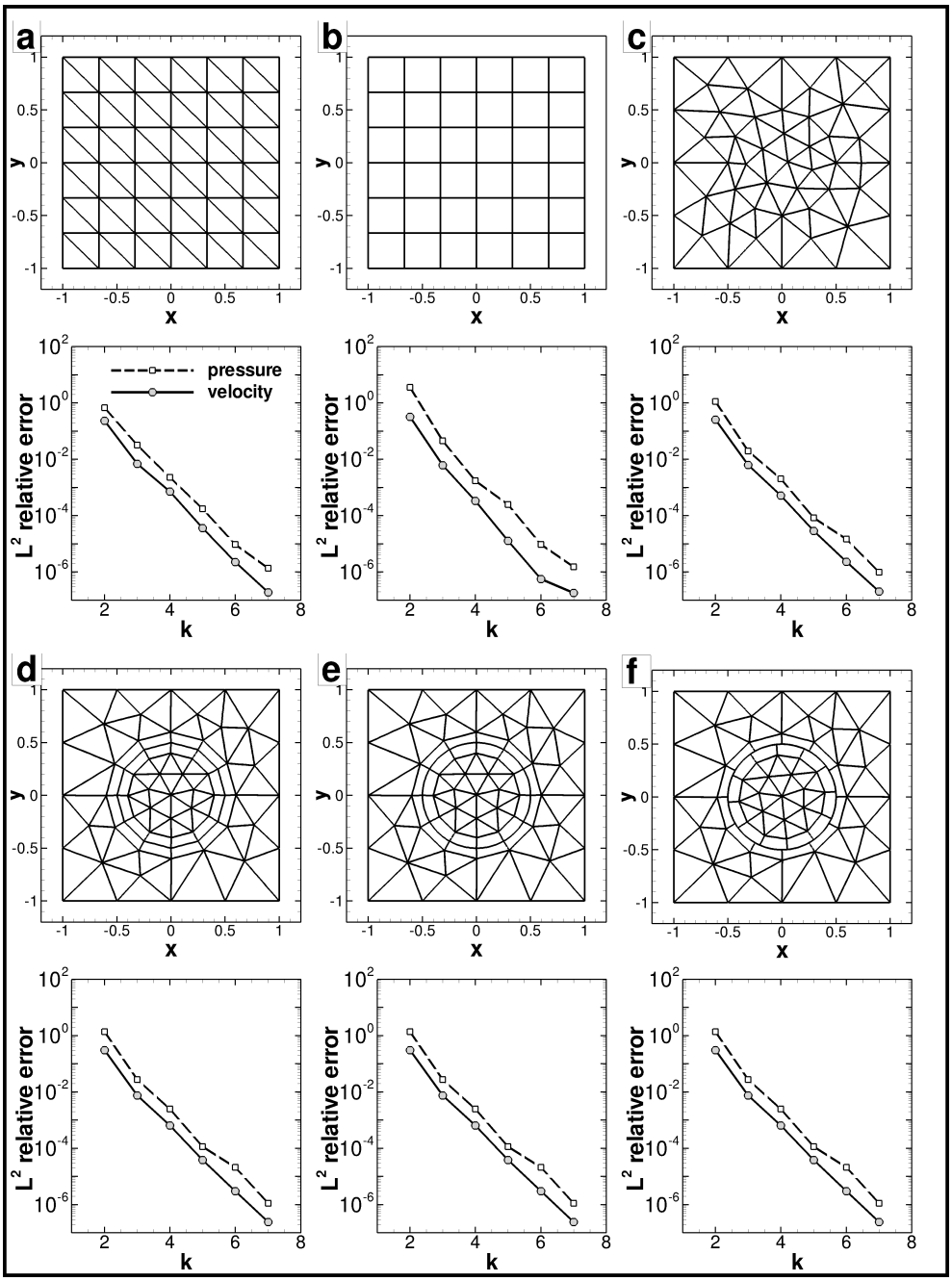

Figure 3.6: L^2 relative error norm of pressure and velocity for the Taylor-vortex problem for p-refinement (polynomial enrichment) in various meshes. k represents the polynomial order.

3.6.e, the methodology used for the imposition of an internal curved edge maintains the high order properties of the method. Finally, Fig. 3.6.f shows that the *geometric incompatibility* arising from the existence of hanging nodes can be handled by the solver

without damaging the exponential decay of the error.

Remark: *In appendix C section C.2.3, the L^2 relative error accuracy for other Interior Penalty methods (i.e. SIPG, IIPG and NIPG) is analysed, for a Poisson problem, on meshes composed of triangular-quadrilateral elements and internal curved edges, showing no degradation on the solution accuracy for p-refinement.*

3.4.1.2 Rotating cases

The mesh depicted in Fig. 3.6.e and 3.6.f is now used together with the ALE formulation and the sliding mesh capability.

First, the behaviour of L^2 relative error is explored for p-refinement for a range of rotational speeds $L\omega/U = 0, 1, 5, 10$ and 15. The time step is fixed to $\triangle t = 0.0005$ for all simulations. Fig. 3.7 shows the results for polynomial enrichment, where it can be seen that exponential convergence in the L^2 norm is obtained for pressure and velocity and for all cases if $k \leq 5$. However, it appears that as the rotational speed is increased, the error for high polynomial orders increases. This is found to be related to the temporal error as explained below.

Let us now fix the polynomial order to $k = 7$ and perform a time step refinement study. Results for the L^2 relative error norm for pressure and velocity and for various rotational speeds are plotted in Fig. 3.8. The slopes in the figure are calculated using the four points corresponding to higher time steps and show second order $\mathcal{O}(\triangle t^2)$ temporal accuracy for pressure and velocity for all rotational speeds.

Theoretical temporal convergence rates, for this splitting scheme, in the L^2 norm for a Stokes problem were obtained in [71]: $\mathcal{O}(\triangle t^{3/2})$ for pressure and $\mathcal{O}(\triangle t^2)$ for velocity, which are in very good agreement for a Stokes problem studied by the author in chapter 2 section 2.5.2. However, in this test case for the full Navier-Stokes equations, an improvement on the convergence rate for the pressure is observed; $\mathcal{O}(\triangle t^2)$ are obtained for pressure and velocity for all rotational speeds. More precisely, the L^2 norm seems to

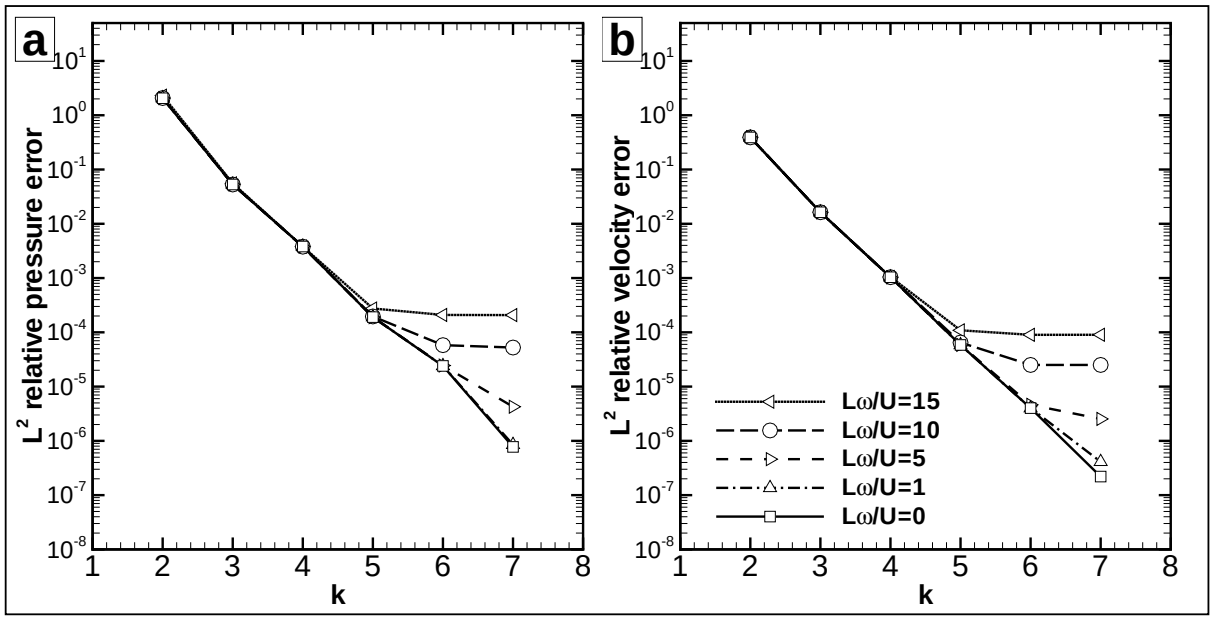

Figure 3.7: L^2 relative error norm for p-refinement, Taylor vortex problem with sliding meshes and various rotational speeds for (a) pressure and (b) velocity. k represents the polynomial order.

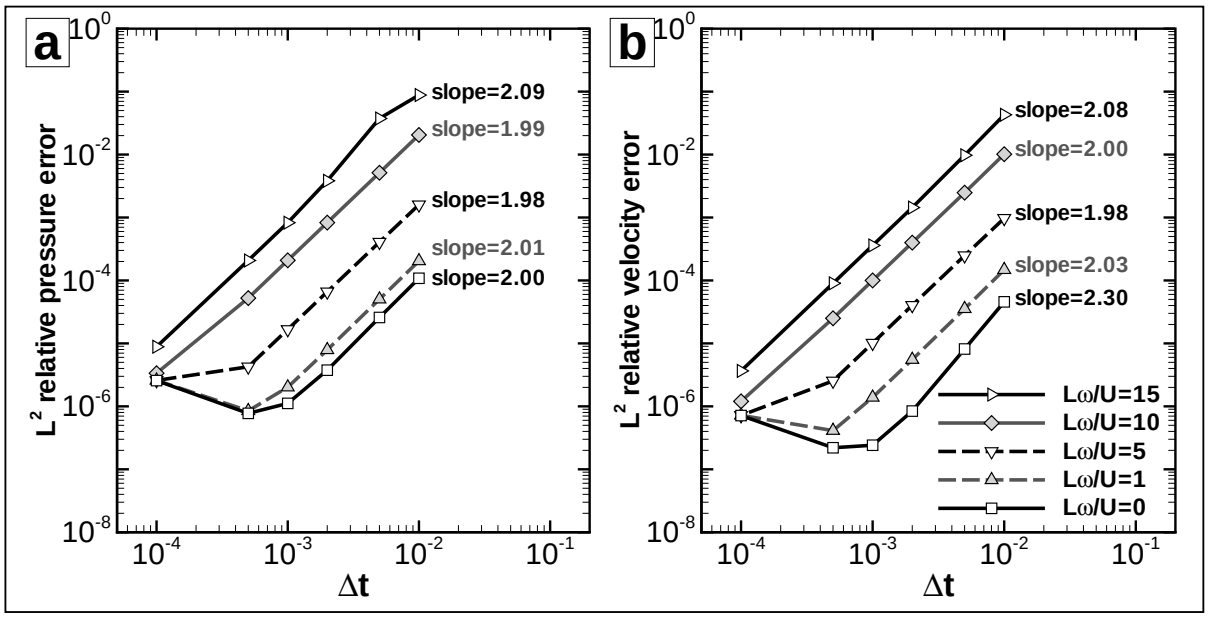

Figure 3.8: L^2 relative error norm for time step refinement, Taylor-vortex problem with sliding meshes and various rotational speeds for (a) pressure and (b) velocity.

behave as: $L^2(\triangle t) = \mathcal{O}((1+\omega^2)\triangle t^2)$, showing that to reach a certain level of accuracy it is necessary to reduce the time step, if a high rotational speed is to be computed.

Returning to the exponential convergence for p-refinement shown in Fig. 3.7, it appears clear from consulting Fig. 3.8 that the limit in exponential convergence for polynomial enrichment for $\omega > 0$ corresponds to the temporal error when this dominates over the spatial error. In addition, let us note that for very small time steps the error increases as the time step decreases, as previously observed and characterised in chapter 2.

To close this section and for completeness, the L^2 error norm for pressure and velocity is plotted against the mesh angle: $\triangle tL\omega/U$ in Fig. 3.9, which demonstrates the second order accuracy of the scheme, consistently with the observations of Fig. 3.8 and confirms the L^2 error estimate: $L^2(\triangle t) = \mathcal{O}((1+\omega^2)\triangle t^2)$.

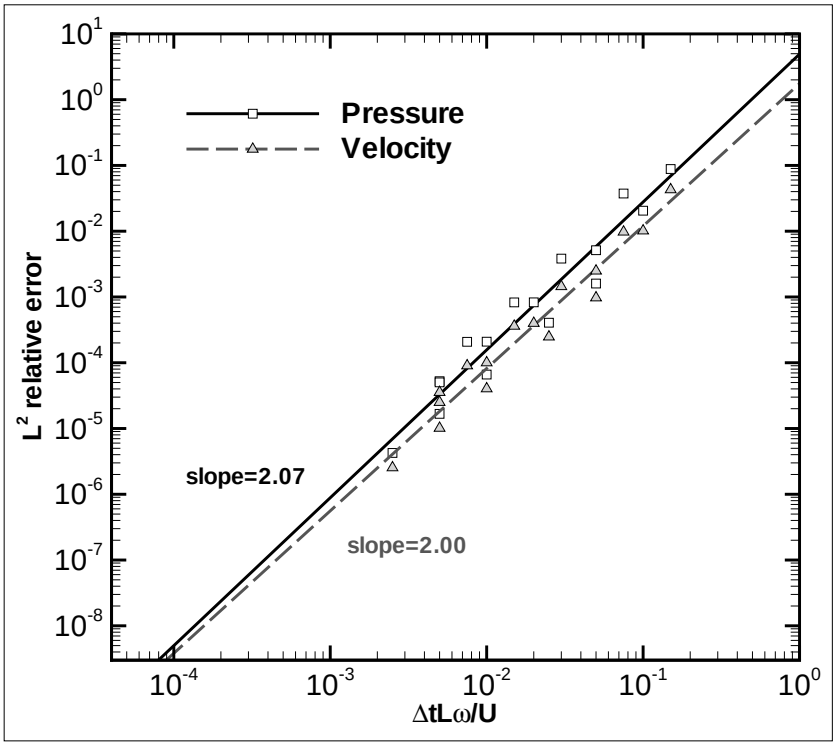

Figure 3.9: L^2 relative error norm of pressure and velocity for non-dimensional mesh angle ($\triangle tL\omega/U$) refinement, Taylor-vortex problem with sliding meshes and various rotational speeds.

3.4.2 Wannier flow using sliding meshes and curved boundary conditions

This section considers the Wannier flow problem [101], [175]. This problem with analytical solution corresponds to creeping flow (Stokes equations without non-linear terms) over a rotating circular cylinder near to a moving wall, as depicted in Figs. 3.10.a, 3.10.b and 3.10.d. In this test problem, a triangular-quadrilateral mesh, with external circular boundary conditions (to fit the geometry of the circular cylinder) and a sliding mesh interface, are used. The exact solution is given by:

$$\begin{aligned} u(x,y) &= U - 2(a_1 + a_0 Y_1)\left[\frac{s+Y_1}{K_1} + \frac{s-Y_1}{K_2}\right] - a_0 \ln(\frac{K_1}{K_2}) \\ &\quad -\frac{a_2}{K_1}\left[s + Y_2 - \frac{(s+Y_1)^2 Y_2}{K_1}\right] - \frac{a_3}{K_2}\left[s - Y_2 + \frac{(s-Y_1)^2 Y_2}{K_2}\right], \\ v(x,y) &= \frac{2x}{K_1 K_2}(a_1 + a_0 Y_1)(K_2 - K_1) - \frac{x a_2 (s+Y_1) Y_2}{K_1^2} - \frac{x a_3 (s-Y_1) Y_2}{K_2^2}, \end{aligned} \tag{3.4.1}$$

with the following functions:

$$\begin{aligned} Y_1(y) &= y + d\,, \quad Y_2(y) = 2Y_1(y), \\ K_1(x,y) &= x^2 + (s + Y_1(y))^2\,, \quad K_1(x,y) = x^2 + (s - Y_1(y))^2, \end{aligned} \tag{3.4.2}$$

and the constants:

$$\begin{aligned} a_0 &= \frac{U}{\ln \Gamma}\,, \quad a_1 = -d\left(a_0 + \frac{r^2 \omega_0}{2s}\right), \\ a_2 &= 2(d+s)\left(a_0 + \frac{r^2\omega_0}{2s}\right)\,, \quad a_3 = 2(d-s)\left(a_0 + \frac{r^2\omega_0}{2s}\right), \end{aligned} \tag{3.4.3}$$

where r represents the cylinder radius, ω_0 the rotational speed of the cylinder, d is the distance from the cylinder centre (located at $(x,y) = (0,0)$) to the moving wall and U represents the velocity of the wall. In addition, the following definitions are used $s^2 = d^2 - r^2$ and $\Gamma = (d+s)/(d-s)$. This test case utilises the same parameters as

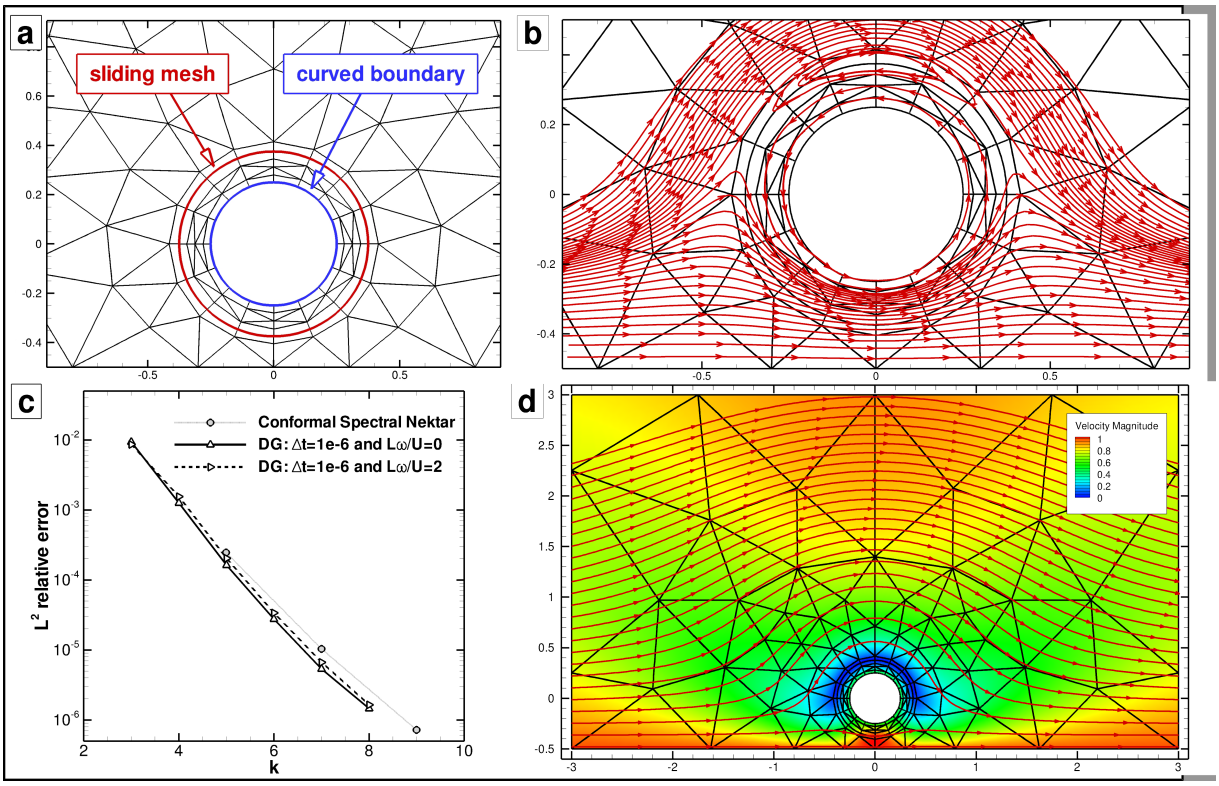

Figure 3.10: Wannier flow results: (a) DG mesh with a sliding mesh interface and a circular curved boundary, (b) DG Streamlines, (c) L^2 relative error norm, for the velocity, for the DG solver using two non-dimensional rotational speeds ($L\omega/U = 0$ and 2) and results for the conformal h/p Spectral code [175] and (d) DG velocity contour and streamlines.

used by Warburton in [175]; namely, $r = 0.25$, $d = 0.5$, $U = 1$ and $\omega_0 = 2$. Fig. 3.10.c compares the exponential decay of the L^2 error for p-refinement for the conformal h/p Spectral code Nektar and the DG solver. Nektar results have been extracted from [175] and were obtained with a different mesh. However, it can be seen that close agreement is obtained between the solvers. Using the DG solver, it is possible to perform an additional simulation, where the inner mesh is rotated using the sliding mesh capability. Fig. 3.10.c shows no degradation in L^2 error when a rotational speed of $L\omega/U = 2$ is selected.

This section has shown that external circular boundary conditions using the developed technique provide exponential convergence and comparable results to a conformal h/p Spectral method. In addition, this test case shows that the sliding mesh technique does not damage the accuracy of the solver if rotation is applied to a mesh region.

3.4.3 Rotating square cylinder

In this section, the sliding mesh capability is used to simulate the development of vortex shedding aft of a rotating cylinder for various rotational speeds and Reynolds numbers. A very coarse mesh (i.e. 142 tri-quad elements) is used for all cases with high polynomial orders $k = 8$. The results reported in this section are preliminary since spatial and temporal convergence have not been addressed. However, analysis of the results show, qualitatively sensible physical phenomena and provide guidelines for further studies. In addition, this section shows an example of the usability of the sliding mesh technique to compute flows around rotating geometries.

The square cylinder section under static conditions (i.e. no rotation) has been studied in chapter 2 section 2.6.2 (and published in [52]) showing very good agreement with experimental data and the conformal h/p Spectral solver Nektar [101]. Although this static case has been explored before (see for example [186]), to the authors knowledge this is the first time that the spectral character of the vortex formation due to rotation is analysed. For static cylinders (i.e. no rotation) there is a limiting Reynolds numbers $\mathrm{Re}_{crit} \approx 54$ [103] below which the cylinder wake remains steady and no vortex shedding is instigated. However, for $\mathrm{Re} > \mathrm{Re}_{crit}$, an unsteady Von Kármán vortex street develops.

Three Reynolds numbers are selected: $\mathrm{Re} = 10$, 50 and 100, to explore the effect of rotation in these different regimes. In addition, the rotational speed is varied: $L\omega/U = 0.1$, 0.5 and 1.0, where L is the square section edge length.

The aim of this study is to analyse the interaction (if any) of the vortices that develop purely due to the rotational motion and those that would naturally develop due to unsteady shedding in the absence of rotation (i.e. Von Kármán vortex street) for $\mathrm{Re} > \mathrm{Re}_{crit}$. Fig. 3.11 shows two snapshots of the solution streamlines for $\mathrm{Re} = 10$ and 100, showing very different flow patterns aft the rotating cylinder for $L\omega/U = 0.1$. The spectral content of the time varying cylinder's lift coefficient, resulting from the simulations, is depicted in Fig. 3.12. For $\mathrm{Re} = 10$ (Fig. 3.12.a1, 3.12.b1 and 3.12.c1), the figures show a unique

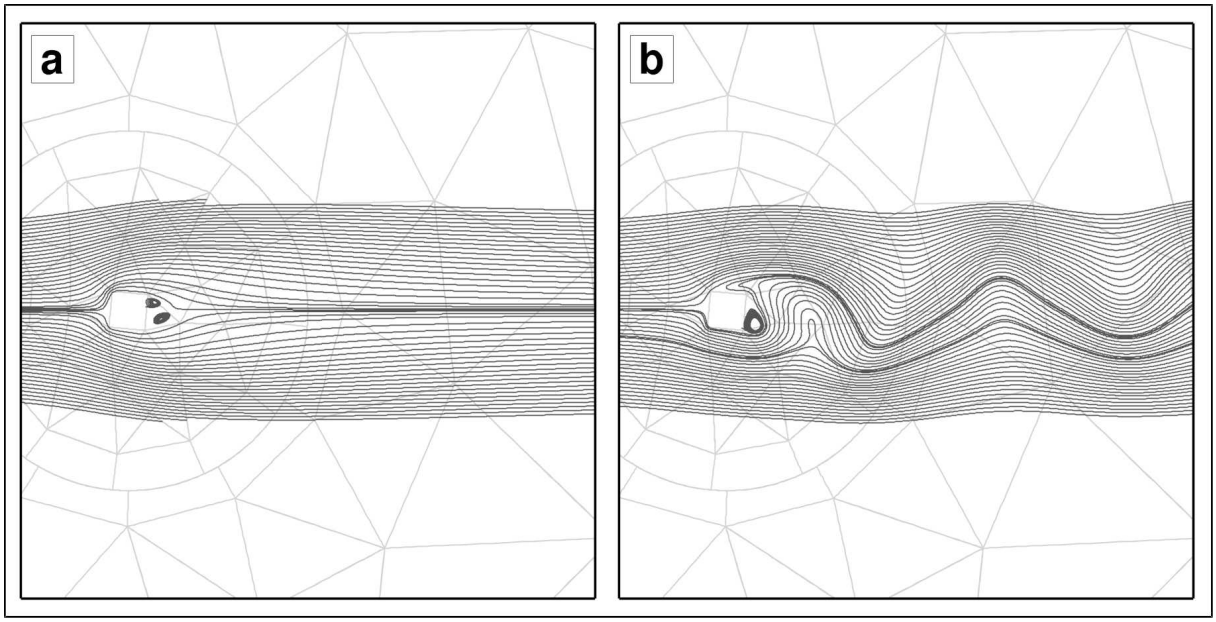

Figure 3.11: Streamlines around a rotating square cylinder with sliding meshes (polynomial order $k = 8$) for a rotational speed $L\omega/U = 0.1$ and two Reynolds numbers (a) Re = 10 and (b) Re = 100.

main frequency driven by the geometric rotation $f_{rot}L/U = 4\frac{L\omega/U}{2\pi}$, which is associated with the flow separation at the front corner of the cylinder (4 times per revolution). Let us note that for Re = 10, the Föppl vortices in the cylinder wake are non-symmetric due to the rotational motion. However, vortex shedding remains suppressed for this low Reynolds number.

As the Reynolds number is increased to Re = 50 and 100, a second frequency peak is observed and appears to be associated with the Von Kármán vortex shedding. The vortex shedding frequency for the non-rotating cylinder at Re = 100, computed using the non-rotating version the present DG method (see chapter 2) is $f_{VK}L/U$=0.144, which is similar to the frequency developing in the presence of rotation for Re = 100. In addition, it can be seen that the vortex shedding frequency is independent of the rotational speed as shown in Fig. 3.12.a3, 3.12.b3 and 3.12.c3. It may be concluded that two weakly interacting mechanisms are driving the frequency spectra: on the one hand, a rotationally driven flow separation at frequency f_{rot} proportional to the rotational speed and,

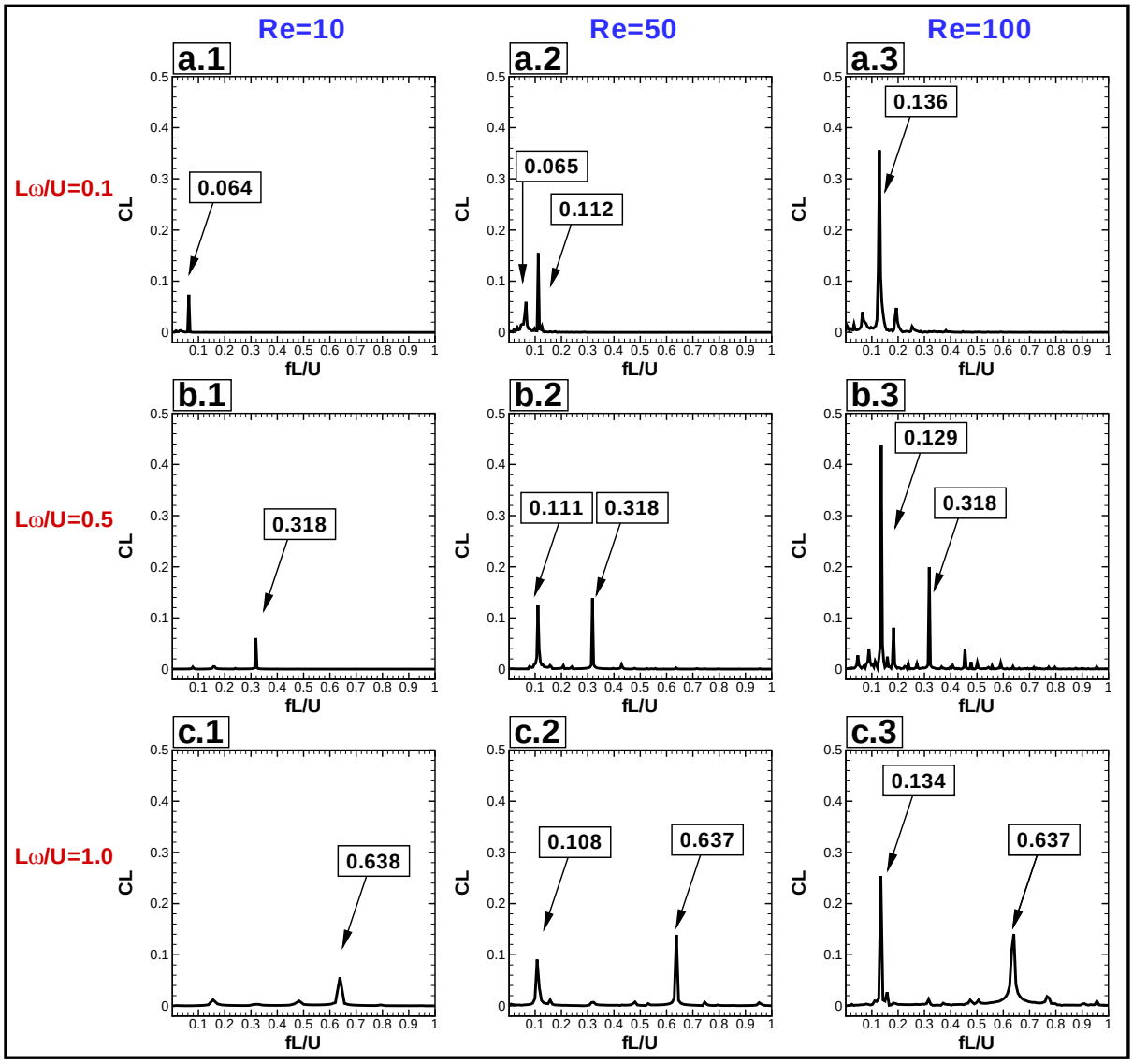

Figure 3.12: Frequency spectra for the lift coefficient ($CL = {F_p}^y + {F_v}^y$) of a rotating square cylinder with sliding meshes (polynomial order $k = 8$) for various rotational speeds and Reynolds numbers; boxes identify peak frequencies.

on the other hand, the unsteady natural Von Kármán vortex shedding mechanism with frequency f_{VK}. The interaction between mechanism appears weaker the closer f_{rot} and f_{VK}, and may be related to a lock-in phenomenon for $f_{rot} \approx f_{VK}$. Finally, let us note that for the case $\mathrm{Re} = 50 < \mathrm{Re}_{crit}$ (Fig. 3.12.a2, 3.12.b2 and 3.12.c2), rotation triggers vortex shedding, which would not have been present for the non-rotating case at the same Reynolds number.

3.4.4 NACA0015 foil

The symmetric NACA0015 foil section is selected for verification. Firstly, static cases are considered to assess the correct implementation and accuracy of the NACA 4-digit boundary condition (see section 3.2.2.3). Secondly, a NACA0015 under rotation is considered, where the sliding mesh capability together with the NACA 4-digit boundary condition, are used.

3.4.4.1 Non-rotating cases

This section verifies the implementation of the NACA 4-digit boundary conditions (Eq. 3.2.16) and investigates the solver convergence properties using a NACA0015 at zero Angle Of Attack (AOA) and two Reynolds numbers (i.e. Re = 100 and 500). For all cases, it is ensured that the inflow boundary is at 6 chords upstream of the foil and the outflow boundary is at 13 chords aft the foil. On the upper and lower surfaces, Dirichlet boundaries for velocity are used, and are located at 8 chords from the foil.

Firstly, results are compared for two simulations using the DG solver for a mesh constituted only of straight edge triangles and a triangular-quadrilateral mesh that uses analytical NACA boundaries to describe the foil geometry. The triangular mesh has a total of 988 elements of which 64 are fitted to the foil surface to define the geometry. The triangular-quadrilateral mesh with curved NACA boundaries requires a total of 825 elements and 48 curved quadrilateral elements to define the foil geometry. This last mesh is shown in Fig. 3.14.

Both cases are simulated with a polynomial order $k = 5$ and results for the pressure coefficient ($Cp = 2p$) and the skin friction coefficient ($Cf = 2\tau_w/\text{Re}$, with τ_w the shear stress at the wall) at Re = 500 and zero angle of attack ($AOA = 0°$), are depicted in Fig. 3.13. Fig. 3.13.b shows a smooth distribution along the chord for both variables when the curved parametrisation is used as opposed to the jagged curve obtained with straight edges Fig. 3.13.a, verifying the necessity of the NACA curved boundaries and its

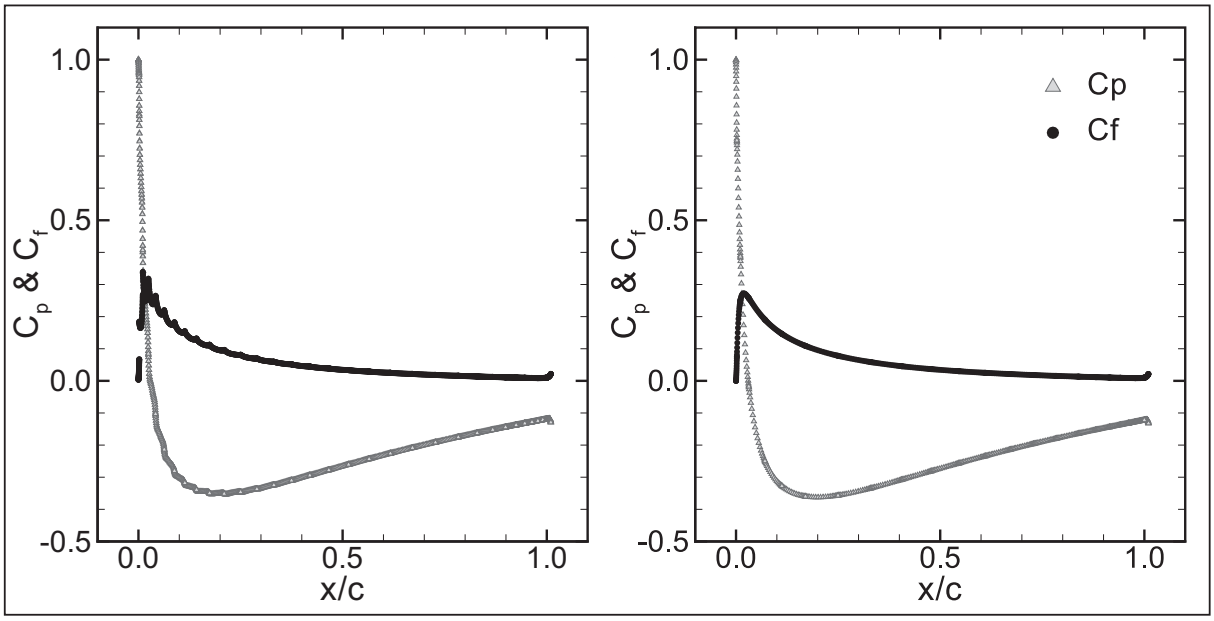

Figure 3.13: Pressure (Cp) and skin friction (Cf) coefficients for the NACA0015 foil at Re = 500, $AOA = 0°$ and polynomial order $k = 5$ for (a) Triangular elements with straight edges and (b) Quadrilateral elements with NACA 4-digit curved boundaries.

correct implementation. It is interesting to note that the implementation of the NACA boundaries enables the use of fewer elements to define the foil geometry whilst increasing the resolution of aerodynamic quantities.

To investigate the solver convergence properties, a non-symmetric (with respect to the horizontal axis) triangular-quadrilateral mesh with curved NACA boundaries (i.e. the 825 elements mesh described previously), is selected. Simulations are performed for two Reynolds numbers: Re =100 and 500, and one angle of attack $AOA = 0°$. Fig. 3.14 depicts the mesh and pressure contours for polynomial orders k =3 and 6 at Re =500. The figure shows that qualitatively good agreement is obtained between the simulations using different polynomial orders and that a smoother solution is obtained when using a higher polynomial.

Since the lift force for a symmetric foil at $AOA = 0°$ is identically zero, a p-refinement study can be performed to assess the solver convergence properties in a realistic case. Fig. 3.15 shows that exponential convergence is obtained for the absolute values of the

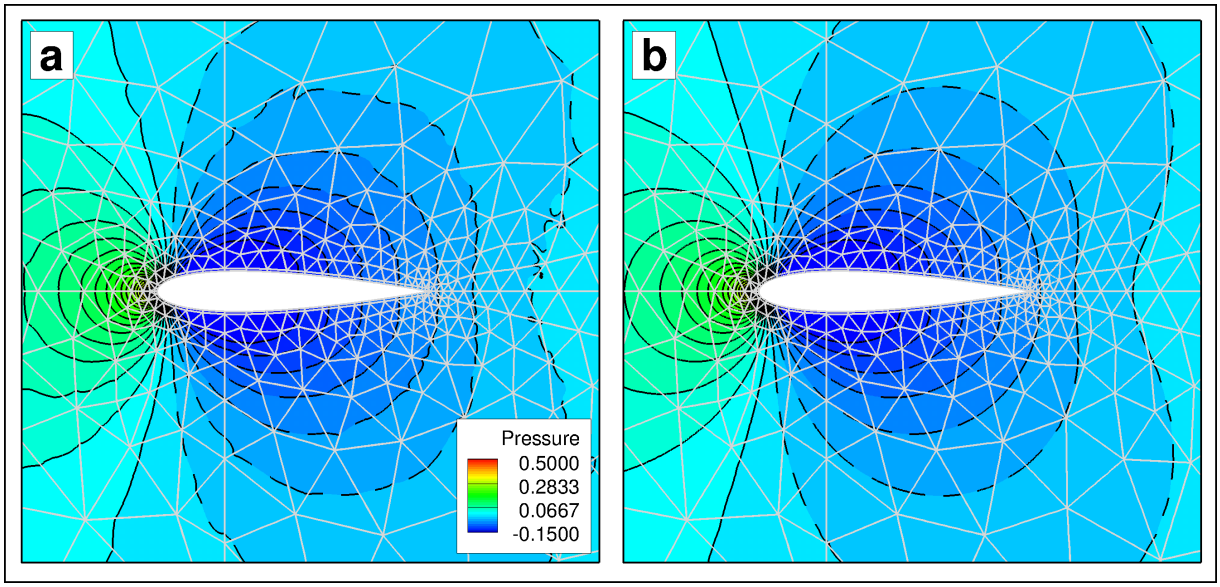

Figure 3.14: NACA0015 foil at Re = 500 and $AOA = 0°$ with 35 pressure contours [-0.15:0.5] (dashed lines show negative values) for polynomial orders (a) $k = 3$ and (b) $k = 6$.

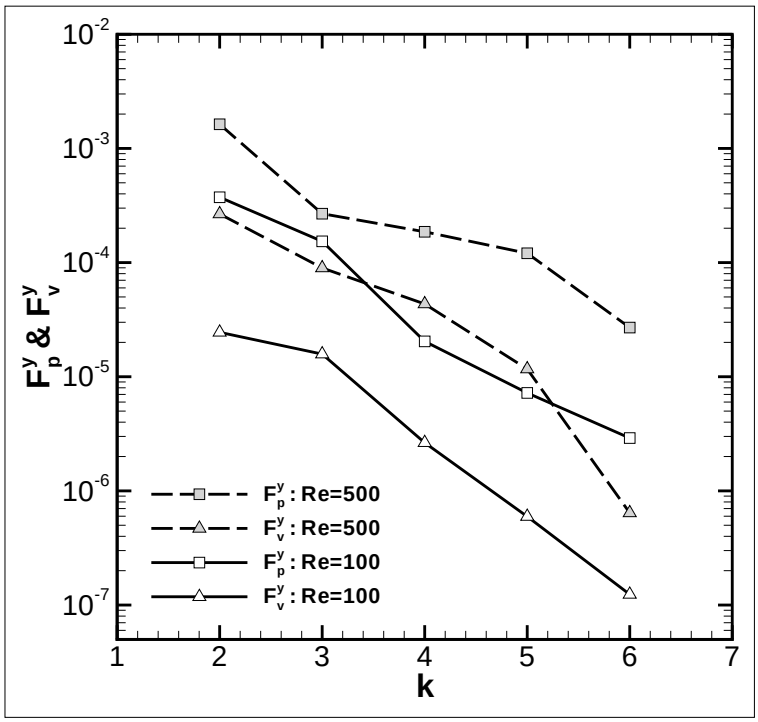

Figure 3.15: Exponential convergence for the absolute value of the pressure and viscous components of the lift force (F_p^y and F_v^y) for the NACA0015 foil at Re =100 and 500, and $AOA = 0°$. k represents the polynomial order.

pressure (F_p^y) and viscous (F_v^y) component of the lift force for the two tested Reynolds numbers.

3.4.4.2 Rotating cases

This section considers cases where the NACA 4-digit boundary is used conjointly with the sliding mesh technique (mesh depicted in Fig. 3.1 and solution snapshots in Fig. 3.16). A NACA0015 foil under impulsive start conditions is pitched around its quarter

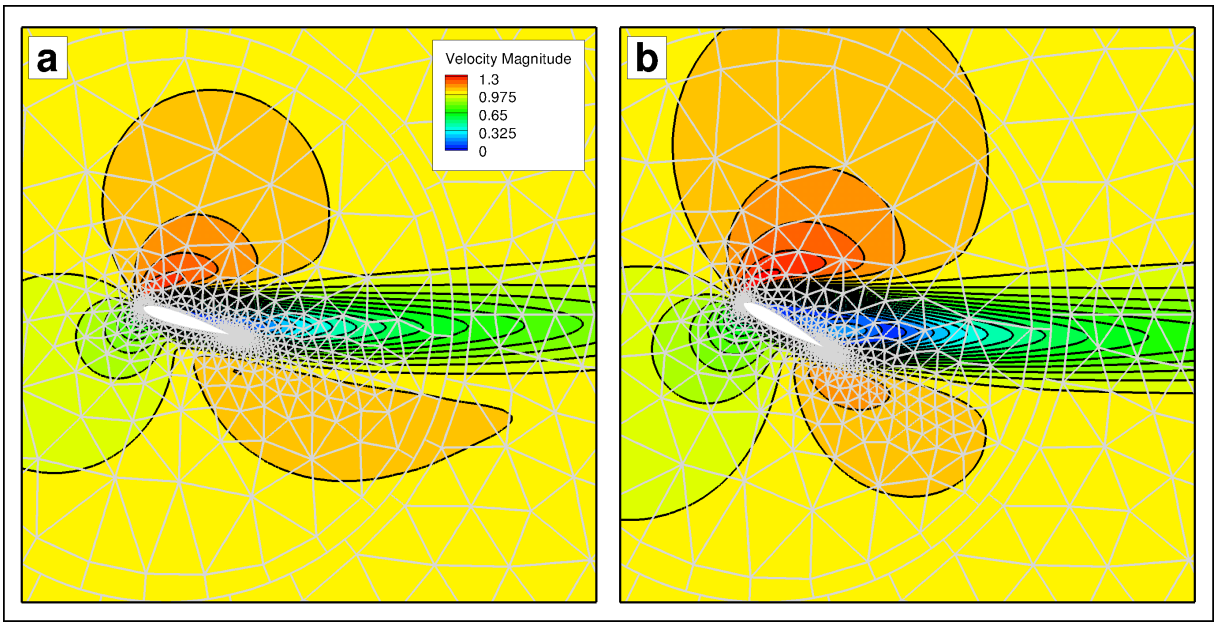

Figure 3.16: Snapshots of the rotating NACA0015 foil with 21 velocity magnitude contours [0:1.3] for polynomial order $k = 5$ and rotational speed $L\omega/U = 0.05$ for (a) geometric $AOA = 17.2°$ and (b) geometric $AOA = 28.6°$.

chord point for a range of geometric angles of attack (i.e. the angle between the free stream velocity and the foil chord line) $0° \leq AOA \leq 35°$ and four rotational speeds $L\omega/U = 0.005$, 0.01, 0.05 and 0.1, where $L = c$ and c represents the foil chord. The Reynolds number is fixed to Re = 100 and a triangular-quadrilateral mesh constituted by 1462 elements is used together with a polynomial order $k = 5$. Fig. 3.16 shows snapshots of the unsteady dynamically rotating simulation for $L\omega/U = 0.05$ at two geometric angles of attack ($AOA = 17.2°$ and 28.6°). It can be seen that the contours are smooth across

the interfaces between triangular and quadrilateral elements and also across the sliding mesh interface.

Finally, Fig. 3.17 depicts the variation of the vertical F^y and horizontal F^x forces (including pressure and viscous components Eq. 2.4.3) against the geometric AOA for the four rotational speeds considered. In addition, the time averaged forces for static cases at $AOA = 0°$, $10°$, $20°$ and $30°$ are included. Comparison between static and dynamic simulation results at low rotational speeds shows very good agreement, demonstrating the low error level introduced by the rotating mesh interface. In addition, it can be seen that dynamic effects cause an increase in the vertical and horizontal components of the forces with increasing rotational speed.

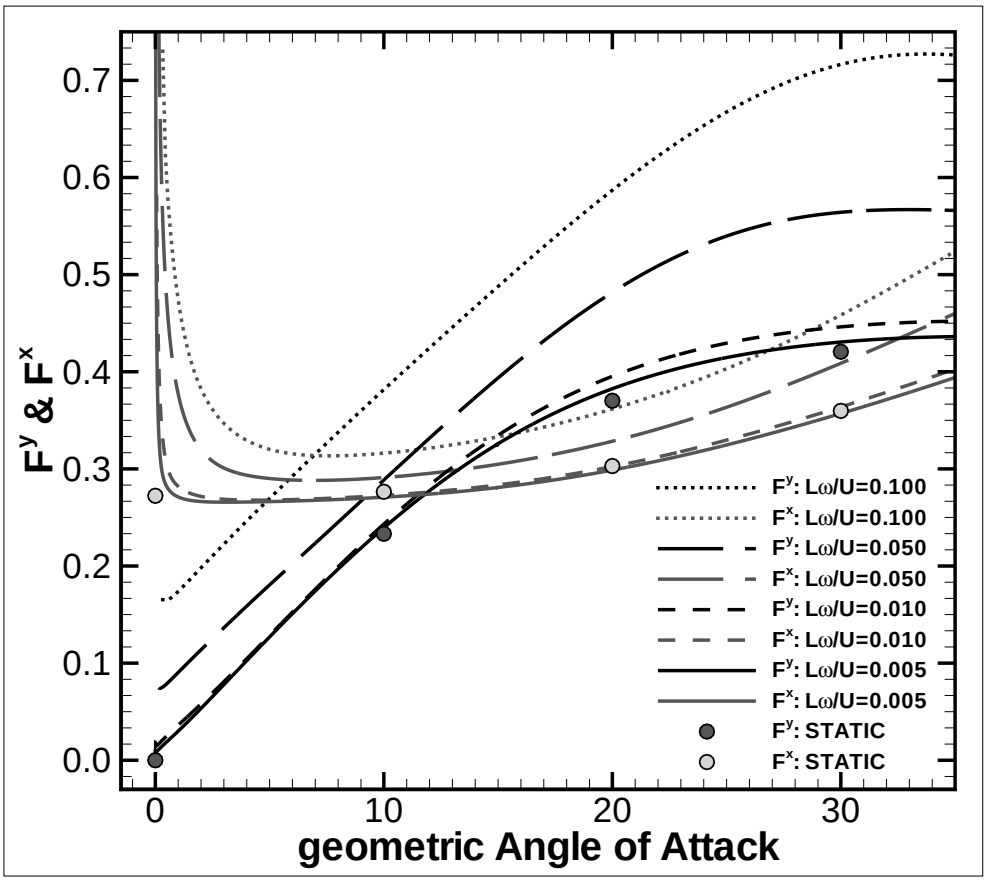

Figure 3.17: Vertical and horizontal forces (F^y and F^x) against the geometric $AOA = 0°$ to $35°$, for the NACA0015 foil at various rotational speeds $L\omega/U = 0.005$, 0.01, 0.05 and 0.1.

Chapter 4

Three dimensional flows and parallelisation

Truly two dimensional flows are rare in nature. Very low Reynolds number flows (e.g. $Re \lesssim 150$ for a circular cylinder, where Re is based on the free stream velocity and the cylinder diameter) or flows where a geometric confinement limits three dimensionality (e.g. shallow water flows) are examples of two dimensional flows. However, when no geometric confinement is considered and the Reynolds number increases such as to trigger three dimensional modes (see section 4.5), flows show generally a three dimensional character. If the Reynolds number is increased further, then a turbulence regime (i.e. regime including an inertial subrange of energy cascade) develops, which is by definition also three dimensional (see chapter 5 for details on turbulent flows). In summary, three dimensional structures are likely to exist in bluff body flows, aerodynamic applications and rotating machinery such as cross-flow turbines.

In this chapter, the two dimensional Discontinuous Galerkin capabilities presented in the previous chapters are extended to account for three dimensional flow features. Let us note that the defined tensorial representation of the mappings from physical to computational space (sections 2.2.2.4 and 3.2.2.3), in combination with tensorial basis functions (sections 2.2.2.5 and 3.2.2.5), allow for 3D extensions of the previously defined method

without further modifications (e.g. 3D DG spatial discretisations). However, a more efficient 3D extension is provided when combining the previous 2D DG formulation with Fourier series to account for three dimensional flow features. This technique enables the use of the Fast Fourier Transform (FFT) algorithm and provides an efficient 3D solver for problems involving spanwise geometric homogeneity. The resulting methodology is named DG-Fourier and is described in this chapter.

To alleviate the increased computational cost, associated with computing three dimensional solutions of the 3D incompressible Navier-Stokes equations, a parallelisation strategy based on the Message Passing Interface (MPI) paradigm is implemented. The chapter provides verification and validation cases for three dimensional flows where the accurate prediction of the flow three dimensionality is essential.

4.1 Chapter outline

Section 4.2 introduces the methodology necessary to extend the two dimensional DG code with sliding meshes and curved external boundary conditions, presented in previous chapters, to a three dimensional DG-Fourier solver. The solver parallelisation using the MPI paradigm is detailed in section 4.3. Section 4.4 provides verification of the method through problems where the exact solution is known: an elliptic problem and a 3D NS problem with exact solution. Finally, section 4.5 presents three dimensional flows for square and circular cylinders. The chapter concludes with a computation of a circular cylinder in the wake of a rotating NACA0012 blade at moderate Reynolds number to illustrate the possibility of coexisting static and rotating 3D geometries in one single mesh domain.

4.2 Methodology

4.2.1 Preliminaries: discrete Fourier transform on periodic domains

In this section, the one dimensional discrete Fourier transformations (FFT) and its inverse $invFFT$ are considered on a periodic domain $[0, L_z[$. These transformations map the physical space z to the computational Fourier space ξ as shown in Fig. 4.1 (further details can be found in [167] or [24]). The uniform discretisation implies $h = L_z/M$,

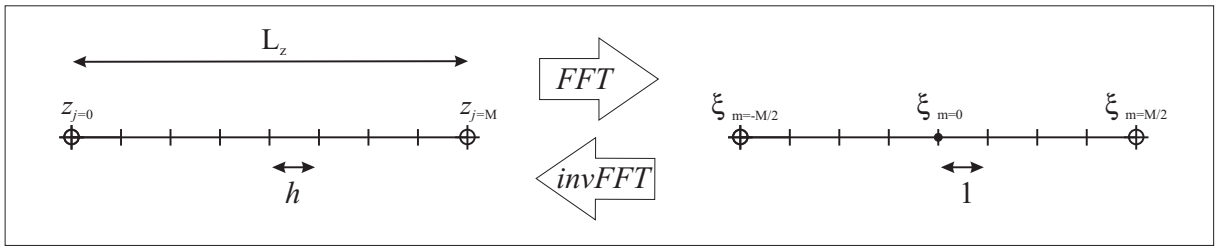

Figure 4.1: Discrete Fourier Transform for periodic domains.

where h is the grid spacing in physical space and L_z defines the periodic length of the domain in the z-direction. M represents the number of planes in physical space and the number of modes in the Fourier space.

The transformations depicted in Fig. 4.1 are defined as:

$$\begin{aligned} \widehat{u}(\xi_m) &= FFT_m\{u(z)\} = \frac{1}{M}\sum_{j=0}^{M-1} u(z_j)e^{-i\beta m z_j}, \quad -M/2 \le m \le M/2-1, \\ u(z_j) &= invFFT_j\{\widehat{u}(\xi)\} = \sum_{m=-M/2}^{M/2-1} \widehat{u}(\xi_m)e^{i\beta m z_j}, \;\; j = 0, 1..., M-1, \end{aligned} \tag{4.2.1}$$

where $i = \sqrt{-1}$ and $\beta = 2\pi/L_z$ is the characteristic wavenumber and βm represents the wavenumber corresponding to the mth mode.

4.2.2 DG-Fourier solver

The use of Fourier series, to efficiently extend two dimensional codes and account for three dimensional flows, has been reported for h/p conformal Spectral methods (e.g. [99], [101] or [19]). This technique leads to a series of 2D DG linear solves that are coupled through the non-linear terms (see details below).

Let us rewrite for clarity the Arbitrary Lagrangian-Eulerian (ALE) form of the incompressible Navier-Stokes equations introduced in chapter 3:

$$\begin{aligned} NS^1(\mathbf{u},p) &= \tfrac{\partial \mathbf{u}}{\partial t} + ((\mathbf{u}-\mathbf{w}).\nabla)\,\mathbf{u} + \nabla p - \tfrac{1}{\mathrm{Re}}\nabla^2\mathbf{u} = 0 \quad \text{in } \Omega(t), \\ NS^2(\mathbf{u}) &= \nabla.\mathbf{u} = 0 \qquad \text{in } \Omega(t), \\ \mathbf{u} &= L_D \qquad \text{on } \partial\Omega_D, \\ \tfrac{1}{\mathrm{Re}}\tfrac{\partial \mathbf{u}}{\partial n} + p\mathbf{n} &= 0 \qquad \text{on } \partial\Omega_N. \end{aligned}$$

The main advantage of this approach is that the Fast Fourier Transform, which has a cost $\mathcal{O}(Mlog_2M)$, can be used rendering the scheme much more efficient when compared to using other discretisation strategies in the third direction (e.g. a DG spatial discretisation). The main disadvantage is that only periodic spanwise boundary conditions and homogeneous geometries (or geometries that can be described through a Fourier transformation) can be simulated.

Using the Discrete Fourier Transform (Eq. 4.2.1), the three dimensional velocity and pressure become:

$$\mathbf{u}(x,y,z) = \sum_{m=-M/2}^{M/2-1} \mathbf{u}_m(x,y)e^{i\beta m z}, \tag{4.2.2}$$

$$p(x,y,z) = \sum_{m=-M/2}^{M/2-1} p_m(x,y)e^{i\beta m z}, \tag{4.2.3}$$

where the notation introduced in 4.2.1 has been simplified by setting: $\widehat{u}(\xi_m) = u_m$, and $u(z_j) = u(z)$. Let us note that $\mathbf{u}_m(x,y)$ and $p_m(x,y)$ have already been defined in previous

chapters (Eq. 2.2.23) and represent the x-y plane DG solution for the mth Fourier mode.

Applying the inverse of the Fourier transform (i.e. $invFFT(\widehat{\bullet})$, see Eq. 4.2.1) to the momentum and continuity equations in ALE form (Eq. 4.2.2 and 4.2.2) and using its linearity property, one obtains:

$$\begin{aligned}
NS^1(\mathbf{u},p) &= invFFT\{\widehat{NS^1(\mathbf{u},p)}\} \\
&= \sum_{m=-M/2}^{M/2-1} \{NS_m(\mathbf{u},p)\} e^{i\beta m z} \\
&= \sum_{m=M/2}^{M/2-1} \left\{ \frac{\partial \mathbf{u}_m}{\partial t} + \widehat{\nabla}_m p_m - \frac{1}{\mathrm{Re}} \widehat{\nabla}^2_m \mathbf{u}_m \right\} e^{i\beta m z} \qquad (4.2.4) \\
&\quad + \sum_{m=M/2}^{M/2-1} \left\{ FFT_m \left\{ ((\mathbf{u}-\mathbf{w}).\nabla)\,\mathbf{u} \right\} \right\} e^{i\beta m z} = 0 \quad \text{in } \Omega(t), \\
NS^2(\mathbf{u}) &= invFFT\{\widehat{NS^2(\mathbf{u})}\} \\
&= \sum_{m=-M/2}^{M/2-1} \{\widehat{\nabla}_m.\mathbf{u}_m\} e^{i\beta m z} = 0 \qquad \text{in } \Omega(t). \qquad (4.2.5)
\end{aligned}$$

where $FFT_m\{\bullet\}$ represent the mth mode of the Fourier transform of the argument and the Fourier operators $\widehat{\bullet}$ are defined as:

$$\begin{aligned}
\widehat{\nabla}_m &= \left(\tfrac{\partial}{\partial x}, \tfrac{\partial}{\partial y}, i\beta m \right)^T = (\nabla_{2D}, i\beta m)^T, \\
\widehat{\nabla}^2_m &= \left(\tfrac{\partial^2}{\partial x^2} + \tfrac{\partial^2}{\partial y^2} - \beta^2 m^2 \right) = \nabla^2{}_{2D} - \beta^2 m^2,
\end{aligned}$$

where ∇_{2D} and $\nabla^2{}_{2D}$ are the two dimensional gradient (nabla) and Laplacian operators, respectively.

The system described by Eq. 4.2.4 and Eq. 4.2.5 is equivalent to solving an indepen-

dent equation for each mode m:

$$\begin{aligned}
\tfrac{\partial \mathbf{u}_m}{\partial t} + FFT_m\left\{((\mathbf{u}-\mathbf{w}).\nabla)\,\mathbf{u}\right\} &= -\widehat{\nabla}_m p_m + \tfrac{1}{\mathrm{Re}}\widehat{\nabla}^2_m \mathbf{u}_m && \text{in } \Omega_m(t),\\
\widehat{\nabla}_m.\mathbf{u}_m &= 0 && \text{in } \Omega_m(t),\\
\mathbf{u}_{m=0} &= L_D\ ,\ \mathbf{u}_{m\neq 0}=0 && \text{on } \partial\Omega_{Dm},\\
\left(\tfrac{1}{\mathrm{Re}}\tfrac{\partial \mathbf{u}}{\partial n} + p\mathbf{n}\right)_{m=0} &= 0 \quad ,\left(\tfrac{1}{\mathrm{Re}}\tfrac{\partial \mathbf{u}}{\partial n} + p\mathbf{n}\right)_{m\neq 0}=0 && \text{on } \partial\Omega_{Nm}.
\end{aligned}$$

It is important to note that the mode coupling is ensured through the non-linear terms, which are computed in physical space and then transformed into Fourier and DG space through the Fourier transform and L^2 projection respectively.

4.2.3 DG-Fourier temporal discretisation

This section extends the numerical splitting scheme introduced in chapter 3 section 3.2.1 to account for the Fourier dimension:

$$\begin{aligned}
\tilde{\mathbf{u}}_m = (\alpha_0/\gamma_0)\mathbf{u}^n_m + (\alpha_1/\gamma_0)\mathbf{u}^{n-1}_m &- (\beta_0\Delta t/\gamma_0)FFT_m\left\{((\mathbf{u}^n-\mathbf{w}^n).\nabla)\,\mathbf{u}^n\right\} && (4.2.6)\\
&- (\beta_1\Delta t/\gamma_0)FFT_m\left\{\left((\mathbf{u}^{n-1}-\mathbf{w}^{n-1}).\nabla\right)\mathbf{u}^{n-1}\right\},\\
\mathbf{x}^{n+1} &= \mathbf{R}(\Delta\theta)(\mathbf{x}^n-\mathbf{x}_0)+\mathbf{x}_0, && (4.2.7)\\
\left(-\nabla^2{}_{2D}+\beta^2 m^2\right)p^{n+1}_m &= -\frac{\gamma_0}{\Delta t}\widehat{\nabla}_m.\tilde{\mathbf{u}}_m, && (4.2.8)\\
\left(-\nabla^2{}_{2D}+\beta^2 m^2+\frac{\mathrm{Re}\gamma_0}{\Delta t}\right)\mathbf{u}^{n+1}_m &= \mathrm{Re}\left(\frac{\gamma_0}{\Delta t}\tilde{\mathbf{u}}_m-\widehat{\nabla}_m p^{n+1}_m\right). && (4.2.9)
\end{aligned}$$

Let us note that the non-linear terms (Eq. 4.2.6) and the mesh rotation step (Eq. 4.2.7) are computed in physical space. The Neumann-Vorticity boundary condition for the Poisson equation becomes:

$$\begin{aligned}
\left(\frac{\partial p}{\partial n}^{n+1}\right)_m &= \mathbf{n}.\left(\frac{\partial \mathbf{u}}{\partial t}^{n+1}\right)_m - \beta_0\mathbf{n}.\left(FFT_m\left\{((\mathbf{u}^n-\mathbf{w}^n).\nabla)\,\mathbf{u}^n\right\}+\frac{1}{\mathrm{Re}}\nabla\times\boldsymbol{\omega}_m{}^n\right)\\
&\quad -\beta_1\mathbf{n}.\left(FFT_m\left\{\left((\mathbf{u}^{n-1}-\mathbf{w}^{n-1}).\nabla\right)\mathbf{u}^{n-1}\right\}+\frac{1}{\mathrm{Re}}\nabla\times\boldsymbol{\omega}_m{}^{n-1}\right) \quad (4.2.10)
\end{aligned}$$

Throughout this work, only homogeneous geometries in the Fourier direction (i.e. the z-direction) have been considered and thus $\mathbf{n} = (n_x, n_y, 0)^T$. Let us note that the normals only contribute to the first mode and thus for $m \neq 0$: $\mathbf{n}_{m\neq 0} = 0$. The formulation for the curl of the vorticity follows:

$$\nabla \times \boldsymbol{\omega}_m = \left(\frac{\partial \omega_z}{\partial y} - i\beta m \omega_y, i\beta m \omega_x - \frac{\partial \omega_z}{\partial x}, \frac{\partial \omega_y}{\partial x} - \frac{\partial \omega_x}{\partial y} \right), \tag{4.2.11}$$

with the vorticity: $\boldsymbol{\omega}_m = \left(\frac{\partial w}{\partial y} - i\beta m v, i\beta m u - \frac{\partial w}{\partial x}, \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right)$, and $\mathbf{u} = (u, v, w)^T$.

4.2.4 Non-linear terms, aliasing error and the 3/2 dealiasing rule

Aliasing may occur when computing the non-linear terms since they represent the product of trigonometric functions, when using a Fourier representation. In particular, this form of numerical error is associated with the use of discrete Fourier transforms (i.e. a truncated series) as opposed to continuous transforms (i.e. with infinite terms). In this section, the author follows [24] to introduce the aliasing phenomenon and the well known 3/2 dealiasing rule.

Let us consider two one-dimensional truncated Fourier expansions:

$$u_j = \sum_{m=-M/2}^{M/2-1} u_m e^{i\beta m z_j}, \quad v_j = \sum_{m=-M/2}^{M/2-1} v_m e^{i\beta m z_j}, \quad j = 0, 1..., M-1, \tag{4.2.12}$$

and define the product in physical space as: $w_j = u_j v_j,\ j = 0, 1..., M-1$. One can use the Fourier relation:

$$\frac{1}{M} \sum_{j=0}^{M-1} e^{-ipz_j} = \begin{cases} 1 & \text{if } p = Ml,\ l = 0, \pm 1, \pm 2, ... \\ 0 & \text{otherwise,} \end{cases}$$

to obtain the expression for the Fourier coefficients for the product:

$$w_q = \sum_{m+n=q} u_m v_n + \sum_{m+n=q\pm M} u_m v_n. \tag{4.2.13}$$

The first term on the right hand side corresponds to the convolution sum and the second term is the aliasing error.

To remove the described aliasing error, various techniques are available and the reader is referred to the monograph by Canuto et al. [24] for further details. Among them, the most popular is, probably, the 3/2 de-aliasing rule, also known as removal by padding or truncation, which is the approach considered in this work.

Let us consider a larger stencil of points $A \geq 3M/2$ to represent u_j,v_j and w_j, where zeros are padded to the extra coefficients $m \geq M/2$. With the extended stencils, Eq. 4.2.13 becomes:

$$w_q = = \sum_{m+n=q} u_m v_n + \sum_{m+n=q\pm A} u_m v_n. \tag{4.2.14}$$

Since only the values of w_q for $|q| \leq M/2$ are of interest and u_m and v_m are zero for $|m| > M/2$, the worst case for the aliasing term to vanish can be found for $m = n = -M/2$ and $q = M/2 - 1$, which leads to: $-M/2 - M/2 \leq M/2 - 1 - A$ (i.e. $m + n \leq q - A$). Therefore an aliasing free solution is provided for $A \geq 3M/2 - 1$. This analysis shows that $3M/2$ points are necessary to avoid aliasing which gives the name to the 3/2 dealiasing rule.

To compute the non linear terms using the described 3/2 dealiasing rule it suffices to:

- Pad zeros to the Fourier coefficient vectors from -3M/2 to -M/2 and from M/2-1 to 3M/2-1 .
- Calculate the $invFFT$ of these coefficients to obtain the expanded physical values of u_j and v_j.
- Perform the multiplication $w_j = u_j v_j$ in physical space

- Calculate the FFT of the product to obtain the coefficient w_q, $-3M/2 < q < 3M/2-1$
- Chop off the coefficient from -3M/2 to -M/2 and from M/2-1 to 3M/2-1.

This simple algorithm produces an aliasing error free product.

4.2.5 Implementation details

4.2.5.1 Real to complex transform

As suggested by the mapping introduced in Fig. 4.1, the use of the Fourier transform that concerns this thesis involves only real numbers in physical space, which implies a symmetric transform in Fourier space. For real to complex transforms, it is only necessary to compute $M/2+1$ modes (as opposed to the M-1 modes necessary for a complex to complex transform). This is a direct consequence of the symmetry of the Fourier transform and implies $f(\xi_m) = f^*(\xi_{-m})$, where f^* denotes the complex conjugate of f and $m = 0, .., M/2$. One can use this property to reduce the computational cost for the evaluation of the Fourier transform (e.g. using the real-to-complex fast Fourier transform and functions $c2r$, $r2c$ from the *fftw-package* [57]). As a consequence of this symmetry and for an even number of Fourier modes M, the modes corresponding to $m = -M/2$ and $m = 0$ should be purely real (i.e. zero imaginary part). This factor needs to be considered when computing first order derivatives. An appropriate formulation for these is given by:

$$\frac{\partial \mathbf{u}(x,y,z)}{\partial z} = \sum_{m=-M/2+1}^{M/2-1} i\beta m \mathbf{u}_m(x,y) e^{i\beta m z}, \tag{4.2.15}$$

$$\frac{\partial p(x,y,z)}{\partial z} = \sum_{m=-M/2+1}^{M/2-1} i\beta m p_m(x,y) e^{i\beta m z}, \tag{4.2.16}$$

where the explicit treatment for the first mode can be seen (the sum starts at $m = -M/2+1$), implying that the first term is multiplied by zero to maintain symmetry.

4.2.5.2 Complex arithmetic

The described Fourier extension introduces complex-number arithmetics. To handle complex-number operations, the developed solver uses the complex-number arithmetic capabilities provided by the C++ language. In addition, the intel-MKL pardiso solver [118] in its complex version of the sparse direct solver (i.e. Cholesky factorization) has been used to solve the required linear systems.

4.3 Parallelisation

Parallelisation is performed through the Message Passing Interface (MPI) paradigm (i.e. OpenMPI libraries [59]). To parallelise the present solver, the author follows the parallelisation strategy proposed in [50] for a conformal h/p Spectral-Fourier solver.
The solution of the incompressible 3D NS equations requires computation of $M/2+1$ independent solves Eq. (4.2.8)-(4.2.9) and of non-linear terms Eq. (4.2.6). A natural parallelisation strategy takes advantage of this decoupling and assigns an equal number of linear solves to each computational node. This strategy ensures that the only necessary communication is performed at the beginning and end of each step and before computation of the non-linear terms. In addition, one can obtain further performance improvements if the computation of the non-linear terms is parallelised. This is achieved, in the present implementation, through a domain decomposition strategy, where each computational node computes the non-linear terms in physical space over an equal number of mesh elements. The described parallelisation strategy is illustrated in Fig. 4.2.

4.4 Verification

Two problems are introduced in this section for verification: an elliptic problem and a 3D NS problem with an exact solution.

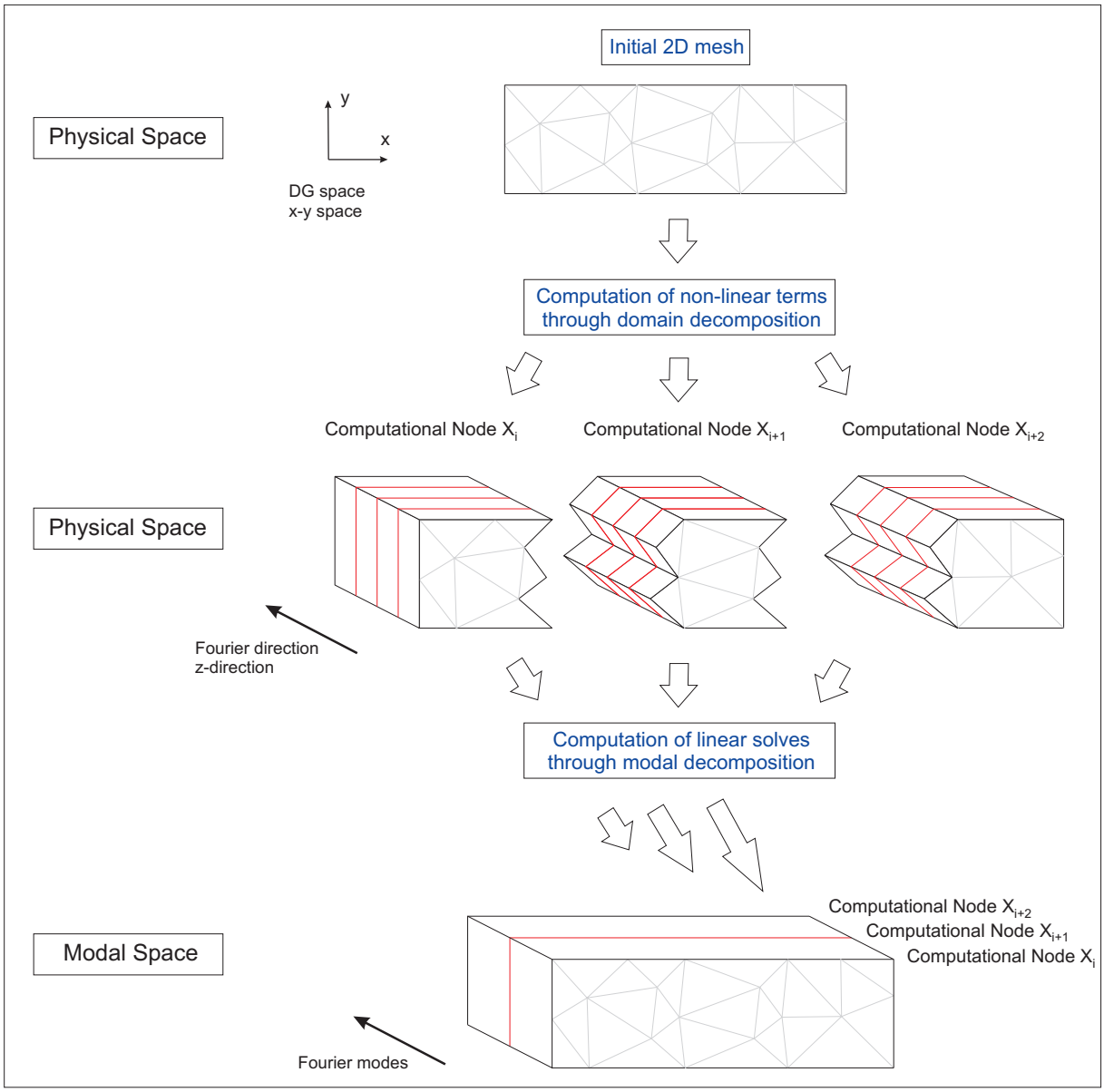

Figure 4.2: Schematic for the parallelisation of the DG-Fourier solver.

4.4.1 Elliptic equation

Fig. 4.3 depicts the DG-Fourier solution for a Poisson equation (i.e. $-\triangle \mathbf{u} = g(x,y)f(z)$) extended in the third direction using various functions $f(z)$. This Poisson problem has been previously introduced in chapter 2 section 2.5.1 and is defined through $g(x,y) = 2\pi^2 \cos(\pi x)\cos(\pi y)$ in $\Omega = [0,1]^2$. The figure shows that three dimensionality is only subject to finding the appropriate function to describe the variability in the third direction.

In previous sections of this chapter, the author has introduced the term "homogeneity in the third direction" to characterise the Fourier extension. However, this example shows that this term can be relaxed to "transformable via FFT". For simplicity and due to time limitations, in this work only homogeneous geometries in the third direction are considered. However, this section shows that more complicated geometries can be considered using the presented method.

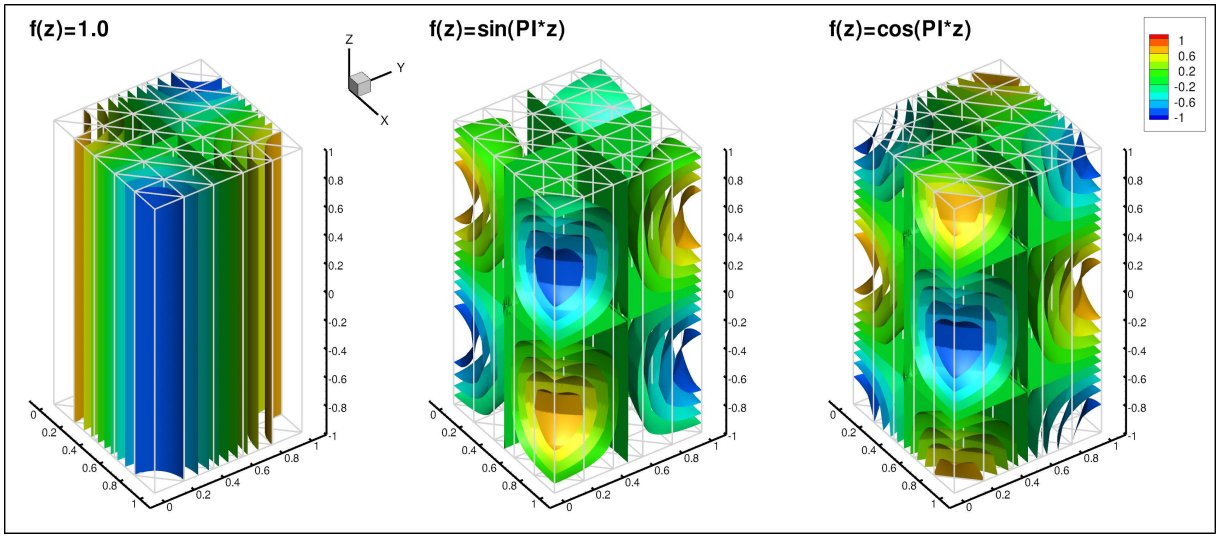

Figure 4.3: DG-Fourier solution for a 3D Poisson problem for various forcing functions $f(z)$. In the z-direction, 16 Fourier planes are used.

4.4.2 Navier-Stokes problem with exact solution

The selected problem to test the full incompressible NS equations in three dimensions is the steady, periodic in the z-direction, problem defined in [28]. This test problem, with limited physical meaning, is obtained by ensuring that the pressure field satisfies the y-momentum equation with zero forcing (i.e. $f_y(x, y, z) = 0$). The exact solution is given by:

$$\begin{aligned}(u, v, w) &= e^{\lambda x}\left(\cos\mu y\cos mz, \sin\mu y\cos mz, -\left[\frac{\lambda+\mu}{m}\right]\cos\mu y\sin mz\right), \\ p &= \frac{\mu}{\mathrm{Re}}e^{\lambda x}\cos\mu y\cos mz\left[1+\frac{m^2}{\mu^2}-\frac{\lambda^2}{\mu^2}\right]+e^{2\lambda x}\frac{\cos^2\mu y}{2}\left[1+\frac{\lambda}{\mu}\right], \quad (4.4.1)\end{aligned}$$

provided that the forcing terms have the following form:

$$\begin{aligned} f_x(x,y,z) &= \frac{1}{\mathrm{Re}} e^{\lambda x}\cos\mu y\cos mz A_1 + e^{2\lambda x}\cos^2\mu y A_2 \\ &\quad +\mu e^{2\lambda x}\left[\cos^2\mu y\sin^2 mz - \cos^2 mz\sin^2\mu y\right], \\ f_y(x,y,z) &= 0, \\ f_z(x,y,z) &= \frac{1}{\mathrm{Re}} e^{\lambda x}\cos\mu y\cos mz B_1 + e^{2\lambda x}\sin 2mz B_2, \end{aligned} \tag{4.4.2}$$

with the constants:

$$\begin{aligned} A_1 &= -\lambda^2 + m^2 - \frac{\lambda^3}{\mu} + \frac{\lambda m^2}{\mu} + \lambda\mu + \mu^2\ , \quad A_2 = 2\lambda + \frac{\lambda^2}{\mu}, \\ B_1 &= \frac{\lambda^3}{m} - \lambda m + \frac{\lambda^2 m}{\mu} - \frac{m^3}{\mu} + \frac{\lambda^2\mu}{m} - 2m\mu - \frac{\lambda\mu^2}{m} - \frac{\mu^3}{m}\ , \quad B_2 = \frac{\lambda\mu+\mu^2}{2m}. \end{aligned}$$

This problem is solved in the rectangular domain $\Omega = [-1,1]\times[0,1]\times[0,1]$, where the

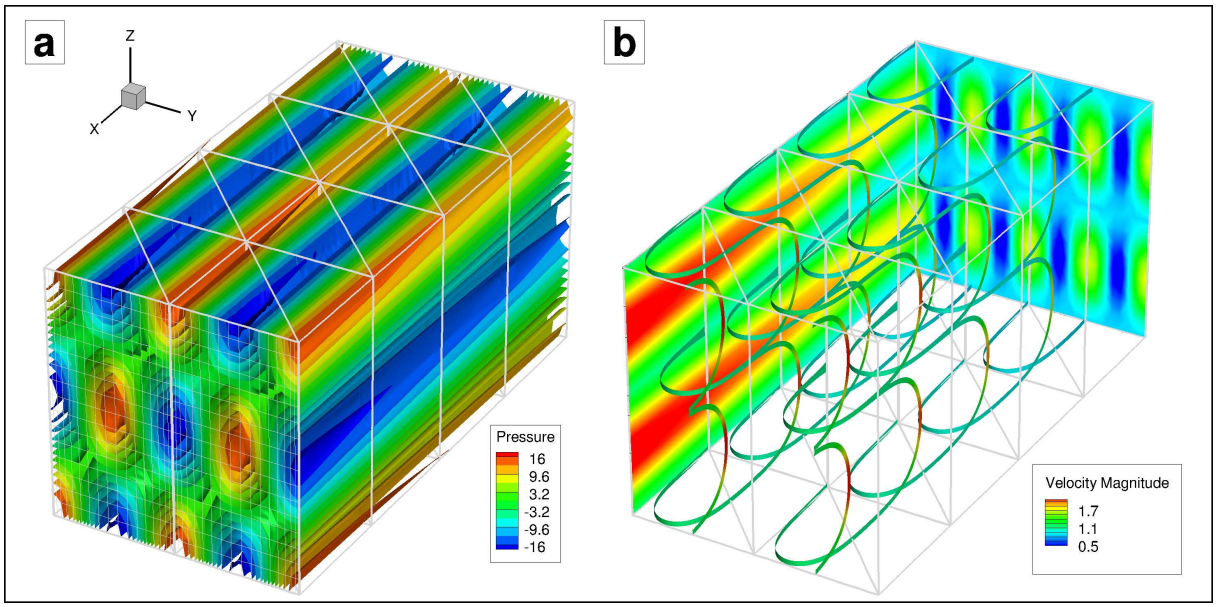

Figure 4.4: DG-Fourier solution for a Navier-Stokes problem with an exact solution (a) Pressure contours and (b) Ribbons and contours colored by velocity magnitude.

periodic $z-$direction is represented through the Fourier expansion and the $x-y$ plane is resolved using the DG approach. The following parameters are set: Re $= 1.0$, $\mu = 4\pi$,

$\lambda = 0.15$ and $m = 2\pi$ and all simulations are run for 10 time steps with a fixed size of $\triangle t = 1 \times 10^{-4}$. The DG solution for polynomial order $k = 5$ in the $x - y$ DG plane and 16 Fourier z-planes is depicted in Fig. 4.4.

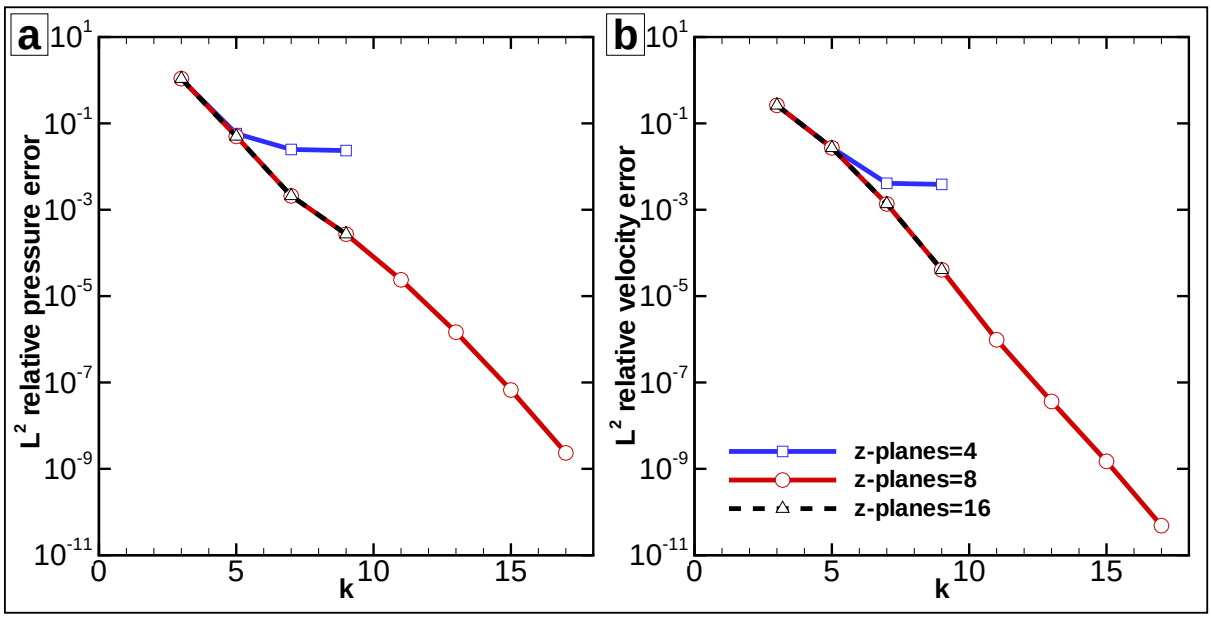

Figure 4.5: DG-Fourier solution for a Navier-Stokes problem with an exact solution: L^2 relative error for p-refinement and three different Fourier resolutions: number of Fourier planes: $z-$plane$= 4, 8$ and 16.

In addition, Fig. 4.5 shows the exponential convergence obtained for the L^2 relative error norm and p-refinement, at the plane $z = 0$ for three resolutions in the Fourier direction: number of $z-$planes=4, 8 and 16. Exponential decay of the error is observed for pressure and velocity when enough resolution in the Fourier direction is considered, here $z-$planes ≥ 8 are required.

4.5 Three dimensional flows

In this section, various three dimensional flows are presented. Qualitative comparisons with published simulations for 3D flows (i.e. square and circular cylinders) provide verification and validation for the DG-Fourier solver. In addition, the section illustrates how

at moderately low Reynolds number Re $>$ 150 (where Re is based on the free stream velocity and characteristic geometric length scale) flows can develop three dimensional flow features that cannot be ignored.

4.5.1 Square and circular cylinders

In [148], Robichaux et al. summarised the three dimensional spanwise modes for square and circular cylinders, together with their characteristic Reynolds number and spanwise wave lengths. These are natural modes found by Floquet analysis (i.e. Floquet modes). Blackburn et al. [19], provided a corrected version of the mechanisms driving some three dimensional flow features and showed that additional modes exist (i.e. SW-TW). Table 4.1 summarises the different 3D modes with their associated Reynolds number (i.e. minimum Reynolds number to trigger the three dimensional mode) and associated wavelengths. The reader is referred to [180] and [19] for a wider explanation and illustrations of the 3D structures for the various spanwise modes.

Mode	Square cylinder		Circular cylinder	
	Re	λ	Re	λ
A	162 ± 12	5.22	188	3.96
B	190 ± 14	1.2	259	0.82
C	-	-	170	1.8
S	200 ± 5	2.8	-	-
SW-TW	216	2.65	377	3.5

Table 4.1: Three dimensional modes for square and circular cylinders.

To provide validation for the DG-Fourier solver, three dimensional flows around a square and a circular cylinder are simulated. The square cylinder has sides of length $L = 1$ and a mesh consisting of 1064 triangular elements. The circular cylinder requires curved quadrilateral elements to account for the circular geometry and a mesh consisting of 1066 tri-quad elements. The diameter of the circular cylinder is $L = 1$. The time step is set to $\triangle t = 0.0075$ with a polynomial order $k = 3$ in the $x - y$ DG plane for

all simulations. To discretise the spanwise direction, 32 Fourier planes are used in the z-direction. The geometric spanwise length (L_z) is different between the simulations to accommodate for the expected wavelengths as shown in table 4.1.

Fig. 4.6 shows streamwise vorticity (ω_x) iso-surfaces for Reynolds numbers corresponding

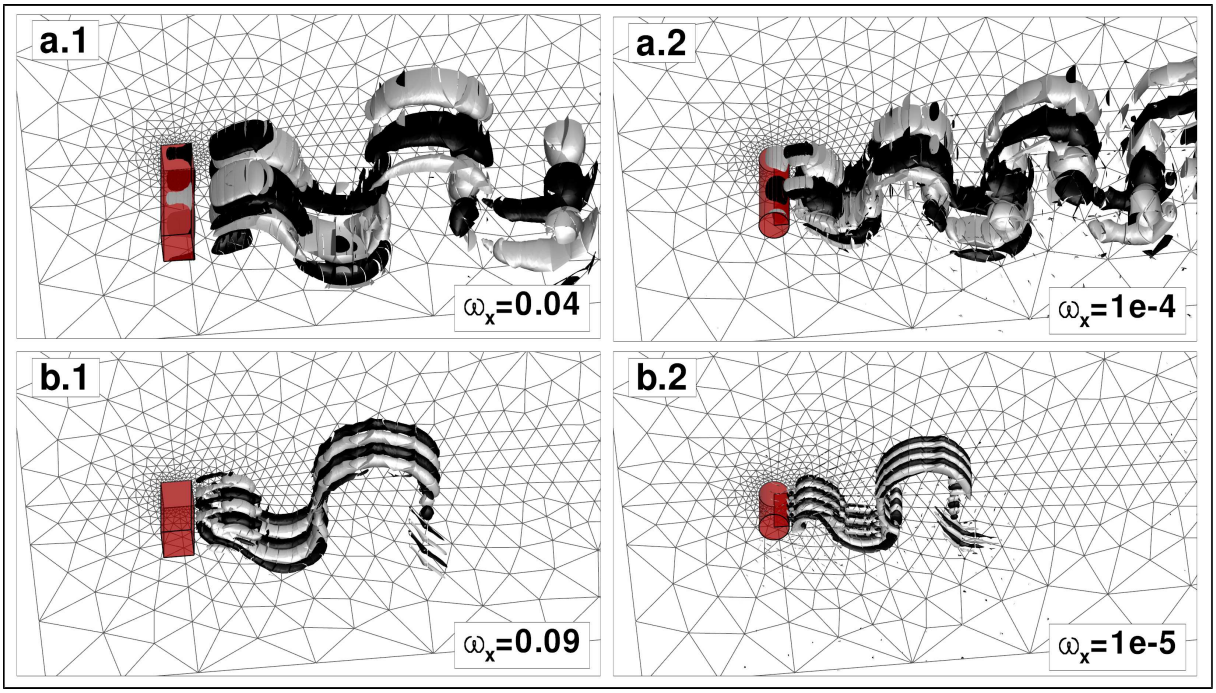

Figure 4.6: DG-Fourier solution for the flow aft square and circular cylinders for (a) mode A and (b) mode B. (a.1) Square cylinder at Re=167, (b.1) Square cylinder at Re=195, (a.2) Circular cylinder at Re=188 and (b.2) Circular cylinder at Re=270. The figures show iso-surfaces of streamwise vorticity (ω_x), gray shows positive and black negative iso-values.

to modes A and B for square and circular cylinders. It can be seen that the two geometries require different Reynolds numbers to excite the same three dimensional modes. The spanwise patterns are comparable, hence showing the same mode behaviour. Let us note that mode A and mode B show different streamwise vorticity patterns. On the one hand, mode A (Fig. 4.6.a.1 and Fig. 4.6.a.2) shows streamwise vortices of one sign that are in a staggered arrangement from neighbouring *braid* regions [180]. On the other hand, mode B shows finer scales and streamwise vortical structures of same sign along the streamwise direction (i.e. there is no change of sign). It can also be seen in Fig. 4.6 that higher

values of streamwise vorticity are obtained for the square cylinder, when compared to the circular geometry, suggesting a more energetic three dimensional flow, which is likely to relate to the geometric sharpness of the bluff body.

In addition, the development of the spanwise SW-TW mode (i.e. Re=400) for a circular

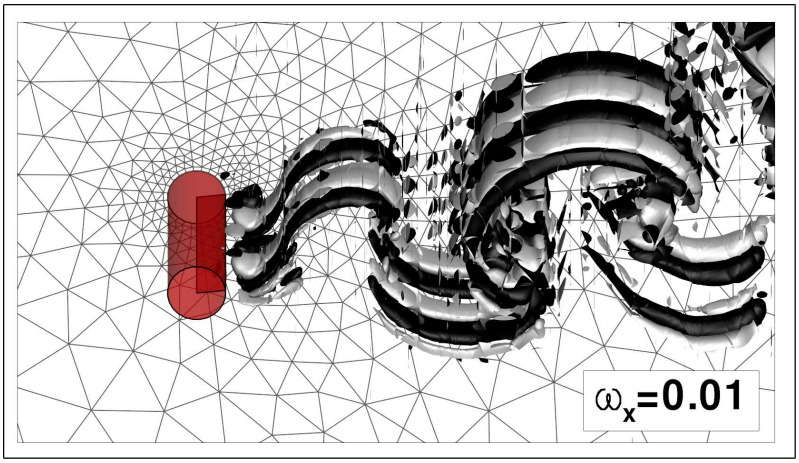

Figure 4.7: DG-Fourier solution for the flow aft a circular cylinder at Re=400 showing the SW-TW spanwise mode. The figure shows iso-surfaces of streamwise vorticity (ω_x), gray shows positive and black negative iso-values.

cylinder is shown in Fig. 4.7. The reader is referred to [19] for comparison of flow structures with published results. This simulation required a polynomial order $k = 4$ in the DG plane.

This validation exercise shows that even with a coarse DG discretisation (i.e. x-y plane), three dimensional effects can be properly captured through the developed DG-Fourier solver. Furthermore, these snapshots provide a flow pattern catalog that may be used to identify spanwise modes in geometries where these are a priori unknown (see following section).

4.5.2 Circular cylinder in the wake of a pitching NACA0012 blade

This chapter concludes with the computation of the 3D flow over a circular cylinder, which is shadowed by a pitching NACA0012. This test case is presented to illustrate the possibility of locating static and rotating 3D geometries together in a single mesh domain. In addition, various features presented in this thesis are included in a single simulation: high order sliding meshes and external curved boundary surfaces for NACA airfoils and circular cylinders.

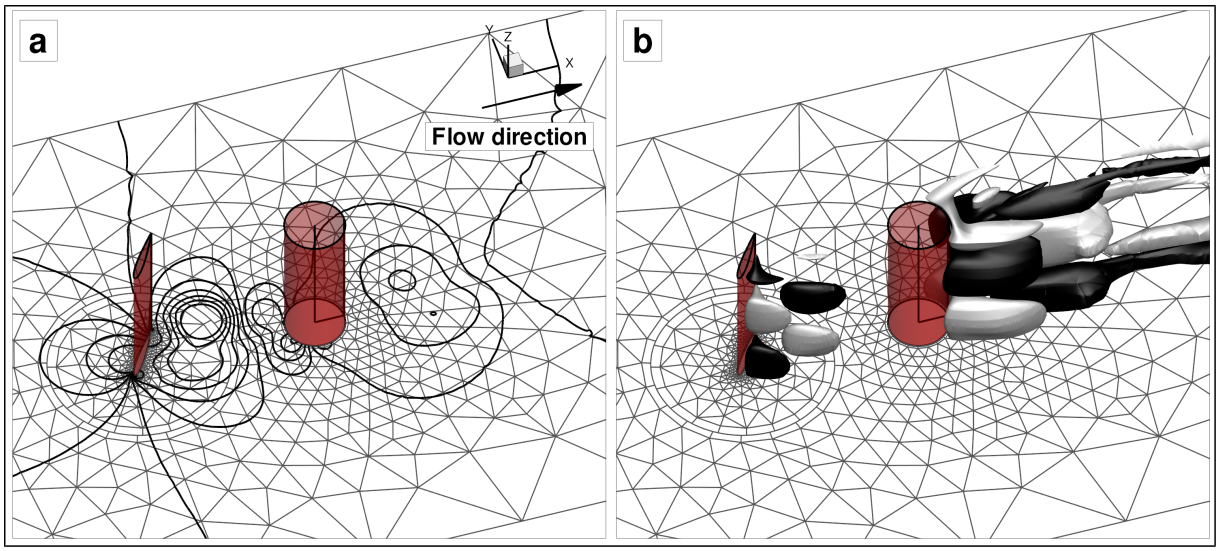

Figure 4.8: DG-Fourier solution snapshot of the flow around a 3D circular cylinder in the wake of a 3D pitching NACA0012 blade: (a) 11 pressure contours [-0.6:0.6] and (b) Iso-surfaces of z-velocity $w = \pm 1.5 \times 10^{-6}$.

The NACA foil is pitching with a rotational speed of $L\omega/U = 0.5$ and the Reynolds number based on the foil chord (which is equal to the cylinder diameter) is Re = 200 (i.e. large enough to trigger three dimensionalities in the flow). The spatial discretisation consists of polynomial orders k=4 in the DG plane (i.e. the x-y plane) and 16 Fourier planes in the z-direction. The time step is set to $\triangle t = 0.001$. Fig. 4.8 shows the pressure contours and iso-surfaces of z-velocity. It can be seen that the sliding mesh technique does not damage the numerical solution and captures the pressure interactions between static

and rotating geometries. In addition, Fig. 4.8.b shows that three dimensionalities are present in the wake of both geometries. Comparing Fig. 4.8 with the three dimensional modes for square and circular cylinders depicted in Fig. 4.6 and Fig. 4.7, it appears that the spanwise modal behaviour of the rotating blade and the circular cylinder present similarities to mode A. This qualitative comparison shows the importance of considering three dimensional flows even at low Reynolds numbers.

Chapter 5

Computing turbulent flows

Turbulent flows may be characterised as unsteady, three dimensional, with high vorticity and with chaotic behaviour. However, the main characteristic of turbulent flows is the presence of a broad spectrum, where a wide range of length and time scales coexist [53]. A turbulent flow spectrum may be recognised by the presence of an *inertial range* (i.e. $E(k) = f(k^{-5/3})$). Within this spectral range, the energy from larger structures (i.e. eddies) is handed down to smaller structures in an inviscid cascade fashion. Eventually, small enough eddies are dissipated by laminar viscosity within the *dissipation range*. The concept of *energy cascade*, which in turbulent flows defines the inertial range, was first introduced by Richardson in 1922 and then by Kolmogorov in 1941 (see [149] and references therein for details). A typical turbulent spectrum showing the various ranges of energy spectra is sketched in Fig. 5.1. The horizontal extent of the turbulent spectrum (i.e. the universal equilibrium range) can be shown (following dimensional arguments) to depend on $Re_\ell^{3/4}$, where $Re_\ell = u_\ell \ell/\nu$ is the Reynolds number based on a typical large eddy with characteristic velocity u_ℓ and length ℓ. It is the multiscale character of turbulent flows, which scales with the Reynolds number, that renders the numerical treatment of turbulence a difficult task.

Different approaches exist to derive suitable equations for the numerical treatment of turbulence and its effects, depending on the degree of computational resolution and mod-

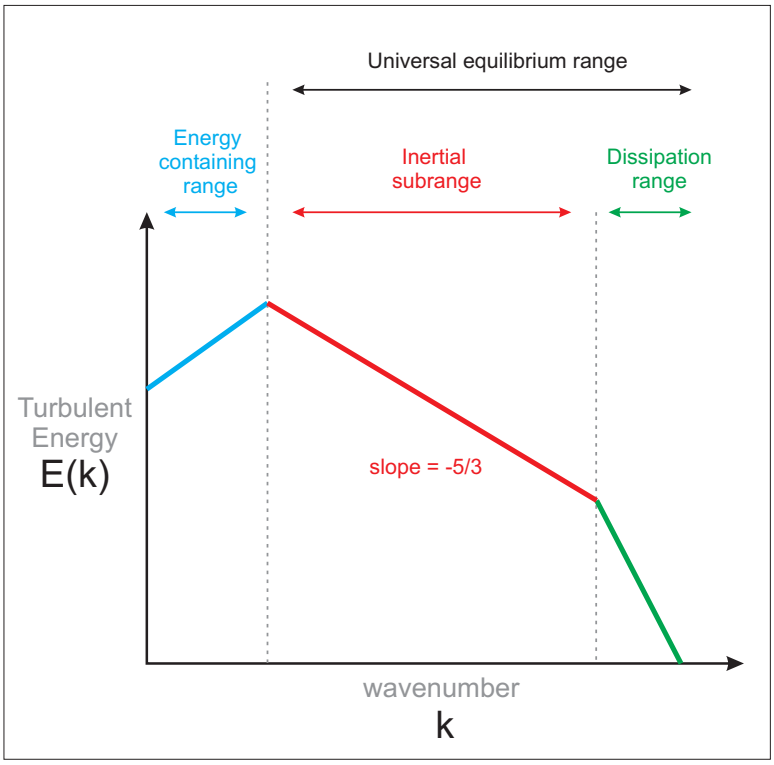

Figure 5.1: Schematic of a typical log-log turbulent spectrum showing the turbulent energy $E(k)$ against wave number k.

elling introduced. The Direct Numerical Simulation (DNS) approach solves the Navier-Stokes (NS) equations (henceforth referred to as DNS-NS) without additional modelling and hence resolves all flow scales. Consequently, this method requires a very high spatial and temporal resolution and is thus restricted to moderate Reynolds numbers and generally to simple geometries (see the review by Moin and Mahesh [121] for details). Since the number of degrees of freedom to compute a fully resolved flow simulation scales as $Re^{9/4}$ (based on the magnitude of the velocity fluctuation and the integral scale in three dimensional flows [54]), it is necessary to introduce a certain degree of modelling when computing turbulent flows. Hence, to obtain a "usable method" (i.e. computable with the available computational resources), it is necessary to develop modelling strategies to account for the effects of turbulence (e.g. on the skin friction), without explicitly resolving its complexity. Two main stream approaches exist: The Reynolds Averaged NS (RANS) approach and Large Eddy Simulation (LES) technique. The first considers time aver-

aging of the NS equations where the instantaneous velocities are decomposed into their temporal mean and fluctuating components (see [179] for a detailed description). The second approach considers spatial filtering (or projection) where the large structures are resolved, reducing modelling to the small turbulent structures (i.e. small eddies), which are considered to behave in a homogeneous isotropic fashion. In this thesis only the LES approach is considered.

The original approach to derive the appropriate equations for the LES formulation consists of applying spatial filtering to the NS equations (a detailed description can be found in the monograph by Sagaut [149]). Filtering allows for scale separation: large scales (i.e. large eddies) to be resolved and small scales (i.e. subgrid scale) to be modelled. It may be noted that scale separation may be considered also in spectral space (i.e. frequency domain), which leads to the so called *spectral models* [149].

In recent years, different frameworks for derivation and analysis of LES type methods have emerged such as, for example, the *variational multiscale method*. This encompasses projection/variational type numerical techniques (e.g. Finite Element, h/p Spectral, Discontinuous Galerkin) and was originally described by Hughes et al. [86] for general variational formulations in the context of computational mechanics. Since then, it has been extended and adapted to fluid dynamic problems by various authors (see [88] or [150] for reviews). This approach does not consider filtering but projection of scales (e.g. resolved large scales, resolved small scales and unresolved scales) on the space spanned by the finite element polynomials as the starting point for derivation of the LES equations. The original method considered the weak formulation of the NS equations but has been also extended to the strong form [150].

The filtered equation concept, led in 1963 to the first LES model, which was presented by Smagorinsky [158]. This model characterises the small eddies (e.g. unresolved scales smaller than the spatial resolution) through a turbulent viscosity (spatially and temporally flow varying viscosity), which relates to the resolved velocity strain in a mixing length

fashion. This approach has limitations, for example, it provides turbulent viscosity values that are non-zero near wall boundaries. This is, typically, a drawback of most basic LES closure models, which may require further modelling (e.g. wall damping functions) to provide physically realistic predictions. More evolved LES subgrid models account, to a certain extent, for these deficiencies (e.g. Wall Adapting Large Eddy (WALE) model, Germano dynamic approach or Germano-Lilly technique are some examples and detailed descriptions can be found in [149]).

To conclude this overview, the Implicit Large Eddy Simulation (ILES) is introduced since it is explored in the current work (see section 5.2.2). This approach, increasingly popular when combined with high order numerical methods, uses the numerical dissipation inherited from the numerical scheme (e.g. from upwinding the non-linear terms or Finite Element stabilisation terms) to account for turbulent effects. The underlying idea of this technique is that it is not necessary to derive new equations to compute turbulence (e.g. filtered NS equations), but that the DNS-NS equations can handle high Reynolds number flows provided that some numerical dissipation (i.e. providing stabilisation) is present when the DNS-NS equations are discretised (e.g. through upwinding of the non-linear terms). Indeed, this technique does not require explicit modelling of the turbulent viscosity and provides valuable results even for challenging problems such as the prediction of transitional flows [168].

A similar concept has been proposed in the context of high order methods (e.g. h/p Spectral) through the Spectral Vanishing Viscosity (SVV) method (e.g. [97] or [107]). This technique introduces an additional dissipative term (only to the highest modes) that enhances stability, vanishes in the laminar limit and provides spectrally accurate solutions (i.e. maintains exponential convergence). This technique is also explored in this thesis and further details of the SSV technique can be found in section 5.2.2.

This chapter examines two different techniques to compute turbulent flows. First, a LES-Smagorinsky (called Smagorinsky-DG) type model is derived for Discontinuous

Galerkin formulations and the considered temporal advancement splitting scheme. This model is restricted to two dimensional simulations in the present implementation since the Fourier extension renders the extension to 3D difficult to solve. However, the method could be suitable for a full DG discretisation and hence it is included in this chapter. Comparison of 2D simulations with experimental data show the potential of this methodology. Second, an ILES-DG SVV-Fourier method is proposed to compute three dimensional turbulent flows. This scheme combines two approaches: the Implicit Large Eddy Simulation (ILES) approach on the DG plane combined with the Spectral Vanishing Viscosity (SVV) method in the Fourier direction. Results for the ILES-DG SVV-Fourier approach show that it provides stable computations and good agreement with experimental data.

5.1 Chapter outline

Section 5.2 introduces two methodologies to compute turbulent flows: a Smagorinsky-DG model and an ILES-DG SVV-Fourier approach. In particular, section 5.2.1 details the derivation of the Smagorinsky-DG technique, including the definition of the filtered LES-NS equations and a description of the subgrid term (i.e. Smagorinsky subgrid term). In section 5.2.1.2, the modifications required to the splitting scheme to incorporate the Smagorinsky model are detailed. Section 5.2.1.3 studies the stability of the scheme to show the stabilising effect of incorporating a turbulent viscosity term. In section 5.2.2, the ILES-DG SVV-Fourier method is introduced to model 3D turbulent flows. Finally, validation cases for a circular cylinder and NACA foils, using the two turbulence models, are presented in section 5.3 together with results from other solvers and experimental data.

5.2 Two methodologies

5.2.1 Smagorinsky-DG model

Let us first consider a Smagorinsky-LES approach where the Boussinesq hypothesis is used to define the subgrid model (see section 5.2.1.1 for details). The classic LES approach requires derivation of new equations. To this end, this section follows the *multiscale separation* description detailed in [150], where the following decomposition for the primitive variables (velocity $\mathbf{u}$ and pressure p) is considered: $f(\mathbf{x},t) = \overline{f}(\mathbf{x},t) + f'(\mathbf{x},t)$, where $\mathbf{x} = (x,y,z)^T$ and t denotes time. Let us define $\overline{f}(\mathbf{x},t) = \mathfrak{P}(f(\mathbf{x},t))$ as the *resolved* part of $f(\mathbf{x},t)$ and $f'(\mathbf{x},t) = (Id - \mathfrak{P})(f(\mathbf{x},t))$, as the *unresolved* part, where Id is the identity operator. The operator $\mathfrak{P}$ is a general scale separation operator (e.g. filtering, projection) which is responsible for decreasing the computational cost of the simulation.

Applying the separator operator $\mathfrak{P}(\bullet)$ to the dimensional incompressible NS equations (i.e. $\mathfrak{P}(NS)$) in conservative form, one obtains the following set of equations in terms of resolved and unresolved fields:

$$\nabla.\overline{\mathbf{u}} = -A_1 \ , \quad \frac{\partial \overline{\mathbf{u}}}{\partial t} + \nabla.(\overline{\mathbf{u}} \otimes \overline{\mathbf{u}}) = -\nabla \overline{p} + \nu \nabla^2 \overline{\mathbf{u}} - (A_2 + A_3 + A_4), \tag{5.2.1}$$

with A_1 to A_4 defined as:

$$\begin{aligned} A_1 &= \mathfrak{P}(\nabla.)\mathbf{u} \ , \quad A_2 = \nabla.\mathfrak{P}(\Upsilon(\mathbf{u},\mathbf{u})) \ , \quad A_4 = \mathfrak{P}\left(\frac{\partial}{\partial t}\right)\mathbf{u}, \\ A_3 &= \mathfrak{P}(\nabla.)\Upsilon(\mathbf{u},\mathbf{u}) + \mathfrak{P}(\nabla)p + \nu\mathfrak{P}(\nabla^2)\mathbf{u}, \end{aligned} \tag{5.2.2}$$

where the non-linear term operator for the conservative form is defined as: $\Upsilon(\mathbf{u},\mathbf{v}) = \mathbf{u} \otimes \mathbf{v}$. The terms A_1, A_3 and A_4 relate to possible *commutation errors* (see [149] for definitions) between the scale separation operator and the spatial and temporal derivatives. The term A_2 is directly related to the non-linearity of the convective term and thus never vanishes. It can be seen that a non zero commutation error would lead to $A_1 \neq 0$

rendering the flow not divergence free.

Remark: *Under this general formulation, the Reynolds Averaged NS (RANS) equations can also be defined. By considering* $\overline{f}(\mathbf{x},t)$ *as the mean (ensemble averaged),* $f'(\mathbf{x},t)$ *as the fluctuating part of* $f(\mathbf{x},t)$ *and* $\mathfrak{P}(f(\mathbf{x},t))$ *a statistical average operator, the RANS equations can be obtained for* $A_1 = A_3 = A_4 = 0$ *(which follows from the commutation property of the statistical average operator) and* $A_2 = \nabla.\tau_{\mathbf{RANS}}$*, with* $\tau_{\mathbf{RANS}} = \mathbf{u}' \otimes \mathbf{u}'$ *referred to as the Reynolds stress tensor.*

The technique considered in this thesis is the Large Eddy Simulation approach. Under this modelling strategy, the operator $\mathfrak{P}(f(\mathbf{x},t))$ is classically defined as a low-pass filter. Assuming that this filter operator commutes with spatial and temporal operators (see [149] for examples), one obtains $A_1 = A_3 = A_4 = 0$. In addition, $A_2 = \nabla.\tau_{\mathbf{LES}}$ is called the subgrid stress tensor, defined through the turbulent stress: $\tau_{\mathbf{LES}} = \overline{\mathbf{u} \otimes \mathbf{u}} - \overline{\mathbf{u}} \otimes \overline{\mathbf{u}}$. This stress tensor accounts for the interactions between resolved and unresolved scales and thus requires modelling to provide mathematical closure.

5.2.1.1 Boussinesq assumption and Smagorinsky subgrid scale model

In 1887, Boussinesq hypothesised that the energy transfer mechanism in turbulent flows (from the resolved to the subgrid scales) is analogous to the molecular diffusion (i.e. Brownian motion) represented by the fluid kinematic viscosity ν. Following this hypothesis, it is possible to represent the subgrid turbulent sink of energy analogously to the drawn of energy that molecular viscosity causes on laminar flows. In mathematical terms, this hypothesis translates into: $-\nabla.{\tau^{\mathbf{d}}}_{LES} = \nabla.(\nu_t(\nabla\overline{\mathbf{u}} + \nabla^T\overline{\mathbf{u}}))$, where ν_t represent the turbulent viscosity (to be subsequently defined). In the last equation, the deviatoric stress tensor is defined as: ${\tau^d}_{ij} = \tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij}$, where δ_{ij} is the Kronecker delta defined as: $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$. Since the term $(\nabla\overline{\mathbf{u}} + \nabla^T\overline{\mathbf{u}})$ has zero trace for incompressible flows, one needs to define a zero trace stress tensor to model. This requirement leads to the previous definition for ${\tau^d}_{ij}$. In practice, this term does not require modelling since it

is added to the resolved (i.e. filtered) static pressure [149]. The modified pressure is for simplicity denoted p in what follows.

To simplify the notation, let us drop henceforth the overbars to represent the resolved scales and set: $\overline{\mathbf{u}} = \mathbf{u}$ and $\overline{p} = p$.

The turbulent viscosity ν_t may be defined through the zero equation model known as Smagorinsky closure and modified for high order h/p methods in [96]:

$$\nu_t = (c_s \triangle)^2 |\mathbf{S}|, \quad |\mathbf{S}| = (2 S_{ij} S_{ij})^{1/2}, \tag{5.2.3}$$

$$\triangle = \left(\mathcal{A} \left(\frac{\pi}{k} \right)^2 \Delta z \right)^{1/3}, \tag{5.2.4}$$

where $\mathcal{A}$ defines the element area in 2D (i.e. the DG-dimension), k is the polynomial order of the element and $c_s = 0.1$ a fixed constant. It can be shown that $l_s = c_s \triangle$ is the Smagorinsky length scale. In addition, let us note that the ratio $\frac{\pi}{k}$ represents the resolved half wavenumber for a given polynomial of order k (see [96] and references therein). It is interesting to note that the Smagorinsky turbulent viscosity is invariant under rotation and hence can be used within the moving mesh without modifications.

As it was mentioned in the introduction, 2D turbulence cannot be really representative of 3D turbulence. Therefore 2D LES models cannot be representative of turbulent flows. However, the Smagorinsky closure model in 2D (setting for example $\Delta z = 1$) can be interpreted as a mixing length model, where the mixing length is defined in terms of the spatial resolution taking into account the element area and polynomial order. The nature of the solution is not of LES type but may be interpreted from an unsteady RANS perspective.

If a 3D formulation is to be used (not in this thesis as explained in the next section) then the Fourier filter length becomes: $\Delta z = L_z/M$, where L_z is the Fourier dimension and M is the number of planes in physical space (see Fig. 4.1 in chapter 4).

5.2.1.2 Modified splitting scheme for turbulent flows

The starting point to treat the NS equations with the additional turbulent terms is similar to the approach suggested by Karamanos and Sherwin [98] and Karamanos et al. [96] but the overall scheme proposed here is different and takes advantage of the present DG framework (i.e. discontinuous functions between elements). As presented in previous chapters, the pressure and laminar viscous terms are treated implicitly and non-linear terms explicitly. As for the turbulent terms, they are split algebraically and treated partially explicitly and partially implicitly. In this chapter, it is advantageous to present the equations in dimensional form, and not in their non-dimensional form as in previous chapters. The discretised momentum equation reads:

$$\begin{aligned} \frac{\gamma_0 \mathbf{u}^{n+1} - \alpha_0 \mathbf{u}^n - \alpha_1 \mathbf{u}^{n-1}}{\Delta t} = & - \nabla p^{n+1} - \beta_0 \mathcal{F}^n(\nu_t, \mathbf{u}, \mathbf{w}) - \beta_1 \mathcal{F}^{n-1}(\nu_t, \mathbf{u}, \mathbf{w}) \\ & + (\nu + \nu_t^{n+1}) \nabla^2 \mathbf{u}^{n+1}, \end{aligned} \tag{5.2.5}$$

where $\mathcal{F}(\nu_t, \mathbf{u}, \mathbf{w})$ account for the non-linear terms and the explicitly treated turbulent terms: $\mathcal{F}(\nu_t, \mathbf{u}, \mathbf{w}) = ((\mathbf{u} - \mathbf{w}).\nabla)\mathbf{u} - (\nabla \nu_t).(\nabla \mathbf{u} + \nabla^T \mathbf{u})$. To split the turbulent terms, the following equality has been used:

$$\nabla.[\nu_t(\nabla \mathbf{u} + \nabla^T \mathbf{u})] = (\nabla \nu_t).(\nabla \mathbf{u} + \nabla^T \mathbf{u}) + \nu_t \nabla^2 \mathbf{u}. \tag{5.2.6}$$

In the present implementation, the first term in 5.2.6 is treated explicitly and the second implicitly, as shown in Eq. 5.2.5.

By introducing an intermediate velocity $\tilde{\mathbf{u}}$, one can define the following splitting algorithm, where as seen previously Eq. 5.2.9 and Eq. 5.2.10 are computed in the DG

variational framework:

$$\begin{aligned}
\tilde{\mathbf{u}} = \alpha_0/\gamma_0 \mathbf{u}^n + \alpha_1/\gamma_0 \mathbf{u}^{n-1} &- (\beta_0 \Delta t/\gamma_0) F^n(\nu_t, \mathbf{u}, \mathbf{w}) \\
&- (\beta_1 \Delta t/\gamma_0) F^{n-1}(\nu_t, \mathbf{u}, \mathbf{w}), \qquad (5.2.7) \\
\mathbf{x}^{n+1} &= \mathbf{R}(\Delta\theta)(\mathbf{x}^n - \mathbf{x}_c) + \mathbf{x}_c, \qquad (5.2.8) \\
-\nabla^2 p^{n+1} &= -\gamma_0 \frac{1}{\Delta t} \nabla . \tilde{\mathbf{u}}, \qquad (5.2.9) \\
-\nabla^2 \mathbf{u}^{n+1} + \frac{\gamma_0}{(\nu + \nu_t^{n+1})\Delta t} \mathbf{u}^{n+1} &= \frac{1}{(\nu + \nu_t^{n+1})} (\frac{\gamma_0}{\Delta t} \tilde{\mathbf{u}} - \nabla p^{n+1}). \qquad (5.2.10)
\end{aligned}$$

As in previous chapters, second order Adams-Bashforth values for the coefficients γ_0, α_0, α_1, β_0 and β_1 are selected. Let us also define a second order approximation for the turbulent viscosity as: $\nu_t^{n+1} = \beta_0 \nu_t^n + \beta_1 \nu_t^{n-1}$.Eq. 5.2.9 defines a Poisson equation for the pressure, which is closed using the following boundary condition at walls and inflows:

$$\begin{aligned}
\frac{\partial p}{\partial n}^{n+1} &= \boldsymbol{\varpi} \times \boldsymbol{\varpi} \times \mathbf{x}^{n+1} - \beta_0 \mathbf{n}. \left(F(\nu_t^n, \mathbf{u}^n, \mathbf{w}^n) + \nu \nabla \times \boldsymbol{\omega}^n\right) \qquad (5.2.11) \\
&\quad -\beta_1 \mathbf{n}. \left(F(\nu_t^{n-1}, \mathbf{u}^{n-1}, \mathbf{w}^{n-1}) + \nu \nabla \times \boldsymbol{\omega}^{n-1}\right),
\end{aligned}$$

where the first term was derived in chapter 3 and is necessary to account for rotating boundary walls and $\boldsymbol{\omega} = \nabla \times \mathbf{u}$. It can be seen that zero turbulent viscosity at inflow and walls (i.e. $\nu_t = 0$) has been imposed. This condition is required by the temporal scheme's compatibility condition for the pressure (derived in [100]): $\int_\Omega \nabla^2 p = \int_{\partial\Omega} \frac{\partial p}{\partial n}$. At outflow boundaries, Dirichlet boundary conditions for pressure are used.

Remark: *The modified turbulent scheme should provide a consistent method to compute turbulent flows when a full 3D DG discretisation (i.e. DG in three directions) is considered. However, this modified scheme is not suitable for the DG-Fourier methodology introduced in the previous chapter. Inspection of the Helmholtz equation (Eq. 5.2.10) shows that when a Fourier discretisation is introduced in the z-direction, the term* $\frac{\boldsymbol{u}^{n+1}}{(\nu+\nu_t^{n+1})\Delta t}$ *results in a convolution in modal space. This convolution couples all modes rendering the*

scheme practically not solvable (i.e. it would require solving a coupled system). For this reason, this scheme is only used for 2D simulations in the present work in order to show its potential. To compute 3D turbulent flows, a different approach is introduced in section 5.2.2.

5.2.1.3 Algebraic system and enhanced stability

This section extends the algebraic system detailed in chapter 2 section 2.5.3.2. In this case, let us consider the complete set of the discretised NS equations Eq. 5.2.7 to 5.2.10 and make use of the matrix definitions of chapter 2 section 2.2.3. Before proceeding, let us define a modified symmetric turbulent mass matrix:

- **Local Turbulent Mass matrix**: $\mathbf{M^{el}_{turb}} := \{M^{el}_{\text{turb}ij} = \int_{el} \frac{1}{\nu+\nu_t}\phi_i(\mathbf{x})\phi_j(\mathbf{x})d\mathbf{x}\}$,

and the modified non-linear terms:

- $\mathcal{F}_{\mathbf{r}}(\nu_t, \mathbf{u}_h, \mathbf{w}) = \sum_{el\in\Omega_h} \mathcal{F}^{el}_r(\nu_t, \mathbf{u}_h, \mathbf{w}) := \{\mathcal{F}^{el}_{r\,j}(\nu_t, \mathbf{u}_h, \mathbf{w}) = \sum_{el\in\Omega_h} \int_{el} \mathcal{F}_r(\nu_t, \mathbf{u}_h, \mathbf{w})\phi_j d\mathbf{x}\}$, with $r = x$, y or z (no-summation).

Remark: *The definition of the Local Turbulent Mass matrix is only possible because the DG framework allows for discontinuous function to be defined in each element and continuity is not required. In a conformal setting, this scheme would not be appropriate.*

The algebraic splitting scheme is summarised as:

$$\mathbf{M}\frac{\gamma_0\tilde{U_r} - \alpha_0 {U_r}^n - \alpha_1 {U_r}^{n-1}}{\Delta t} = -\beta_0 \mathcal{F}^n_r(\nu_t, U, \mathbf{w}) - \beta_1 \mathcal{F}^{n-1}_r(\nu_t, U, \mathbf{w}), \tag{5.2.12}$$

$$\mathbf{A}^{Pois}_{(\alpha=0,\sigma)} P^{n+1} = -\frac{\gamma_0}{\Delta t}\mathbf{K_r}\tilde{U_r} + \mathbf{RHS}^{Pois}_{(g=0,\sigma)}, \tag{5.2.13}$$

$$\left(\mathbf{A}^{Helm}_{(\alpha=0,\sigma)} + \frac{\gamma_0}{\Delta t}\mathbf{M_{turb}}\right) {U_r}^{n+1} = \frac{\gamma_0}{\Delta t}\mathbf{M_{turb}}\left(\tilde{U_r} - \frac{\Delta t}{\gamma_0}\mathbf{M}^{-1}{\mathbf{K_r}}^T P^{n+1}\right) + \mathbf{RHS}^{Helm}_{(g=0,\sigma)}, \tag{5.2.14}$$

where $r = x, y, z$ (no-summation in Eq. 5.2.12 and Eq. 5.2.14, but summation over r in Eq. 5.2.13). To obtain the last Helmholtz equation Eq. 5.2.14, one can use the

original scheme defined in chapter 2 section 2.2.1 to express the intermediary divergence free velocity $\tilde{\tilde{U}}$ as a function of the intermediate velocity $\tilde{U}$ and the pressure P:

$$\tilde{\tilde{U}}_r = -\frac{\Delta t}{\gamma_0}\mathbf{M}^{-1}{\mathbf{K_r}}^T P^{n+1} + \tilde{U}_r. \tag{5.2.15}$$

As presented in chapter 2 section 2.5.3.2, one can recast the splitting scheme in matrix form when the steady case without non-linear terms is considered. The resulting system reads:

$$\mathbf{S}^{\mathbf{algTurb}}_{\mathbf{split}\ \mathbf{h}} = \begin{pmatrix} {\mathbf{M_{turb}}}^{-1}\mathbf{MA}^{Helm}_{(\alpha=0,\sigma)} & \mathbf{K}^T \\ \mathbf{K} & \frac{\triangle t^*}{\gamma_0}\mathbf{A}^{Pois}_{(\alpha=0,\sigma)} \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} {\mathbf{M_{turb}}}^{-1}\mathbf{MRHS}^{Helm}_{(g=0,\sigma)} \\ \mathbf{RHS}^{Pois}_{(g=0,\sigma)} \end{pmatrix}. \tag{5.2.16}$$

As in chapter 2, the solvability of this system is explored though the pressure Schur complement of the system defined by Eq. 5.2.16:

$$Schur(\mathbf{S}^{\mathbf{algTurb}}_{\mathbf{split}\ \mathbf{h}}) = \frac{\triangle t^*}{\gamma_0}\mathbf{A}^{Pois}_{(\alpha=0,\sigma)} - \mathbf{K}[{\mathbf{M_{turb}}}^{-1}\mathbf{MA}^{Helm}_{(\alpha=0,\sigma)}]^{-1}\mathbf{K}^T, \tag{5.2.17}$$

following heuristic arguments, let us simplify the problem to analyse its numerical properties by setting: $\mathbf{M_{turb}} \approx \frac{1}{\nu+\nu_t}\mathbf{M}$, which is only true if ν_t is constant in space. Under this assumption, one can rewrite the equivalent $Schur(\mathbf{S}^{\mathbf{algTurb}}_{\mathbf{split}\ \mathbf{h}})$ complement as:

$$Schur(\mathbf{S}^{\mathbf{algTurb}}_{\mathbf{split}\ \mathbf{h}}) = \frac{\triangle t^*(\nu+\nu_t)}{\gamma_0}\mathbf{A}^{Pois}_{(\alpha=0,\sigma)} - \mathbf{K}[\mathbf{A}^{Helm}_{(\alpha=0,\sigma)}]^{-1}\mathbf{K}^T. \tag{5.2.18}$$

Following the same reasoning detailed in chapter 2 section 2.5.3.2, one obtains a limit for stability by bounding the Schur complement through its L^2 norm. The new limit for

stability becomes:

$$\lim_{\nu\triangle t^* \to 0} \frac{(\nu + \nu_t)\triangle t^*}{\gamma_0}\mathcal{O}(k^{3.2}) \longrightarrow 0,$$
$$\lim_{\nu\triangle t^* \to 0} \frac{\nu_t \triangle t^*}{\gamma_0}\mathcal{O}(k^{3.2}) \longrightarrow 0. \tag{5.2.19}$$

When comparing this result to the limit obtained in chapter 2, it appears that the turbulent viscosity ν_t may provide stabilisation for problems where small time steps or low laminar viscosities are used.

It is interesting to note that turbulent stabilisation arises from the semi-implicit treatment of the turbulent terms and that this would not have been the case if a fully explicit treatment of the turbulent terms had been considered. In addition, one may note that this turbulent stabilisation is provided through the modification of the Helmholtz equation. It is interesting to note that the Spectral Vanishing Viscosity (SVV) technique, which is detailed in section 5.2.2, also modifies the Helmholtz equation (when combined with the present splitting scheme as in [107]) in a similar manner to the proposed Smagorinsky-DG scheme. It is thus likely that SVV methods provide enhanced stability properties.

5.2.2 ILES-DG SVV-Fourier solver to compute 3D turbulent flows

This section presents the necessary modifications to extend the DG-Fourier solver detailed in the previous chapter to compute three dimensional turbulent flows. The proposed scheme combines two approaches: an Implicit Large Eddy Simulation (ILES) approach on the DG plane combined with a Spectral Vanishing Viscosity (SVV) method in the Fourier direction. To the author's knowledge this combination has not been attempted previously.

As it was mentioned in the introduction of this chapter, the ILES approach relies on the numerical dissipation introduced by the spatial discretisation to stabilise the DNS-NS

equations. Since high order techniques, like the DG method considered in this thesis, present low numerical errors (e.g. numerical dissipation), any attempt to solve the DNS-NS equations presented in previous chapters, at moderate Reynolds numbers on coarse meshes, would result in unstable calculations. To stabilise the equations, numerical dissipation needs to be introduced through, for example, upwinding of the non-linear terms. The proposed approach considers the Lesaint-Raviart non-linear form introduced in chapter 2 section 2.2.2.2 (Eq. 2.2.13). The additional numerical viscosity required to stabilise the DNS-NS equations should be provided by the fluxes (i.e. surface integrals in Eq. 2.2.13). It is interesting to note some properties of these fluxes:

- As the polynomial order increases, the fluxes vanish for a smooth solution.
- Inversely, the coarser the spatial resolution, the more important the effect of the fluxes becomes.
- The fluxes become active if discontinuities arise in the numerical solutions.
- Since hierarchical modal basis functions have been selected in this thesis, higher modes should require more upwinding than lower modes if a coarse mesh is selected. Therefore the fluxes should be more active at higher modes for poorly spatially resolved regions, which implicitly states that numerical dissipation may be applied mainly to the highest modes. Let us note that this characteristic is the main driving force of the SVV approach (see following section).

It may be argued that these properties are suitable for LES type modelling. Firstly, fluxes with these characteristics provide numerical dissipation for coarse discretisations and particularly in regions where discontinuities arise. Secondly, the influence of the fluxes should not affect well resolved regions of the simulation. In particular, near no-slip boundaries, where the mesh is finer, the implementation should provide the correct laminar viscosity solution (note that is is not true for the previously introduced Smagorinsky model). For well enough resolved simulations, this characteristic allows for the computation of flows where laminar and turbulent regions coexist and may be able capture laminar-turbulent

transition.

To provide further stabilisation, the SVV technique is incorporated in the Fourier direction. The following section details the implementation of this model.

5.2.2.1 Spectral Vanishing Viscosity technique for Fourier discretisations

The main idea behind the SVV technique is to add controlled artificial viscosity only to the highest modes in order to provide stabilisation in high order numerical methods. This technique does not modify the lowest modes and hence preserves the spectral accuracy of the method (see [135] or [107] for a detailed explanation).

The SVV technique was originally conceived by Tadmor [162] to regularise the solution (i.e. avoid oscillatory phenomena) in the inviscid Burgers equation:

$$\frac{\partial}{\partial t}u(x,t) + \frac{\partial}{\partial x}\left[\frac{u^2(x,t)}{2}\right] = 0, \tag{5.2.20}$$

subject to appropriate initial and boundary conditions.

Tadmor introduced in Eq. 5.2.20 a convolution term that only acts on the high wavenumbers, leaving low wavenumbers unchanged. The discrete form of the previous Burgers equation with the regularisation term reads:

$$\frac{\partial}{\partial t}u_N(x,t) + \frac{\partial}{\partial x}\left[\mathcal{P}_N\left(\frac{u^2(x,t)}{2}\right)\right] = \epsilon\frac{\partial}{\partial x}\left[Q_N * \frac{\partial u_N(x,t)}{\partial x}\right], \tag{5.2.21}$$

where $u_N(x,t)$ is the Nth order modal approximation of $u(x,t)$, $\mathcal{P}_N$ is a projection operator and $*$ represents the convolution. In addition, Q_N is a damping function (also called *viscosity kernel*) that becomes active for high wavenumbers only.

In the particular case of a Fourier discretisation (i.e. the z-direction in the DG-Fourier implementation), the SVV regularisation term becomes:

$$\epsilon\frac{\partial}{\partial x}\left[Q_N * \frac{\partial u_N(x,t)}{\partial x}\right] = -\epsilon\sum_{P_{cut}\leq|k|\leq N} k^2\widehat{Q}_k\widehat{\mathbf{u}}_k e^{ikx}, \tag{5.2.22}$$

where k is the wave number, N is the number of Fourier modes and P_{cut} is the wavenumber for the damping function to become active.

Different definitions for $\widehat{Q}_k$ have been previously proposed (e.g. Tadmor [162]) but in this work, the expression by Maday [117] is retained:

$$\widehat{Q}_k = exp\left[-\frac{(k-N)^2}{(k-P_{cut})^2}\right], \quad k > P_{cut}. \tag{5.2.23}$$

Finally, it is necessary to define the cut-off wavenumber P_{cut} and the spectral viscosity ϵ. As in [117] or [107], the present implementation uses: $P_{cut} = C_{SVV}\sqrt{N}$ and $\epsilon = \frac{\nu_{SVV}}{N}$, where C_{SVV} is a constant that controls the wavenumber cut off level for the damping and ν_{SVV} controls the amount of SVV viscosity. Fig. 5.2 shows the effect of the parameter C_{SVV} on the SVV damping function (i.e. *viscosity kernel*). Let us note that Pasquetti [135] compared various definitions for the cutting wavenumber level P_{cut} (e.g. $P_{cut} = \sqrt{N}$, $P_{cut} = N/2$) and the spectral viscosity ϵ (e.g. $\epsilon = \frac{1}{N}$, $\epsilon = \frac{4}{N}$) and showed that these lead to different flow solutions. Namely, the solution captured finer flow structures with a smaller ϵ and a larger cut-off wavenumber P_{cut}. In addition, it was shown that if these values are too restrictive (i.e. ϵ too small and/or P_{cut} too large), the method became unstable.

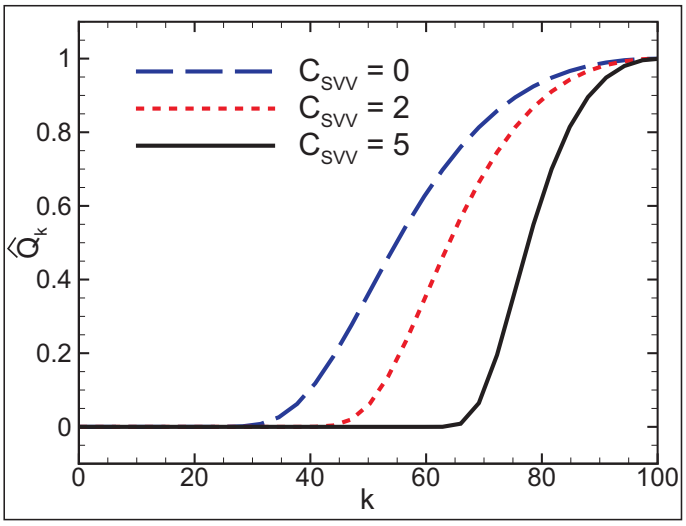

Figure 5.2: SVV damping function ($\widehat{Q}_k$) against wavenumber (k) for two values of the SVV constant (C_{SVV}) controlling the cut-off wavenumber.

To incorporate the SVV term in the DG-Fourier solver, it suffices to consider the Helmholtz equation corresponding to the Fourier direction (i.e. $z-$direction associated to the $z-$velocity component w). The original Helmhotlz equation, that was derived in chapter 4 section 4.2.3, is regularised through the SVV convolution term, which is added to the right hand side:

$$\nu\left(-\nabla^2{}_{2D}+\beta^2m^2+\frac{\gamma_0}{\nu\Delta t}\right)w_m^{n+1} \;=\; \left(\frac{\gamma_0}{\Delta t}\tilde{w}_m-\widehat{\nabla}_m p_m^{n+1}\right)-\epsilon\beta^2m^2\widehat{Q}_m\widehat{w}_m^{n+1}, \tag{5.2.24}$$

where the wavenumber has been defined as $k=m\beta$ (following the notation introduced in chapter 4). Rearranging Eq. 5.2.24, one obtains a SVV-Helmholtz equation for the Fourier direction:

$$\left(-\nabla^2{}_{2D}+\beta^2m^2\left[1+\frac{\epsilon\widehat{Q}_m}{\nu}\right]+\frac{\gamma_0}{\nu\Delta t}\right)w_m^{n+1} \;=\; \frac{1}{\nu}\left(\frac{\gamma_0}{\Delta t}\tilde{w}_m-\widehat{\nabla}_m p_m^{n+1}\right). \tag{5.2.25}$$

In most SVV implementations, including the author's, the added SVV-viscosity depends only on the wavenumber (i.e. $\epsilon\approx\frac{1}{N}$, where N is the number of Fourier modes). Ideally, this parameter should relate to the numerical solution and adapt depending on the local spatial damping required. This idea was introduced in [97] and explored in the context of compressible NS formulations in [104]. However, in the present work and due to time constraints, this variable SVV-viscosity extension has not been explored.

Let us finally note that the computational cost of including the additional SVV terms is almost negligible. This method only requires modification of the diagonal terms in the matrix relating to the Helmholtz step.

5.3 Validations and comparisons

This section presents examples of the usability of the described methods: the Smagorinsky-DG and the ILES-DG SVV-Fourier. In addition, it provides quantification of the accuracy of the two techniques. The objective of this section is twofold:

- Demonstrate that these methods are stable on relatively coarse meshes at moderate Reynolds number flows (i.e. Re = 3900 and 10000).
- Compare the results to experimental data and other numerical techniques (i.e. DNS, unsteady RANS) to provide verification and validation for the described methods.

Note that this section is not devoted to providing an analysis of turbulence physics. Instead, it attempts to show that the two turbulent techniques are stable and provide reasonable results with the coarse resolutions selected. In addition, this chapter sets the bases for further developments.

5.3.1 Circular cylinder

This section presents preliminary results for a circular cylinder at Re=3900. Turbulent simulations for circular cylinders at this particular Reynolds number can be found in [116], where an extensive analysis was performed using a conformal h/p Spectral solver (i.e. Nektar). Reported results are for DNS and LES simulations. In addition, let us note that a very complete review of experimental and numerical data for circular cylinder flows has been compiled by Norberg in [129].

The results included in this section have been obtained using the ILES-DG SVV-Fourier technique on a mesh consisting of 1066 mixed tri-quad elements (with 24 curved quad elements to capture the circular geometry) and a polynomial order $k = 5$ in the $x - y$ DG plane. In the Fourier direction (i.e. $z-$direction) 16 planes are used, which extend a spanwise length of $L_z/D = 1$, where D is the cylinder diameter. The time step is set to $\triangle t = 0.0005$ and the SVV constants to $C_{SVV} = 1$ and $\nu_{SVV} = 1$. Fig. 5.3 shows

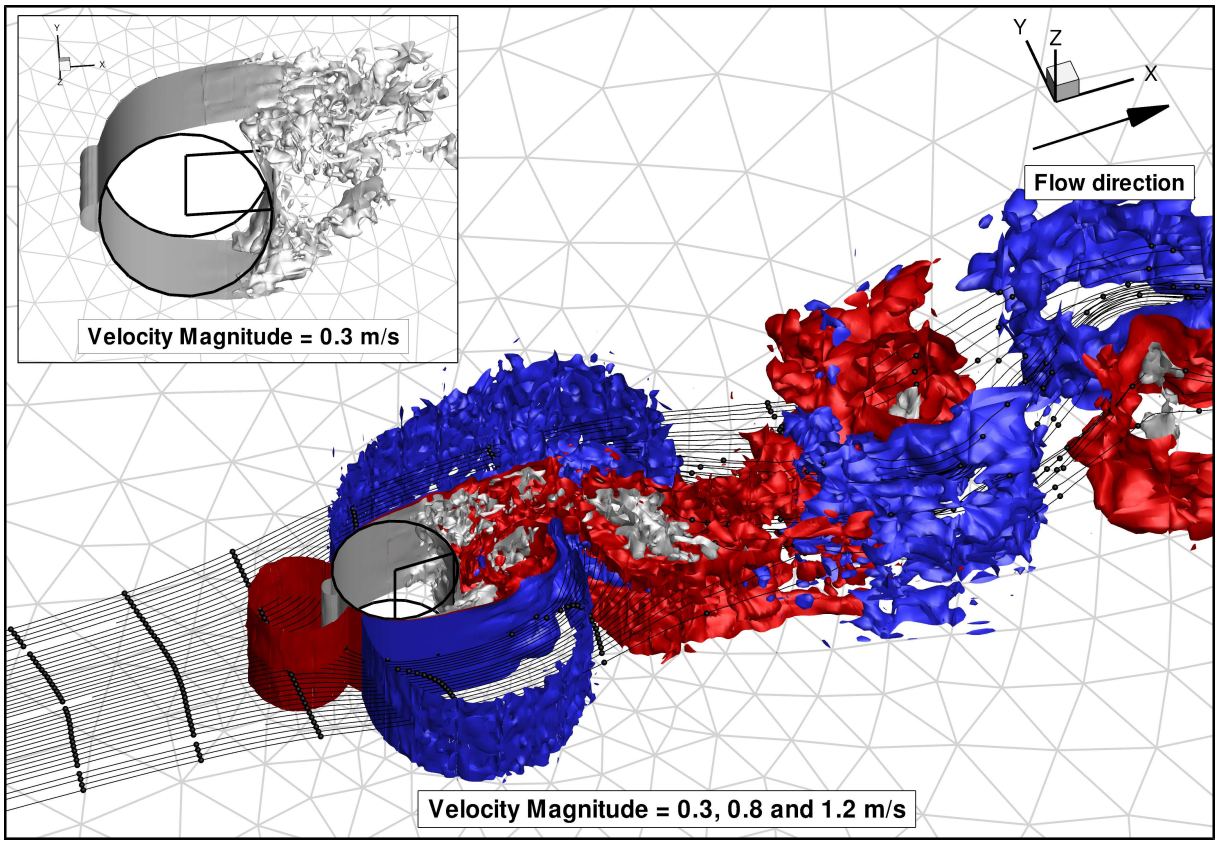

Figure 5.3: ILES-DG SVV-Fourier solution for a circular cylinder at Re =3900 showing iso-surfaces of velocity magnitude (grey $|\mathbf{u}| = 0.3$ m/s, red $|\mathbf{u}| = 0.8$ m/s and blue $|\mathbf{u}| = 1.2$ m/s). Overlaid in the main figure are streamlines showing the flow trajectory. Inset figure shows details of the flow structures near the circular cylinder (velocity magnitude iso-surface $|\mathbf{u}| = 0.3$ m/s).

a snapshot of velocity magnitude iso-surfaces after four vortex cycles. It can be seen that a turbulent flow develops in the wake of the cylinder together with vortex shedding. In addition, a transitional shear layer (see monograph by Zdravkovic [187] for definitions) can be observed in the inset figure.

To quantify the obtained results, the time averaged pressure coefficient distribution on the cylinder surface is depicted in Fig. 5.4. Included in the figure are experimental data [128] and simulation results from a conformal h/p Spectral code using DNS and LES techniques [116]. Let us note that previous numerical [116] and experimental [129] studies have shown an important dependency of the pressure distribution with the cylinder spanwise length. The presented simulation has a shorter spanwise length ($L_z/D = 1$) than

the simulated published data [116], where the spanwise lengths vary as: $\pi \leq L_z/D \leq 2\pi$. The aim of this section reduces to show the potential of the ILES-DG SVV-Fourier model to capture the general flow features and the presented results should only be considered as preliminary. Fig. 5.4 shows reasonable agreement between the ILES-DG SVV-Fourier

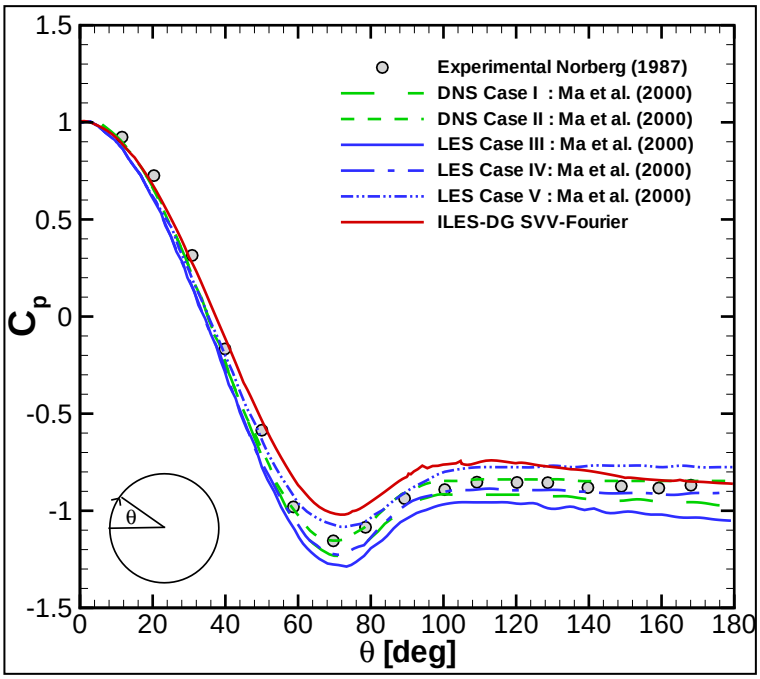

Figure 5.4: Time averaged pressure coefficient (C_p) distribution on the circular cylinder at Re=3900. ILES-DG SVV-Fourier results, experimental data [128] and published numerical results [116] using a conformal h/p Spectral code (i.e. DNS and LES computations).

computations and published results. The discrepancies may be explained by the coarse spatial resolution, the shorter spanwise length and the short integration time (i.e. one vortex cycle) used to compute the time averaged values in the present simulation. Further computations may be required to assess in depth the accuracy of the ILES-DG SVV-Fourier technique, however, the present computation demonstrates the potential of this novel technique.

5.3.2 NACA foils

This section considers symmetric NACA 4-digit foils and compares results from the two developed turbulence techniques to experimental data and other numerical techniques.

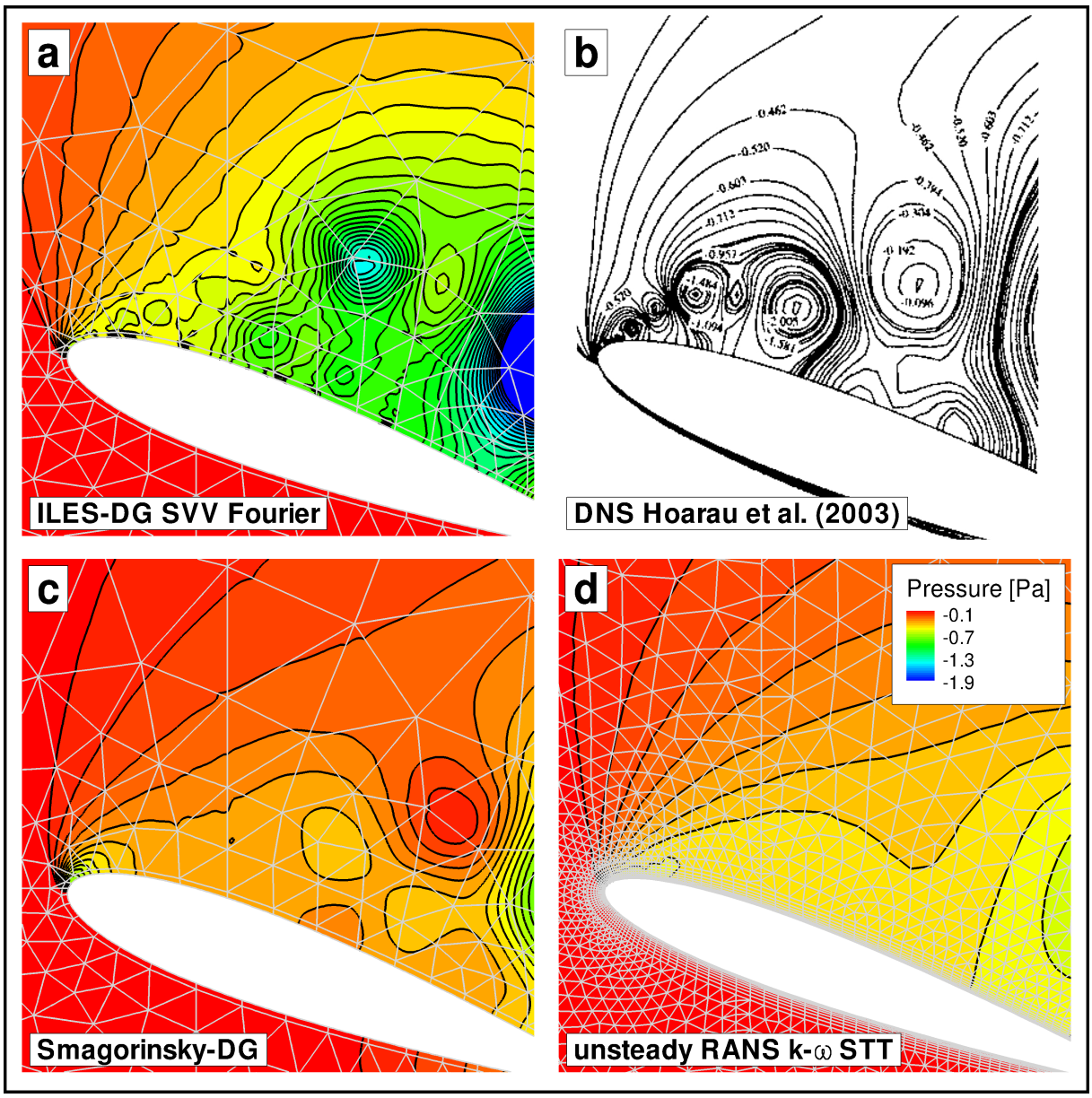

Figure 5.5: Flow field snapshot of the pressure field at Re =10000 ; (a) NACA0015 3D ILES-DG SVV-Fourier $k = 5$ and 4 Fourier planes, (b) NACA0012 DNS finite volume [79], (c) NACA0015 2D Smagorinsky-DG $k = 5$ and (d) NACA0012 2D low order finite volume Fluent [56] ($k - \omega$ SST).

Firstly, Fig. 5.5 provides a qualitative comparison of the pressure fields for NACA

foils at Re = 10000 and a fixed Angle Of Attack (AOA) $AOA = 20\,^{\circ}$. The figure includes results from four different numerical techniques:

- **3D ILES-DG SVV-Fourier**: Fig. 5.5.a shows the pressure field for a NACA0015 using the 3D ILES-DG SVV-Fourier solver with polynomials of order $k = 5$ in the $x - y$ DG plane and 4 Fourier planes in the $z-$direction. The snapshot has been taken before flow three dimensionality develops.
- **Published DNS**: Fig. 5.5.b shows a snapshot of the pressure field using a finite volume code and the Direct Numerical Simulation approach [79]. This published result is for a NACA0012.
- **2D Smagorinsky-DG**: Fig. 5.5.c shows the pressure field for a NACA0015 using the developed 2D Smagorinsky-DG model. The DG discretisation uses polynomials of order $k = 5$.
- **Unsteady RANS**: Fig. 5.5.d shows the pressure field for a NACA0012 obtained with the low order finite volume code Fluent [56] using the $k - \omega$ SST turbulence model. Second order time and space discretisation have been used for this unsteady calculation.

Let us note that the figure presents two different foils: a NACA0015 for the ILES-DG SVV-Fourier and the Smagorinsky-DG simulations and the NACA0012 for the published data and unsteady RANS results. However, it may be hypothesised that at this high AOA with mainly separated flow on the upper surface, the foil thickness should have minimal impact on the flow field, thus allowing for qualitative comparisons of these foils.

As described in [79] and shown in Fig. 5.5.b, at this AOA the flow is characterised by Kelvin-Helmholtz shear layer vortices that lead eventually to a turbulent regime. It can be seen that the flow field solutions show similar flow patterns and all show separated flow on the upper surface. Fig. 5.5.a shows that the ILES-DG SVV-Fourier technique captures, to some extent, the Kelvin-Helmholtz shear layer vortices and most of the vortical structures developing within the separated region. The 2D Smagorinsky-DG simulation (Fig.

5.5.c) captured the general flow pattern but fails to capture the Kelvin-Helmholtz shear layer vortices. Note that the 2D Smagorinsky-DG and the 3D ILES-DG SVV-Fourier simulations used the same spatial resolution in the $x-y$ plane (i.e. identical meshes and polynomial order $k = 5$). Finally, Fig. 5.5.d shows that the unsteady RANS simulation using the low order finite volume code Fluent cannot capture either the Kelvin-Helmholtz shear layer vortices or the vortical structures developing within the separated flow region. The discrepancies seen for the Smagorinsky-DG method and the unsteady RANS simulations may be explained by the overly dissipative character of the turbulent viscosity approach present in both models.

The second part of this section considers only the NACA0012 foil at Re=10000 for a range of angles of attack $0\,^\circ \leq AOA \leq 20\,^\circ$. Solution snapshots using the 3D ILES-DG

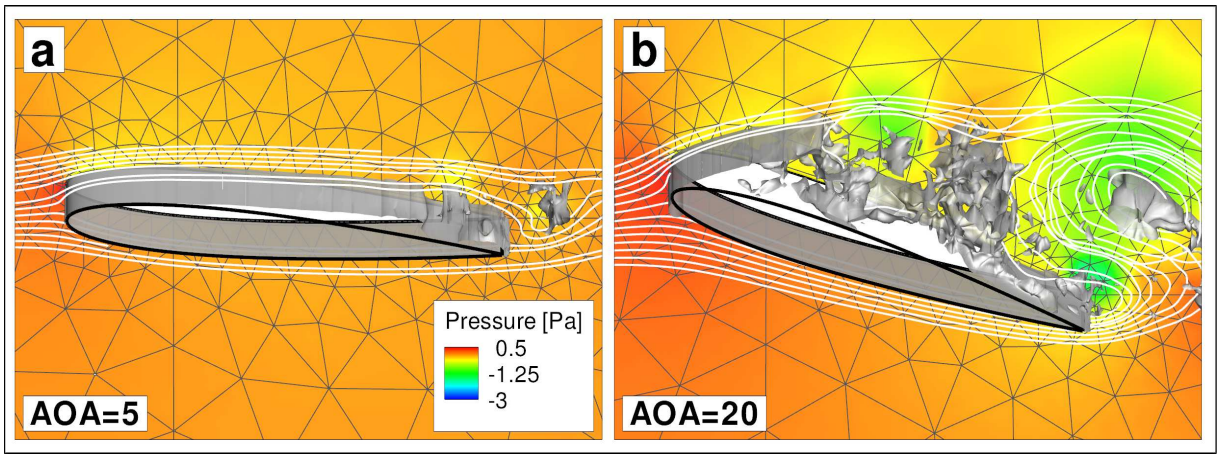

Figure 5.6: 3D ILES-DG SVV-Fourier flow field snapshot for the NACA0012 at Re =10000 with 4 Fourier planes and $L_z/c = 0.2$ for: (a) $AOA = 5\,^\circ$, $k = 3$, (b) $AOA = 20\,^\circ$, $k = 5$. Pressure field is shown at the plane $z = 0$. Overlaid are streamlines showing flow trajectory and an iso-surface of velocity magnitude with iso-values: (a) $|\mathbf{u}| = 0.15$ m/s and (b) $|\mathbf{u}| = 0.3$ m/s.

SVV-Fourier technique for this foil at $AOA = 5\,^\circ$ and $20\,^\circ$ are depicted in Fig. 5.6. The DG discretisation consist of polynomial orders $k = 3$ for $AOA = 5\,^\circ$ and $k = 5$ for the $20\,^\circ$ case. The figure shows that the ILES-DG SVV-Fourier method provides details of 3D flow features. It can be seen that three dimensionality is more important for the highest angle of attack. In addition, let us note that the expected flow features are captured by

the solver. Namely, the solver predicts separated flow on the foil's suction side and a larger separated region for the higher AOA case.

This section concludes with comparisons of computed aerodynamic coefficients, using the 2D Smagorinsky-DG and the 3D ILES-DG SVV-Fourier methods, to experimental data [189] and [160] for a range of angles of attacks: $AOA = 0\,^\circ$ to $20\,^\circ$. In addition, 2D results from the low order finite volume code Fluent [56] with the $k-\omega$ SST turbulence model are reported, referred to as: unsteady RANS. Computed results include lift and drag coefficients (mean and standard deviations) for the NACA0012, which is compared to two sets of published experimental data for the same foil [189] and [160]. Table 5.1 summarises the flow conditions and available data from computations and experiments. Experimental and simulated aerodynamic coefficients are depicted in Fig. 5.7. Mean values for the lift and drag coefficients are reported in Fig. 5.7.a and Fig. 5.7.b respectively. The associated standard deviations (SD) are shown in Fig. 5.7.c and Fig. 5.7.d. Fig. 5.7 shows a general good agreement between the numerical results and experimental data presented for all numerical techniques. A detailed discussion follows:

	NACA foil	**Re/10^4**	**AOA** [$^\circ$]	**Coefficients**
2D Smagorinsky-DG	0012	1.00	0, 5, 10, 20	Cl_{Mean}, Cd_{Mean} Cl_{SD}, Cd_{SD}
3D ILES-DG SVV-Fourier	0012	1.00	0, 5, 10, 20	Cl_{Mean}, Cd_{Mean} Cl_{SD}, Cd_{SD}
Unsteady RANS $k-\omega$ SST	0012	1.00	0, 5, 10, 20	Cl_{Mean}, Cd_{Mean} Cl_{SD}, Cd_{SD}
Exp. Zhou et al. (2011) [189]	0012	0.53, 1.05, 5.10	0, 10, 20	Cl_{Mean}, Cd_{Mean} Cl_{SD}, Cd_{SD}
Exp. Sunada et al. (1997) [160]	0012	0.40	0 to 21	Cl_{Mean}

Table 5.1: Summary of simulated and experimental aerodynamic coefficients for the NACA0012 foil at moderate Reynolds numbers.

- **Experimental data**: The selected experimental data for the NACA0012 (i.e. [189] and [160]) show generally low scatter, particularly for low AOAs for the mean components. However let us note that for $AOA = 10\,^\circ$, the mean lift seems to be very

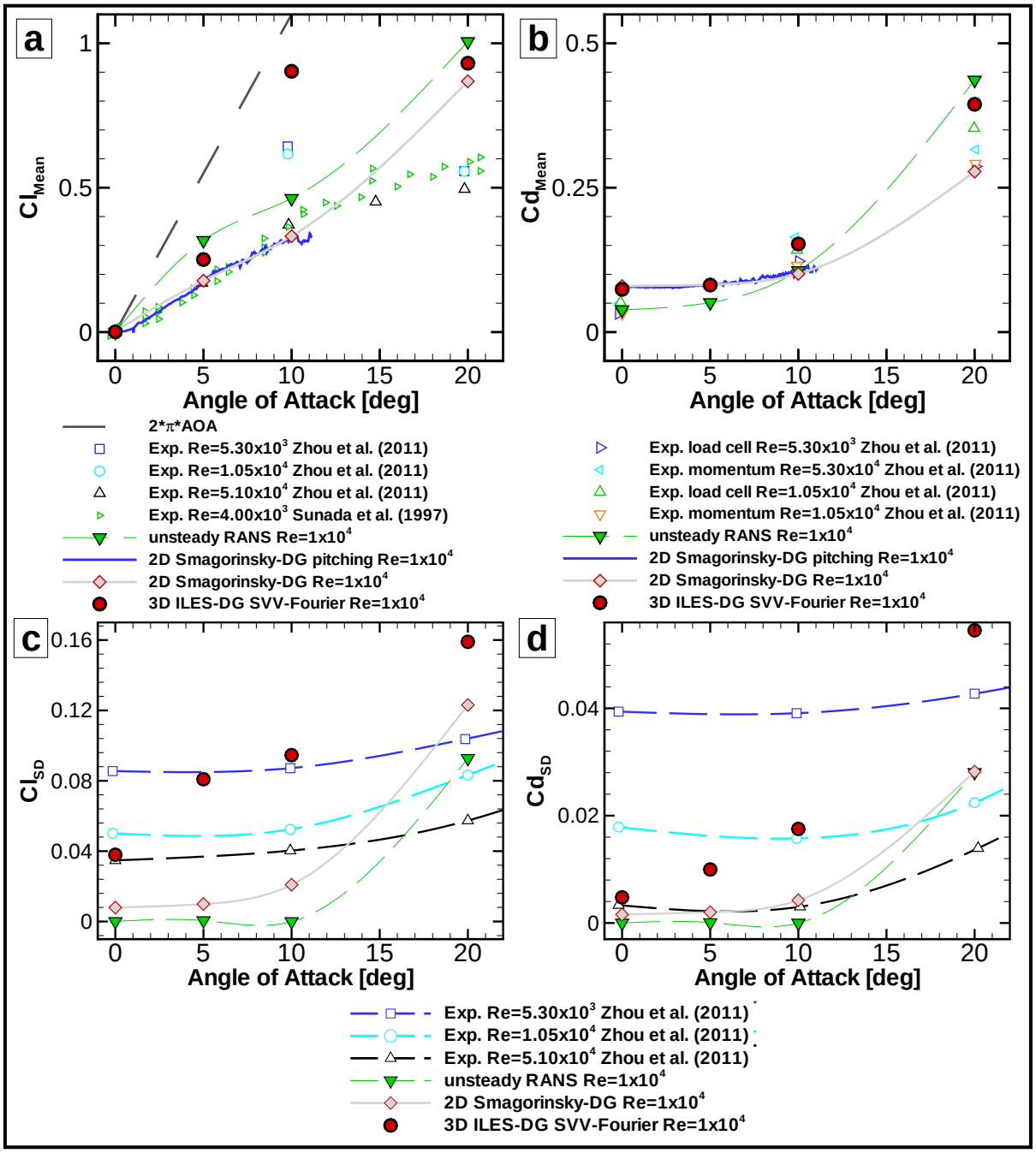

Figure 5.7: NACA0012 validation: 2D Smagorinsky-DG ($k = 3$ static and pitching), 3D ILES-DG SVV-Fourier ($k = 3$ in the DG plane for $AOA = 0\,^\circ$ and $5\,^\circ$ and $k = 5$ for $AOA = 10\,^\circ$ and $20\,^\circ$, all with 4 Fourier planes and $L_z/c = 0.2$), 2D second order Finite Volume Fluent [56] (unsteady RANS: $k-\omega$ SST) and Experimental data [189] [160]; (a) Mean lift coefficient, Cl_{Mean} (b) Mean drag coefficient, Cd_{Mean} and (c), (d) Standard deviation for the lift and drag coefficients respectively: Cl_{SD} and Cd_{SD}.

sensitive to the Reynolds number (Exp. Zhou et all. [189]).

It is interesting to note that at this moderate Reynolds number the slope of the mean lift curve is far from the theoretical slope of $\frac{d\,Cl}{d\,AOA} = 2\pi$ derived from inviscid thin airfoil theory [4]. In addition, the standard deviations (Fig. 5.7.c and Fig. 5.7.d) increase with the AOA and show a not negligible inverse dependency of the fluctuating components with the Reynolds number. The reader is referred to the original paper [189] for possible explanations and discussion of these phenomena.

- **Unsteady RANS**: unsteady RANS results have been obtained using second order discretisations in space and time and the $k-\omega$ SST turbulence model (unsteady RANS) with a mesh constituted of 8000 mixed tri-quad elements (similar to the mesh depicted in Fig. 5.5.d).

  Fig. 5.7.a and Fig. 5.7.b (green gradient symbols) show that the unsteady RANS simulations provide reasonable predictions for low AOAs ($AOA = 0\,^\circ$, $5\,^\circ$ and $10\,^\circ$) for the mean lift and drag forces but over-predict these forces at the highest angle $AOA = 20\,^\circ$. In addition, Fig. 5.7.c and Fig. 5.7.d show negligible fluctuations for $AOA = 0\,^\circ, 5\,^\circ$ and $10\,^\circ$, compared to the experimental data. These under-predictions may be related to the over-dissipative character of these unsteady RANS solutions, which would tend to suppress unsteady flow features. It is interesting to note that although the mean lift and drag components are over-predicted for the highest AOA, very good agreement is obtained for the fluctuating components.

- **2D Smagorinsky-DG**: Two dimensional static results (grey diamonds) are compared to a pitching simulation using the sliding mesh capability (blue line) for the NACA0012, both simulated using a polynomial order $k = 3$. The static results correspond to averaged values over time, whilst instantaneous values are depicted for the pitching case (rotating around the quarter chord with a rotational velocity $\omega = 0.01$ rad/s). Mean values for the static and the pitching simulations are in very good agreement with each other providing further verification of the sliding mesh

technique.

Predicted values for the mean lift, shown in 5.7.a, are in very good agreement with experimental data for static and pitching cases. However, the simulated mean lift appears higher for $AOA = 20\,^\circ$ than experimental results and similar to the unsteady RANS predictions. Fig. 5.7.b shows that the Smagorinsky-DG method provides results that are in very good agreement for the mean drag for higher AOAs ($AOA = 10\,^\circ$ and $20\,^\circ$) but over-predicts the mean drag for at $AOA = 0\,^\circ$. This discrepancy can be explained by the lack of turbulent viscosity damping near the wall (e.g. Van Driest damping [96]) and also because these simulations consider only turbulent flow (no laminar-turbulent transition has been considered), providing an increased drag coefficient through increased wall shear stress and a thicker boundary layer. In addition, Fig. 5.7.c and 5.7.d show small values for the fluctuating lift and drag forces when compared to the experimental data but larger than predicted by the unsteady RANS simulations (at $AOA = 0\,^\circ$, $5\,^\circ$ and $10\,^\circ$). These damped fluctuations may be explained by the low spatial resolution of the simulation, the overly dissipative character of the Smagorinsky model (using this spatial resolution) and the lack of turbulent viscous damping near walls (e.g. Van Driest damping) in the present implementation. Finally, let us note that the influence of the Smagorinsky constant c_s (defined in section 5.2.1.1) on the aerodynamics coefficient has not been explored and is likely to have influence on the presented results.

- **3D ILES-DG SVV-Fourier**: All the simulations computed with the ILES-DG SVV-Fourier technique consider a periodic length in the Fourier direction $L_z/c = 0.2$ and use 4 Fourier planes with the SVV parameters set to $C_{SVV} = 2$ and $\nu_{SVV} = 1$. The DG discretisation consists of polynomial orders $k = 3$ for $AOA = 0\,^\circ$, $5\,^\circ$ whilst $k = 5$ was used for $AOA = 10\,^\circ$ and $20\,^\circ$. The plotted results have been obtained by spatially averaging spanwise sectional forces, which are subsequently time averaged. These time averaged values have been computed after the development of three di-

mensional flow.

Simulated mean values show generally good agreement with experimental data for all AOAs. Although as shown in Fig. 5.7.a, this model over-predicts the mean lift at $AOA = 10\,^\circ$ and $20\,^\circ$. Let us note that for $AOA = 10\,^\circ$, the experimental data presents large scatter, which may suggest a limiting flow case that is very sensitive to testing conditions (e.g. Reynolds number, trailing edge sharpness) and may explain the discrepancies between, both simulated and experimental results. In addition, it can be seen that the mean lift at $AOA = 20\,^\circ$ is similar to the results provided by the other numerical techniques but larger than the experimental data. Preliminary calculations (not shown) suggest that this over-prediction relates to the coarse spatial resolution used in the DG plane. The effect of spanwise resolution has not been assessed but may influence these results and requires further investigation.

Fig. 5.7.b shows that the simulated mean drag coefficients are in very good agreement with experimental data for all AOAs, but is over-predicted (and similar to the Smagorinsky-DG) for the lowest angle $AOA = 0\,^\circ$. A spatial refinement study and a parametric study on the SVV constants may provide insight into the causes of this mismatch.

Fluctuating values for the lift coefficient, Fig. 5.7.c, show good agreement with experimental data and appear to be larger than predicted by the other numerical techniques. Let us note that this is consistent with the observation drawn from the snapshots of the pressure fields at $AOA = 20\,^\circ$ (Fig. 5.5). It can also be observed that the rate of increase of fluctuating quantities with the angle of attack is larger than seen for the experimental data. Preliminary comparisons for two spatial resolutions in the DG plane (not shown), suggest that the over-prediction of fluctuating forces may be related to a low spatial resolution. The dependency of spatial resolution and the influence of the SVV constant in the aerodynamic quantities require further exploration. However, it may be concluded that the 3D ILES-DG

SVV-Fourier technique provides generally accurate aerodynamic functionals for the present cases.

Overall, it has been shown that both methods, the 2D Smagorinsky-DG and the 3D ILES-DG SVV-Fourier, provide reasonable predictions of aerodynamic functionals for the cases tested despite the coarse resolutions used. It is concluded that these results provide verification and validation of the developed techniques showing the correct implementation and usability of the two novel approaches presented in this chapter to stabilise the high order code and compute turbulent flows at moderate Reynolds numbers.

Chapter 6

Simulations of cross-flow turbines

The previous chapters detailed the numerical formulation, implementation, verification and validation of an appropriate solver to simulate cross-flow turbine flows. In the introductory chapter 1, several aspects of cross-flow turbine flow physics were introduced and some of them are explored in this chapter using the developed solver. The chapter details 2D and 3D simulations for one and three bladed turbines under laminar flow regimes. Cross-flow turbine computations under turbulent flow conditions with the capabilities detailed in chapter 5 have not been explored due to time constraints. Although the Reynolds number considered in this chapter are low, when compared to operational cross-flow turbines, the results and analysis presented herein provide useful insight to turbine flow phenomena at higher Reynolds numbers. In addition, these cases explore the capabilities of the developed solver and show possible further uses for the numerical techniques formulated in this thesis.

6.1 Chapter outline

Section 6.2 introduces the necessary definitions used to characterise cross-flow turbines. Two dimensional simulations, at low Reynolds numbers, where three dimensional effect may be disregarded, are presented in section 6.3. This section includes computations for

one and three bladed turbines. In addition, a ducted three bladed turbine is simulated. Particular emphasis is placed on the analysis of blade-vortex (also referred to as blade-wake) interactions. In addition, analytical estimates to bound these interactions are provided and compared to the simulated results. Finally, in section 6.4, three dimensional simulations are presented for one and three bladed turbines, showing three dimensional wake structures. In addition, analysis of Reynolds number effects and turbine taking off conditions are reported.

6.2 Cross-flow turbine definitions

Fig. 6.1 shows a sketch of a typical straight bladed cross-flow turbine, where the essential parameters to characterise its geometry and operational regimes are included. The figure introduces the azimuthal angle θ, the free stream velocity $\mathbf{U}$ with magnitude $|\mathbf{U}| = U$, the rotational velocity $\mathbf{U_{rot}}$ with magnitude $|\mathbf{U_{rot}}| = \omega R$, where R is the turbine radius and ω the rotational speed, the effective velocity $\mathbf{U_{eff}} = \mathbf{U} + \mathbf{U_{rot}}$ and the resulting geometric angle of attack (AOA). In addition, Fig. 6.1 defines the Lift (L) and Drag (D) force components and their associated polar projections: Tangential (T) and Normal (N) forces. One may characterise the turbine by the tip speed ratio $\lambda = \omega R/U$, and the solidity $\sigma = Nc/R$, with N the number of blades, and c represents the foil chord. Lastly, the blade geometric velocity and the geometric angle of attack may be defined as: $U_{eff} = U\sqrt{1 + 2\lambda\cos\theta + \lambda^2}$ and $AOA = tan^{-1}(\frac{\sin\theta}{\lambda+\cos\theta})$.

Remark: *The velocity U_{eff} and the AOA are defined as "geometric" since they do not account for induction factors (e.g. related to blockage or wake effects) and only depend on geometric considerations.*

The geometries of the turbines studied in this chapter are characterised by their radius $R = 2c$ and their solidities $\sigma = Nc/R = 1/2$ with $N = 1$ for the one bladed turbine, and $\sigma = 3/2$ with $N = 3$ for the three bladed turbine. The blade forces are non-dimensionalised using the free stream velocity magnitude: $C_i = \frac{2F_i}{\rho c U^2}$, where $i = N, T$ and

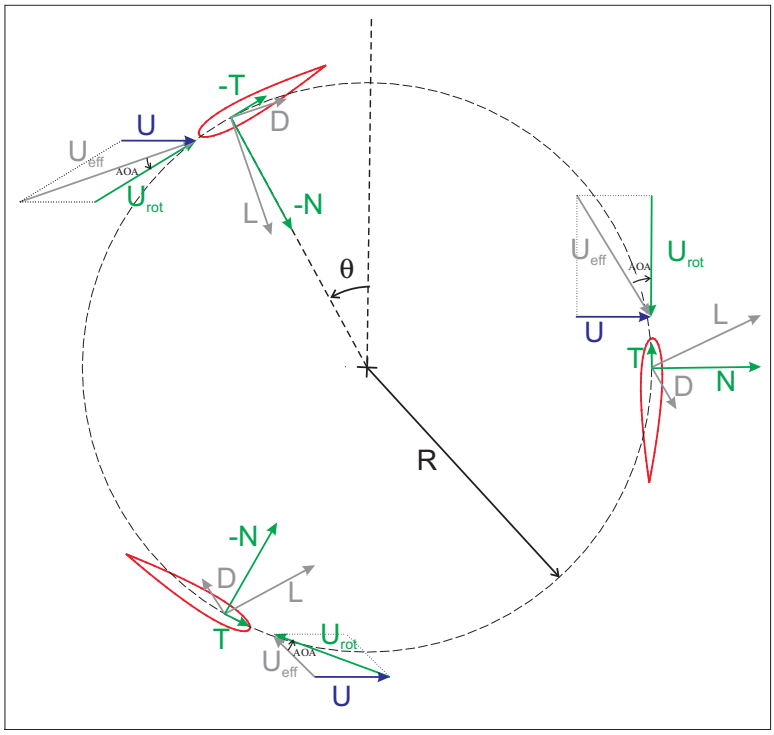

Figure 6.1: Schematic of the three bladed cross-flow turbine.

ρ represents the fluid density.

6.3 Two dimensional simulations

This section considers 2D cross-flow turbines since the Reynolds number selected is low enough to ignore three dimensional effects. The Reynolds number, based on the blade chord and the free stream velocity U (i.e. without accounting for the rotational velocity), is $Re \approx 100$ and it is bounded through the rotational cycle, when considering the effective U_{eff} and all the tip speed ratios, as: $0 < Re_{eff} = cU_{eff}/\nu < 200$.

Three distinct but complementary problems are studied:

1. **One bladed turbine**: analysis of the influence of the wake generated through the upstream on the downstream blade passage: blade-wake or blade-vortex interactions.
2. **One bladed versus three bladed turbine**: analysis of the aerodynamic differ-

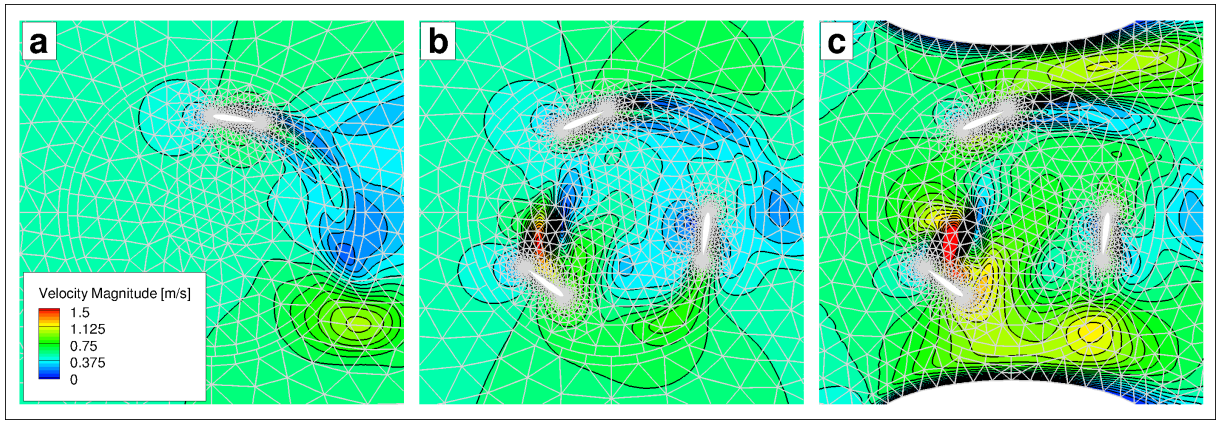

Figure 6.2: DG flow field snapshots of velocity magnitude for $\lambda = 2$; (a) one bladed turbine, (b) three bladed turbine and (c) ducted three bladed turbine (tidal fence).

ences between configurations and blade-wake interactions.

3. **Ducted case for a three bladed turbine**: analysis of the influence of blockage (elliptical cross-sectional support columns) on the aerodynamic performance of the three bladed turbine. The ducted case may be representative of the tidal fence concept, which is under consideration to enhance the energy extraction resulting from the tides.

Fig. 6.2 shows DG results (using a polynomial order $k = 3$) for the three configurations considered. It can be seen that the code preserves the wake structures even for the coarse mesh and the low polynomial order used. All meshes are constituted of triangular-quadrilateral elements: 1794 elements for the one bladed turbine mesh depicted partially in Fig. 6.2.a, 3320 for the unconstrained three bladed turbine in Fig. 6.2.b and 3614 for the ducted three bladed turbine in Fig. 6.2.c. The three cases show substantially different flow fields which will be analysed in the following sections.

Table 6.1 summarises the flow conditions and turbine characteristics considered in this section. Namely, three tip speed ratios (λ) are studied by varying the free stream speed (U), while maintaining constant the rotational speed (ω). This parameter space can be related to variations in free stream velocities for a set operational rotational speed.

Free stream velocity U [m/s]	**Rotational speed** ω [rad/s]	**Tip speed ratio** λ
0.2	0.5	5
0.5	0.5	2
1.0	0.5	1

Table 6.1: Flow conditions and turbine characteristics for 2D simulations: varying the free stream velocity.

6.3.1 One bladed turbine

Before proceeding to the physical investigations, the necessary spatial resolution is investigated. Namely, the required polynomial order k for the given mesh to obtain accurate results (unchanged under increased spatial resolution). To this end, Fig. 6.3 compares the tangential and normal force traces against azimuth for a one bladed turbine (with solidity $\sigma = 1/2$) for two polynomial orders $k = 2$ and 3, for two values of the free stream velocity U leading to two distinct tip speed ratios $\lambda = 1$ and 2. It can be seen that the results are

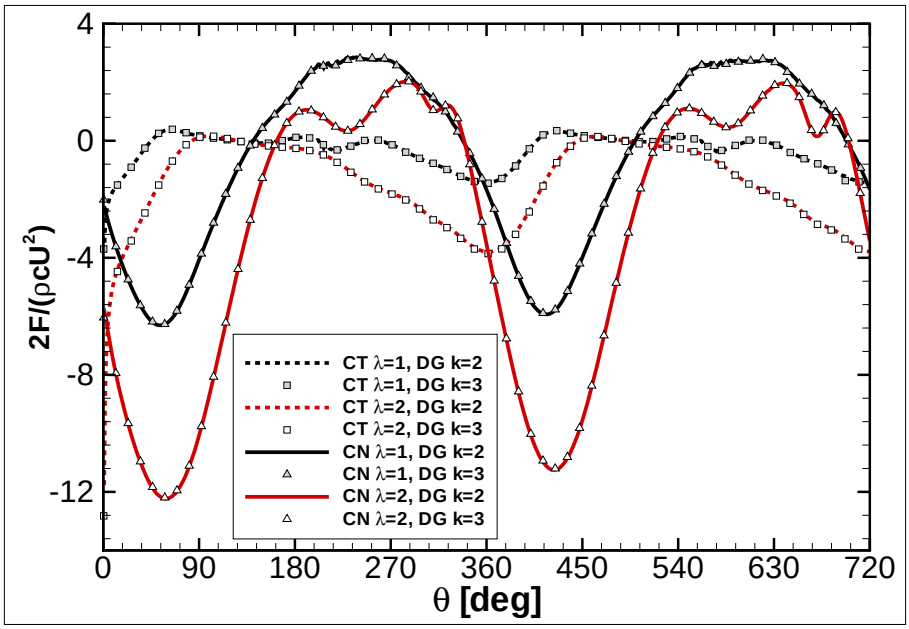

Figure 6.3: Tangential and normal force coefficients against azimuth, one bladed turbine for tip speed ratios: $\lambda = 1$ and 2 and polynomial orders: $k = 2$ and 3.

unchanged by the use of a higher resolution, showing that even for the low polynomial order $k = 2$, the DG solver provides accurate solutions for these cases, where dettached

flow over the blades and blade-wake interactions are present (as discussed below).

Let us now extend the results for $k = 2$ and the normal force (C_N) to 11 revolutions for $\lambda = 1$, 2 and 5 and focus attention on the influence of the upstream blade passage ($0^\circ < \theta < 180^\circ$) on the downstream blade passage ($180^\circ < \theta < 360^\circ$). Results are shown in Fig. 6.4. First, it can be observed that for all tip speed ratios the forces on the

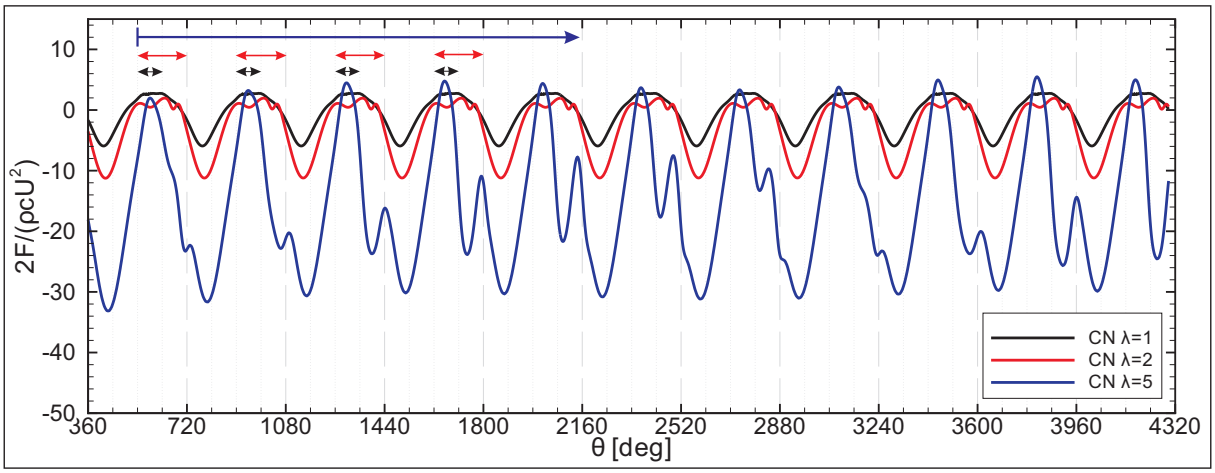

Figure 6.4: Normal force coefficient against azimuth, one bladed turbine for $\lambda = 1$, 2 and 5. 11 revolutions are shown.

downstream passage are not the same as that on the upstream passage, as might naively be expected for a one bladed turbine. Furthermore, the differences between the upstream and downstream passage increase with tip speed ratio λ.

To analyse these observations, one needs to consider and compare two phenomena: firstly, the vortex or wake generation at the upstream blade passage and its convection; secondly, the blade rotation speed. It is the relative times of wake convection to blade rotation that gives rise to different types of interactions. Fig. 6.5 shows the three types of interactions expected, when varying the blade rotational speed with respect to the wake convection speed, in terms of the tip speed ratio λ and various limiting cases, which are subsequently defined.

A blade traveling through the upstream passage generates a wake (Fig. 6.5.a) that is convected downstream with the speed $U_{axial} = \beta U$, where U_{axial} is the mean streamwise velocity component within the turbine and β is the axial induction factor related to the

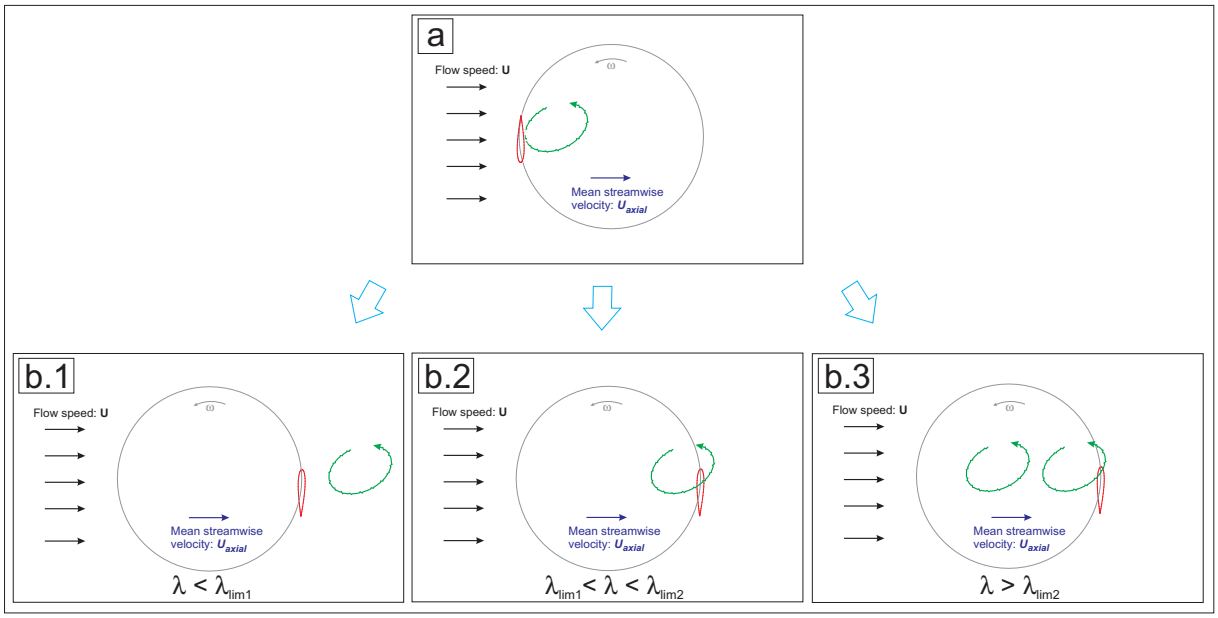

Figure 6.5: Schematic of the three limiting conditions for blade-wake or blade-vortex interactions: (a) initial wake or vortex shedding, (b.1) no blade-wake interactions, (b.2) one wake interacts with the blade at the rear passage and (b.2) two wakes interact with the blade at the rear passage.

front half of the turbine (i.e. $\beta \approx 1$ for the one bladed turbine). Considering the speed of convection and the turbine geometrical diameter, the time for the wake to reach the downstream section can be estimated as $t_{wake} \approx 2R/\beta U$. Regarding the blade rotation, the time for the blade to transition from $\theta = 90\,^\circ$ to $\theta = 270\,^\circ$ is $t_{foil} = R\pi/(R\omega)$.

When comparing these times it becomes clear that for $t_{wake} < t_{foil}$, the blade wake shed on the upstream passage is convected aft of the turbine before the blade completes its downstream pass and there is no blade-wake interaction (Fig. 6.5.b.1). However, if $t_{wake} > t_{foil}$ the blade on the downstream section sees a disturbed upstream flow that leads to a change in its aerodynamic performance (e.g. Fig. 6.5.b.2). One can note that the first limiting case (Fig. 6.5.b.2) occurs when the ratio $t_{wake}/t_{foil} = 2\lambda/\beta\pi \approx 1$, leading to interactions for values of $\lambda_{lim1} \gtrsim \pi/2 \approx 1.6$ (assuming $\beta \approx 1$).

However, for values of $\lambda < \lambda_{lim1} \approx \beta\pi/2$ one can expect interactions for $180\,^\circ < \theta < 270\,^\circ$ as seen in Fig. 6.4 for $\lambda = 1$ (black arrow) due to the proximity of the blade to the previously shed wake in the bottom half of the circular path.

This last case can be analytically analysed by considering a chord of the circle (and not the diameter) $chr = 2R\sin(\varphi/2)$ with associated arc length $arc = \varphi R$ for an arbitrary angle $\varphi < 180\,^\circ$. A new generalised limiting case for interactions for azimuthal angles within $180\,^\circ < \theta < 270\,^\circ$ can then be estimated using the relationship $t_{wake}/t_{foil} = \frac{2R\sin(\varphi/2)}{\varphi R}\frac{R\omega}{\beta U} = 2\lambda\sin(\varphi/2)/(\beta\varphi) \approx 1$. For small enough angles φ, a Taylor series approximation of $\sin(\varphi/2) \approx \varphi/2$ leads to $\lambda_{lim1-chord-based} = \beta$. Finally, it can be concluded that for values of $\lambda < \lambda_{lim1-chord-based} = \beta$ no interactions at all are present. For values of the tip speed ratio: $\lambda_{lim1-chord-based} < \lambda < \lambda_{lim1}$, interactions should only occur at the bottom half of the circular path, (i.e. azimuthal angle: $180\,^\circ < \theta < 270\,^\circ$). Inspection of the results depicted in Fig. 6.4 shows limited interactions for $\lambda = 1$ (black arrow), which suggests that for the one bladed turbine the induction factor β is smaller than unity (i.e. $\beta < 1$).

Following the same reasoning as for the derivation of λ_{lim1} (i.e. considering $\theta = 90\,^\circ$ and $\theta = 270\,^\circ$), a second limiting case can be found: $\lambda_{lim2} > 3\pi/2 \approx 4.7$ (i.e. assuming $\beta \approx 1$ and $t_{wake} \approx 2R/\beta U$ and $t_{foil} = 3R\pi/(R\omega)$). This corresponds to the blade travelling 1.5 revolutions (from $\theta = 90\,^\circ$ to $\theta = 630\,^\circ$) and the first wake (shed at $\theta = 90\,^\circ$) not having been convected aft of the turbine as shown in Fig. 6.5.b.3. For this last case, a cumulative wake effect (i.e. two wakes have been shed) is to be expected, since the wakes being shed by the foil at $\theta = 90\,^\circ$ and $\theta = 450\,^\circ$ have an impact on the blade performance once it reaches $\theta = 630\,^\circ$.

The force traces depicted in Fig. 6.4 illustrate the previous discussed limiting cases: for $\lambda = 1 < \lambda_{lim1}$, interactions are limited to $180\,^\circ < \theta < 270\,^\circ$ (black arrow). For $\lambda_{lim1} < \lambda = 2 < \lambda_{lim2}$, interactions are limited to $180\,^\circ < \theta < 360\,^\circ$ (red arrow) (as sketched in Fig. 6.5.b.2). For $\lambda_{lim2} < \lambda = 5$, interactions are of non-linear type (i.e. cumulative wake effect) showing no clear periodic character (blue arrow) (see sketch Fig. 6.5.b.3).

6.3.2 One bladed vs three bladed turbine

This section compares one and three bladed turbines for the tip speed ratios $\lambda = 1$ and 2. The resulting solidities are $\sigma = 1/2$ and $3/2$ respectively. Fig. 6.6 depicts the force traces for a unique blade and 4 revolutions (starting on the second). The figure shows that for the

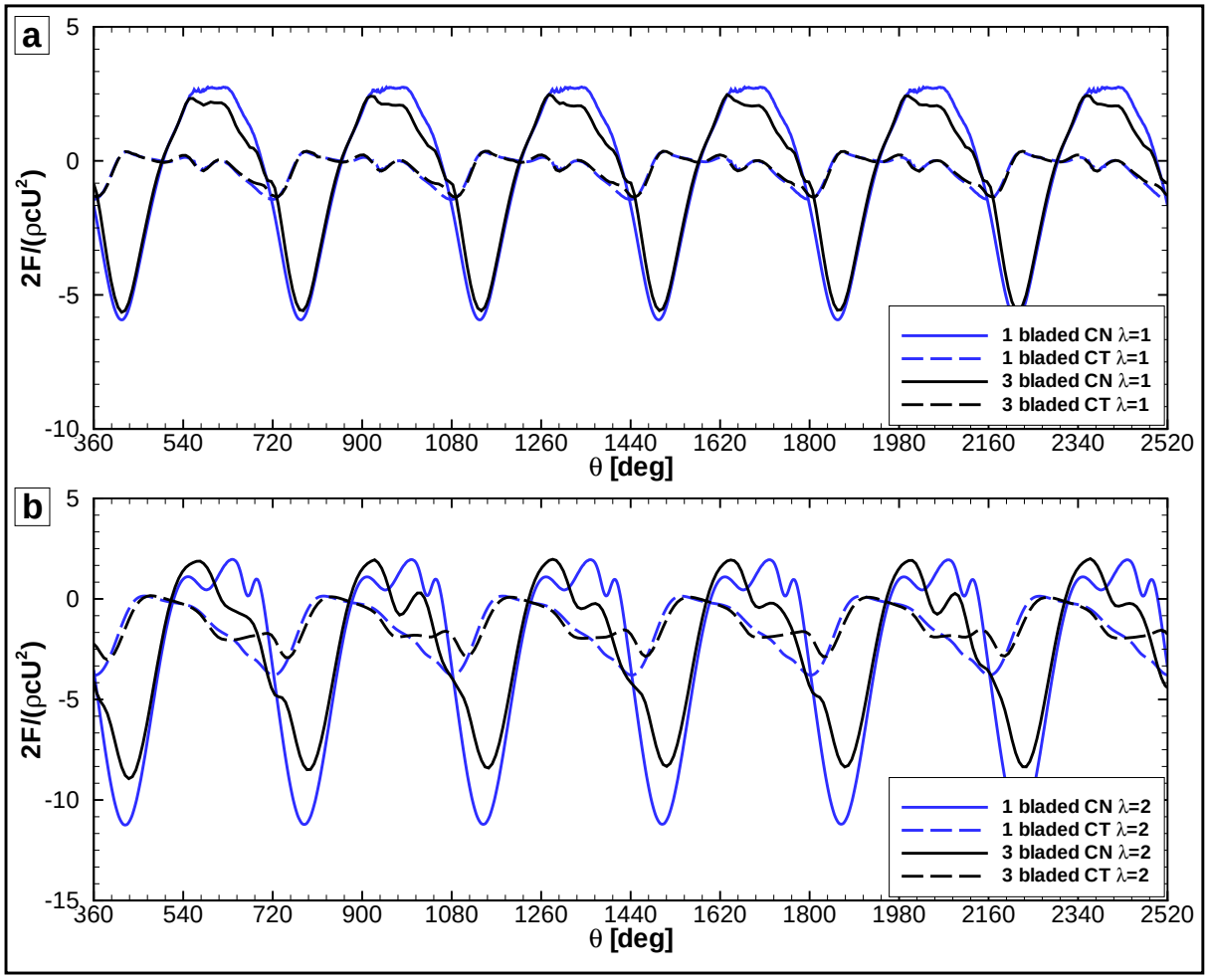

Figure 6.6: Tangential and normal force coefficients against azimuth, one and three bladed turbines (a) $\lambda = 1$ and (b) $\lambda = 2$. Forces shown are for a single blade.

upstream passage ($360\,^\circ < \theta < 540\,^\circ$), the three bladed turbine presents lower forces than observed for the one bladed turbine. These lower forces can be explained by considering the greater streamline deflection (increased axial induction factor) for the three bladed turbine due to its higher solidity (effectively higher flow resistance). Deflection of the flow leads to a lower induced AOA which leads to reduced loads. However, this difference is

larger for the higher tip speed ratio, showing that the effective blockage (resistance to the flow) relates to both the solidity σ and the tip speed ratio λ (i.e. $\beta = f(\sigma, \lambda)$). On the downstream passage, the blade of the three bladed turbine sees a more perturbed and slower flow than the one bladed turbine, resulting in lower forces.

One can also obtain estimates for the blade-wake interaction limits for the three bladed turbine, as in the previous section considering the three bladed geometry:

- $\lambda_{lim1} > \beta\pi/6 \approx 0.5\beta$: blade interacts with one wake. In this case, the wake shed by a front passing blade at $\theta \approx 90\,^\circ$ can be intercepted by another blade after rotating an angle $\triangle\theta = 60\,^\circ$ (i.e. $t_{wake} \approx 2R/\beta U$ and $t_{foil} = R\pi/(3R\omega)$).
- $\lambda_{lim2} > \beta\pi/2 \approx 1.6\beta$: blade interacting with two wakes. In this case, the blade that has shed the wake during the front passage interacts with its wake when reaching $\theta \approx 90\,^\circ$.
- $\lambda_{lim3} > \beta 5\pi/6 \approx 2.6\beta$: blade interacting with three wakes. For this case: $t_{wake} \approx 2R/\beta U$ and $t_{foil} = 5R\pi/(3R\omega)$.

Remark: *It is interesting to note that for the three bladed turbines, the induction factor β is smaller than unity due to increased solidity and that the limiting tip speed ratios are lower than for the one bladed turbine.*

Fig. 6.6.a shows, for $\lambda_{lim1} < \lambda = 1 < \lambda_{lim2}$, blade-wake interactions are limited to $180\,^\circ < \theta < 360\,^\circ$. For the higher tip speed ratio $\lambda_{lim2} < \lambda = 2 < \lambda_{lim3}$, depicted in Fig. 6.6.b, a similar non-linear effect as observed for the one bladed turbine at $\lambda = 5$ in Fig. 6.4 is shown.

Let us finally note that, to the author's knowledge, the effect of blade-wake interactions on three bladed turbines has not been fully assessed in the past. A possible explanation is that, generally, integrated or averaged forces for the three blades are studied. The integration or averaging process can, indeed, mask blade-wake interaction effects on individual blades. In the present work, the force traces have been studied for a single blade even when considering three bladed turbine configurations, which has allowed the exploration

of these interactions.

6.3.3 Ducted case or tidal fence for a three bladed turbine

This section analyses the modified physics of blocking the flow (or ducting as shown in Fig. 6.2.c, i.e. tidal fence concept) of the previously studied unconstrained three bladed turbine. The concept of ducting is under consideration as an alternative to tidal barrages, with the potential for reducing impact of energy extraction on the environment. Fig. 6.7 depicts the tangential and normal force traces for both unconstrained and ducted configurations. It can be seen that a general increase in both tangential and normal

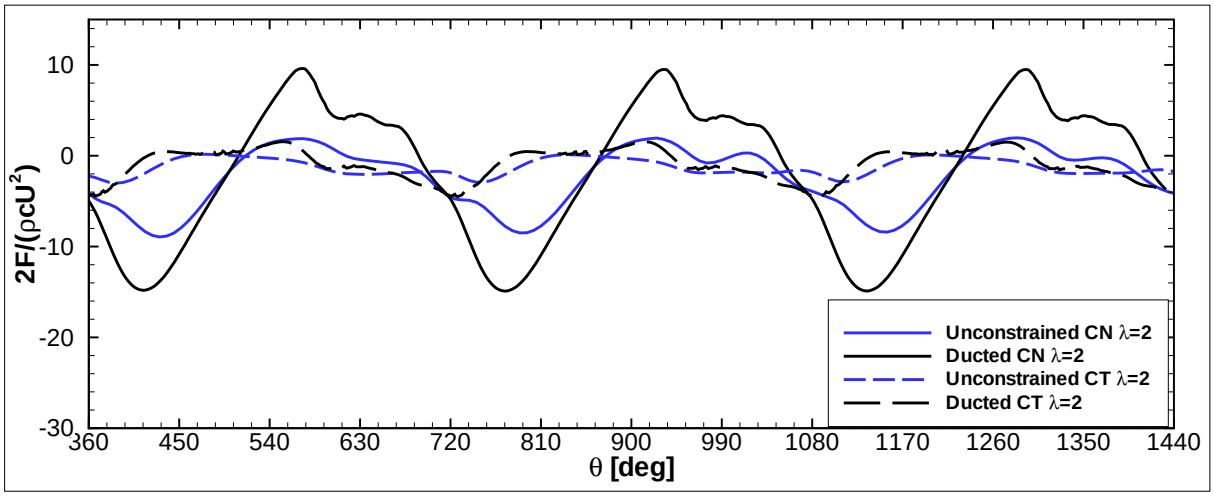

Figure 6.7: Tangential and normal force coefficients against azimuth, three bladed turbine $\lambda = 2$, unconstrained and ducted cases. Forces shown are for a single blade.

forces occurs for all azimuth angles, showing that ducting might result advantageously in increasing the power generated. The increased velocity also increases wake convection (i.e. induction factor $\beta > 1$), limiting the blade-wake interaction to the downstream passage, $180\,^\circ < \theta < 360\,^\circ$ for the given tip speed ratio $\lambda = 2$ (the non-linear character has been suppressed).

6.4 Three dimensional simulations

In this section, three dimensional flows for one and three bladed turbines are presented. The same turbine geometries defined in the previous section 6.3 are used and extended using the Fourier extension with a periodic length of $L_z/c = 2$ (see chapter 4 for details) to account for 3D flow features.

6.4.1 One bladed turbine

This section explores two physical phenomena. First, various tip speed ratios (λ) are studied by varying the rotational speed (ω), whilst maintaining a constant free stream velocity (U). This parameter space can be related to taking off conditions (i.e. starting conditions) of the turbine. Let us note that this parameter space is different from the space defined in section 6.3, where the free stream velocity was varied whilst maintaining constant the rotational speed. Second, the effect of increasing the Reynolds number (based on the free stream velocity and the foil chord): Re=100, 200 and 500, is analysed for a fixed tip speed ratio ($\lambda = 1$), showing distinct flow behaviours and force traces.

For the first set of simulations, the Reynolds number based on the free stream velocity is fixed to 200 for the three simulations presented and the rotational speed is varied to $\omega = 0.1$, 0.3 and 0.5 rad/s. Provided the free stream velocity is fixed to $U = 1$ m/s, these rotational speeds lead to tip speed ratios of $\lambda = 0.2$, 0.6 and 1 respectively. Table 6.2 summarises the flow conditions and turbine characteristics considered here.

The one bladed turbine required a spatial discretisation composed of 1794 DG tri-quad

Free stream velocity U [m/s]	**Rotational speed** ω [rad/s]	**Tip speed ratio** λ
1	0.1	0.2
1	0.3	0.6
1	0.5	1

Table 6.2: Flow conditions and turbine characteristics for 3D simulations: varying the rotational speed.

elements with polynomial order $k = 3$ in the $x-y$-plane and 16 Fourier planes to discretise the z-direction. The non-dimensional time step is set to $\triangle t = 0.001$.

Fig. 6.8 shows snapshots after one revolution (i.e. $\theta \approx 360\,^\circ$) and the three tip speed ratios considered. It can be seen that three dimensionalities develop in all simulations but that different flow patterns arise in the wakes, which are dependent on the rotational speed. In addition, it can be seen that the variation of the tip speed ratio provides a different path for the wake. In particular, it is noticeable the tilting towards the negative y-direction for the highest tip speed ratio $\lambda = 1.0$, which may be related to the effect of flow blockage by the rotating blade (i.e. rotating blade shadowing the wake) and the blade-wake interaction for this particular tip speed ratio (as discussed below).

At these low Reynolds numbers, the three dimensional velocities depicted in Fig. 6.8 are found to be weak. Inspection of the force traces revealed that these 3D effects do not alter the forces along the span significantly (not shown). Hence, the normal and tangential forces on the rotating blades are only shown for one single spanwise section in Fig. 6.9. In Fig. 6.9.a, the Reynolds number is fixed to Re=200 and the tip speed ratios are varied as λ =0.2, 0.6 and 1.0, which correspond to the snapshots previously depicted in Fig. 6.8. It can be seen that the blade experiences different regimes: at λ =0.2, the blade shows vortex shedding that translates into oscillating forces near $\theta = 90\,^\circ$ and $\theta = 180\,^\circ$ due to the high angle of attack experienced. For higher tip speed ratios (i.e. λ =0.6 and 1.0) these oscillations disappear, showing a more benign regime of partially attached flow. These flow conditions correspond to the turbine "take off condition" (i.e. ramping rotational speed) and show that these transient states may produce undesirable transient loads (e.g. reducing fatigue live).

Returning to the blade-wake interaction analysis for one bladed turbines developed in section 6.3.1, it was derived that the tip speed ratio limit for interactions to disappear is: $\lambda < \lambda_{lim1-chord-based} = \beta$. Inspection of the force traces shown in Fig. 6.9.a, suggests that the induction factor, for this one bladed turbine, can be approximated as $\beta \approx 0.6$, since

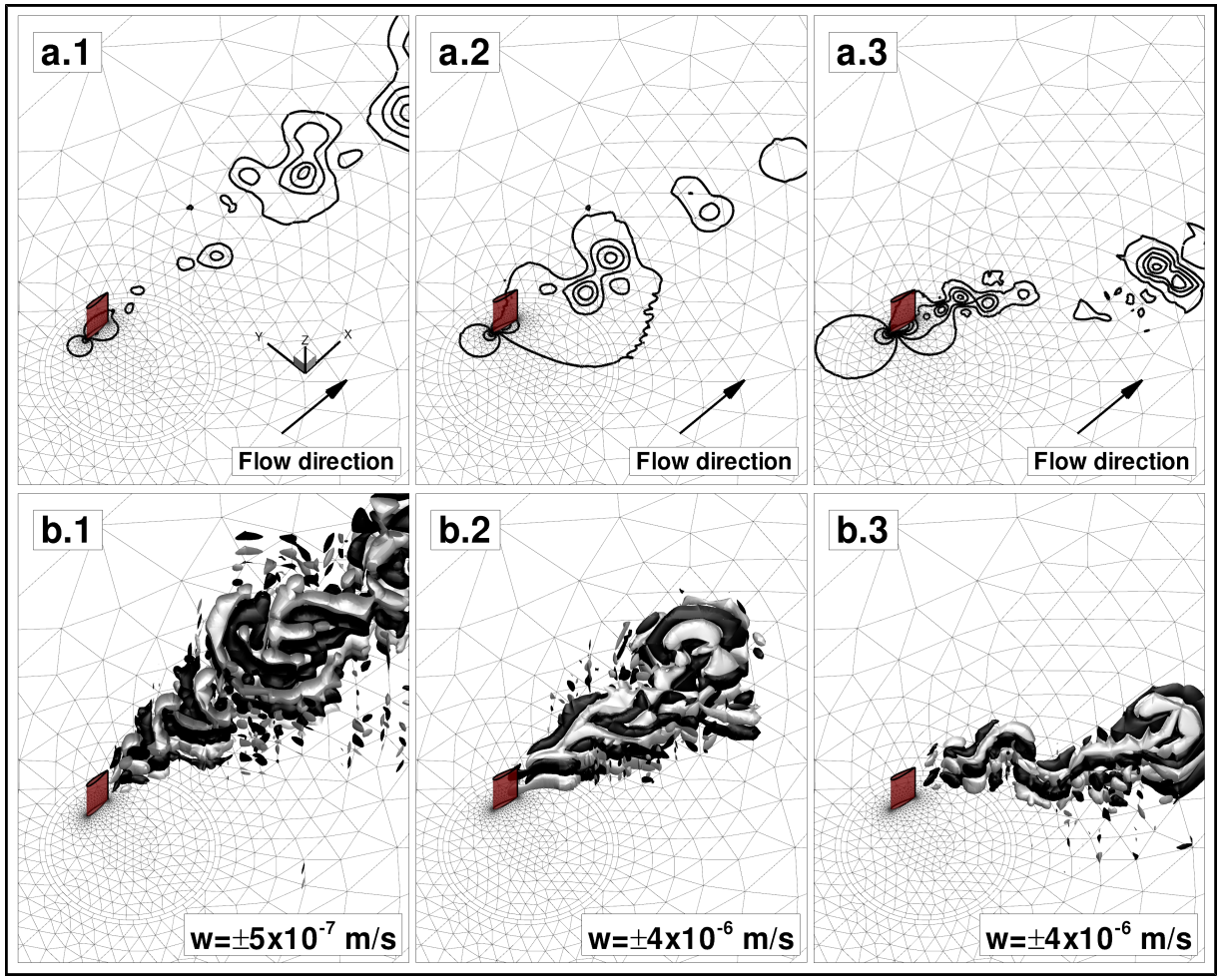

Figure 6.8: DG-Fourier solution snapshots for a rotating blade after one revolution (i.e. $\theta \approx 360\,^{\circ}$) and three tip speed ratios. The figures show (a) 11 pressure contour [-0.8:1] in Pascals and (b) iso-surface of z-velocity w (black negative and gray positive) for: (1) $\lambda = 0.2$, (2) $\lambda = 0.6$ and (3) $\lambda = 1.0$.

for $\lambda = 0.6$ the red curve is almost antisymmetric with respect to the azimuthal angle $\theta = 180\,^{\circ}$, showing very weak blade-wake interactions.

The second set of simulations explores the effect of increasing the Reynolds number with a constant tip sped ratio. Fig. 6.9.b shows this effect for Re=100, 200 and 500 (i.e. Reynolds number based on the free stream velocity U) and a fixed tip speed ratio $\lambda = 1.0$. The figure shows that as the Reynolds number increases, both tangential and normal forces increase. It can also be seen that an increase in the Reynolds number results in a more abrupt stall behaviour and reattachment (elbow like shape near $\theta = 90\,^{\circ}$

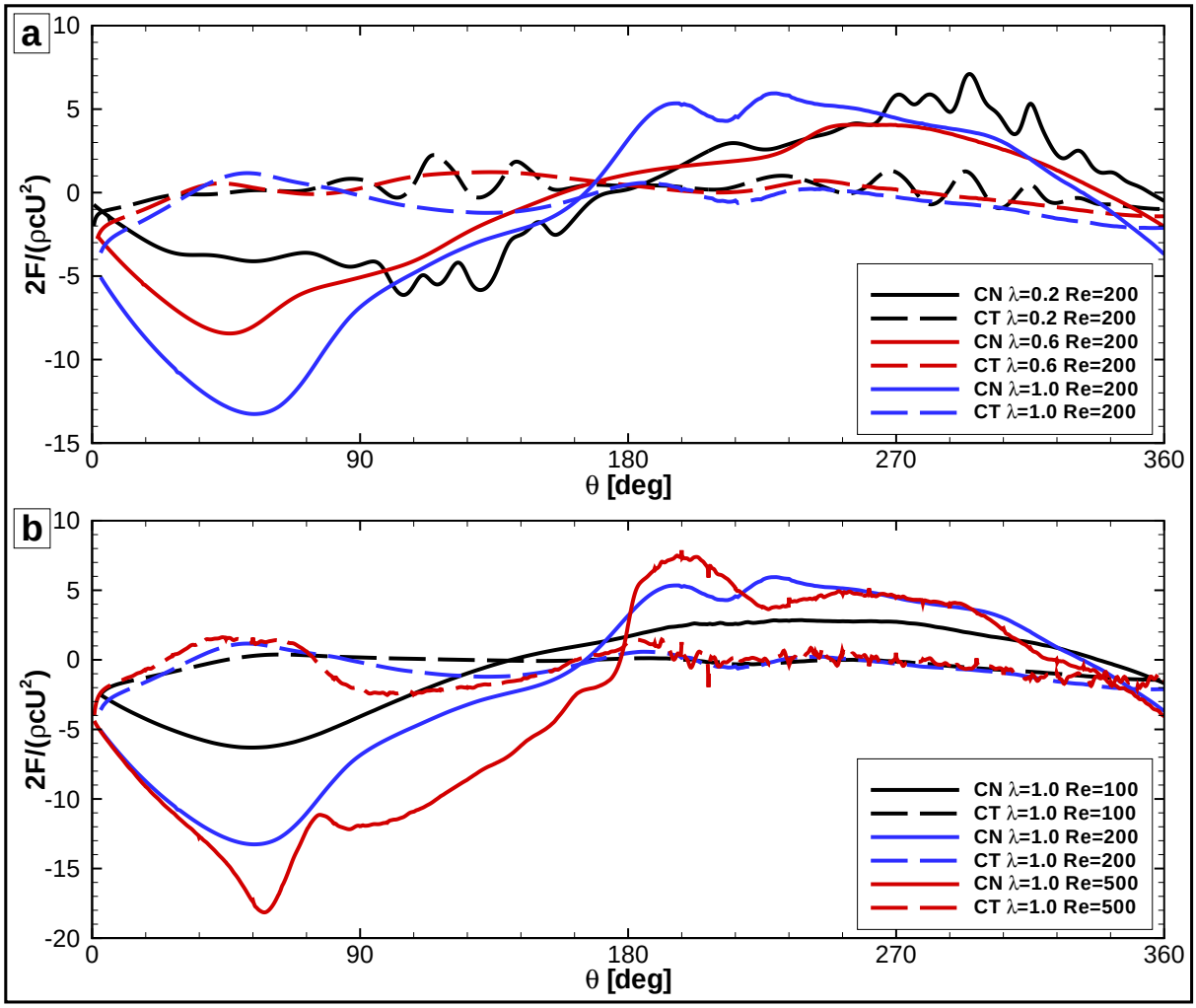

Figure 6.9: Tangential and normal force coefficients against azimuth, one bladed turbine: (a) Fixed Reynolds number Re=200 and tip speed ratios λ =0.2, 0.6 and 1.0 and (b) Fixed tip speed ratio λ =1.0 and Reynolds numbers Re=100, 200 and 500.

and $\theta = 200\,^{\circ}$). In addition, the oscillations presented by the force traces corresponding to Re=500 (red curve) for the rear passage, suggests that the wake interacting with the blade has finer flow structures as expected for a higher Reynolds number flow.

6.4.2 Three bladed turbine

This section presents a 3D simulation for a rotating three bladed turbine and shows the complex 3D character of the resulting flow. This 3D simulation requires a spatial

discretisation of 3320 DG tri-quad elements with polynomial order $k = 4$ for $x - y$-plane and 16 Fourier planes to discretise the z-direction to discretise a spanwise length of $L_z/c = 2$, where c is the blade chord. The Reynolds number, based on the free stream velocity, is Re=250 and the rotational speed is fixed to $\omega = 0.1$ rad/s leading to a tip speed ratio of $\lambda = 0.2$ (i.e. $U = 1$ m/s). The non-dimensional time step is set to $\triangle t = 0.00025$. Fig. 6.10.a shows the pressure contours and Fig. 6.10.b depicts iso-surfaces of z-velocity showing the three dimensional character of the flow simulation. This simulation intends to demonstrate the potential of the developed DG-Fourier solver to compute three dimensional flows with relation to three bladed cross-flows turbines.

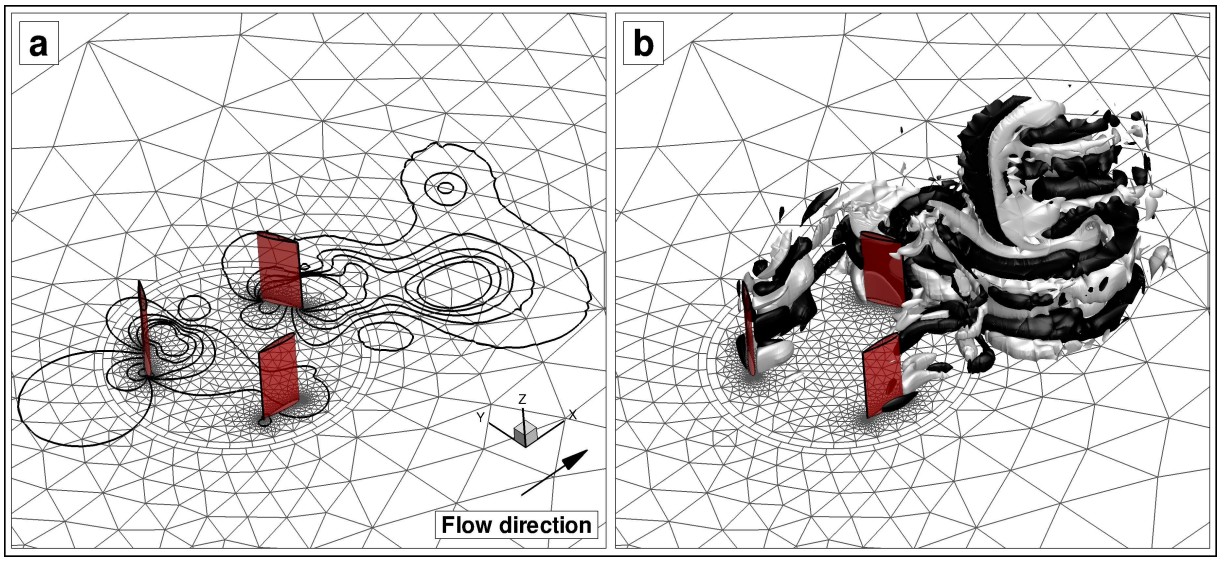

Figure 6.10: DG-Fourier solution snapshot for a 3D simulation of a three bladed turbine at a tip speed ratio $\lambda = 0.2$: (a) 11 pressure contours [-1.6:0.8] in Pascals and (b) Iso-surfaces of z-velocity $w = \pm 1 \times 10^{-6}$ m/s.

Chapter 7

Conclusions

This thesis details the formulation, implementation, verification and validation of a novel approach, which has been implemented from scratch, to solve the three dimensional incompressible Navier-Stokes (NS) equations using a high order ($order \geq 3$) Discontinuous Galerkin - Fourier (DG-Fourier) formulation with rotating sliding meshes and curved boundary conditions. Laminar and turbulent flows can be simulated by means of Direct Numerical Simulation (DNS) or Large Eddy Simulation (LES) closure models.

The solver has been verified and validated for a range of flows including flows where the analytical solution is known, bluff body flows, foil aerodynamics and cross-flow turbine flows. It is concluded that this novel approach provides a promising framework for studying fluid-structure interaction problems involving relative body motion. The following section details the main conclusions and results obtained in each of the chapters included in this thesis. In addition, section 7.2 details future directions to enhance the solver capabilities and presents some research lines that the author has not been able to fully explore within the scope of this work.

7.1 Conclusions by chapter

7.1.1 Chapter 1: Introduction

The introductory chapter presents the driving forces for the numerical developments detailed in this thesis: the simulation of cross-flow turbine (CFT) flows for renewable energy generation. In particular, the chapter details and characterises CFT flows in terms of time, length and velocity scales for the different phenomena encountered when simulating this type of devices. CFTs show complex flow phenomena such as stalled flows, vortex shedding and blade-vortex interactions. In addition, the necessity of accuracy when computing CFT flows is addressed. Finally, an analysis of the numerical requirements to compute CFT flows identifies the Discontinuous Galerkin method as the most appropriate.

7.1.2 Chapter 2: A high order Discontinuous Galerkin solver for incompressible flows

An unstructured two-dimensional incompressible Navier-Stokes (NS) code that uses a dual splitting scheme for time discretisation and a modal Discontinuous Galerkin Symmetric Interior Penalty (DG-SIPG) method for spatial discretisation on triangular element meshes is implemented, verified and validated. The DG-SIPG spatial discretisation has been verified and the accuracy assessed for the Helmholtz equations with varying wave numbers. It has been shown that p-refinement provides exponential decay of the error while providing optimal convergence rates (i.e. $\mathcal{O}(h^{k+1})$) for h-refinement.

For the unsteady Stokes problem, it has been observed that the error in the L^2 norm is of the form $\mathcal{O}(\triangle t^2 + h^{k+1}/\triangle t)$ such that both spatial and temporal refinement are required to reduce the overall error. $\triangle t_{min}$; the time step below which no further reduction in error can be achieved for a given spatial discretisation, has been estimated. In addition, it has been shown that there is no conflict for the selection of the time step, when this condition is considered together with the Courant-Friedrichs-Lewy (CFL) condition, providing a

useful incompressible NS solver. Stability estimates (for equal order approximation for velocity and pressure spaces) related to the inf-sup condition of the equivalent matrix system for the splitting temporal scheme, have shown that high polynomial orders are required to avoid stability problems when small time steps or high Reynolds numbers need to be simulated. In addition, a stablilised algorithm for low spatial resolution cases, that relies on an increased penalty parameter for the pressure step, has been proposed.

The full NS test case (Taylor Vortex) showed that the method provides exponential convergence and optimal convergence rates (i.e. $\mathcal{O}(h^{k+1})$) for h-refinement. The solver has been validated for the flow past a square section cylinder, showing excellent agreement with experimental data and very similar results and convergence trends to the continuous h/p Spectral code Nektar. Comparisons include a steady wake case (Re = 10) and a vortex shedding case (Re = 100).

7.1.3 Chapter 3: High order sliding meshes and curved elements

The formulation, implementation and verification of a novel approach to solve the two-dimensional incompressible NS equations using a high order DG formulation with rotating sliding meshes has been presented. The solver allows accurate solutions with curved elements that conform to either arced surfaces (internal or external) and the profile of NACA 4-digit foils. Hybrid unstructured meshes (i.e. with triangular and quadrilateral elements) can be handled without damaging the solver's high order numerical properties. To account for relative motion, the NS equations are written in Arbitrary Lagrangian-Eulerian form and the sliding interface is given special treatment that maintains the temporal convergence rates and spectral properties of the code when rotational motion is applied to a region of the mesh. High order properties are shown to be maintained when using body fitted NACA 4-digit boundaries.

It is demonstrated that *geometric* and *functional incompatibilities* do not represent a problem in the DG context and that sliding mesh interfaces can be handled naturally

without the necessity to use *mortar* type techniques.

The solver accuracy has been verified with various test cases: the Taylor vortex problem, the Wannier problem, a rotating square cylinder and a static and rotating NACA0015 foil. These test cases have enabled verification of the code as well as demonstrating its capabilities to simulate flow problems in which accuracy is mandatory.

To the author's knowledge, this chapter presents a novel framework for studying fluid-structure interaction problem involving rotation.

7.1.4 Chapter 4: Three dimensional flows and parallelisation

The 2D DG solver has been extended using Fourier series to account for 3D flow features. This extension provides an efficient way to handle three dimensional flows at a tractable computational cost, through the use of the fast Fourier transform. This Fourier extension allows high order 3D solutions for rotating geometries that present a spanwise geometric homogeneity. In addition, it is shown not to damage the capabilities presented for the 2D DG solver in previous chapters. A parallelisation strategy based on the Message Passing Interface (MPI) paradigm is introduced to alleviate the computational cost associated with solving the 3D NS equations.

Various 3D verification and validation cases are presented: an elliptic problems, a full NS problem with exact solution and three dimensional flows aft square and circular cylinders. In addition, the 3D flow over a circular cylinder which is shadowed by a pitching NACA0012 blade is also reported to illustrate the capability for simulating 3D flows where static and rotating geometries coexist in one single domain.

7.1.5 Chapter 5: Computing turbulent flows

Two Large Eddy Simulation (LES) approaches are presented to compute flows at moderate Reynolds numbers. The first method is derived through the filtered LES equations and incorporates a Smagorinsky term (modified for high order methods) to account for

turbulent effects of unresolved flow scales. The incorporation of this closure model in the temporal advancement splitting scheme is detailed and takes advantage of the DG framework (i.e. piecewise discontinuous functions). This model is named Smagorinsky-DG. In addition, analysis shows that this model may provide enhanced inf-sup stability capabilities. Although its formulation generalises to three dimensional flows, the present implementation is restricted to 2D solutions. This restriction relates to the coupling introduced by the Fourier extension, which does not allow for an easy implementation of this closure model in 3D using the DG-Fourier approach.

The second technique, referred to as ILES-DG SVV-Fourier, combines an Implicit LES (ILES) approach in the DG plane with a Spectral Vanishing Viscosity (SVV) technique in the Fourier direction. In the present implementation, the ILES-DG approach relies on upwinding of the non-linear terms (i.e. Lesaint-Raviart formulation) to stabilise the solution at moderate Reynolds numbers. In the Fourier plane, the SVV method provides enhanced stabilisation for moderate Reynolds numbers by introducing additional artificial viscosity to the highest modes.

Both techniques have been tested and shown to provide stable solutions at moderate Reynolds numbers (i.e. $\mathrm{Re} = \mathcal{O}(10^4)$) on coarse meshes. Qualitative and quantitative results are reported for a circular cylinder at $\mathrm{Re} = 3900$ and symmetric NACA foils at $\mathrm{Re} = 10000$. The simulations are compared to other numerical techniques (i.e. DNS, unsteady RANS) and experimental data, showing generally good agreement. It is concluded that these techniques show promise and may be explored further in future work.

7.1.6 Chapter 6: Simulations of cross-flow turbines

The developed solver has been used to simulate cross-flow turbines (CFT) under laminar flow conditions. The high order solver has shown to provide accurate solutions for the complex flows encountered when simulating CFT. In particular, the sliding mesh capability does not perturb the flow solution and can handle difficult flow phenomena as

encountered in CFT flows.

The laminar simulations presented provide understanding to physical phenomena that are to be expected under real operational conditions (higher Reynolds number flows). Two-dimensional simulations helped to analyse various physical phenomena. Firstly, the occurrence of blade-wake (or blade-vortex) interactions on the downstream passage of one and three bladed turbines, secondly, the increased effective flow resistance of three bladed turbines, which relates to the solidity and the tip speed ratio, and finally the increased forces experienced under ducted conditions.

Three dimensional simulations of rotating blades have shown that the DG-Fourier solver provides accurate 3D flow features. In addition, it is shown that at low Reynolds numbers, weak three dimensionalities are present in rotating one and three bladed turbines but that these are not large enough to significantly modify the forces along the span. In addition, the rotational speed of the blade (i.e. tip speed ratio) is shown to influence the wake development and structure of the three dimensional wake. The effect of increasing the rotational speed for a given free stream velocity and the effect of increasing the Reynolds number whilst fixing the rotational speed are also reported for one bladed turbines. The chapter concludes with the computation of the 3D flow through a rotating three bladed turbine.

It may be concluded that the developed solver provides unique capabilities for the study of complex flow phenomena and is a valuable tool for the study of wind and tidal cross-flow turbine flow physics.

7.2 Future Directions

The developed solver provides a new framework for further developments in the topics of numerical solution of the NS equations, bluff body flows, turbulence modelling and rotating machinery such as wind and tidal turbines. Due to the time constraint imposed by the doctoral studies, the author has not been able to fully explore some of the topics

summarised in this section, which aims to provide research guidelines for further developments.

7.2.1 Solver efficiency and faster iterations

The presented work has dealt with new numerical strategies to compute flows aft dynamically rotating geometries. Although the author has tried to program a "fast enough" code, computational efficiency has not been the main focuss of this thesis, and therefore there is scope for improvement in this area (i.e. reduce the computational cost). Two particular improvements are outlined here and could significantly improve the computational speed of the solver.

On the one hand, it has been noted in section 3.3, that when computing rotating meshes, volume and flux contributions to the bilinear matrices are computed on the fly for the rotating mesh zone. This is clearly a bottleneck that can be improved by pre-computing various un-rotated matrices and updated at each iteration through the rotation matrix (i.e. matrix-matrix multiplication). It is important to note that the metrics (e.g. determinant of the Jacobian, gradients and normals) require updating after rotation, which renders this enhancement non trivial to implement. For this reason, this has not yet been incorporated in the current version of the solver.

On the other hand, to compute the linear solves, associated with the Poisson and Helmholtz steps, the intel MKL-Pardiso direct-iterative solver [118] has been used. This choice was mainly driven by the availability of these libraries and its capability to compute linear solves using matrices in sparse format. Furthermore, these libraries allow for symmetric and non-symmetric matrix systems to be solved with minor modifications, which has proven to be useful when comparing various Interior Penalty formulations (i.e. SIPG, IIPG and NIPG results reported in appendix C) in the first stages of this thesis. The iterative variant of the solver allows for LU preconditioning, which has shown to provide accurate results, however its performance has not been fully assessed. Further speed

improvements can be obtained by comparing, benchmarking and analysing different preconditioning strategies and other linear solvers to compute the required linear systems. Furthermore, it has been shown that multigrid strategies (either h or p multigrid) can decrease the computational cost considerably (e.g. [6] or [14]), but these have not been considered here. These topics are currently under development by various groups and could improve the solver efficiency dramatically.

Further computational speed up could be achieved by considering new hardware architectures such as Graphical Processing Units (GPUs). For example, the CULA Package [89] provides a sparse *Conjugate Gradient* solver with incomplete LU preconditioning that runs on GPU's, whilst other parts of the code may run on the CPU system. The limitation of this approach is that hybrid CPU-GPU hardware systems are required, which have not been available to the author throughout this thesis (only classic CPU based Beowulf clusters have been used) .

7.2.2 Self adjusting time step and faster simulations

An attractive approach to decrease the computational time per simulation (not per time step), is provided by a variable time stepping discretisation algorithm. For problems where the geometry changes with time and the flow evolves substantially with the geometrical changes (as in cross-flow turbine simulations), a variable time step solver can prove to be very advantageous. In [174], Wang and Ruuth extended various implicit-explicit (IMEX) linear multistep methods to account for varying time steps. In particular, they propose a time varying scheme referred to as *variable step-size second-order semi-implicit BDF* (VSSBDF2), which represents the variable time step extension of the scheme adopted in this thesis. The original scheme for fixed time steps (Eq. 7.2.1) and the modified time

step varying scheme (Eq. 7.2.2) follow:

$$Fixed\ \ \Delta t: \qquad \frac{1}{\Delta t}\left[\frac{3}{2}U^{n+1} - 2U^{n} + \frac{1}{2}U^{n-1}\right] = 2f(U^{n}) - f(U^{n-1}) + g(U^{n+1}), \tag{7.2.1}$$

$$Varying\ \ \Delta t: \qquad \frac{1}{\Delta t^{n}}\left[\frac{1+2w_n}{1+w_n}U^{n+1} - (1+w_n)U^{n} + \frac{{w^2}_n}{1+w_n}U^{n-1}\right] = (1+w_n)f(U^{n}) - w_n f(U^{n-1}) + g(U^{n+1}), \tag{7.2.2}$$

where $w_i = \Delta t^i/\Delta t^{i-1}$ accounts for the time step variation.

This scheme has been implemented in the 2D DG version of the solver with sliding meshes and preliminary results are reported herein. The strategy adopted, to adjust the time step, requires computation of the new time step (Δt^i) based on the maximum time step allowable by the CFL condition (i.e. maximum allowed time step for a stable simulation), following the consideration presented in section 3.2.2.2. Fig. 7.1 depicts preliminary results for a one bladed turbine with a tip speed ratio $\lambda = 2$ and polynomial order $k = 2$ (detailed results for fixed time step can be found in chapter 6 and are included in the figure for comparison). For varying time steps, the CFL number has been limited to 2, since this has shown to provide a stable time advancement procedure (see Fig. 7.1.c). It can be seen in Fig. 7.1.b that the varying time step procedure allows for larger time steps to be used, hence decreasing considerably the overall time for a complete calculation. However, inspection of Fig. 7.1.a, shows that the sudden time step change introduces noise when computing the forces, which is clearly an undesired property. For this reason, the current time varying implementation is only recommended when the simulation is to be advanced until a steady state (if this exists) and should be avoided for unsteady flow phenomena calculations. To date, the author cannot provide an explanation of this undesired effect associated with the change in the time step size. This requires further investigation. It can be concluded, nonetheless, that the varying time step approach is very promising in reducing the computational time for problems where the flow evolves considerably as time

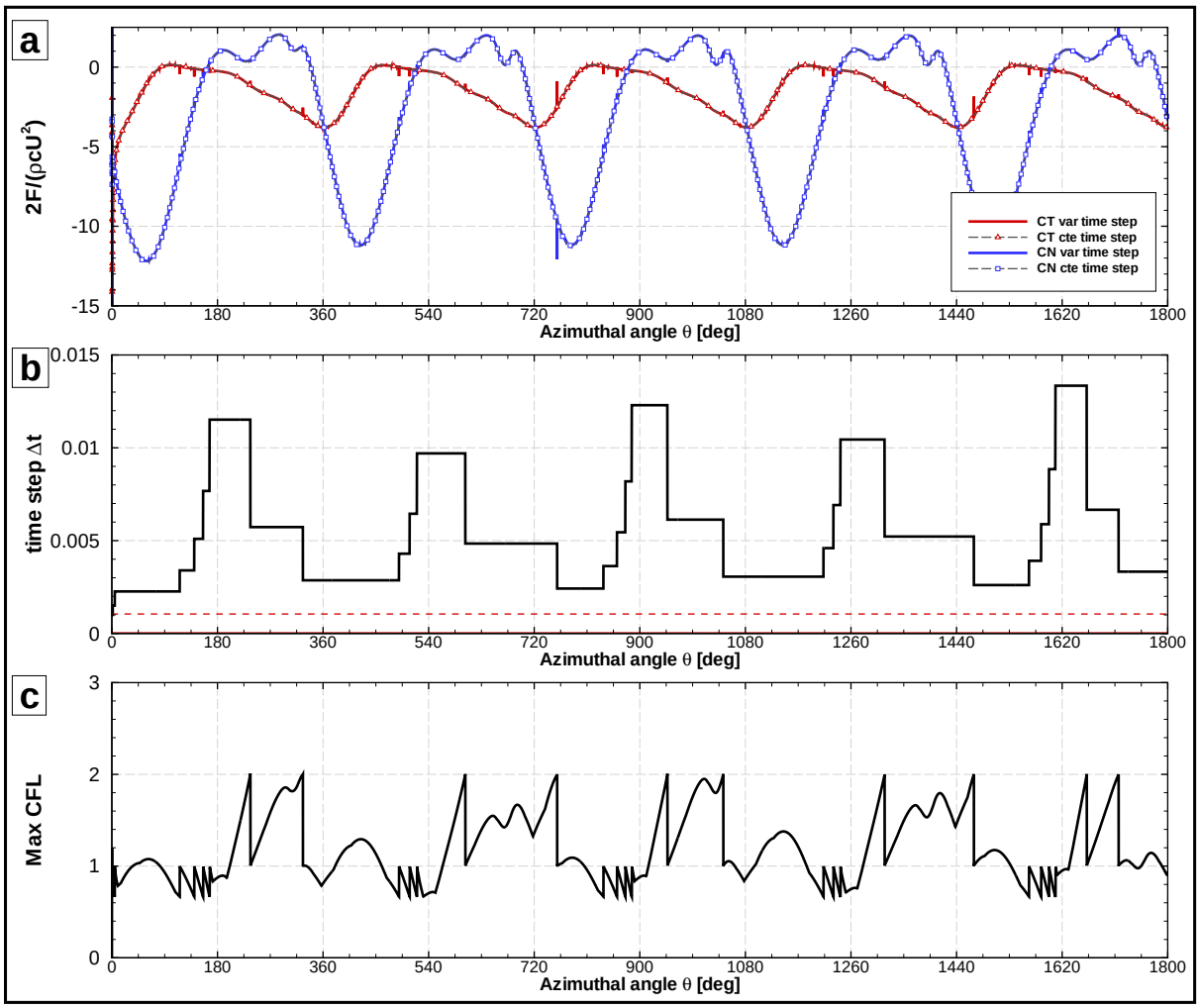

Figure 7.1: One bladed turbine computed using fixed and varying time step algorithms: (a) Tangential and normal force coefficients against azimuth, (b) varying time step (black) and fixed time step (red) and (c) CFL estimate for the varying time step simulation.

advances.

7.2.3 Exploration and improvement of time marching methods

The existing code uses a fractional-step method (i.e. dual splitting technique) that has been shown to provide stable solution under particular conditions, when equal order spaces for pressure and velocity are used (see chapter 2). This stability condition requires high order spatial discretisations (e.g. high polynomial orders), which is not always easy to impose for large problems since the computational effort may become overwhelming.

A future research line could include further examination of this inf-sup condition and exploration of other fractional-step methods. A starting point may be provided by the method developed in [70], which is claimed to be unconditionally inf-sup stable.

7.2.4 Local p-refinement

Since DG methods do not require continuity across elements, and can handle *functional incompatibilities* naturally, local p-refinement (i.e. neighbouring elements have different numbers of local degrees of freedom) can be implemented without compromising the numerical properties of the scheme. Future work could explore p-refinement strategies, based on posteriori error estimates, with particular emphasis to aerodynamic functionals as proposed for compressible DG codes in [76]. The use of variable polynomial orders in the same mesh should provide a faster solver. Furthermore, flows where a wide range of scales are present (see introductory chapter 1), as in cross-flow turbine flows, should benefit from a p-adaption strategy to provide accurate cost effective solutions.

7.2.5 Full 3D extension

To date, the DG-Fourier solver allows for 3D solutions with a homogeneous direction (i.e. the Fourier direction). It could prove useful to expand this code to a full 3D DG version (i.e. no homogeneity required). A full 3D DG solver would allow for computations of other turbine geometries as horizontal axis type turbines. The most time consuming part of this task is predicted to be the definition of high order curved surfaces since most mesh generators do not account for high order surface boundary conditions. However, let us note that the analytical approach introduced in this thesis, to decouple flow solution spaces and geometry definition, can be extended to general 3D geometries. Finally, if an analytical expression for the external curved boundary is not known a-priori, then an analytical expression can be obtained through approximation (e.g. polynomial, NURBS).

7.2.6 Turbulence modelling

Two LES strategies have been studied in this thesis. However, time constraints have not allowed for a deep investigation of these approaches. In particular, these turbulent techniques have not been studied in the context of cross-flow turbine simulations. Future research in this area could include coupling an appropriate p-refinement strategy with the ILES-DG SVV-Fourier technique presented in this thesis, which should provide an accurate and efficient strategy to compute turbulent flows. More specifically, the ILES-DG SVV-Fourier method can be extended using *Variational Multiscale* techniques to accommodate the local spatial resolution required, as suggested by the work of Ramakrishnan and Collis [35].

In addition, if a full 3D extension becomes available, the proposed Smagorinsky-DG scheme proposed in chapter 5 should provide an interesting starting point for further developments and study of 3D turbulence.

7.2.7 Free surface simulation for tidal turbines

When dealing with tidal turbine applications, flow confinement between the free surface (i.e. water-air interface) and the ground may be of importance (see [45] or [119]). To account for blockage effects, a free surface model may be incorporated into the DG-Fourier solver to account for the interaction of the varying static pressure field in the vicinity of the turbine and the free surface deformation. A possible numerical approach is to define the free surface as an external boundary condition (e.g. slip wall) and use the ALE framework (in the context of deforming mesh elements) to move the boundary as time evolves. To do so, one can, for example, relate the boundary deformation to the variations on the hydrodynamic pressure extrapolated from previous time steps.

Appendix A

Metrics and Jacobians for triangular and quadrilateral elements

This appendix details the derivation of metrics and determinant of Jacobians $|\mathbf{J}_\Psi|$ required to map volume and surface integrals from physical to computational space for the different types of elements considered:

- Triangular elements with straight edges
- Quadrilateral elements with straight edges
- Quadrilateral elements with curved edges, including circular arced edges and NACA 4-digit edges.

Figure A.1 depicts the three mappings.

A.1 Triangular elements with straight edges

The transformation for triangular elements is constituted by two changes. The map $1b$ transforms the general straight sided triangle (x,y) to a unit triangle (η_1, η_2) followed by the map $1a$ which defines the mapping from the unit triangle (η_1, η_2) to the unit square (ξ_1, ξ_2). Since the mapping requires two transformations, the chain rule for differentiation

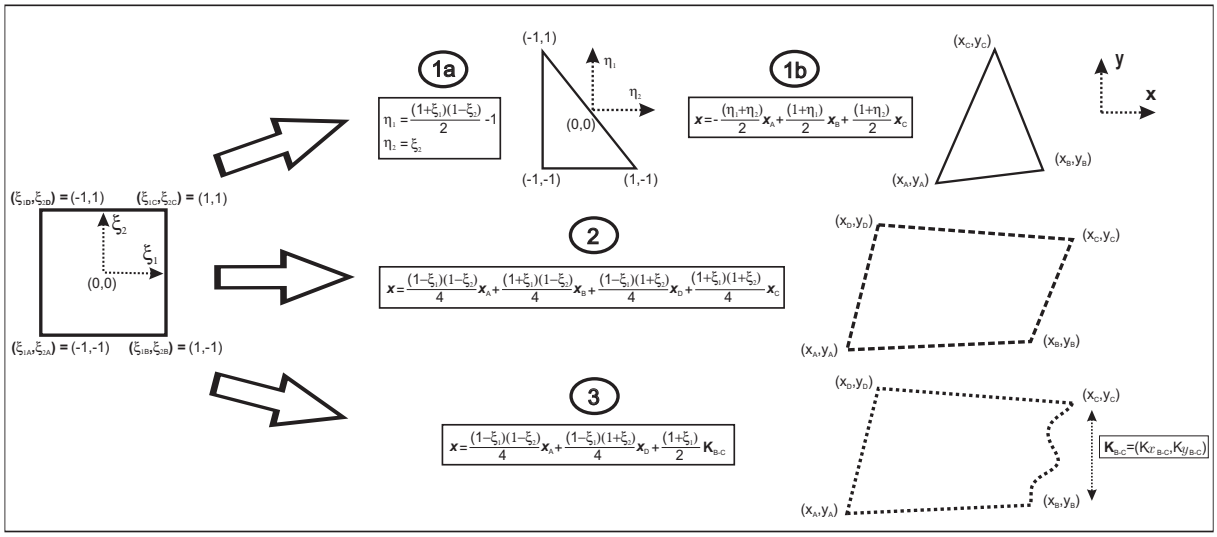

Figure A.1: Mapping from computational space to physical space for straight sided triangular, straight sided quadrilateral and quadrilateral elements with one curved edge.

is required, details can be found in [101], but the basic concepts are detailed here. The determinant of the Jacobians for the transformations $|J_1| = |J_{1a}||J_{1b}|$ can be obtained by differentiation of the mapping provided in Fig. A.1:

$$\begin{aligned} |J_{1b}| &= |J(\eta_1, \eta_2)| = \left| \frac{\partial(x, y)}{\partial(\eta_1, \eta_2)} \right| = \left| \left(\frac{\partial x}{\partial \eta_1} \frac{\partial y}{\partial \eta_2} - \frac{\partial y}{\partial \eta_1} \frac{\partial x}{\partial \eta_2} \right) \right| \\ &= \frac{1}{4} \left[(x_B - x_A)(y_C - y_A) - (y_B - y_A)(x_C - x_A) \right] = \frac{1}{2} \mathcal{A}, \end{aligned} \tag{A.1.1}$$

where $\mathcal{A}$ represents the triangle area and

$$|J_{1a}| = |J(\xi_1, \xi_2)| = \left| \frac{\partial(\eta_1, \eta_2)}{\partial(\xi_1, \xi_2)} \right| = \left| \left(\frac{\partial \eta_1}{\partial \xi_1} \frac{\partial \eta_2}{\partial \xi_2} - \frac{\partial \eta_2}{\partial \xi_1} \frac{\partial \eta_1}{\partial \xi_2} \right) \right| = \frac{1}{2}(1 - \xi_2). \tag{A.1.2}$$

When considering the last mapping (from the unit triangle to the unit square), it is important to note that $|J_{1a}|$ becomes zero for $\xi_2 = 1$. It is thus necessary to select an appropriate quadrature rule to avoid this singularity. In the present implementation, Gauss-Legendre quadrature is used, which does not include the interval ends and hence avoids the problematic point.

Let us now introduce the necessary metrics (i.e. gradients). Using the chain rule for differentiation one can obtain:

$$\frac{\partial \bullet}{\partial x} = \left[\frac{\partial \bullet}{\partial \xi_1}\frac{\partial \xi_1}{\partial \eta_1} + \frac{\partial \bullet}{\partial \xi_2}\frac{\partial \xi_2}{\partial \eta_1}\right]\frac{\partial \eta_1}{\partial x} + \left[\frac{\partial \bullet}{\partial \xi_1}\frac{\partial \xi_1}{\partial \eta_2} + \frac{\partial \bullet}{\partial \xi_2}\frac{\partial \xi_2}{\partial \eta_2}\right]\frac{\partial \eta_2}{\partial x}, \tag{A.1.3}$$

$$\frac{\partial \bullet}{\partial y} = \left[\frac{\partial \bullet}{\partial \xi_1}\frac{\partial \xi_1}{\partial \eta_1} + \frac{\partial \bullet}{\partial \xi_2}\frac{\partial \xi_2}{\partial \eta_1}\right]\frac{\partial \eta_1}{\partial y} + \left[\frac{\partial \bullet}{\partial \xi_1}\frac{\partial \xi_1}{\partial \eta_2} + \frac{\partial \bullet}{\partial \xi_2}\frac{\partial \xi_2}{\partial \eta_2}\right]\frac{\partial \eta_2}{\partial y}. \tag{A.1.4}$$

This differentiation has provided a map involving $\frac{\partial \eta_i}{\partial x}$ and $\frac{\partial \eta_i}{\partial y}$, however a more useful map would involve the inverse transformations $\frac{\partial x}{\partial \eta_i}$ and $\frac{\partial y}{\partial \eta_i}$. To obtain the required metrics, let us evaluate the total change as:

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \eta_1} & \frac{\partial x}{\partial \eta_2} \\ \frac{\partial y}{\partial \eta_1} & \frac{\partial y}{\partial \eta_2} \end{bmatrix} \begin{bmatrix} d\eta_1 \\ d\eta_2 \end{bmatrix}.$$

This matrix may be inverted provided $|J(\eta_1, \eta_2)| \neq 0$, $\forall \eta_1, \eta_2$ (i.e. bijective transformation):

$$\begin{bmatrix} d\eta_1 \\ d\eta_2 \end{bmatrix} = \frac{1}{|J_{1b}|} \begin{bmatrix} \frac{\partial y}{\partial \eta_2} & -\frac{\partial x}{\partial \eta_2} \\ -\frac{\partial y}{\partial \eta_1} & \frac{\partial x}{\partial \eta_1} \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix}. \tag{A.1.5}$$

Alternatively, it is possible to apply the chain rule directly to η_1 and η_2:

$$\begin{bmatrix} d\eta_1 \\ d\eta_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial \eta_1}{\partial x} & \frac{\partial \eta_1}{\partial y} \\ \frac{\partial \eta_2}{\partial x} & \frac{\partial \eta_2}{\partial y} \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix}. \tag{A.1.6}$$

Equating Eq. A.1.5 and Eq. A.1.6, one obtains:

$$\frac{\partial \eta_1}{\partial x} = \frac{1}{|J_{1a}|}\frac{\partial y}{\partial \eta_2}, \quad \frac{\partial \eta_1}{\partial y} = -\frac{1}{|J_{1a}|}\frac{\partial x}{\partial \eta_2}, \quad \frac{\partial \eta_2}{\partial x} = -\frac{1}{|J_{1a}|}\frac{\partial y}{\partial \eta_1}, \quad \frac{\partial \eta_2}{\partial y} = \frac{1}{|J_{1a}|}\frac{\partial x}{\partial \eta_1}, \tag{A.1.7}$$

and similarly, for the transformation from the unit triangle to the unit square (step 1a):

$$\frac{\partial \xi_1}{\partial \eta_1} = \frac{1}{|J_{1b}|}\frac{\partial \eta_2}{\partial \xi_2}, \quad \frac{\partial \xi_1}{\partial \eta_2} = -\frac{1}{|J_{1b}|}\frac{\partial \eta_2}{\partial \xi_1}, \quad \frac{\partial \xi_2}{\partial \eta_1} = -\frac{1}{|J_{1b}|}\frac{\partial \eta_1}{\partial \xi_2}, \quad \frac{\partial \xi_2}{\partial \eta_2} = \frac{1}{|J_{1b}|}\frac{\partial \eta_1}{\partial \xi_1}. \tag{A.1.8}$$

Summarising, Eq. A.1.3 and Eq. A.1.4 can be evaluated using the following expressions for the metrics:

$$\begin{aligned}
\frac{\partial \bullet}{\partial x} &= \frac{1}{|J_{1a}||J_{1b}|}\left\{\left[\frac{\partial \bullet}{\partial \xi_1}\frac{\partial \eta_2}{\partial \xi_2}-\frac{\partial \bullet}{\partial \xi_2}\frac{\partial \eta_1}{\partial \xi_2}\right]\frac{\partial y}{\partial \eta_2}+\left[\frac{\partial \bullet}{\partial \xi_1}\frac{\partial \eta_2}{\partial \xi_1}-\frac{\partial \bullet}{\partial \xi_2}\frac{\partial \eta_1}{\partial \xi_1}\right]\frac{\partial y}{\partial \eta_1}\right\} \qquad \text{(A.1.9)}\\
&= \frac{2}{\mathcal{A}(1-\xi_2)}\left\{\left[\frac{\partial \bullet}{\partial \xi_1}+\frac{\partial \bullet}{\partial \xi_2}\frac{1-\xi_1}{2}\right](y_C-y_A)+\left[\frac{\partial \bullet}{\partial \xi_2}\frac{-1+\xi_2}{2}\right](y_B-y_A)\right\},\\
\frac{\partial \bullet}{\partial y} &= \frac{-1}{|J_{1a}||J_{1b}|}\left\{\left[\frac{\partial \bullet}{\partial \xi_1}\frac{\partial \eta_2}{\partial \xi_2}-\frac{\partial \bullet}{\partial \xi_2}\frac{\partial \eta_1}{\partial \xi_2}\right]\frac{\partial x}{\partial \eta_2}+\left[\frac{\partial \bullet}{\partial \xi_1}\frac{\partial \eta_2}{\partial \xi_1}-\frac{\partial \bullet}{\partial \xi_2}\frac{\partial \eta_1}{\partial \xi_1}\right]\frac{\partial x}{\partial \eta_1}\right\} \qquad \text{(A.1.10)}\\
&= \frac{2}{\mathcal{A}(1-\xi_2)}\left\{\left[\frac{\partial \bullet}{\partial \xi_1}+\frac{\partial \bullet}{\partial \xi_2}\frac{1-\xi_1}{2}\right](x_A-x_C)+\left[\frac{\partial \bullet}{\partial \xi_2}\frac{-1+\xi_2}{2}\right](x_A-x_B)\right\}.
\end{aligned}$$

A.2 Quadrilateral elements with straight edges

In the case of quadrilateral elements, the previous transformation simplifies since only one step is required. The map transforms the general straight sided quadrilateral (x,y) to the unit square (ξ_1, ξ_2). The Jacobian is defined as follows:

$$\begin{aligned}
|J_2| &= |J(\xi_1,\xi_2)| = \left|\frac{\partial(x,y)}{\partial(\xi_1,\xi_2)}\right| = \left|\left(\frac{\partial x}{\partial \xi_1}\frac{\partial y}{\partial \xi_2}-\frac{\partial y}{\partial \xi_1}\frac{\partial x}{\partial \xi_2}\right)\right| \qquad \text{(A.2.1)}\\
&= \frac{1}{4}\left[(1-\xi_2)(x_B-x_A)+(1+\xi_2)(x_C-x_D)\right]\left[(1-\xi_1)(y_D-y_A)+(1+\xi_1)(y_C-y_B)\right]\\
&\quad -\frac{1}{4}\left[(1-\xi_2)(y_B-y_A)+(1+\xi_2)(y_C-y_D)\right]\left[(1-\xi_1)(x_D-x_A)+(1+\xi_1)(x_C-x_B)\right].
\end{aligned}$$

In the general case, the determinant of the Jacobian is non constant. For quadrilaterals elements with two parallel edges, it can be shown that the Jacobian becomes constant (e.g. if $\overrightarrow{BC}=\overrightarrow{AD}$, then $|J_2|=(x_B-x_A)(y_D-y_A)-(y_B-y_A)(x_D-x_A)$). However, in the present implementation, quadrilateral elements are used mainly when curved edges are present, and hence the general definition for the non-constant determinant of the Jacobians is retained. Following a similar reasoning as for triangular elements, a definition for the

metrics for quadrilateral elements with straight edges can be obtained:

$$\frac{\partial \bullet}{\partial x} = \frac{1}{|J_2|}\left[\frac{\partial \bullet}{\partial \xi_1}\frac{\partial y}{\partial \xi_2} - \frac{\partial \bullet}{\partial \xi_2}\frac{\partial y}{\partial \xi_1}\right] \ , \ \frac{\partial \bullet}{\partial y} = \frac{-1}{|J_2|}\left[\frac{\partial \bullet}{\partial \xi_1}\frac{\partial x}{\partial \xi_2} - \frac{\partial \bullet}{\partial \xi_2}\frac{\partial x}{\partial \xi_1}\right]. \tag{A.2.2}$$

A.3 Quadrilateral elements with a curved edge

The previous definitions can be generalised to account for quadrilateral elements with one curved edge. In this case, let us only consider the edge B-C being curved which corresponds to the edge $\boldsymbol{\xi} = (\xi_1 = 1, \xi_2)$ in the computational space as shown in Fig. A.1. This can be ensured by the appropriate node numbering when generating the mesh (but generalisation to other edges is direct). Using the definition of an arbitrarily shaped curved edge: $\boldsymbol{K}_{B-C}(\xi_2) = (K_{x_{B-C}}(\xi_2), K_{y_{B-C}}(\xi_2))$, the determinant of the Jacobian reads:

$$\begin{aligned}
|J_3| &= |J(\xi_1,\xi_2)| = \left|\frac{\partial(x,y)}{\partial(\xi_1,\xi_2)}\right| = \left|\left(\frac{\partial x}{\partial \xi_1}\frac{\partial y}{\partial \xi_2} - \frac{\partial y}{\partial \xi_1}\frac{\partial x}{\partial \xi_2}\right)\right| \\
&= \left\{\frac{1}{4}[(1-\xi_2)(-x_A) + (1+\xi_2)(-x_D)] + \frac{1}{2}K_{x_{B-C}}(\xi_2)\right\} \\
&\quad \left\{\frac{1}{4}[(1-\xi_1)(y_D - y_A)] + \frac{(1+\xi_1)}{2}\frac{dK_{y_{B-C}}(\xi_2)}{d\xi_2}\right\} \\
&\quad - \left\{\frac{1}{4}[(1-\xi_2)(-y_A) + (1+\xi_2)(-y_D)] + \frac{1}{2}K_{y_{B-C}}(\xi_2)\right\} \\
&\quad \left\{\frac{1}{4}[(1-\xi_1)(x_D - x_A)] + \frac{(1+\xi_1)}{2}\frac{dK_{x_{B-C}}(\xi_2)}{d\xi_2}\right\}.
\end{aligned} \tag{A.3.1}$$

The required metrics can be calculated using Eq. A.2.2. In addition, the local normals corresponding to the curved edge can be calculated using these metrics:

$$\begin{aligned}
N_x(\xi_2) &= \left(\frac{dK_{y_{B-C}}(\xi_2)}{d\xi_2}\right)/\tau(\xi_2), \\
N_y(\xi_2) &= -\left(\frac{dK_{x_{B-C}}(\xi_2)}{d\xi_2}\right)/\tau(\xi_2),
\end{aligned} \tag{A.3.2}$$

where $\tau(\xi_2) = |\mathbf{J_C}(\xi_2)|$, the differential length defined as:

$$\tau(\xi_2) = \sqrt{\left(\frac{dK_{x_{B-C}}(\xi_2)}{d\xi_2}\right)^2 + \left(\frac{dK_{y_{B-C}}(\xi_2)}{d\xi_2}\right)^2} d\xi_2. \tag{A.3.3}$$

In the next sections, the function $\boldsymbol{K}_{B-C}(\xi_2)$ and its gradient is particularised for circular arcs and NACA 4-digit edges (as defined in chapter 3).

A.3.1 Circular arced edge

The circular parametrisation is defined as:

$$\begin{aligned} K^{CIRC}_{x_{B-C}}(\xi_2) &= R\cos\left(\frac{1-\xi_2}{2}\theta_B + \frac{1+\xi_2}{2}\theta_C\right), \\ K^{CIRC}_{y_{B-C}}(\xi_2) &= R\sin\left(\frac{1-\xi_2}{2}\theta_B + \frac{1+\xi_2}{2}\theta_C\right), \end{aligned} \tag{A.3.4}$$

where $R = R_B = R_C$ and θ_B, θ_C are the polar-coordinates of nodes B and C with respect to an origin at the centre of curvature. The gradients of this parametrisation read:

$$\begin{aligned} \frac{dK^{CIRC}_{x_{B-C}}(\xi_2)}{d\xi_2} &= \frac{R}{2}(\theta_B - \theta_C)\sin\left(\frac{1-\xi_2}{2}\theta_B + \frac{1+\xi_2}{2}\theta_C\right), \\ \frac{dK^{CIRC}_{y_{B-C}}(\xi_2)}{d\xi_2} &= \frac{R}{2}(\theta_C - \theta_B)\cos\left(\frac{1-\xi_2}{2}\theta_B + \frac{1+\xi_2}{2}\theta_C\right). \end{aligned} \tag{A.3.5}$$

The Jacobian can be evaluated using Eq. A.3.1, Eq. A.3.4 and Eq. A.3.5.

A.3.2 Symmetric NACA 4-digit edge

For the NACA 4-digits, the mapping is defined as:

$$\begin{aligned} K^{NACA}_{x_{B-C}}(\xi_2) &= \frac{1-\xi_2}{2}x_B + \frac{1+\xi_2}{2}x_C, \\ K^{NACA}_{y_{B-C}}(\xi_2) &= \pm\frac{\zeta}{0.2}(\tau_0\sqrt{K^{NACA}_{x_{B-C}}(\xi_2)} - \tau_1[K^{NACA}_{x_{B-C}}(\xi_2)] \\ &\quad -\tau_2[K^{NACA}_{x_{B-C}}(\xi_2)]^2 - \tau_3[K^{NACA}_{x_{B-C}}(\xi_2)]^3 - \tau_4[K^{NACA}_{x_{B-C}}(\xi_2)]^4), \end{aligned} \tag{A.3.6}$$

where ζ is the foil maximum thickness (with respect to unit chord $c = 1$) and $\tau_0 = 0.29690$, $\tau_1 = 0.12600$, $\tau_2 = 0.35160$, $\tau_3 = 0.28430$ and $\tau_4 = 0.20250$ are the NACA constants required to define the profile geometry for symmetric NACA foils. To define the foil nose, the circular mapping Eq. A.3.4 with radius $r_{nose} = 1.1019\zeta^2$ is used . The gradients of this parametrisation read:

$$\begin{aligned}
\frac{dK^{NACA}_{x_{B-C}}(\xi_2)}{d\xi_2} &= \frac{x_C - x_B}{2}, \\
\frac{dK^{NACA}_{y_{B-C}}(\xi_2)}{d\xi_2} &= \pm\frac{\zeta}{0.2}\frac{x_C - x_B}{2}\Big(\frac{\tau_0}{2K^{NACA}_{x_{B-C}}(\xi_2)^{1/2}} - \tau_1 \\
&\quad -2\tau_2[K^{NACA}_{x_{B-C}}(\xi_2)] - 3\tau_3[K^{NACA}_{x_{B-C}}(\xi_2)]^2 - 4\tau_4[K^{NACA}_{x_{B-C}}(\xi_2)]^3\Big).
\end{aligned} \tag{A.3.7}$$

A.3.3 Cambered NACA 4-digits edge

The previous definition for the symmetric NACA can be extended to account for camber [1]. To this end, the coordinates for the mean camber line y_{camb} and the coordinates for the upper and lower foil surfaces $(K^{CAMB-UPP}_{x_{B-C}}(\xi_2), K^{CAMB-UPP}_{y_{B-C}}(\xi_2))$ and $(K^{CAMB-LOW}_{x_{B-C}}(\xi_2), K^{CAMB-LOW}_{y_{B-C}}(\xi_2))$, are introduced. The previous definition for the symmetric NACA $(K^{NACA}_{x_{B-C}}(\xi_2), K^{NACA}_{y_{B-C}}(\xi_2))$ can be used to define the coordinates for the cambered profile:

$$\begin{aligned}
K^{CAMB-UPP}_{x_{B-C}}(\xi_2) &= K^{NACA}_{x_{B-C}}(\xi_2) - \Big(K^{NACA}_{y_{B-C}}(\xi_2)\sin\delta(\xi_2)\Big), \\
K^{CAMB-UPP}_{y_{B-C}}(\xi_2) &= y_{camb}(\xi_2) + \Big(K^{NACA}_{y_{B-C}}(\xi_2)\cos\delta(\xi_2)\Big), \\
K^{CAMB-LOW}_{x_{B-C}}(\xi_2) &= K^{NACA}_{x_{B-C}}(\xi_2) + \Big(K^{NACA}_{y_{B-C}}(\xi_2)\sin\delta(\xi_2)\Big), \\
K^{CAMB-LOW}_{y_{B-C}}(\xi_2) &= y_{camb}(\xi_2) - \Big(K^{NACA}_{y_{B-C}}(\xi_2)\cos\delta(\xi_2)\Big),
\end{aligned} \tag{A.3.8}$$

where $y_{camb}(\xi_2)$ and $\delta(\xi_2)$ are defined as:

$$y_{camb}(\xi_2) = \begin{cases} \left(\frac{m}{p^2}\right) K^{NACA}_{x_{B-C}}(\xi_2) \left(2p - \frac{K^{NACA}_{x_{B-C}}(\xi_2)}{c}\right) & , \ 0 \leq K^{NACA}_{x_{B-C}}(\xi_2) \leq pc \\ \left(\frac{m}{(1-p)^2}\right) \left(c - K^{NACA}_{x_{B-C}}(\xi_2)\right) \left(1 + \frac{K^{NACA}_{x_{B-C}}(\xi_2)}{c} - 2p\right) & , \ pc \leq K^{NACA}_{x_{B-C}}(\xi_2) \leq c \end{cases}, \tag{A.3.9}$$

and

$$\delta(\xi_2) = atan\left(\frac{dy_{camb}(\xi_2)}{dK^{NACA}_{x_{B-C}}(\xi_2)} \frac{dK^{NACA}_{x_{B-C}}(\xi_2)}{d\xi_2}\right). \tag{A.3.10}$$

To define the cambered NACA, the maximum camber (m) and the location of maximum camber (p), have been introduced. In this case, the foil chord may be different than unity. Finally, it can seen that for $m = 0$ this definition reduces to the symmetric NACA formulation. The gradients can be calculated as functions of the gradient of the symmetric parametrisation defined in Eq. A.3.7:

$$\begin{aligned}
\frac{dK^{CAMB-UPP}_{x_{B-C}}(\xi_2)}{d\xi_2} &= \frac{dK^{NACA}_{x_{B-C}}(\xi_2)}{d\xi_2} - \left(\frac{dK^{NACA}_{y_{B-C}}(\xi_2)}{d\xi_2} \sin\delta(\xi_2) - \frac{\partial\delta(\xi_2)}{\partial\xi_2} K^{NACA}_{y_{B-C}}(\xi_2) \cos\delta(\xi_2)\right), \\
\frac{dK^{CAMB-UPP}_{y_{B-C}}(\xi_2)}{d\xi_2} &= \frac{\partial y_{camb}(\xi_2)}{\partial\xi_2} + \left(\frac{dK^{NACA}_{y_{B-C}}(\xi_2)}{d\xi_2} \cos\delta(\xi_2) - \frac{\partial\delta(\xi_2)}{\partial\xi_2} K^{NACA}_{y_{B-C}}(\xi_2) \sin\delta(\xi_2)\right), \\
\frac{dK^{CAMB-LOW}_{x_{B-C}}(\xi_2)}{d\xi_2} &= \frac{dK^{NACA}_{x_{B-C}}(\xi_2)}{d\xi_2} + \left(\frac{dK^{NACA}_{y_{B-C}}(\xi_2)}{d\xi_2} \sin\delta(\xi_2) + \frac{\partial\delta(\xi_2)}{\partial\xi_2} K^{NACA}_{y_{B-C}}(\xi_2) \cos\delta(\xi_2)\right), \\
\frac{dK^{CAMB-LOW}_{y_{B-C}}(\xi_2)}{d\xi_2} &= \frac{\partial y_{camb}(\xi_2)}{\partial\xi_2} - \left(\frac{dK^{NACA}_{y_{B-C}}(\xi_2)}{d\xi_2} \cos\delta(\xi_2) + \frac{\partial\delta(\xi_2)}{\partial\xi_2} K^{NACA}_{y_{B-C}}(\xi_2) \sin\delta(\xi_2)\right).
\end{aligned} \tag{A.3.11}$$

A usable definition for these metrics may be obtained using symbolic differentiation software (e.g. Matlab).

Appendix B

Derivation of Interior Penalty Discontinuous Galerkin methods

A general framework for derivation and analysis of Discontinuous Galerkin methods for elliptic equations (e.g. Interior Penalty, Local Discontinuous Galerkin, Bassi-Rebay) was derived in [9], where it was shown that all known methods can be compared through the so called *primal formulation*. A detailed derivation of the Interior Penalty method following this approach can be found in [9], or [74]. Alternatively, in [22] a different view on the derivation of DG methods is proposed.

This appendix however, shows the derivation of various Interior Penalty Discontinuous Galerkin methods following the original approach used for their creation since the author believes that this approach provides insight into the main features of the method (further details can be found in [146] or [60]).

B.1 Continuous Problem

Let us consider the general elliptic problem:

$$\begin{aligned} -\triangle u + \alpha u &= g \quad \text{in } \Omega, \\ u &= L_D \quad \text{on } \Gamma_D, \\ \nabla u.\mathbf{n} &= L_N \quad \text{on } \Gamma_N, \end{aligned} \tag{B.1.1}$$

where $u \in H^1(\Omega)$ is a scalar solution (but extension to the vector formulation is direct), $\sqrt{\alpha}$ is real and represents the wave number for the Helmholtz equation, $\mathbf{n}$ is the outward pointing unit normal vector on $\partial\Omega$, $g \in L^2(\Omega)$ is the forcing term, and $L_D \in H^{1/2}(\Gamma_D)$ and $L_N \in L^2(\Gamma_N)$ represent the boundary conditions. This form defines the Laplace, Poisson and Helmholtz equation as follows:

- **Laplace**: forcing term $g = 0$ and convective term $\alpha = 0$,
- **Poisson**: forcing term $g \neq 0$ and convective term $\alpha = 0$,
- **Helmholtz**: forcing term $g \neq 0$ and convective term $\alpha \neq 0$.

B.2 Local formulation and weak form

Let $\Omega = \Omega_{sta} \cup \Omega_{rot}$ be a domain in $\mathbb{R}^2$, with boundaries $\partial\Omega$ of Dirichlet (Γ_D) or Neumann (Γ_N) type, where $\Gamma_D \cap \Gamma_N = \emptyset$. Let us introduce a triangular-quadrilateral tessellation of N^{el} elements: $\Omega_h = \{el^{tri}, el^{quad}\}$ of Ω with external boundaries $\partial\Omega_h$ and interior edges Γ_h.

To obtain the weak formulation of the general elliptic problem, one considers a broken Sobolev space (i.e. elementwise continuous) $H^m(\Omega_h)$ consisting of functions $v \in L^2(\Omega)$ whose restriction to each element $el \in \Omega_h$ belongs to the Sobolev space $H^m(el)$, and analogously the vector form defined by $[H^m(\Omega_h)]^d$, where d denotes the d-vector valued broken Sobolev space. Formally, the broken space $H^m(\Omega_h)$ (i.e. $m > 1$ for two dimensional

formulations and $m > 3/2$ in three dimensions [146]) is defined as:

$$H^m(\Omega_h) = \{v \in L^2(\Omega) : \forall el \in \Omega_h, v|_{el} \in H^m(el)\}. \tag{B.2.1}$$

Remark: *It is important to note that the starting point for the development of DG methods is substantially different from the derivation of a weak form for the Continuous Galerkin method. Since DG methods consider broken spaces, the equations are discretised at a local or elemental level, in contrast to the Continuous Galerkin methods that considers continuous spaces and therefore the weak formulation is directly derived for the entire domain. This starting point, shows one of the main advantages of DG against CG methods, which is that the equations satisfy conservation locally for each element (i.e. if the numerical fluxes are chosen appropriately). It also shows the main disadvantage: discontinuous elements imply double valued degrees of freedom at element boundaries increasing the computational cost of the method.*

Multiplying Eq. B.1.1 by the test function $v \in H^1(\Omega_h)$ and integrating by parts, the weak form (or variational formulation) of the problem for the element el reduces to finding $u \in H^1(\Omega_h)$ such that:

$$\int_{el} (\nabla u.\nabla v + \alpha u v) d\mathbf{x} - \oint_{\partial el} (\nabla u.\mathbf{n}_{el} v) ds = \int_{el} g v d\mathbf{x} \quad \forall v \in H^1(\Omega_h), \tag{B.2.2}$$

where $\mathbf{n}_{el}$ is the unit outward normal to ∂el. Considering all elements $el \in \Omega_h$ and excluding the external domain boundaries $\partial\Omega_h$, one obtains:

$$\sum_{el \in \Omega_h} \int_{el} (\nabla u.\nabla v + \alpha u v) d\mathbf{x} - \sum_{\Gamma \in \Gamma_h} \int_{\Gamma} [[\nabla u.\mathbf{n}_{\Gamma} v]] ds = \sum_{el \in \Omega_h} \int_{el} g v d\mathbf{x}, \tag{B.2.3}$$

where Γ represents a positively oriented (i.e. anticlockwise orientation) single surface of the element boundaries ∂el, and Γ_h represents all the interior boundaries of Ω_h, excluding $\partial\Omega_h$. The normal vector $\mathbf{n}_{\Gamma}$ is defined as the outward pointing unit normal vector to the

direction of Γ. The term in square brackets is the jump (e.g. $[[\bullet]] = (\bullet|_{el^1}) - (\bullet|_{el^2})$, where el^1 and el^2 are neighbouring elements) and is responsible for the transfer of information between elements. This *numerical flux* arises from the discontinuity between elements (i.e. the variable is not single valued at each element boundaries) and highlights the difference between continuous and discontinuous finite element methods.

To obtain a valid method, the *numerical flux* is required to fulfill the following properties (these definitions are taken from [29]):

- *Consistency*: A numerical flux is said to be consistent if it coincides with the variable it approximates when all the functions are continuous.
- *Conservativeness*: The numerical flux is conservative when it can be defined independent of the order of the basis functions used within either side of the interface.
- *Coercivity*: The numerical flux provides a unique solution.

Methods that use this type of numerical flux are referred to as *locally conservative* and it becomes critical to ensure that any DG method fulfills these properties since as noted by Cockburn in [29]: *the fluxes couple the degrees of freedom between elements and must be carefully defined since they dramatically influence the stability and accuracy of the method.*

B.2.1 Interior penalty method

In this section, the formulation of the Interior Penalty DG numerical flux, which under the appropriate conditions can be shown to be *consistent*, *conservative* and *coercive*, is derived. In order to compute the numerical fluxes, it is necessary to introduce the definitions of the jump and averages across discontinuous elements.

Although various definitions for the jump in the DG context exist, the definition retained here is: $[[\bullet]] = (\bullet|_{el^1}) - (\bullet|_{el^2})$. This definition has the advantage to yield a vector if the considered variable is a vector and a scalar if the parameter is a scalar [146].

The most general definition for the averages [72], [122] is: $\{\{\bullet\}\} = \kappa_i(\bullet|_{el^i}) + \kappa_j(\bullet|_{el^j})$ with $\kappa_i + \kappa_j = 1$. However, in the current work, only the particular case $\kappa_i = \kappa_j = 1/2$ is

retained, leading to: $\{\{\bullet\}\} = \frac{1}{2}(\bullet|_{el^1}) + \frac{1}{2}(\bullet|_{el^2})$ (i.e. $\kappa_i = \kappa_j = 1/2$).

Let us now consider the boundary term of Eq. B.2.3 and note that the normal vector can be taken outside the jump: $[[\nabla u.\mathbf{n}_\Gamma v]] = [[\nabla u\ v]].\mathbf{n}_\Gamma$. It may also be shown that:

$$[[\nabla u\ v]] = \{\{\nabla u\}\}[[v]] + [[\nabla u]]\{\{v\}\}. \tag{B.2.4}$$

Since a *consistent* scheme is required, the exact solution should satisfy: $[[\nabla u^{exact}]] = 0$ and hence the second term in Eq. B.2.4 vanishes. Furthermore, the exact solution should also satisfy $[[u^{exact}]] = 0$. It is thus justified to rewrite the above expression with the additional term $\{\{\nabla v\}\}[[u]]$ which vanishes as the approximate solution approaches the exact solution (i.e. $[[u^{exact}]] = [[\nabla u^{exact}]] = 0$).

$$\begin{aligned}\sum_{el\in\Omega_h}\int_{el}(\nabla u.\nabla v + \alpha v u)d\mathbf{x} \;&-\; \sum_{\Gamma\in\Gamma_h}\int_\Gamma \{\{\nabla u\}\}.\mathbf{n}_\Gamma[[v]]ds + \varepsilon\sum_{\Gamma\in\Gamma_h}\int_\Gamma\{\{\nabla v\}\}.\mathbf{n}_\Gamma[[u]]ds \\ &= \sum_{el\in\Omega_h}\int_{el} g v d\mathbf{x}.\end{aligned} \tag{B.2.5}$$

Eq. B.2.5 is the general expression of the *Interior Penalty formulation* for the DG method. From the definition of ε various submethods can be identified. Before identifying the different submethods, it should be noted that some of them require penalisation for the scheme to be *coercive* [146]. Two types of penalisation have been conceived and are introduced here: $J_0^{\sigma^0,\beta_0}(u,v)$ and $J_1^{\sigma^1,\beta_1}(u,v)$. The first penalty term penalises the jump in the function across elements:

$$J_0^{\sigma^0,\beta_0}(u,v) = \sum_{\Gamma\in\Gamma_h}\frac{\sigma_{el}^0}{|trace|^{\beta_0}}\int_\Gamma [[u]][[v]]ds. \tag{B.2.6}$$

The second penalty term may be used to penalise the jump in the normal derivatives.

$$J_1^{\sigma^1,\beta_1}(u,v) = \sum_{\Gamma\in\Gamma_h}\frac{\sigma_{el}^1}{|trace|^{\beta_1}}\int_\Gamma [[\nabla u.\mathbf{n}_\Gamma]][[\nabla v.\mathbf{n}_\Gamma]]ds, \tag{B.2.7}$$

where σ^0, σ^1, β_0 and β_1 are positive real parameters and $|trace|$ is defined as the length of the element edge. Rewriting all the defined terms, one can retrieve the bilinear form, which will be used hereafter:

$$\begin{aligned} a_\varepsilon(u,v) &= \sum_{el\in\Omega_h}\int_{el}(\nabla u.\nabla v+\alpha v u)d\mathbf{x}-\sum_{\Gamma\in\Gamma_h}\int_\Gamma\{\{\nabla u\}\}.\mathbf{n}_\Gamma[[v]]ds \\ &\quad+\varepsilon\sum_{\Gamma\in\Gamma_h}\int_\Gamma\{\{\nabla v\}\}.\mathbf{n}_\Gamma[[u]]ds \\ &\quad+J_0^{\sigma^0,\beta_0}(u,v)+J_1^{\sigma^1,\beta_1}(u,v). \end{aligned} \tag{B.2.8}$$

The choice of ε in Eq. B.2.8 defines the method and has consequences on the choice of the penalty parameters, as detailed in [146]:

- $\varepsilon=-1$: The method is called the Symmetric Interior Penalty Galerkin (SIPG) and was introduced by Wheeler [177] in the late 1970s and Arnold [8] in the early 1980s. It can be shown that the method converges for a large enough value of σ_{el}^0. The combination $\epsilon=-1$ and $\sigma_{el}^0=\sigma_{el}^1=0$ was introduced by Delves et al. [42] in 1979, but provides a method that is unstable.
- $\varepsilon=+1$: The method is called Nonsymmetric Interior Penalty (NIPG) and was introduced by Riviere et al. [147] in 1999 and converges for any non negative values of σ_{el}^0. In the particular case of $\sigma_{el}^0=\sigma_{el}^1=0$ (introduced by Oden et al. [130] in 1998), the method is called OBB or NIPG 0.
- $\varepsilon=0$: The method is called Incomplete Interior Penalty Galerkin (IIPG). It was introduced by Dawson et al. [41] in 2004. As the SIPG, the method converges for large enough values of σ_{el}^0.

The stability of all the methods is independent of the penalty parameter on normal derivatives $J_1^{\sigma^1,\beta_1}(u,v)$ and hence this will not be considered hereafter.
It is important to note that some combinations of ε, σ_{el}^0, σ_{el}^1 do not provide stable solutions. For instance, the simplest method would use $\epsilon=0$ and $\sigma_{el}^0=0$, $\sigma_{el}^1=0$, however it can be

proved that this method is unstable, does not converge and solution uniqueness cannot be proved [146].

Table B.1 provides the conditions for each method to yield a unique stable solution for the elliptic problem defined in Eq. B.1.1. Let us note that $\alpha \neq 0$ identifies the equation as a Helmholtz problem and k defined the polynomial order. For the SIPG and IIPG

method	ε	k	α		σ^0
SIPG	-1	≥ 1	$\forall\alpha$	*and*	large
IIPG	0	≥ 1	$\forall\alpha$	*and*	large
NIPG	1	≥ 1	$\alpha > 0$	*or*	> 0
NIPG 0	1	≥ 2	$= 0$	*and*	$= 0$

Table B.1: Interior Penalty methods and conditions for stability

methods a large value for the penalty is required. The theory provides a lower bound for this parameter that depends on the mesh topology [48] and has been used in chapter 2 section 2.2.2.3 to derive an appropriate expression for the penalty parameter (i.e. $\sigma = 3k(k+1)$, where k is the polynomial order). A different explicit formulation for the penalty parameter exists for unstructured triangular elements and the SIPG method and was derived by Shahbazi in [155] and used to solve the incompressible NS equations in [156]. This form for the penalty parameter Eq. B.2.9 is compared to other formulations in appendix C and called is SIPG-Shahbazi.

$$J_0^{\sigma^0,\beta_0}(u,v) = \sum_{\Gamma\in\Gamma_h} \frac{(k+1)(k+d)}{d} \left[MAX\left(\frac{S_{el_1}}{V_{el_1}}, \frac{S_{el_2}}{V_{el_2}}\right)\right] \int_\Gamma [[u]][[v]] ds, \tag{B.2.9}$$

where k is the polynomial order, $d = 2$ corresponds to the dimension of the problem, S_{el} and V_{el} are the perimeter and area (in two dimensions) of each of the elements el_1 and el_2 sharing the edge Γ. This expression does not require any arbitrary penalisation term (σ, β combination), but automatically adjusts the penalisation term according to the underlying mesh geometry.

Castillo [25] performed a comparison of various DG methods including the NIPG and

the SIPG methods and showed that the condition number of the system scales with the penalty parameter. A large penalty parameter thus can considerably degrade the performance of the iterative method used to solve the system and even provide less accurate results. He also showed that for large values of the penalty parameter, all IP sub-methods have the same accuracy. Similar conclusions are found by the author for all the considered IP methods (see appendix C).

It can be seen, by inspecting Eq. B.2.8, that only the SIPG method provides a symmetric bilinear form. The symmetry of the bilinear form implies that the resulting matrix is symmetric positive definite. This has advantages when solving the resulting matrix system since it allows for the use of fast iterative solvers (e.g. Conjugate Gradient, Cholesky factorisation preconditioning). The IIPG and NIPG methods require solvers for non-symmetric matrices (e.g. GMRES) which are generally slower.

B.2.2 Boundary conditions

B.2.2.1 Dirichlet boundary

The general approach within the DG community is to treat Dirichlet boundary conditions weakly (i.e. the boundary condition is satisfied in the weak sense and not point-wise). Strong imposition of boundary condition is possible [146] but uncommon and are not considered in this work.

To define the treatment of this type of boundary condition, one needs to consider the jump and averages at external boundaries (i.e. $\Gamma = \partial el_1 \cap \partial\Omega_h$). To do so, let us distinguish between the jump and the average of the variable u and of the test function v. The latter function is only defined within the element and hence the jump and average will be defined as $[[v]] = \{\{v\}\} = (v|_{el^1})$ where el^1 is the element on $\partial\Omega_h$.

On the other hand, the jump and average of u, have to satisfy boundary conditions and are therefore written as $[[u]] = u|_{el^1} - L_D$ and $\{\{u\}\} = \frac{1}{2}u|_{el^1} + \frac{1}{2}L_D$. At Dirichlet boundaries one can also set: $\{\{\nabla u\}\} = \nabla u|_{el^1}$. Using these definitions, the previously

introduced bilinear form of Eq. B.2.8 can be extended to account for Dirichlet boundaries:

$$\begin{aligned} a_\varepsilon(u,v) &= \sum_{el\in\Omega_h}\int_{el}(\nabla u.\nabla v+\alpha v u)d\mathbf{x}-\sum_{\Gamma\in\Gamma_h\cup\Gamma_D}\int_\Gamma\{\{\nabla u\}\}.\mathbf{n}_\Gamma[[v]]ds \\ &\quad+\varepsilon\sum_{\Gamma\in\Gamma_h\cup\Gamma_D}\int_\Gamma\{\{\nabla v\}\}.\mathbf{n}_\Gamma[[u]]ds+J_0^{\sigma^0,\beta_0}(u,v)+J_1^{\sigma^1,\beta_1}(u,v). \end{aligned} \tag{B.2.10}$$

The penalty term $J_0^{\sigma^0,\beta_0}$ is also modified:

$$J_0^{\sigma^0,\beta_0}(u,v)=\sum_{\Gamma\in\Gamma_h\cup\Gamma_D}\frac{\sigma_{el}^0}{|trace|^{\beta_0}}\int_\Gamma[[u]][[v]]ds. \tag{B.2.11}$$

Using these definitions and expanding the jumps and averages at external boundaries, there is a contribution to the element matrix due to $u|_{el^1}v|_{el^1}$, and two terms involving L_D that can be moved from the bilinear form to the right hand side $\ell(v)$:

$$\sum_{\Gamma\in\Gamma_D}\int_\Gamma(\varepsilon\nabla v.\mathbf{n}_\Gamma+\frac{\sigma_{el}^0}{|trace|^{\beta_0}}v)L_D ds. \tag{B.2.12}$$

B.2.2.2 Neumann boundary

The Neumann boundary condition is defined as $\nabla u.\mathbf{n}=L_N$, which directly satisfies the surface integral term:

$$\sum_{\Gamma\in\Gamma_N}\int_\Gamma\nabla u.\mathbf{n}_\Gamma v ds=\sum_{\Gamma\in\Gamma_N}\int_\Gamma L_N v ds. \tag{B.2.13}$$

Hence this term can be moved to the right hand side and makes no contribution to the bilinear form.

B.3 Summary of the Interior Penalty DG method for elliptic problems

The DG scheme is now complete and will provide a weak solution to the elliptic problem: find $u \in H^1(\Omega_h)$ such as: $a_\varepsilon(u,\upsilon) = \ell(\upsilon)\ ,\ \forall \upsilon \in H^1(\Omega_h)$, with the bilinear form:

$$\begin{aligned} a_\varepsilon(u,\upsilon) &= \sum_{el\in\Omega_h}\int_{el}(\nabla u.\nabla\upsilon)d\mathbf{x} + \sum_{el\in\Omega_h}\int_{el}(\alpha u\upsilon)d\mathbf{x} \\ &\quad - \sum_{\Gamma\in\Gamma_h\cup\Gamma_D}\int_\Gamma\{\{\nabla u\}\}.\mathbf{n}_\Gamma[[\upsilon]]ds - \sum_{\Gamma\in\Gamma_h\cup\Gamma_D}\int_\Gamma\{\{\nabla\upsilon\}\}.\mathbf{n}_\Gamma[[u]]ds \\ &\quad + J_0^{\sigma^0,\beta_0}(u,\upsilon) + J_1^{\sigma^1,\beta_1}(u,\upsilon), \end{aligned} \tag{B.3.1}$$

and the following right hand side that completes the scheme:

$$\begin{aligned} \ell(\upsilon) &= \sum_{el\in\Omega_h}\int_{el} g\upsilon d\mathbf{x} - \sum_{\Gamma\in\Gamma_D}\int_\Gamma(\nabla\upsilon.\mathbf{n}_\Gamma - \frac{\sigma}{|trace|^\beta}\upsilon)L_D ds \\ &\quad + \sum_{\Gamma\in\Gamma_N}\int_\Gamma \upsilon L_N ds. \end{aligned} \tag{B.3.2}$$

B.4 Finite element spaces and discretised IP-DG method

Finally, the finite element form can be derived by considering the finite dimensional subspaces $D_k(\Omega_h) = D_k^{tri}(\Omega_h) \cup D_k^{quad}(\Omega_h)$ of the broken Sobolev space $H^1(\Omega_h)$ as:

$$\begin{aligned} D_k^{tri}(\Omega_h) &= \{\upsilon_h \in L^2(\Omega) : \forall el^{tri} \in \Omega_h, \upsilon_h|_{el^{tri}} \in \mathbb{P}_k(el^{tri})\}, \\ D_k^{quad}(\Omega_h) &= \{\upsilon_h \in L^2(\Omega) : \forall el^{quad} \in \Omega_h, \upsilon_h|_{el^{quad}} \in \mathbb{Q}_k(el^{quad})\}, \end{aligned}$$

where $\mathbb{P}_k(el^{tri})$ and $\mathbb{Q}_k(el^{quad})$ denote the space of polynomials of order less than or equal to k for triangular and quadrilateral elements respectively.

The IP-DG scheme provides an approximate solution by finding $u_h \in D_k(\Omega_h) \subset$

$H^1(\Omega_h)$ provided by: $a_\varepsilon(u_h, \upsilon_h) = \ell(\upsilon_h) \quad \forall \upsilon_h \in D_k(\Omega_h)$ with:

$$\begin{aligned} a_\varepsilon(u_h, \upsilon_h) &= \sum_{el \in \Omega_h} \int_{el} (\nabla u_h . \nabla \upsilon_h) d\mathbf{x} + \sum_{el \in \Omega_h} \int_{el} (\alpha u_h \upsilon_h) d\mathbf{x} \\ &\quad - \sum_{\Gamma \in \Gamma_h \cup \Gamma_D} \int_\Gamma \{\{\nabla u_h\}\} . \mathbf{n}_\Gamma [[\upsilon_h]] ds - \sum_{\Gamma \in \Gamma_h \cup \Gamma_D} \int_\Gamma \{\{\nabla \upsilon_h\}\} . \mathbf{n}_\Gamma [[u_h]] ds \\ &\quad + \sum_{\Gamma \in \Gamma_h \cup \Gamma_D} \frac{\sigma}{|trace|^\beta} \int_\Gamma [[u_h]][[\upsilon_h]] ds, \end{aligned} \tag{B.4.1}$$

and the following right hand side that completes the scheme:

$$\begin{aligned} \ell(\upsilon_h) &= \sum_{el \in \Omega_h} \int_{el} g \upsilon_h d\mathbf{x} - \sum_{\Gamma \in \Gamma_D} \int_\Gamma (\nabla \upsilon_h . \mathbf{n}_\Gamma - \frac{\sigma}{|trace|^\beta} \upsilon_h) L_D ds \\ &\quad + \sum_{\Gamma \in \Gamma_N} \int_\Gamma \upsilon_h L_N ds. \end{aligned} \tag{B.4.2}$$

Appendix C

Comparison of Interior Penalty methods for elliptic problems

The derivation of various Interior Penalty (IP) methods has been detailed in appendix B. This appendix is concerned with the verification of the implementation of the different IP methods and the characterisation of their numerical properties when solving elliptic equations. In particular, as detailed in appendix B, IP methods are characterised by the necessity of tuning a penalty parameter. The solution accuracy has been found to be influenced by this parameter [25] and therefore it becomes necessary to study its influence. The final expression for the bilinear and linear forms issued from the IP-DG discretisation of general elliptic equations (see appendix B for details) is repeated here for clarity.

The IP-DG scheme provides and approximate solution by finding $u_h \in D_k(\Omega_h)$ provided by: $a_\varepsilon(u_h, \upsilon_h) = \ell(\upsilon_h) \quad \forall \upsilon \in D_k(\Omega_h)$ with:

$$\begin{aligned} a_\varepsilon(u_h, \upsilon_h) = \quad & \sum_{el\in\Omega_h} \int_{el} (\nabla u_h . \nabla \upsilon_h) d\mathbf{x} + \sum_{el\in\Omega_h} \int_{el} (\alpha u_h \upsilon_h) d\mathbf{x} \\ & - \sum_{\Gamma\in\Gamma_h\cup\Gamma_D} \int_\Gamma \{\{\nabla u_h\}\}.\mathbf{n}_\Gamma [[\upsilon_h]] ds - \sum_{\Gamma\in\Gamma_h\cup\Gamma_D} \int_\Gamma \{\{\nabla \upsilon_h\}\}.\mathbf{n}_\Gamma [[u_h]] ds \\ & + \sum_{\Gamma\in\Gamma_h\cup\Gamma_D} \frac{\sigma}{|trace|^\beta} \int_\Gamma [[u_h]][[\upsilon_h]] ds, \end{aligned} \tag{C.0.1}$$

and the following right hand side that completes the scheme:

$$\ell(\upsilon_h) = \sum_{el\in\Omega_h}\int_{el} g\upsilon_h d\mathbf{x} - \sum_{\Gamma\in\Gamma_D}\int_{\Gamma}(\nabla\upsilon_h.\mathbf{n}_\Gamma - \frac{\sigma}{|trace|^\beta}\upsilon_h)L_D ds + \sum_{\Gamma\in\Gamma_N}\int_{\Gamma}\upsilon_h L_N ds. \tag{C.0.2}$$

C.1 Summary of a-priori error estimates for IP-DG methods

A-priori error estimates, in the L^2 norm, for the IP methods derived in the appendix B are summarised in table C.1. Error estimates for all IP methods in the Energy norm can be derived and details of the analysis can be found in [146] or [74].

Optimal error estimates in the L^2 norm for the SIPG method can be derived using Aubin-Nitsche technique as shown in [146]. However, for a long time, optimality for odd polynomials in the L^2 norm could not be theoretically proved for non-symmetric methods (i.e. NIPG and IIPG) and has remained an open question for a many years. Recently, Ayuso et al. [10] proved optimal estimates in the L^2 norm for the IIPG method in multiple dimensions. To the author's knowledge, optimal error estimates for the NIPG method in the L^2 norm have been proved for strongly imposed boundary conditions [26] but not in the general case. Therefore the L^2 estimates for the NIPG method shown in table C.1 are based on observations. The table shows that the SIPG method provides optimal convergence, i.e. $O(h^{k+1})$, in the L^2 norm independently of the penalisation used as long as this is large enough. The IIPG and NIPG methods provide optimal convergence only for odd polynomials if no super-penalisation is used, i.e. $\beta = 1$, but optimality is recovered for all polynomials when using super-penalisation, i.e. $\beta = 3$.

IP Method	ε	σ	L_2 Convergence $\beta = 1$	L_2 Convergence $\beta = 3$
SIPG	-1	large	$O(h^{k+1})$	$O(h^{k+1})$
SIPG shahbazi	-1	-	$O(h^{k+1})$	$O(h^{k+1})$
IIPG	0	large	$O(h^{k+1})$, k odd $O(h^k)$, k even	$O(h^{k+1})$
NIPG 0	+1	$= 0$	$O(h^{k+1})$, k odd $O(h^k)$, k even	–
NIPG 1	+1	$= 1$	$O(h^{k+1})$, k odd $O(h^k)$, k even	$O(h^{k+1})$

Table C.1: Penalty parameter requirements and theoretical convergence rates for various IP methods.

C.2 Verification and comparison of IP-DG methods

This section presents convergence rates for various IP methods and studies the influence of the penalty parameter in the accuracy of the different techniques. Generally, super-penalisation (i.e. $\beta = 3$) is not used and the penalty parameter β is set to one for all IP formulation. If not stated otherwise, the IIPG and the SIPG methods use $\sigma = 3k(k+1)$ as derived in chapter 2 section 2.2.2.3. The SIPG-Shahbazi method determines the penalty parameter through Eq. B.2.9 detailed in appendix B.

C.2.1 Poisson equation on triangular elements

The Poisson problem, i.e. $-\triangle u = g$, is solved in $\Omega = [-1, 1]^2$. The problem is defined with the forcing term $g(x, y) = 2\pi^2 sin(\pi x)sin(\pi y)$ and with Dirichlet boundary conditions $L_D = 0$. The analytical solution is $u^{exact}(x, y) = sin(\pi x)sin(\pi y)$. Fig. C.1.a shows the exponential convergence obtained using p-refinement on a grid with 200 triangular elements which is similar to the mesh depicted in Fig. C.2.a. The convergence rates (i.e. slope of the curve $L^2 - \sqrt{N^{el}}$, where N^{el} is the number of elements) for h-refinement using four meshes constituted of 50, 200 and 800 triangular elements, can be found in table C.2. The slopes agree well with the previously observed error estimates (see table C.1).

However, it can be seen that the convergence rates for the IIPG method are better than predicted by the theory for even k.

In addition, let us explore the influence of the penalty parameter, σ, on the solution

IP Method	k	mesh elements N_{el}			Slope
		50	200	800	
IIPG	2	3.338E-02	5.364E-03	1.049E-03	2.505
	3	4.158E-03	2.521E-04	1.540E-05	4.038
	4	4.626E-04	1.584E-05	5.955E-07	4.807
	5	4.619E-05	7.505E-07	1.183E-08	5.966
	6	3.809E-06	3.146E-08	2.691E-10	6.898
NIPG 0	2	6.982E-02	2.007E-02	5.332E-03	1.853
	3	1.084E-02	7.964E-04	5.217E-05	3.847
	4	6.619E-04	3.660E-05	2.512E-06	4.021
	5	8.529E-05	1.468E-06	2.346E-08	5.914
	6	5.296E-06	6.073E-08	8.902E-10	6.276
NIPG 1	2	6.412E-02	1.804E-02	4.718E-03	1.880
	3	9.474E-03	6.622E-04	4.246E-05	3.899
	4	6.282E-04	3.528E-05	2.391E-06	4.019
	5	8.208E-05	1.401E-06	2.230E-08	5.923
	6	5.202E-06	5.935E-08	8.655E-10	6.283
SIPG	2	2.707E-02	3.215E-03	3.931E-04	3.054
	3	3.807E-03	2.360E-04	1.455E-05	4.016
	4	4.399E-04	1.438E-05	4.554E-07	4.957
	5	4.183E-05	6.833E-07	1.082E-08	5.958
	6	3.627E-06	2.935E-08	2.317E-10	6.966
SIPG Shahbazi	2	3.079E-02	3.774E-03	4.676E-04	2.993
	3	4.078E-03	2.479E-04	1.513E-05	4.043
	4	4.585E-04	1.506E-05	4.780E-07	4.942
	5	4.510E-05	7.261E-07	6.401E-08	5.976
	6	3.736E-06	3.028E-08	2.392E-10	6.960

Table C.2: Convergence rates for a Poisson problem when performing h-refinement

accuracy for this problem and all IP formulations. Fixing the polynomial order to $k = 2$, since the effect of the penalty parameter is more apparent for low polynomial orders, one can increase the penalty parameter and explore the behaviour of the error in the L^2 norm. Results are depicted in in Fig. C.1.b. It can be seen that, for this particular test case, the IIPG and NIPG methods show decay of the error as σ is increased, but for the SIPG method the error increases. It is also observed that, as the penalty increases, all the methods converge to the same result. This fact was noted by Castillo [25] when

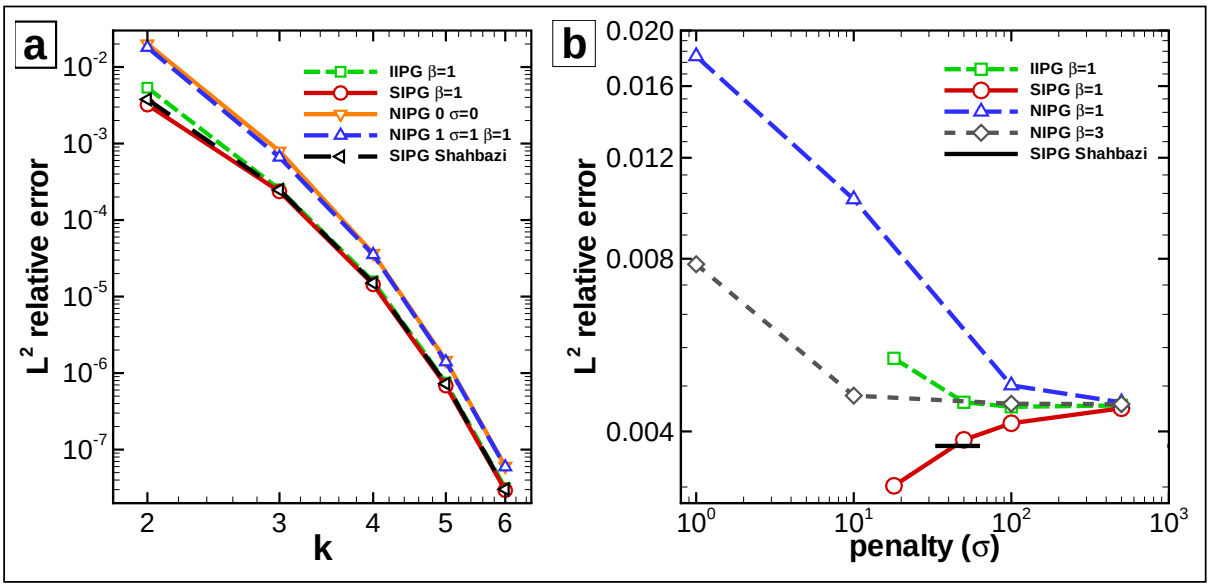

Figure C.1: (a) L^2 relative error norm for a Poisson problem and p-refinement; (b) Penalty parameter (σ) influence on the L^2 error norm for a Poisson problem and polynomial order $k = 2$.

comparing the NIPG and the SIPG methods. Finally, it can be seen in Fig. C.1.b that Shahbazi's expression (i.e. intersection of black line with SIPG curve) provides enough penalisation when compared to the SIPG method, but more than provided by the expression $\sigma = 3k(k+1)$, which corresponds to the first point defining the SIPG curve.

C.2.2 Helmholtz equation on triangular elements

The inhomogeneous Helmholtz equation, i.e. $-\triangle u + \alpha u = g$, is solved in $\Omega = [0,1]^2$. The wave number $\sqrt{\alpha}$ can be chosen arbitrarily provided that the forcing term is of the form $g(x,y) = (\alpha + 2\pi^2)cos(\pi x)cos(\pi y)$. The problem is closed using Dirichlet boundary conditions derived from the exact solution $u^{exact}(x,y) = cos(\pi x)cos(\pi y)$. Let us fix the square of the wave number to $\alpha = 10$, the penalty parameter $\beta = 1$ for all IP formulation and $\sigma = 3k(k+1)$ for the IIPG and SIPG. As previously, the convergence rates for h-refinement using three meshes with 50, 200 and 338 elements are tabulated in C.3. As noted in the previous section, the IIPG method provides better convergence rates than

IP Method	k	mesh elements N_{el}			Slope
		50	200	338	
IIPG	2	1.635E-03	2.225E-04	1.074E-04	2.858
	3	1.243E-04	7.566E-06	2.631E-06	4.035
	4	7.573E-06	2.458E-07	6.795E-08	4.937
	5	3.697E-07	5.827E-09	1.208E-09	5.989
	6	1.545E-08	1.243E-10	2.017E-11	6.953
NIPG 0	2	3.201E-03	5.564E-04	3.165E-04	2.452
	3	2.530E-04	1.675E-05	5.930E-06	3.925
	4	1.060E-05	5.267E-07	1.824E-07	4.275
	5	5.989E-07	9.413E-09	1.949E-09	5.994
	6	2.300E-08	2.508E-10	4.851E-11	6.469
NIPG 1	2	2.375E-03	4.959E-04	2.930E-04	2.211
	3	2.146E-04	1.384E-05	4.867E-06	3.960
	4	1.022E-05	5.096E-07	1.757E-07	4.274
	5	5.787E-07	9.081E-09	1.879E-09	5.996
	6	2.259E-08	2.454E-10	4.737E-11	6.475
SIPG	2	1.560E-03	1.951E-04	8.876E-05	3.000
	3	1.189E-04	7.330E-06	2.556E-06	4.019
	4	7.177E-06	2.276E-07	6.143E-08	4.981
	5	3.394E-07	5.393E-09	1.121E-09	5.978
	6	1.463E-08	1.156E-10	1.847E-11	6.984
SIPG Shahbazi	2	1.808E-03	2.303E-04	1.052E-04	2.957
	3	1.257E-04	7.656E-06	2.662E-06	4.038
	4	7.538E-06	2.391E-07	6.454E-08	4.975
	5	3.631E-07	5.708E-09	1.183E-09	5.993
	6	1.512E-08	1.195E-10	1.908E-11	6.982

Table C.3: Convergence rates for a Helmholtz problem with $\alpha = 10$ when performing h-refinement

predicted by the theory, showing optimal convergence for all polynomial orders. The meshes used, which are similar to those used to solve the previous Poisson problem, are of the type shown in Fig. C.2.a.

In addition, let us investigate the effect of the penalty parameter on the L^2 error accuracy for the SIPG method and a fixed polynomial order $k = 2$ and $\beta = 1$. This time, the square of the wave number is varied as $0 \leq \alpha \leq 10^5$ and the penalty bounded by: $18 \leq \sigma \leq 10^4$. Results are depicted in Fig. C.2.b. As for the Poisson problem, an increase of the L^2 error norm as σ increases can be observed. It can also be seen that for large wave numbers, increasing the penalty parameter has greater effect.

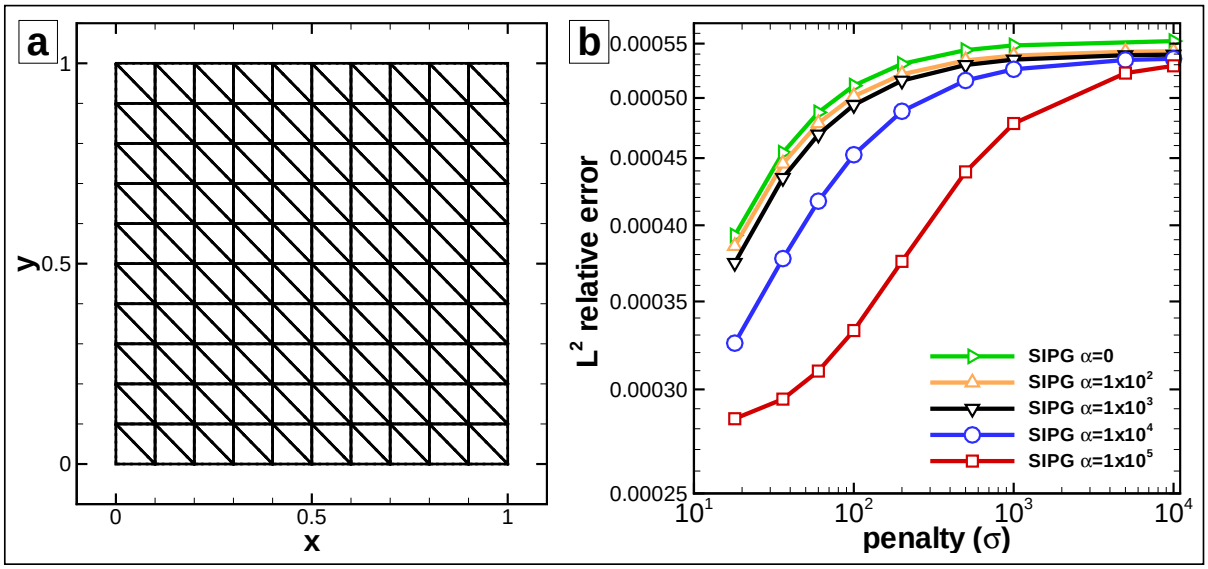

Figure C.2: (a) 200 element mesh used to study the influence of the penalty parameter σ on the solution of a Helmholtz problem; (b) Penalty parameter (σ) influence on the L^2 error norm for a Helmholtz problem using the SIPG method for various wave numbers (α) and a fixed polynomial order $k = 2$.

C.2.3 Poisson equation on mixed triangular-quadrilateral and interior curved edges with hanging nodes

In this section, the accuracy of the solver to handle curved edges and hanging nodes is studied for the Poisson problem presented in section C.2.1. Fig C.3 shows solution contours and the L^2 relative error for p-refinement for the IIPG, NIPG 0, NIPG 10 and SIPG methods. It can be seen that there is no degradation of the solver accuracy (i.e. overlapping curves), when dealing with mixed triangular-quadrilateral elements and interior curved edges with hanging nodes, for any of the Interior Penalty formulation.

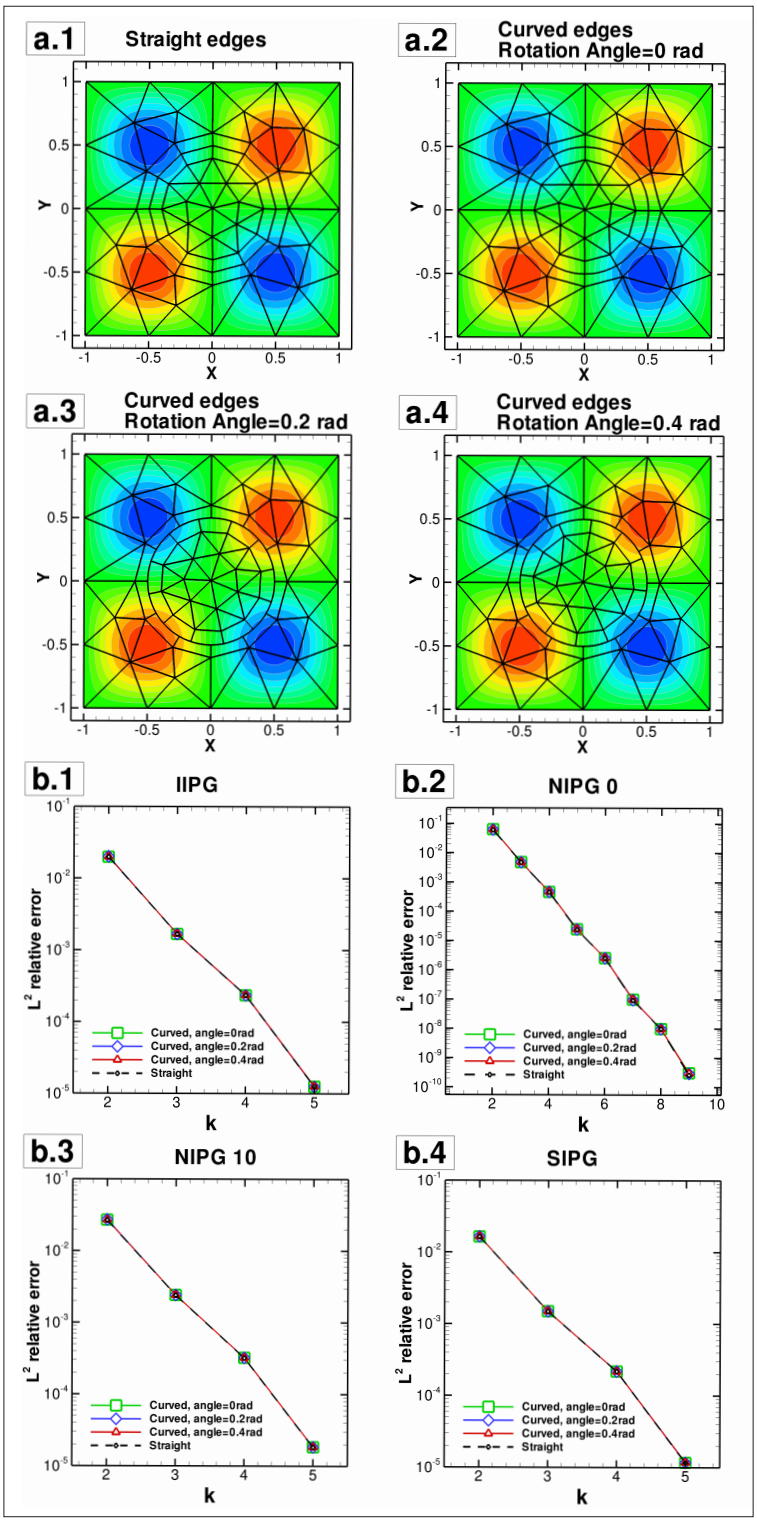

Figure C.3: Comparison of IP formulations for a Poisson problem on meshes with curved elements and hanging nodes: (a) Solution contours for NIPG 0 method an polynomial order $k = 3$ (b) L^2 relative error norm for p-refinement and IIPG, NIPG 0, NIPG 10 and SIPG methods .

Bibliography

[1] I. Abbott and A. Von-Doenhoff. *Theory of wing sections : including a summary of airfoil data*. New York : Dover, 1959.

[2] P. Alotto, A. Bertoni, I. Perugia, and D. Schötzau. Efficient use of the local discontinuous Galerkin method for meshes sliding on a circular boundary. *IEEE Transactions on Magnetics*, 38(2):405 –408, March 2002.

[3] G. Anagnostou. *Nonconforming sliding spectral element methods for the unsteady incompressible Navier–Stokes equations*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1991.

[4] J.D. Anderson. *Fundamentals of aerodynamics*. McGraw-Hill, 4 edition, 2005.

[5] S. Antheaume, T. Maitre, and J.L. Achard. Hydraulic darrieus turbines efficiency for free fluid flow conditions versus power farms conditions. *Renewable Energy*, 33(10):2186 – 2198, 2008.

[6] P. Antonietti and P. Houston. A class of domain decomposition preconditioners forh/p discontinuous Galerkin finite element methods. *Journal of Scientific Computing*, 46:124–149, 2011.

[7] D.N. Arnold. *An interior penalty finite element method with discontinuous elements*. PhD thesis, The University of Chicago, Chicago, IL, 1979.

[8] D.N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM Journal on Numerical Analysis*, 19(4):742–760, 1982.

[9] D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal of Numerical Analysis*, 39(5):1749–1779, 2001.

[10] B. Ayuso de Dios, F. Brezzi, O. Havle, and L.D. Marini. L2-estimates for the DG IIPG-0 scheme. *Numerical Methods for Partial Differential Equations*, 2011.

[11] I. Babuška. The finite element method with penalty. *Mathematics of Computation*, 27:221–228, 1973.

[12] S. Badia and R. Codina. Algebraic pressure segregation methods for the incompressible Navier–Stokes equations. *Archives of Computational Methods in Engineering*, 15:1–52, 2007. 10.1007/BF03024946.

[13] F. Bassi, A. Crivellini, D.A. Di Pietro, and S. Rebay. An artificial compressibility flux for the discontinuous Galerkin solution of the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 218(2):794 – 815, 2006.

[14] F. Bassi, N. Franchina, A. Ghidoni, and S. Rebay. Spectral p-multigrid discontinuous Galerkin solution of the Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 67(11):1540–1558, 2011.

[15] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. *Journal of Computational Physics*, 131(2), 1997.

[16] C.E. Baumann. *An hp-adaptive discontinuous finite element method for computational fluid dynamics*. PhD thesis, University of Texas at Austin, 1997.

[17] A. Beskok and T.C. Warburton. An unstructured h/p finite-element scheme for fluid flow and heat transfer in moving domains. *Journal of Computational Physics*, 174(2):492 – 509, 2001.

[18] A Betz. Das maximum der theoretisch möglichen ausnützung des windes durch windmotoren. *Zeitschrift für das gesamte Turbinenwesen*, 26(26):307–309, 1920.

[19] H.M. Blackburn, F. Marques, and J.M. Lopez. Symmetry breaking of two-dimensional time-periodic wakes. *Journal of Fluid Mechanics*, 522:395–411, January 2005.

[20] M.T. Brahimi, A. Allet, and I. Paraschivoiu. Aerodynamic analysis models for vertical–axis wind turbines. *International Journal of Rotating Machinery*, 2(1):15–21, 1995.

[21] F. Brezzi and K.J. Bathe. A discourse on the stability conditions for mixed finite element formulations. *Computational Methods in Applied Mechanics and Engineering*, 82:27–57, September 1990.

[22] F. Brezzi, B. Cockburn, L.D. Marini, and E. Süli. Stabilization mechanisms in discontinuous Galerkin finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 195(25-28):3293 – 3310, 2006.

[23] A. Buffa, I. Perugia, and T. Warburton. The mortar-discontinuous Galerkin method for the 2D Maxwell eigenproblem. *Journal of Scientific Computing*, 40:86–114, July 2009.

[24] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral methods in fluid dynamics*. Springer-Verlag, 1988.

[25] P. Castillo. Performance of discontinuous Galerkin methods for elliptic PDEs. *SIAM Journal of Scientific Computing*, 24(2):524–547, 2002.

[26] H. Chen. Superconvergence properties of discontinuous Galerkin methods for two-point boundary value problems. *International Journal of Numerical Analysis Model*, 3(2):163–185, 2006.

[27] A.J. Chorin. Numerical solution of the Navier–Stokes equations. *Math. Comp*, 22:742–762, 1968.

[28] D. Chu, R. Henderson, and G.E. Karniadakis. Parallel spectral-element Fourier simulation of turbulent flow over riblet-mounted surfaces. *Theoretical and Computational Fluid Dynamics*, 3:219–229, 1992. 10.1007/BF00417914.

[29] B. Cockburn, G. Kanschat, and D. Schotzau. The local discontinuous Galerkin method for linearized incompressible fluid flow: a review. *Computers & Fluids*, 34(4-5):491 – 506, 2005.

[30] B. Cockburn, G. Kanschat, and D. Schötzau. An equal-order DG method for the incompressible Navier–Stokes equations. *Journal of Scientific Computing*, 40:188–210, 2009.

[31] B. Cockburn and C. Shu. Runge–Kutta discontinuous Galerkin methods for convection dominated problems. *Journal of Scientific Computing*, 16(3):173–261, 2001.

[32] B. Cockburn and C.W. Shu. The local discontinuous Galerkin method for time dependent convection-diffusion systems. *SIAM Journal of Numerical Analysis*, 35:2440–2463, 1998.

[33] R. Codina. Pressure stability in fractional step finite element methods for incompressible flows. *Journal of Computational Physics*, 140(1):112–140, 2001.

[34] S.S. Collis. The DG/VMS method for unified turbulence simulation. In *Thirty-second AIAA Fluid Dynamics Conference and Exhibit, St. Louis, MO*, June 2002.

[35] S.S. Collis and S. Ramakrishnan. The local variational multi-scale method for turbulence simulation. Technical Report SAND2005-2733, Sandia National Laboratory, 2005.

[36] C.A. Consul. *Hydrodynamic analysis and design optimisation of a marine tidal cross-flow turbine*. PhD thesis, University of Oxford, 2008.

[37] C.A. Consul, R.H.J. Willden, E. Ferrer, and M.D. McCulloch. Influence of solidity on the performance of a cross-flow turbine. *Proceedings of the 8th European Wave and Tidal Energy Conference*, 2009.

[38] J.A. Cottrell, T.J.R. Hughes, and Y.Bazilevs. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons, Ltd, 2009.

[39] Y.M. Dai and W. Lam. Numerical study of straight-bladed Darrieus-type tidal turbine. *Proceedings of the Institution of Civil Engineers*, Energy 162, 2009.

[40] R.M. Darekar and S.J. Sherwin. Flow past a square-section cylinder with a wavy stagnation face. *Journal of Fluid Mechanics*, 426:263–295, 2001.

[41] C. Dawson, S. Sun, and M.F. Wheeler. Compatible algorithms for coupled flow and transport. *Computer Methods in Applied Mechanics and Engineering*, 193(23-26):2565 – 2580, 2004.

[42] L.M. Delves and C.A. Hall. An implicit matching principle for global element calculations. *IMA J Appl Math*, 23(2):223–234, 1979.

[43] M.O. Deville, P.F. Fischer, and E.H. Mund. *High-order methods for incompressible fluid flow*. Cambridge monographs on applied and computational mathematics. Cambridge University Press, 2002.

[44] J. Donea and A. Huerta. *Finite element methods for flow problems*. John Wiley & Sons Ltd., 2005.

[45] S. Draper. *Tidal stream energy extraction in coastal basins*. PhD thesis, University of Oxford, 2011.

[46] DTI. *Energy White Paper - Our Energy future - Creating a Low Carbon Economy*. 2003.

[47] H.C. Elman, D.J. Silvester, and A.J. Wathen. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Oxford University Press, 2005.

[48] Y. Epshteyn and B. Rivière. Estimation of penalty parameters for symmetric interior penalty Galerkin methods. *Journal of Computational and Applied Mathematics*, 206:843–872, September 2007.

[49] S. Eriksson, H. Bernhoff, and M. Leijon. Evaluation of different turbine concepts for wind power. *Renewable and Sustainable Energy Reviews*, 12(5):1419–1434, 2008.

[50] C. Evangelinos and G.E. Karniadakis. Communication patterns and models in prism: a spectral element-Fourier parallel Navier–Stokes solver. In *Proceedings of the 1996 ACM/IEEE conference on Supercomputing (CDROM)*, Supercomputing '96. IEEE Computer Society, 1996.

[51] C.J. Simo Ferreira, H. Bijl, Van Bussel, and G. Kuik. Simulating dynamic stall in a 2D VAWT: modeling strategy, verification and validation with particle image velocimetry data. *Journal of Physics Conference Series*, 75:012023, 2007.

[52] E. Ferrer and R.H.J. Willden. A high order discontinuous Galerkin finite element solver for the incompressible Navier–Stokes equations. *Computers & Fluids*, 46(1):224–230, 2011.

[53] J. Ferziger. The physics and simulation of turbulence. Technical report, Von Karman Institute for Fluid Dynamics, 2004.

[54] J.H. Ferziger and M. Peric. *Computational methods for fluid dynamics*. Springer, 1997.

[55] Fluent Ansys Inc. Gambit User's guide. Lebanon NH, 1998.

[56] Fluent Release 12.0. Ansys Inc.

[57] M. Frigo and S.G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation".

[58] R.E. Froude. *Transactions, Institute of Naval Architects*, 30(390), 1889.

[59] E. Gabriel, G.E. Fagg, G. Bosilca, T. Angskun, J.J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D.J. Daniel, R.L. Graham, and T.S. Woodall. Open MPI: Goals, concept, and design of a next generation MPI implementation. In *Proceedings, 11th European PVM/MPI Users' Group Meeting*, pages 97–104, Budapest, Hungary, September 2004.

[60] V. Girault and F. Wheeler. Discontinuous Galerkin methods. *Computational Methods in Applied Sciences*, 16(1):3–26, 2008.

[61] H. Glauert. *The elements of aerofoil and airscrew theory*. Cambridge University Press, Cambridge, second edition edition, 1947.

[62] W.J. Gordon and C.A. Hall. Transfinite element methods: blending-function interpolation over arbitrary curved element domains. *Numerische Mathematik*, 21:109–129, 1973.

[63] D. Gottlieb and S.A. Orszag. *Numerical analysis of spectral methods : theory and applications*. Society for Industrial and Applied Mathematics, Philadelphia :, 1977.

[64] G.I. Gretton. *The hydrodynamic analysis of a vertical axis tidal current turbine*. PhD thesis, The University of Edinburgh, 2009.

[65] G.I. Gretton and T. Bruce. Preliminary results from analytical and numerical models of a variable-pitch vertical-axis tidal current turbine. In *6th European Wave and Tidal Energy Conference*, Glasgow, UK, September 2005.

[66] G.I. Gretton and T. Bruce. Hydrodynamic modelling of a vertical-axis tidal current turbine using a Navier–Stokes solver. In *Proceedings of the 9th World Renewable Energy Congress*, Florence, Italy, 2006.

[67] G.I. Gretton, T. Bruce, and D.M. Ingram. *Hydrodynamic modelling of a vertical axis tidal current turbine using CFD*, pages 468–476. 2009.

[68] J.L. Guermond, P. Minev, and J. Shen. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 195(44-47):6011 – 6045, 2006.

[69] J.L. Guermond and L. Quartapelle. On stability and convergence of projection methods based on pressure Poisson equation. *International Journal of Numerical Methods in Fluids*, 26:1039–1053, 1998.

[70] J.L. Guermond and J. Shen. A new class of truly consistent splitting schemes for incompressible flows. *Journal of Computational Physics*, 192:262–276, November 2003.

[71] J.L. Guermond and J. Shen. Velocity-correction projection methods for incompressible flows. *SIAM Journal of Numerical Analysis*, 41:112–134, January 2003.

[72] A. Hansbo and P. Hansbo. A finite element method for the simulation of strong and weak discontinuities in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 193(33-35):3523 – 3540, 2004.

[73] M.O.L. Hansen and D.N. Sorensen. CFD model for vertical axis wind turbine. In *Proceedings of the European Wind Energy Conference, Copenhagen, Denmark*, 2001.

[74] R. Hartmann. Numerical analysis of higher order discontinuous Galerkin finite element methods. In H. Deconinck, editor, *VKI LS 2008-08: CFD - ADIGMA course on very high order discretization methods, Oct. 13-17, 2008*. Von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium, 2008.

[75] R. Hartmann, J. Held, and T. Leicht. Adjoint-based error estimation and adaptive mesh refinement for the RANS and k-ω turbulence model equations. *Journal of Computational Physics*, 230(11):4268–4284, 2011.

[76] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *Journal of Computational Physics*, 183:508–532, December 2002.

[77] J.S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods – algorithms, analysis, and applications*. Springer, 2008.

[78] C.W. Hirt, A.A. Amsden, and J.L. Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14(3):227 – 253, 1974.

[79] Y. Hoarau, M. Braza, Y. Ventikos, D. Faghani, and G. Tzabiras. Organized modes and the three-dimensional transition to turbulence in the incompressible flow around a NACA0012 wing. *Journal of Fluid Mechanics*, 496:63–72, 2003.

[80] K.A. Hoffmann and S.T. Chiang. *Computational fluid dynamics volume I*. Engineering Education System, fourth edition edition, 2001.

[81] K.A. Hoffmann and S.T. Chiang. *Computational fluid dynamics volume II*. Engineering Education System, fourth edition edition, 2001.

[82] K. Horiuchi, I. Ushiyama, and K. Seki. Straight wing vertical axis wind turbines: a flow analysis. *Wind Engineering*, 29(3):243252, 2005.

[83] G.T. Houlsby, S. Draper, and M.L.G Oldefield. Application of linear momentum actuator disc theory to open channel flows. Technical report, Oxford University Enginnering Library Report, 2008.

[84] R. Howell, N. Qin, J. Edwards, and N. Durrani. Wind tunnel and numerical study of a small vertical axis wind turbine. *Renewable Energy*, 35(2):412 – 422, 2010.

[85] Ning Hu, Xian-Zhong Guo, and I. Norman Katz. Bounds for eigenvalues and condition numbers in the p-version of the finite element method. *Mathematics of Computation*, 67(224):1423–1450, 1998.

[86] T.J.R. Hughes, G.R. Feijao, L. Mazzei, and J.B. Quincy. The variational multiscale method: a paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166(1-2):3 – 24, 1998.

[87] T.J.R. Hughes, W.K. Liu, and T.K. Zimmermann. Lagrangian-Eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 29(3):329 – 349, 1981.

[88] T.J.R. Hughes, G. Scovazzi, and L.P. Franca. *Multiscale and stabilized methods.* Encyclopedia of Computational Mechanics. John Wiley & Sons, Ltd, 2004.

[89] J. R. Humphrey, D.K. Price, K.E. Spagnoli, A.L. Paolini, and E.J. Kelmelis. CULA: hybrid GPU accelerated linear algebra routines. volume 7705, 2010.

[90] I. Hwang, S. Min, I. Jeong, Y. Lee, and S. Kim. Efficiency improvement of a new vertical axis wind turbine by individual active control of blade motion. In *Proceedings of the SPIE*, volume 6173, 2006.

[91] S.R. Idelsohn and E. Oñate. Finite volumes and finite elements: two good friends. *International Journal for Numerical Methods in Engineering*, 37(19):3323–3341, 1994.

[92] IEA. *International Energy Agency, World Energy Outlook 2008.*

[93] K. Ishimatsu, K. Kage, and T. Okubayashi. Numerical trial for Darrieus-type alternating flow turbine. *Flux, The International Society of Offshore and Polar Engineers*, 3:568–574, 2002.

[94] M. Islam, D.S.K. Ting, and A. Fartaj. Aerodynamic models for Darrieus-type straight-bladed vertical axis wind turbines. *Renewable and Sustainable Energy Reviews*, 12(4):1087 – 1109, 2008.

[95] P.B. Johnson, G.I. Gretton, and T. McCombers. Numerical modelling of cross-flow turines: a direct comparison of four prediction techniques. In *3rd International Conference on Ocean Energy*, October 2010.

[96] G.S. Karamanos. *Large eddy simulation using unstructured spectral h/p elements.* PhD thesis, Imperial College, London, 1999.

[97] G.S. Karamanos and G.E. Karniadakis. A spectral vanishing viscosity method for large eddy simulations. *Journal of Computational Physics*, 163(1):22 – 50, 2000.

[98] G.S. Karamanos and S.J. Sherwin. A high order splitting scheme for the Navier–Stokes equations with variable viscosity. *Applied Numerical Mathematics*, 33(1-4):455–462, May 2000.

[99] G.E. Karniadakis. Spectral element-Fourier methods for incompressible turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 80(1-3):367 – 380, 1990.

[100] G.E. Karniadakis, M. Israeli, and S.A. Orszag. High-order splitting methods for incompressible Navier–Stokes equations. *Journal of Computational Physics*, 97:414–443, 1991.

[101] G.E. Karniadakis and S.J. Sherwin. *Spectral h/p element methods for computational fluid dynamics*. Oxford Science Publications, 2005.

[102] M. Kawaguti. Numerical solution of the Navier–Stokes equations for the flow around a circular cylinder at Reynolds number 40. *Journal of Physical Society of Japan*, 8:747–757, 1953.

[103] K.M. Kelkar and S.V. Patankar. Numerical prediction of vortex shedding behind a square cylinder. *International Journal of Numerical Methods in Fluids*, 14(3):327–341, 1992.

[104] R.M. Kirby and G.E. Karniadakis. Coarse resolution turbulence simulations with spectral vanishing viscosity–large eddy simulations (SVV-LES). *Journal of Fluids Engineering*, 124(4):886–891, 2002.

[105] R.M. Kirby and G.E. Karniadakis. De-aliasing on non-uniform grids: algorithms and applications. *Journal of Computational Physics*, 191:249–264, 2003.

[106] R.M. Kirby and S.J. Sherwin. Aliasing errors due to quadratic non-linearities on triangular spectral h/p element discretisations. *Journal of Engineering Mathematics*, 56(3):273–288, 2006.

[107] R.M. Kirby and S.J. Sherwin. Stabilisation of spectral h/p element methods through spectral vanishing viscosity: application to fluid mechanics modelling. *Computer Methods in Applied Mechanics and Engineering*, 195(23-24):3128 – 3144, 2006.

[108] C.M. Klaij, J.J.W. van der Vegt, and H. van der Ven. Space-time discontinuous Galerkin method for the compressible Navier–Stokes equations. *Journal of Computational Physics*, 217(2):589 – 611, 2006.

[109] V.R. Klaptocz, G.W. Rawlings, Y. Nabavi, M. Alidadi, Y. Li, and S.M. Calisal. Numerical and experimental investigation of a ducted vertical axis tidal current turbine. In *Proceedings of the 7th European Wave and Tidal Energy conference,Porto, Portugal*, 2007.

[110] F.W. Lanchester. A contribution to the theory of propulsion and the screw propeller. *Journal of the American Society for Naval Engineers*, 27(2):509–510, 1915.

[111] B. Landmann, M. Kessler, S. Wagner, and E. Krämer. A parallel high-order discontinuous Galerkin code for laminar and turbulent flows. *Computers & Fluids*, 37(4):427 – 438, 2008.

[112] J.G. Leishman and T.S. Beddoes. A semi-empirical model for dynamic stall. *Journal of the American Helicopter Society*, 34(3):3–17, 1989.

[113] B.Q. Li. *Discontinuous finite elements in fluid dynamics and heat transfer*. Springer, 2006.

[114] J.L. Lions. Problèmes aux limites non homogèenes à donées irrégulières: Une méthode d'approximation. *Numerical Analysis of Partial Differential Equations*, pages 283–292, C.I.M.E. 2 Ciclo, Ispra, 1967.

[115] I. Lomtev, R.M. Kirby, and G.E. Karniadakis. A discontinuous Galerkin ALE method for compressible viscous flows in moving domains. *Journal of Computational Physics*, 155(1):128 – 159, 1999.

[116] X. Ma, G.S. Karamanos, and G.E. Karniadakis. Dynamics and low-dimensionality of a turbulent near wake. *Journal of Fluid Mechanics*, 410:29–65, 2000.

[117] Y. Maday, S.M. Ould Kaber, and E. Tadmor. Legendre pseudospectral viscosity method for nonlinear conservation laws. *SIAM Journal on Numerical Analysis*, 30(2):321–342, 1993.

[118] Reference Manual Math Kernel Library. Intel, 2008.

[119] R. McAdam. *Studies into the technical feasibility of the tranverse horizontal axis water turbine*. PhD thesis, University of Oxford, 2011.

[120] K. McLaren, S. Tullis, and S. Ziada. Computational fluid dynamics simulation of the aerodynamics of a high solidity, small-scale vertical axis wind turbine. *Wind Energy*, 2011.

[121] P. Moin and K. Mahesh. Direct Numerical Simulation: a tool in turbulence research. *Annual Review of Fluid Mechanics*, 30:539–578, 1998.

[122] A. Montlaur, S. Fernandez-Mendez, and A. Huerta. Discontinuous Galerkin methods for the Stokes equations using divergence-free approximations. *International Journal for Numerical Methods in Fluids*, 57(9):1071–1092, 2008.

[123] B.G. Newman. Actuator-disc theory for vertical-axis wind turbines. *Journal of Wind Engineering and Industrial Aerodynamics*, 15(1-3):347 – 355, 1983.

[124] N.C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 230(4):1147 – 1170, 2011.

[125] N.C. Nguyen, P.O. Persson, and J. Peraire. RANS solutions using high order discontinuous Galerkin methods. *45th AIAA Aerospace Science Meeting and Exhibit, Reno, Nevada*, 2007.

[126] V.T. Nguyen. An arbitrary Lagrangian-Eulerian discontinuous Galerkin method for simulations of flows over variable geometries. *Journal of Fluids and Structures*, 26(2):312 – 329, 2010.

[127] J.A. Nitsche. Über ein variationsprinzip zur lösung dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unteworfen sind, abh. *Mathematischen Seminar der Universität Hamburg*, 36, 1971.

[128] C. Norberg. Effect of Reynolds number and low-intensity freestream turbulence on the flow around a circular cylinder. Technical Report 87-2, Department of Applied Thermodynamics and Fluid Mechanics, Chalmer University of Technology, Sweden, 1987.

[129] C. Norberg. Fluctuating lift on a circular cylinder: review and new measurements. *Journal of Fluids and Structures*, 17(1):57 – 96, 2003.

[130] J.T. Oden, I. Babuška, and C.E. Baumann. A discontinuous hp finite element method for diffusion problems. *Journal of Computational Physics*, 146(2):491–519, 1998.

[131] A. Okajima. Strouhal numbers of rectangular cylinders. *Journal of Fluid Mechanics*, 123:379–398, 1982.

[132] T.A. Oliver and D.L. Darmofal. An unsteady adaptation algorithm for discontinuous Galerkin discretizations of the RANS equations. *18th AIAA Computational Fluid Dynamics Conference, Reno, Nevada*, 2007.

[133] United Nations Framework Convention on Climate Change. Kyoto protocol: Status of ratification. Technical report, January 2009.

[134] I. Paraschivoiu. *Wind turbine design with emphasis on Darrieus concept*. Polytechnic International Press, Montreal, Quebec, Canada, 2002.

[135] R. Pasquetti. Spectral vanishing viscosity method for LES: sensitivity to the SVV control parameters. *Journal of Turbulence*, 6, 2005.

[136] A.T. Patera. A spectral element method for fluid dynamics: laminar flow in a channel expansion. *Journal of Computational Physics*, 54(3):468 – 488, 1984.

[137] J. Peiro and S.J. Sherwin. *Handbook of materials: volume 1: methods and models*, chapter Finite difference, finite element and finite volume methods for partial differential equations, pages 1–32. Springer, 2005.

[138] P.O. Persson, J. Peraire, and J. Bonet. Discontinuous Galerkin solution of the Navier–Stokes equations on deformable domains. In 45th AIAA Aerospace Science Meeting, editor, *AIAA Conference*, 2007.

[139] D.A. Di Pietro and A. Ern. Mathematical aspects of discontinuous Galerkin methods, 2012.

[140] United Nations Environment Programme. Industrialized countries to cut greenhouse gas emissions by 5.2%. Technical report, December 1997.

[141] QinetiQ. Cycloidal tidal power generation: Phase II. Technical Report URN 04/1102, Report for the DTI, 2004.

[142] S. Ramakrishnan and S.S. Collis. Multiscale modeling for turbulence simulation in complex geometries. In *Forty-second AIAA Aerospace Sciences Meeting and Exhibit Conference and Exhibit,Reno, NV*, January 2004.

[143] W.J. Rankine. *Transactions, Institute of Naval Architects*, 6(13), 1865.

[144] H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[145] S. Rhebergen and B. Cockburn. A space-time hybridizable discontinuous Galerkin method for incompressible flows on deforming domains. *Journal of Computational Physics*, 223(11):4185–4204, 2012.

[146] B. Riviere. *Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.

[147] B. Rivire, M.F. Wheeler, and V. Girault. Part I: Improved energy estimates for interior penalty, constrained and discontinuous Galerkin methods for elliptic problems. *Computational Geosciences*, 3(3-4):337–360, december 1999.

[148] J. Robichaux, S. Balachandar, and S.P. Vanka. Three-dimensional floquet instability of the wake of square cylinder. *Physics of Fluids*, 11(3):560–578, 1999.

[149] P. Sagaut. *Large eddy simulation for incompressible flows: an introduction*. Scientific computation. Springer, 2001.

[150] P. Sagaut, S. Deck, and M. Terracol. *Multiscale and multiresolution approaches in turbulence*. Imperial College, 2006.

[151] H. Schlichting. *Boundary layer theory*. McGraw-Hill, New-York, 1979.

[152] K. Sengupta and F. Mashayek. Large-eddy simulation using a discontinuous Galerkin spectral element method. In AIAA-2007-402, editor, *Forty-fifth AIAA Aerospace Sciences Meeting and Exhibit,Reno, NV*, January 2004.

[153] R. Sevilla and S. Fernández-Méndez. Numerical integration over 2D NURBS-shaped domains with applications to NURBS-enhanced FEM. *Finite Elements in Analysis and Design*, 47(10):1209–1220, 2011.

[154] R. Sevilla, S. Fernandez-Mendez, and A. Huerta. NURBS-enhanced finite element method for Euler equations. *International Journal of Numerical Methods in Fluids*, 00:1–28, 2007.

[155] K. Shahbazi. An explicit expression for the penalty parameter of the interior penalty method. *Journal of Computational Physics*, 205(2):401–407, 2005.

[156] K. Shahbazi, P.F. Fischer, and C.R. Ethier. A high-order discontinuous Galerkin method for the unsteady incompressible Navier–Stokes equations. *Journal of Computational Physics*, 222(1):391 – 407, 2007.

[157] J. Shen. Pseudo-compressibility methods for the unsteady incompressible Navier–Stokes equations. In Guo B, editor, *Proceedings of the 1994 Beijing Symposium on Nonlinear Evolution Equations and Infinite Dynamical Systems*, pages 68–78. ZhongShan University Press, 1997.

[158] J. Smagorinsky. General circulation experiments with the primitive equations. *Monthly Weather Review*, 91(3):99–164, 1963.

[159] J.H. Strickland. Darrieus turbine: a performance prediction model using multiple streamtubes. Technical Report Report No. SAND-75-0431, Sandia Labs, 1975.

[160] S. Sunada, A. Sakaguchi, and K. Kawachi. Airfoil section characteristics at a low Reynolds number. *Journal of Fluids Engineering*, 119(1):129–135, 1997.

[161] B. Szabó and I. Babuška. *Finite element analysis*. Wiley, New York, 1991.

[162] E. Tadmor. Convergence of spectral methods for nonlinear conservation laws. *SIAM Journal of Numerical Analysis*, 26:30–44, 1989.

[163] Tecplot.Inc. 2009. www.tecplot.com.

[164] A.R. Temam. Une méthode d'approximation de la solution des équations de Navier–Stokes. *Bulletin de la Société Mathématique de France*, 96:115–152, 1968.

[165] R.J. Templin. Aerodynamic performance theory for the NRC vertical-axis wind turbine. *NASA STI/Recon Technical Report N*, 76, 1974.

[166] A. Thom. The flow past circular cylinders at low speeds. *Proceedings of the Royal Sociaty London*, A141:651–666, 1933.

[167] L.N. Trefethen. *Spectral methods in MatLab*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[168] A. Uranga, P.O. Persson, M. Drela, and J. Peraire. Implicit large eddy simulation of transition to turbulence at low Reynolds numbers using a discontinuous Galerkin method. *International Journal for Numerical Methods in Engineering*, 87(1-5), 2011.

[169] J.J.W. van der Vegt and Y. Xu. Space-time discontinuous Galerkin method for nonlinear water waves. *Journal of Computational Physics*, 224(1):17 – 39, 2007.

[170] J.C. Vassberg, A.K. Gopinath, and A. Jameson. Revisiting the vertical-axis wind-turbine design using advanced computational fluid dynamics. *AIAA Paper*, 2005.

[171] L.J. Vermeer, J.N. Sørensen, and A. Crespo. Wind turbine wake aerodynamics. *Progress in Aerospace Sciences*, 39(6-7):467 – 510, 2003.

[172] H.K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Longman Scientific & Technical, 1995.

[173] P.E.J. Vos, S.J. Sherwin, and R.M. Kirby. From h to p efficiently: Implementing finite and spectral h/p element methods to achieve optimal performance for low- and high-order discretisations. *Journal of Computational Physics*, 229(13):5161 – 5181, 2010.

[174] D. Wang and S.J. Ruuth. Variable step-size implicit-explicit linear multistep methods for time-dependent partial differential equations. *Journal of Computational Mathematics*, 26(6):838855, 2008.

[175] T. Warburton. *Spectral h/p methods on polymorphic multi–domains: algorithms and applications*. PhD thesis, Brown University, 1999.

[176] M.F. Wheeler. An elliptic collocation-finite element method with interior penalties. *SIAM Journal of Numerical Analysis*, 15:152–161, 1978.

[177] M.F. Wheeler. An elliptic collocation-finite element method with interior penalties. *SIAM Journal on Numerical Analysis*, 15(1):152–161, 1978.

[178] J. Whelan, M. Thomson, J.M.R. Graham, and J. Peiró. Modelling of free surface proximity and wave induced velocities around a horizontal axis tidal stream turbine. In *7th European Wave and Tidal Energy Conference*, Porto, Portugal, September 2007.

[179] D.C. Wilcox. *Turbulence modeling for CFD*. DCW Industries Inc., La Cañada, CA, 1993.

[180] C.H.K. Williamson. Vortex dynamics in the cylinder wake. *Annual review of fluid mechanics* , 28:477–539, 1996.

[181] R.E. Wilson and P.B.S. Lissaman. Applied aerodynamics of wind power machines. *NASA STI/Recon Technical Report*, 75:22669, July 1974.

[182] www.doxygen.org.

[183] www.keplerenergy.co.uk.

[184] www.top500.org.

[185] Z. Xu, D. Stanescu, and J.W. Naughton. Development of a spectral element DNS/LES method for turbulent flow simulations. In *Fifth Joint ASME/JSME Fluids Engineering Conference, San Diego, CA*, August 2007.

[186] T.G. Zaki, M.S., and M. Gad-El-Hak. Numerical and experimental investigation of flow past a freely rotatable square cylinder. *Journal of Fluids and Structures*, 8(7):555 – 582, 1994.

[187] M.M. Zdravkovich. *Flow around circular cylinders: fundamentals*. Oxford Science Publications. Oxford University Press, 1997.

[188] Qingdao Zeyu. Wind power generator co. ltd. http://www.windpowerzeyu.com.

[189] Y. Zhou, M. Mahbub Alam, H.X. Yang, H. Guo, and D.H. Wood. Fluid forces on a very low Reynolds number airfoil and their prediction. *International Journal of Heat and Fluid Flow*, 32(1):329 – 339, 2011.

[190] O.C. Zienkiewicz and K. Morgan. *Finite elements and approximation*. John Wiley & Sons Inc, 1983.

[191] O.C. Zienkiewicz, R.L. Taylor, and P. Nithiarasu. *The finite element method for fluid dynamics*. Elsevier Butterworth-Heinemann, 2005.

[192] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. *The finite element method : its basis and fundamentals*. Elsevier Butterworth-Heinemann, 6th edition edition, 2005.