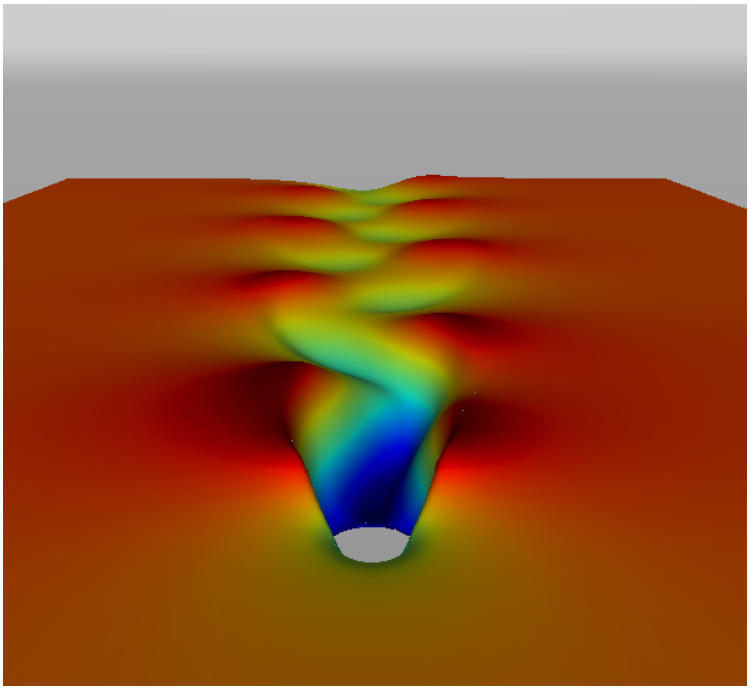
Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Energy and Process Engineering

Tobias Winter

Higher Order Method for Flow Simulations in Complex Geometries

Master's thesis in MTPROD
Supervisor: Bernhard Müller
June 2023



NTNU
Norwegian University of
Science and Technology

Tobias Winter

Higher Order Method for Flow Simulations in Complex Geometries

Master's thesis in MTPROD
Supervisor: Bernhard Müller
June 2023

Norwegian University of Science and Technology
Faculty of Engineering
Department of Energy and Process Engineering

NTNU
Norwegian University of
Science and Technology

Abstract

The increasing capabilities of higher order methods used in computational fluid dynamics enables the analysis of previously little understood phenomena. One such case is the intricate flow mechanism causing obstructive sleep apnea. In working towards a better understanding of the flow mechanism, a arbitrarily high order Discontinuous-Galerkin Immersed boundary method has been developed for the 2D compressible Navier-Stokes equations. The arbitrarily high order is achieved using a novel approach to the construction of basis functions. The method is tested on the case of a flow over a circular cylinder. Results are obtained which indicate that further investigation is needed, but that the method is indeed capable of qualitatively reproducing the expected behaviour.

Acknowledgements

I want to thank Bernhard Müller, my academic supervisor. I was led to believe that i was lacking in my capabilities by the institutions of higher education. My neurodivergent mind was simply incompatible with the archaic way in which examinations were done. Then i met you. You listened to my ramblings about mesh generation and Navier-Stokes solvers and provided me with well thought out answers to my unending stream of questions. You took me seriously and for that I'm grateful. It has been a pleasure to work with you.

I also want to thank my family, including Maša Paunov and Laila Mjelde, for their love and support. I am especially grateful for them believing in me the times when I couldn't muster do so myself.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

CFD	Computational Fluid Dynamics
DDG	Direct Discontinuous-Galerkin
DG	Discontinuous-Galerkin
ENO	Essentially Non-Oscillatory
GC	Ghost Cell IBM
IBM	Immersed Boundary Method
LDG	Local Discontinuous-Galerkin
RBF	Radial Basis Functions
WENO	Weighted Essentially Non-Oscillatory

Symbols

$\boldsymbol{\ell}$	Lagrange basis
γ	Adiabatic index
τ	Viscous stress tensor
$\mathbf{a}$	Polynomial basis
$\mathbf{b}$	Monomial basis
$\mathbf{F}$	Flux
$\mathbf{p}$	Legendre basis
$\mathbf{U}$	Conserved variables

$\mathbf{u}$	Velocity
$\mathcal{R}$	Gas constant of air
$\otimes$	Kronecker product
ρ	Density
A	Coefficient matrix of polynomial basis $\mathbf{a}$
C_{CFL}	Convective CFL number
C_{DFL}	Diffusive CFL number
D	Differentiation matrix
E	Specific total energy
$G_{\mathbf{a}}$	Gram matrix of polynomial basis $\mathbf{a}$
H	Specific total enthalpy
p	Pressure
T	Temperature
u	First velocity component
V	Vandermonde matrix
v	Second velocity component

Chapter 1

Introduction

1.1 Motivation

The field of computational fluid dynamics have undergone rapid development during the last decades and has now matured to the point where it has become an integral part of research and industries such as weather forecasting, aviation, and process engineering. These advancements, as well as advancements in computational infrastructure, means that the envelope of possible use cases of CFD is expanding. The research project VirtuOSA[49], which this thesis is a part of, aims to apply CFD to one such use case. Namely, to use CFD to better understand the complex flow phenomena occurring in the upper airways of patients affected by the sleep disorder obstructive sleep apnea. The tools developed as a part of the research project is intended to aid clinicians in providing patient specific diagnostics and treatment. The objective is for this to result in an improved prognosis for the patient.

Multiple interrelated mechanisms and phenomena seem to be responsible for the onset of obstructive sleep apnea. An extensive overview of these can be found in the works of Moxness [39]. Moxness identified the complex geometry of the human upper airways and its effects on, and interaction with, the air flow as a point in need of better understanding. In an advancement towards the goal of providing such an understanding, the aim of this thesis is to develop a method for high accuracy flow simulations in complex geometries. Several objectives were formulated by me and by supervisor Bernhard Müller[40].

1. Review the literature for higher order discretization methods like the discontinuous Galerkin method and other higher order methods

2. Investigate a higher order method for flow in complex geometries

3. Develop, implement and test a higher order method for the 2D compressible Navier-Stokes equations in complex geometries

4. Verify and validate the higher order method for benchmark test cases

1.2 State of the art

Any graduate student taking an introductory class in CFD is likely to encounter various first- and even second order schemes. However, higher order methods, which in the CFD community is regarded as any method of order 3 or higher[51], are seldomly explored in much detail except for perhaps a passing remark acknowledging their existence. While the basics must absolutely come first, this do in fact mean that many people's knowledge of CFD ends at second order. These people then go on to jobs in the various industries that use CFD, and take with them a bias towards using lower order methods. This affinity for lower order methods as well as increased computational capabilities have lead to a situation where large parts of various industries opt for a "brute-force"-strategy when employing CFD. Thus, higher-order methods have to some extent been relegated to a niche research community. Of course, the people involved with higher-order methods cannot be absolved completely of their effect on the extent of their adoption either. The field do seem quite daunting for those uninitiated due to its heavy use of mathematics, notation, and comprehensive analysis. All of these factors have lead to the proliferation of some misconceptions regarding higher-order methods.

1. They are computationally expensive.

2. The level of accuracy obtained by using higher order methods are beyond the current demands of most industrial applications.

3. They are complicated and the added coding effort of implementing them means that their not worthwhile.

4. They are less robust than lower order methods.

While it is true that higher-order methods generally require more work per mesh element, this varies depending on what higher-order method you use[10]. However, computational cost alone is an insufficient measure when assessing a method. This is where the order of accuracy, which has hitherto been referred to as simply the order, becomes relevant. Given a mesh with a characteristic mesh size h on which some method is used to find an approximate solution u_h to the exact solution u, then we determine the error norm

as

$$e_h = \|u - u_h\|. \tag{1.1}$$

The order of accuracy p is then determined by the behaviour of e_h w.r.t. changes in h. That is

$$e_h \propto h^p. \tag{1.2}$$

Of course, p is rarely constant across the entire range of possible values of h, however, meaningful estimates of its asymptotic behaviour can still be derived. For further details, the reader is referred to [50]. Lets define the characteristic number of mesh units as

$$n = \frac{1}{h}. \tag{1.3}$$

Let us also introduce the cost per mesh unit c. The cost c_h of obtaining u_h on a mesh with d spatial dimensions is therefore

$$c_h = cn^d = c. \tag{1.4}$$

Note that this implies that

$$\frac{c_{h,HO}}{c_{h,LO}} = \frac{c_{HO}}{c_{LO}}, \tag{1.5}$$

i.e., the ratio of the computational cost of the higher order method to that of the lower order method on the same mesh remains constant w.r.t h. However, the error norm does not. In fact, on a two-dimensional mesh, halving the characteristic mesh size results in four times the total computational cost. The error of a first order method would on this mesh be reduced by a factor of 4, while the error of a fourth order method is reduced by a factor of 16. This illustrative example shows that by decreasing the mesh size, higher order methods quickly outpaces lower order ones in terms of error. Much more accurate solutions can therefore be obtained with less refined meshes. So, in response to misconception 1, higher order methods might be more expensive per mesh unit, but they require a lot fewer mesh units to reach comparable levels of accuracy than lower order methods. Thus, the CPU-time spent on reaching the same solution is likely to be lower for a higher order method. This is demonstrated well eloquently in [10].

Misconception 2 is also just that, a misconception. Low-order methods combined with turbulence models can yield modestly accurate estimates of key quantities within reasonable time frames. However, there are several cases for which they simply do not suffice. As pointed out by [51], a 5% error in one solution variable might yield a 20% error in a derived quantity. This is

especially critical wherever you have flow behaviour that depends on highly detailed structures such as vortexes or acoustic waves[51]. An example of such a flow is the flow over a helicopter. The aerodynamic forces acting on the body of the helicopter is highly dependent on the vortices produced at the wing tips. It of paramount importance that these vortexes are not erroneously dissipated before they reach the helicopter body. The solution then becomes highly dependent on the ability of the method to accurately resolve finer flow structures. For the reasons explained in regards to misconception 1, resolving these flow structures accurately quickly becomes infeasible with lower-order methods.

This leads quite readily into misconception 3. While most higher order methods are indeed more complicated and requires some added coding effort, the upfront cost is quickly recuperated in terms of the efficiency of the method once implemented. Summarized, although a pair of scissors are much cheaper than a lawn mower, one would be wise to chose the latter if one wishes to mow a lawn of any discernible size.

The last misconception is dependent on what higher order method is in question. A more thorough discussion of the distinct methods can be found later in this chapter, or in [10]. The statement that higher order methods are less robust need not necessarily be a problem. Although lower order methods might be able to produce a solution for a wider range of inputs, this comes at the cost of the accuracy of those solutions. Essentially, what good is a solution if it is wrong.

Now that some reasoning to why one would want to use a higher order method has been given, one also needs to decide which method to use. In the context of hyperbolic conservation laws, such as the compressible Navier-Stokes, there are two classes of higher order methods which are of particular prominence within the CFD community. Namely the ENO/WENO schemes and the Discontinuous-Galerkin method[1, 10, 9]. Therefore, these will be the focus of the ensuing discussion of the present text. However, one should not infer from this that there doesn't exist other higher-order methods.

1.2.1 ENO/WENO

In order to make the Godunov's scheme[24] second order accurate in space, van Leer developed the MUSCL scheme in his 1979 paper [48]. The idea behind the MUSCL scheme was to approximate the state within each finite volume as linear, rather than constant as in the classical Godunov's scheme. As a consequence a wider stencil is required in order to accurately reconstruct the linear slope within the cell. A finite difference or finite volume scheme employing a large fixed stencil is able to provide high accuracy as long as

the solution is smooth. Unfortunately, using such a stencil when solving a hyperbolic equation leaves the scheme susceptible to spurious oscillations near discontinuities or shocks[45]. In the early days of higher order methods, there existed mainly two ways of dealing with such oscillations. The first approach was to add artificial viscosity near discontinuities such that the oscillations could be suppressed. The second approach was to apply limiters intended to reduce the order near discontinuities. Both of these approaches required carefull tuning, often being quite dependent on the specific problem at hand. Then the MUSCL scheme was extended to arbitrary order by Harten et. al. in their 1986 paper[25]. In contrast to the large fixed stencils of the previously discussed traditional finite difference methods, the scheme used an adaptive stencil. This lead to the scheme being able to be highly accurate in smooth regions while avoiding spurious oscillations at discontinuities. This property ended up being the namesake of this class of schemes, which would be named Essentially Non-oscillatory Schemes, or ENO schemes for short. The ENO schemes utilises one out of several candidate stencils based on some smoothness criterion [10]. Thus, stencils which would not introduce spurious oscillations to reconstruction can be chosen over stencils that would. This idea is then developed further in the 1994 paper by Liu et. al. in 1994 [36] in which the weighted ENO scheme, or WENO, is introduced. Instead of choosing only one out of multiple candidate stencils such as with the ENO schemes, the WENO schemes utilise a weighted convex combination of all candidate stencils. The essentially non-oscillatory property is obtained by having the weights depend on the smoothness of the respective stencils such that stencils affected by discontinuities essentially have no contribution to the final convex combination. The resulting reconstruction lead to an increase in the order of accuracy of one compared to that of the ENO schemes.

ENO-/WENO schemes can be formulated as both finite difference schemes and as finite volume schemes [45]. These two formulations have their own sets of advantages and disadvantages. Noteworthy of the FDM formulation of the WENO schemes is

1. Low computational cost compared to other higher-order methods.[10]

2. They are rather Straightforward to implementation.[9]

3. They require a uniform or sufficiently smooth curvilinear mesh[45]. Ideally, the mesh needs to be of the same smoothness as the reconstruction if the desired order is to be achieved[9].

The FVM formulation has one major advantage over the FDM formulation. Namely that the FVM formulation is capable of being applied to arbitrary

triangulations, making them significantly more flexible than the FDM formulation[9]. However, this comes at a cost. Some properties of the FVM formulation of the WENO schemes is summarized below.

1. They can solved on arbitrary triangulations[9][45].

2. They are more complicated to implement and have a higher computational cost than their finite difference counterparts when solving higher dimensional problems[10].

The ENO- and WENO schemes have been of continued research interest within the CFD community. Attempts have been made to make them more compact through the development of methods such as the Central-WENO (CWENO) described in [19, 47]. Although CWENO schemes have been shown to be more robust and compact than their non-central counterparts, the crux of the matter is that these methods all require moderately large stencils due to their very nature. This is in stark contrast to the highly local Discontinuous-Galerkin method which is to be discussed next.

1.2.2 Discontinuous-Galerkin method

The Discontinuous-Galerkin method was first introduced in the seminal work of Hill and Reed (1973) [43] where they used the method as a means to solve the neutron transport equations on triangular grids. The method was based on the Galerkin FEM method where continuous operators are discretized using a weak formulation w.r.t some trial space. Instead of using a continuous trial space, Reed and Hill used test functions which were continuous within cells, but which could be discontinuous across cell interfaces. This lead to the global mass matrix becoming diagonal and thus trivially invertible. It is for this reason that the DG-method shares several features in common with traditional finite volume methods.

One of these features is that explicit time discretizations can be used to advance in time, thus circumventing the difficulties of implicit time integration. Chavent and Salzano developed such a method in their 1982 paper [7]. A one-dimensional scalar conservation law was discretized using piecewise linear functions for the spatial discretization and explicit Euler for the temporal discretization. Unfortunately, this discretization proved to have rather poor stability properties[12]. This scheme was later improved upon by Cockburn and Chavent in 1989[8] when they introduced suitable slope limiters to the scheme. Further, these slope limiters were generalized to higher order by Cockburn and Shu in 1989[16], discretizing using higher order polynomials

and Runge-Kutta time stepping. This scheme would be named the Runge-Kutta DG method, or RKDG. These extensions proved to make the scheme capable of simulating hyperbolic scalar conservation laws with a high order of accuracy in smooth regions while avoiding spurious oscillations near discontinuities[12]. An extension of the scheme to one dimensional hyperbolic systems followed in 1989 by Cockburn, Lin, and Shu[13], to multiple dimensions in 1990 by Cockburn, Hou, and Shu [11], and ultimately to hyperbolic systems in multiple dimensions in 1998 by Cockburn and Shu [15]. While the RKDG was being developed for hyperbolic systems, Bassi and Rebay were in 1997[2] already on their way to extend the method to work with convection diffusion equations such as the Navier-Stokes equations. They introduced the Local DG method (LDG), where the Navier-Stokes equations are rewritten as a first order system by having the solution gradients occupy the same function space as the solution. Since then, numerous DG methods have been developed. The reader is referred to [12, 14, 26] for more details. Some properties of the DG method is summarized below.

1. It has great parallel efficiency, meaning that computational times can be reduced by using architectures capable of parallel computing such as multi-core CPUs or GPUs.[12]

2. It is local, meaning that any finite cell need only communicate with cells with which it shares an interface.

3. Care must be taken when using the method for which shocks and other discontinuities are present.

1.2.3 Immersed Boundary

As their namesake would suggest, higher order methods are capable of achieving high solution accuracy. That is, as long as the problem geometry can be discretized accurately. One approach is to discretize the spatial domain using a mesh consisting of higher order curved elements. However, high order mesh generation of complex geometries are considered time consuming, insufficiently robust, and in general somewhat of a bottleneck inn regards to the use of higher order methods[50].

A second approach is to use a class of methods known as immersed boundary methods. The fundamental idea behind the immersed boundary method, virtual boundary method, embedded boundary method, or any other moniker used to describe it, is to fulfill a boundary condition through some means other than having a computational mesh conform to the aforementioned

boundary. In other words, the geometric representation of the problem geometry is decoupled from mesh. The immersed boundary method was first introduced by Peskin in his 1972 paper[41]. In the paper, Peskin outlines a method for simulating the flow around a heart valve on a cartesian grid using a fictitious force field.

The IBM has had a steady increase in popularity since it was introduced by Peskin, and there now exist several variants of the method. Among these methods are the subclass of the so called ghost cell immersed boundary method (GCIBM) developed in the works of Tseng and Ferziger[46], Fedkiw, Aslam, Merriman, and Osher[20], and Ghias, Mittal, Dong[23]. The GCIBM enforces boundary conditions by using so called ghost cells. These ghost cells are cells located in parts of the computational domain enclosed by an immersed boundary where no or different governing equations than the ones in the rest of the domain are solved. They're oftentimes located close to the immersed boundary. Since the solution variables at the ghost cells hold no physical meaning, they need to be explicitly set. The idea behind the GCIBM is to set the values at the ghost points such that a boundary condition is satisfied at the immersed boundary through the numerical method. The GCIBM is capable of recreating sharp interfaces[23], which is beneficial if one wants to recreate complex geometry accurately. Another benefit of the GCIBM is that it can be implemented in such a way that no special treatment of the ghost cells are needed once their values are set. The fact that the ghost cells require no special makes it a lot simpler to parallelize the method.

In recent years, the IBM has been used in combination with higher order numerical methods. The previously discussed ENO scheme was used in conjunction with the GCIBM of [20] and in 2007, Fidkowski and Darmofal developed a Discontinious-Galerkin IBM[21]. Since then, multiple other DG-IBM methods have followed such as the ones described in [34, 3, 32, 53, 52, 22, 31].

1.2.4 Conclusions

This review of higher order methods and boundary treatments is by no means exhaustive, but highlights what was deemed the most promising candidates to achieve the objectives stated in section 1.1 at the time of writing this thesis. Ultimately, the DG-method in combination with the GCIBM was chosen over the ENO/WENO schemes. The locality and parallelizability of the DG-method makes it an attractive scheme to coupled with the GCIBM. It is believed that parallelizability is of key importance in reducing the computational time to the point where patient specific preoperative CFD be-

comes a feasible diagnostic tool. The ENO-/WENO schemes do have some favorable properties especially regarding their ability to handle shocks and discontinuities. However, the flow conditions in the human upper airway are believed to be sufficiently below the limit for which such phenomena would arise. The justification for using the compressible Navier-Stokes in lieu of the incompressible variant, is the strong relation between OSA and acoustic phenomena. A fact anyone sharing a bed with an OSA patient can likely attest to.

1.3 Outline

I chapter 2 the compressible Navier-Stokes equations are presented along with an equation of state and various assumptions. Additionally, some boundary conditions that will later be discretized are given. In chapter 3, a method for constructing basis polynomials is detailed, followed by the construction of a Discontinuous-Galerkin scheme. Then numerical discretizations of the aforementioned boundary conditions are given. Chapter 3 ends on the construction of a ghost point immersed boundary method using radial basis functions. In chapter 4 a test case meant for testing the developed method is presented and the results and discussion of the testing is found in chapter 5. Some concluding remarks regarding the results and some recommendations regarding further work is given in chapter 6 and 7 respectively.

Chapter 2

Governing Equations

2.1 The Compressible Navier-Stokes equations

As this thesis pertains to model aerodynamic flow in complex geometries, the compressible Navier-Stokes equations were chosen as suitable governing equations. They describe the temporal evolution of mass, momentum, and energy. The set of equations is oftentimes accompanied by an equation of state in order to close the system. The following presentation of the compressible Navier-Stokes will be done in two spatial dimensions.

$$\mathbf{U}_t + \nabla \cdot \mathbf{F}(\mathbf{U}, \nabla\mathbf{U}) = \mathbf{0} \tag{2.1}$$

Where $\mathbf{U}$ is the conserved variables and $\nabla \cdot \mathbf{F}$ the divergence of the flux. The subscript t denotes the temporal derivative. The conserved variables are given as

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \tag{2.2}$$

where ρ, u, v, and E are the density, the velocity components, and the specific total energy. The flux is regarded as consisting of a convective term and a viscous term. They are denoted by superscript c and v respectively.

$$\mathbf{F} = \mathbf{F}^c - \mathbf{F}^v \tag{2.3}$$

Note that the dependence of the flux on $\mathbf{U}$ and $\nabla\mathbf{U}$ is hereafter omitted for brevity's sake. The subscripts 1 and 2 are used to denote the spatial

directionality of the flux. The convective flux is given as

$$\mathbf{F}^c_{1,2} = \left(\begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho H u \end{bmatrix}, \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho H v \end{bmatrix} \right) \tag{2.4}$$

and the viscous flux as

$$\mathbf{F}^v_{1,2} = \left(\begin{bmatrix} 0 \\ \tau_{11} \\ \tau_{21} \\ \tau_{11}u + \tau_{12}v + kT_x \end{bmatrix}, \begin{bmatrix} 0 \\ \tau_{12} \\ \tau_{22} \\ \tau_{21}u + \tau_{22}v + kT_y \end{bmatrix} \right), \tag{2.5}$$

where the viscous stress tensor, τ, is given by

$$\tau = \mu \begin{bmatrix} \frac{4}{3}u_x - \frac{2}{3}v_y & u_y + v_x \\ u_y + v_x & \frac{4}{3}v_y - \frac{2}{3}u_x \end{bmatrix}. \tag{2.6}$$

The specific total enthalpy is given by $H \equiv E + \frac{p}{\rho}$ and the specific total energy by $E \equiv e + \frac{1}{2}\|\mathbf{u}\|^2$, where e is the specific internal energy. The ideal gas law is used to close the system by introducing an explicit expression for the pressure

$$p = \rho e(\gamma - 1). \tag{2.7}$$

The specific heat capacities, c_p, c_v, and as such, the adiabatic index $\gamma = c_p/c_v$ is also assumed to be constant. Unless otherwise is specified, a value of $\gamma = 1.4$ it is a good fit for dry air at standard conditions. The ideal gas law is further used to express the quantities of the speed of sound c and the temperature T as

$$c = \sqrt{\frac{\gamma p}{\rho}} \tag{2.8}$$

and

$$T = \frac{\gamma - 1}{\mathcal{R}} e. \tag{2.9}$$

Here, $\mathcal{R}$ is the gas constant of air and is assumed to have a value of $\mathcal{R} = 287.052874 \frac{m^2}{Ks^2}$ unless otherwise is stated.

Using the Reynolds-, Mach, and Prandtl number

$$Re_0 = \frac{\rho_0 u_0 L_0}{\mu_0}, \quad Ma_0 = \frac{u_0}{c_0}, \quad Pr_0 = \frac{\gamma \mathcal{R}}{\gamma - 1} \frac{\mu_0}{k_0}, \tag{2.10}$$

in combination with a reference speed u_0 and a characteristic length scale, a nondimensionalization of the flow variables is formulated.

$$t^* = \frac{u_0}{L_0}t, \quad x^* = \frac{x}{L_0}, \quad y^* = \frac{y}{L_0}, \quad u^* = \frac{u}{u_0}, \quad v^* = \frac{v}{u_0}, \tag{2.11}$$

$$E^* = \frac{E}{u_0^2}, \quad H^* = \frac{H}{u_0^2}, \quad T^* = \frac{\gamma\mathcal{R}}{(\gamma-1)u_0^2}T, \quad p^* = \frac{\gamma M a_0}{\rho_0 u_0^2}p \tag{2.12}$$

$$\mu^* = \frac{Re_0}{L_0}\mu, \quad k^* = \frac{(\gamma-1)Pr_0 Re_0}{\gamma\mathcal{R}L_0}k \tag{2.13}$$

The superscript $*$ will hereafter be omitted and any flow variable should be taken to be nondimensionalized unless information is provided to the contrary.

2.2 Boundary conditions

In addition to the governing equations, boundary conditions are also needed to ensure that a problem is well posed. This section will describe some physical boundary conditions that will later be used for the discretization of the numerical method and test-cases. The governing equations dictates what occurs on some domain Ω with a boundary $\Gamma = \partial\Omega$. The state just inside and outside the boundary Γ is given as $\mathbf{U}_\Gamma^-$ and $\mathbf{U}_\Gamma^+$ respectively. Where the state exactly at the boundary is referenced, the $-$ and $+$ is omitted.

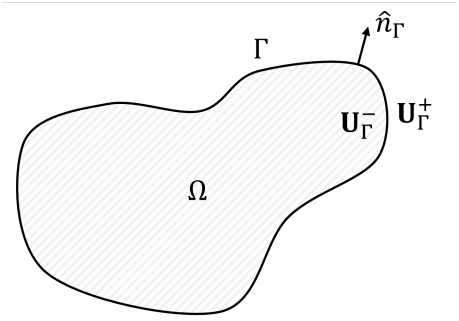


Figure 2.1: Domain and boundary

2.2.1 Free stream

The free stream boundary condition simply presumes that the flow beyond the boundary is at some prescribed state known as the free stream, i.e.

$$\mathbf{U}_\Gamma^+ = \mathbf{U}_\infty. \tag{2.14}$$

2.2.2 No slip

The no slip condition is assumed to apply at solid boundaries. This means that fluid at a solid boundary is assumed to have no velocity relative to the boundary. If the boundary is fixed (which it will be for all test-cases in thesis), then the no slip condition states that

$$\mathbf{u}_\Gamma = \mathbf{0}, \tag{2.15}$$

or in terms of the conserved variables

$$[\rho\mathbf{u}]_\Gamma = \mathbf{0}. \tag{2.16}$$

2.2.3 Periodic

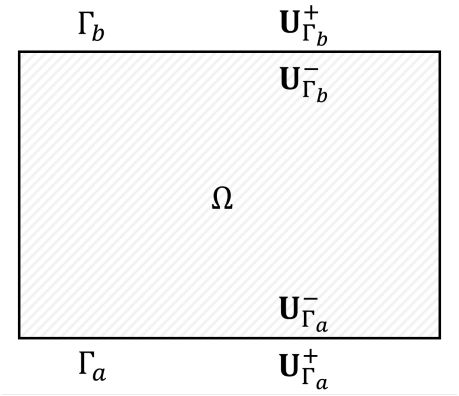


Figure 2.2: A domain with periodic boundaries.

The periodic boundary couples two or more segments of the domain boundary such that they behave as if they were connected. Stated more plainly, what flows into one boundary flows out of the coupled boundary, and vice versa. Two boundary segments, Γ_a and Γ_b, are coupled by

$$\begin{cases} \mathbf{U}^+_{\Gamma_a} = \mathbf{U}^-_{\Gamma_b} \\ \mathbf{U}^+_{\Gamma_b} = \mathbf{U}^-_{\Gamma_a} \end{cases}. \tag{2.17}$$

It is assumed that periodic boundaries are parallel to one another.

2.2.4 Adiabatic

An adiabatic wall is a wall where there is no heat flux. This implies that the following relation holds for the density and the energy.

$$\begin{cases} \frac{\partial \rho}{\partial n} = 0 \\ \frac{\partial E}{\partial n} = 0 \end{cases} \tag{2.18}$$

Chapter 3

Numerical Method

3.1 Construction of basis

3.1.1 Polynomial basis

A polynomial, $a(\eta)$, of order equal to or less than N_p can be regarded as a linear combination of terms from the monomial basis of order N_p.

$$\mathbf{b}(\eta) = \{\eta^i\}_{i=0}^{N_p} \tag{3.1}$$

Therefore a polynomial basis with N_p polynomials $\{a_i\}_{i=0}^{N_p}$ of order equal to or less than N_p can be expressed as a set of dot products between vectors containing the polynomial coefficients, $\{\mathbf{a}_i\}_{i=0}^{N_p}$, and the monomial basis.

$$\begin{aligned}
a_0(\eta) &= \mathbf{a}_0 \cdot \mathbf{b}(\eta) = \Sigma_{i=0}^{N_p} a_{0,i}\eta^i \\
a_1(\eta) &= \mathbf{a}_1 \cdot \mathbf{b}(\eta) = \Sigma_{i=0}^{N_p} a_{1,i}\eta^i \\
a_2(\eta) &= \mathbf{a}_2 \cdot \mathbf{b}(\eta) = \Sigma_{i=0}^{N_p} a_{2,i}\eta^i \\
&\vdots \\
a_{N_p}(\eta) &= \mathbf{a}_{N_p} \cdot \mathbf{b}(\eta) = \Sigma_{i=0}^{N_p} a_{N_p,i}\eta^i
\end{aligned}$$

By introducing a coefficient matrix, the previous operation becomes

$$A = \begin{bmatrix} \mathbf{a}_0^T \\ \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_{N_p}^T \end{bmatrix} \implies \mathbf{a}(\eta) = A\mathbf{b}(\eta) = \begin{bmatrix} a_0(\eta) \\ a_1(\eta) \\ a_2(\eta) \\ \vdots \\ a_{N_p}(\eta) \\ , \end{bmatrix} \tag{3.2}$$

thus, allowing us to express any polynomial basis as the product of a coefficient matrix and a monomial basis.

We define the Gram matrix[1] of some function space w.r.t. the spatial domain Ω as

$$G_{\mathbf{a}} = \int_{\Omega} \mathbf{a}(\eta)\mathbf{a}^T(\eta)d\eta. \tag{3.3}$$

By using the formulation of (3.2) along with the definition of the Gram matrix, one is left with

$$\begin{aligned} G_{\mathbf{a}} &= \int_{\Omega} \mathbf{a}(\eta)\mathbf{a}^T(\eta)d\eta \\ &= \int_{\Omega} A\mathbf{b}(\eta)[A\mathbf{b}(\eta)]^T d\eta \\ &= A\int_{\Omega} \mathbf{b}(\eta)\mathbf{b}^T(\eta)d\eta A^T \\ &= AG_{\mathbf{b}}A^T. \end{aligned}$$

The gram matrix of any polynomial basis can therefore be found if the coefficient matrix and the Gram matrix of the monomial basis is known.

$$G_{\mathbf{a}} = AG_{\mathbf{b}}A^T \tag{3.4}$$

By taking the spatial domain Ω to be $[\eta_a, \eta_b]$ and introducing the shorthand $k = i + j + 1$ an analytical expression of the Gram matrix of the monomial basis is trivially arrived at.

$$G_{\mathbf{b}} = \int_{\Omega} \mathbf{b}(\eta)\mathbf{b}^T(\eta)d\eta = \left\{ \int_{\eta_a}^{\eta_b} \eta^{i+j} d\eta \right\}_{i,j=0}^{Np,Np} = \left\{ \frac{\eta_b^k - \eta_a^k}{k} \right\}_{i,j=0}^{Np,Np} \tag{3.5}$$

Finding the derivative of some polynomial basis w.r.t. η is achieved by transforming the coefficient matrix. To perform this transformation the matrix D is introduced as

$$D = \left[\begin{array}{c|c} 0 & 0 \\ \hline \{i\}_{i=1}^{N_p} I & 0 \end{array}\right] \tag{3.6}$$

such that

$$\frac{d\mathbf{a}(\eta)}{d\eta} = \mathbf{a}_{\eta}(\eta) = P_{\eta}\mathbf{b}(\eta) = (PD)\mathbf{b}(\eta). \tag{3.7}$$

[1]The way that the Gram matrix is defined here might differ from other sources where this definition would technically be the transpose of the Gram matrix, however, the Gram matrix being symmetric means that these definitions are equivalent.

Extending a basis to two dimensions is done using the Kronecker product, denoted $\otimes$.

$$\mathbf{a}^{2D}(\eta,\xi) = \mathbf{a}(\eta)\otimes\mathbf{a}(\xi) = (A\mathbf{b}(\eta))\otimes(A\mathbf{b}(\xi)) = (A\otimes A)(\mathbf{b}(\eta)\otimes\mathbf{b}(\xi)) \quad (3.8)$$

This extension let us define 2-dimensional[2] coefficient matrices and monomial bases

$$A^{2D} = A\otimes A, \quad \mathbf{b}^{2D}(\eta,\xi) = \mathbf{b}(\eta)\otimes\mathbf{b}(\xi) \quad (3.9)$$

and even partial derivatives

$$\mathbf{a}_\eta^{2D}(\eta,\xi) = (AD\otimes A)\mathbf{b}^{2D}(\eta,\xi)$$
$$\mathbf{a}_\xi^{2D}(\eta,\xi) = (A\otimes AD)\mathbf{b}^{2D}(\eta,\xi).$$

3.1.2 Legendre basis

An orthonormal polynomial basis $\mathbf{p}(\eta) = P\mathbf{b}(\eta)$ is used in the numerical method. Requiring the basis to be orthonormal is equivalent to requiring that it's Gram matrix is the identity matrix, i.e.,

$$G_{\mathbf{p}} = PG_{\mathbf{b}}P^T = I. \quad (3.10)$$

Since $G_{\mathbf{b}}$ is Hermitian, Cholesky-decomposition can be used to factor it as

$$G_{\mathbf{b}} = BB^T, \quad (3.11)$$

where B is a lower triangular matrix. This implies that

$$G_{\mathbf{p}} = PBB^TP^T = (PB)(PB)^T. \quad (3.12)$$

Thus, if P satisfies $PB = I$, then $G_{\mathbf{p}} = II^T = I$ and P is determined by solving the linear system

$$PB = I. \quad (3.13)$$

Unsurprisingly, this yields normalized Legendre polynomials. These polynomials will be used as a basis for the modal expansion of the numerical method.

[2]The subscripts $1D$ and $2D$ are omitted where dimensionality can be inferred from context, i.e. you might encounter $\mathbf{a}^{2D} = \mathbf{a}\otimes\mathbf{a}$ and $\mathbf{a} = \mathbf{a}^{1D}\otimes\mathbf{a}^{1D}$ but rarely $\mathbf{a}^{2D} = \mathbf{a}^{1D}\otimes\mathbf{a}^{1D}$.

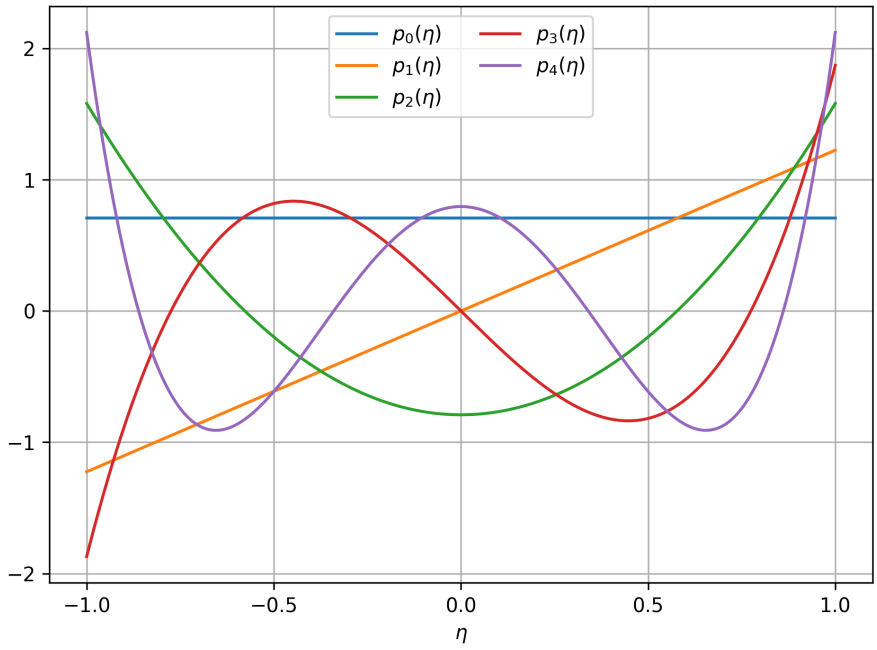


Figure 3.1: The first five normalized Legendre polynomials.

3.1.3 Lagrange basis

The Legendre basis can be used for modal expansion of our solution, however, for certain operations a basis $\boldsymbol{\ell}(\eta) = L\mathbf{b}(\eta)$ which is exact at certain locations, or nodes $\boldsymbol{\eta}$, is desired. We first define the Vandermonde matrix[3].

$$V = \mathbf{b}(\boldsymbol{\eta}^T) = \begin{bmatrix} (\boldsymbol{\eta}^T)^0 \\ (\boldsymbol{\eta}^T)^1 \\ (\boldsymbol{\eta}^T)^2 \\ \vdots \\ (\boldsymbol{\eta}^T)^{N_p} \end{bmatrix} = \begin{bmatrix} \eta_0^0 & \eta_1^0 & \eta_2^0 & \dots & \eta_{N_p}^0 \\ \eta_0^1 & \eta_1^1 & \eta_2^1 & \dots & \eta_{N_p}^1 \\ \eta_0^2 & \eta_1^2 & \eta_2^2 & \dots & \eta_{N_p}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \eta_0^{N_p} & \eta_1^{N_p} & \eta_2^{N_p} & \dots & \eta_{N_p}^{N_p} \end{bmatrix} \tag{3.14}$$

This is done such that we can evaluate our new basis at the aforementioned nodes. Our basis thus becomes

$$\boldsymbol{\ell}(\boldsymbol{\eta}) = L\mathbf{b}(\boldsymbol{\eta}) = LV. \tag{3.15}$$

Ultimately, we would like to evaluate our solution approximation as a weighted sum of the basis polynomials. Therefore, if we require each basis

[3]One might encounter the Vandermonde matrix, defined in other works, as the transpose of how it's defined in this text.

polynomial to be unity at it's corresponding node and to vanish at all other nodes, our solution approximation would be exactly that of it's corresponding weight at that node. Thus, to find this basis, we solve the following linear system.

$$LV = I \tag{3.16}$$

This yields the Lagrange polynomials which can be seen in 3.2.

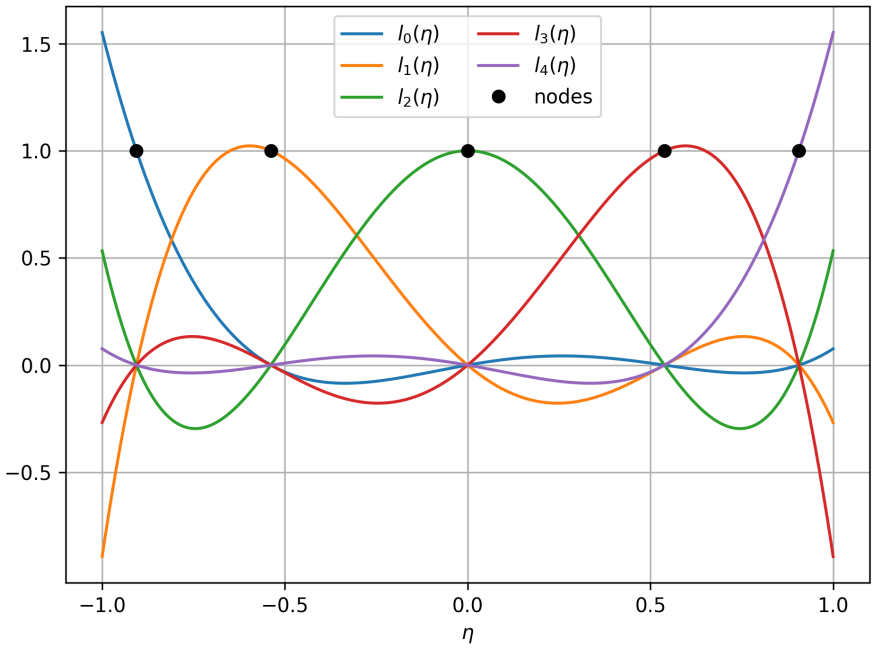


Figure 3.2: The first five Lagrange polynomials using Legendre nodes.

The roots of the $N_p + 1$th Legendre polynomial is used as nodes for the Lagrange interpolation.

3.2 Discontinuous-Galerkin Method

3.2.1 Spatial discretization

We begin by rewriting equation 2.1 as a system of equations.

$$\begin{cases} \mathbf{U}_t + \nabla \cdot \mathbf{F}(\mathbf{U}, \mathbf{S}) = \mathbf{0} \\ \mathbf{S} - \nabla \mathbf{U} = \mathbf{0} \end{cases}. \tag{3.17}$$

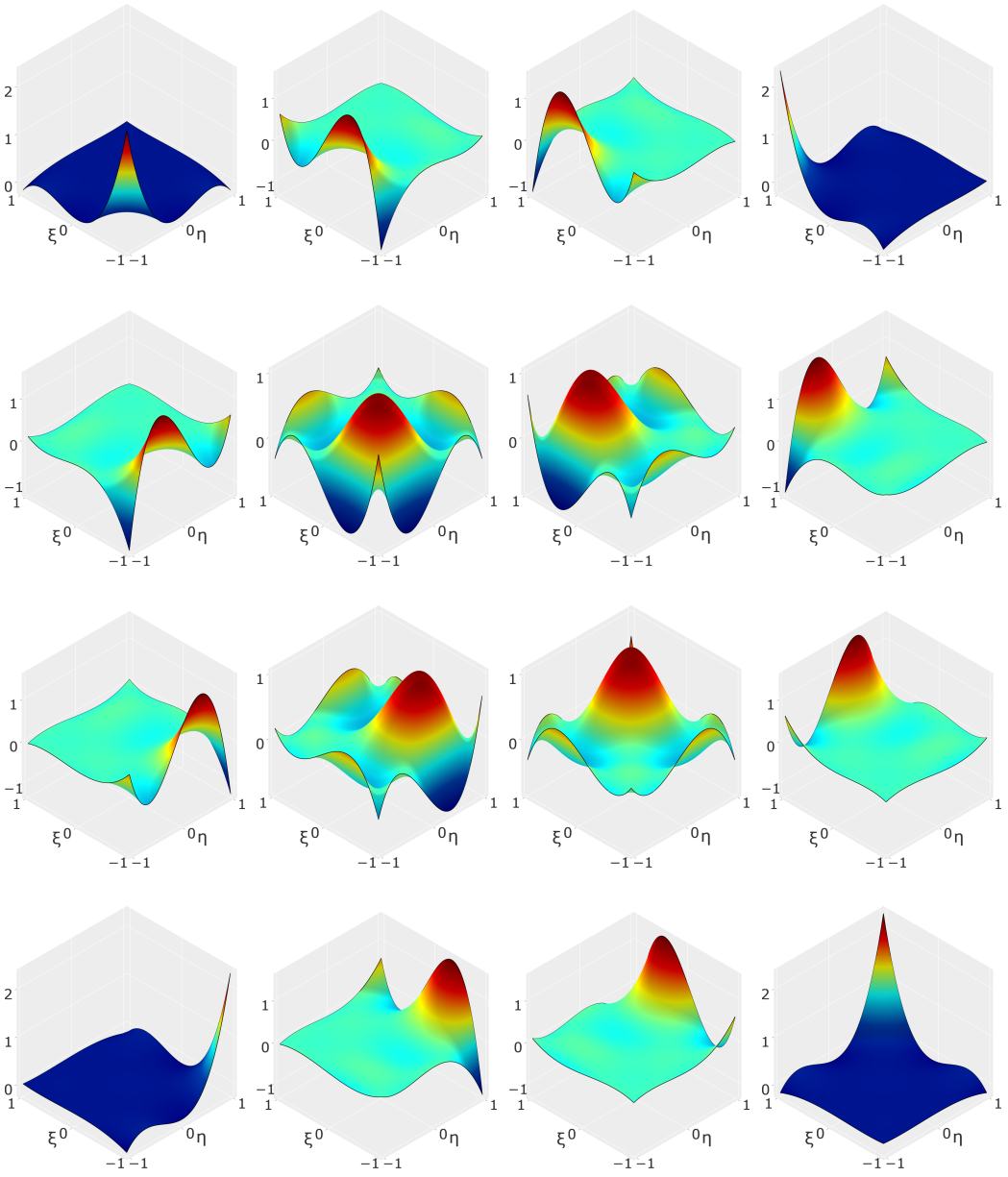


Figure 3.3: A fourth order 2D Lagrange basis made using the construction methods outlined in section 3.1.

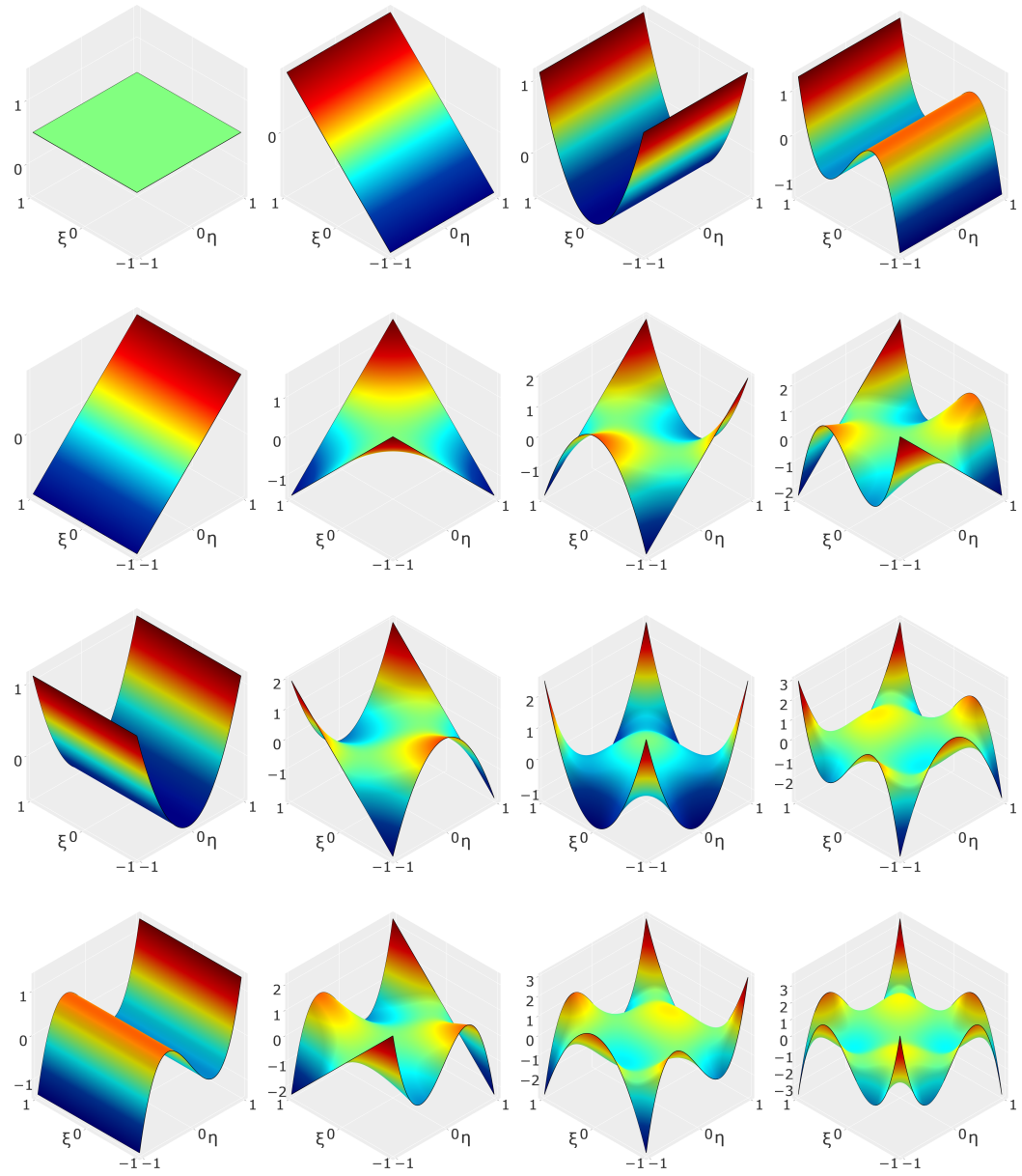


Figure 3.4: A fourth order 2D Legendre basis made using the construction methods outlined in section 3.1.

Similarly to flux, the auxiliary variable $\mathbf{S}$ also has two components.

$$\mathbf{S}_{1,2} = \left(\frac{\partial \mathbf{U}}{\partial x}, \frac{\partial \mathbf{U}}{\partial y} \right) \tag{3.18}$$

. Given some domain Ω for which the dynamics is governed by equation 3.17, the equation can be transformed to a weak formulation by multiplying by a trial space of test functions $\boldsymbol{\psi}$ and integrating over the domain.

$$\iint_{\Omega} \boldsymbol{\psi} \mathbf{U}_t dxdy + \iint_{\Omega} \boldsymbol{\psi} \left(\nabla_{xy} \cdot \mathbf{F} \right) dxdy = \mathbf{0} \tag{3.19}$$

Rather than integrating over the entire domain Ω all at once, we subdivide the domain into smaller subdomains, or cells, Ω^h for which we assume our solution to be continuous.

$$\iint_{\Omega^h} \boldsymbol{\psi}^h \mathbf{U}_t dxdy + \iint_{\Omega^h} \boldsymbol{\psi}^h \left(\nabla_{xy} \cdot \mathbf{F} \right) dxdy = \mathbf{0} \tag{3.20}$$

The specific trial spaces of each cell are all generated from a reference trial space $\boldsymbol{\psi}^*$ over a reference cell Ω^*. The only thing distinguishing the specific trial space from the reference trial space is a coordinate mapping specific to each cell.

$$\boldsymbol{\psi}^h(x, y) = \boldsymbol{\psi}^*(\eta^h(x), \xi^h(y)) \tag{3.21}$$

The coordinate mapping that will be used in this thesis assumes that the domain Ω is subdivided into into a perfectly cartesian grid, thus simplifying the following derivation significantly. The coordinate mapping of cell h is as follows.

$$\begin{cases} \eta^h(x) = \frac{2}{x_b^h - x_a^h} \left[x - \frac{x_a^h + x_b^h}{2} \right] \\ \xi^h(y) = \frac{2}{y_b^h - y_a^h} \left[y - \frac{y_a^h + y_b^h}{2} \right] \end{cases} \tag{3.22}$$

The integration can therefore be transformed such that it is evaluated on Ω^* instead of Ω^h. We do this by rewriting the differentials as

$$\begin{cases} dx = \frac{dx}{d\eta} d\eta = \frac{1}{\eta_x^h} d\eta \\ dy = \frac{dy}{d\xi} d\xi = \frac{1}{\xi_y^h} d\xi \end{cases} \tag{3.23}$$

such that equation 3.20 becomes

$$\frac{1}{\eta_x^h \xi_y^h} \left(\iint_{\Omega^*} \boldsymbol{\psi}^* \mathbf{U}_t d\eta d\xi + \iint_{\Omega^*} \boldsymbol{\psi}^* \left(\nabla_{xy} \cdot \mathbf{F} \right) d\eta d\xi \right) = \mathbf{0}, \tag{3.24}$$

which is equivalent to

$$\iint_{\Omega^*} \boldsymbol{\psi}^* \mathbf{U}_t d\eta d\xi + \iint_{\Omega^*} \boldsymbol{\psi}^* \left(\nabla_{xy} \cdot \mathbf{F}\right) d\eta d\xi = \mathbf{0}. \tag{3.25}$$

Similarly, the gradient operator ∇_{xy} is transformed by

$$\nabla_{xy} = J_h^{-1} \nabla_{\eta\xi}, \qquad J_h^{-1} = \begin{bmatrix} \eta_x^h & 0 \\ 0 & \xi_y^h \end{bmatrix}. \tag{3.26}$$

Transforming the gradient, and using Green's identity leaves us with

$$\iint_{\Omega^*} \boldsymbol{\psi}^* \mathbf{U}_t d\eta d\xi = \int_{\Omega^*} J_h^{-1} \nabla_{\eta\xi} \boldsymbol{\psi}^* \cdot \mathbf{F} d\eta d\xi - \oint_{\chi} \boldsymbol{\psi}^* \left.\mathbf{F}\right|_{\boldsymbol{\Gamma}} \cdot J_h^{-1} d\boldsymbol{\chi}. \tag{3.27}$$

$\boldsymbol{\Gamma}$ is a parameterization of the cell boundary with dependence on the parameter χ. $d\boldsymbol{\chi}$ is a shorthand for $\hat{\mathbf{n}}_\Gamma d\chi$, $\hat{\mathbf{n}}_\Gamma$ being the outwards pointing normal w.r.t. boundary Γ.

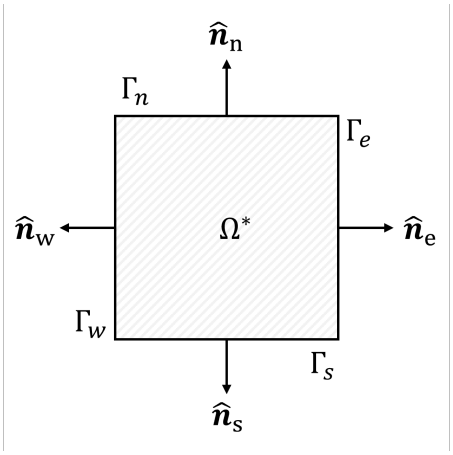


Figure 3.5: A depiction of the reference cell and it's boundary normals.

The boundary of the reference cell is parameterized as

$$\boldsymbol{\Gamma} : \begin{cases} \boldsymbol{\Gamma}_e(\chi) = \quad [1, \chi]^T \\ \boldsymbol{\Gamma}_w(\chi) = -[1, \chi]^T \\ \boldsymbol{\Gamma}_n(\chi) = \quad [\chi, 1]^T \\ \boldsymbol{\Gamma}_s(\chi) = -[\chi, 1]^T \end{cases}, \tag{3.28}$$

and the normal is given by

$$\hat{\mathbf{n}}_\Gamma = \frac{\partial}{\partial \chi} \begin{bmatrix} \Gamma_2 \\ -\Gamma_1 \end{bmatrix}. \tag{3.29}$$

3.2.2 Solution approximations

Even though we have arrived at an equation, our solution is still unknown to us (if it weren't this thesis would be moot). Since equation 3.27 describes the change in our solution with respect to time, we want to find some way of keeping track of our solution within each cell, and how it changes. We do this by introducing appropriate approximations for the quantities found in equation 3.27. The quantities we must approximate are the conserved variables $\mathbf{U}$, the flux within the cell $\mathbf{F}$, and the flux across its boundary $\mathbf{F}_\Gamma$.

We approximate $\mathbf{U}$ as the piecewise continuous union of the solution approximations of each individual cell. The solution in cell h is approximated as the weighted sum of the Legendre polynomials as a modal basis $\mathbf{p}^h(\eta, \xi)$ with the time dependent weights $\mathbf{U}_M^h(t)$.

$$\tilde{\mathbf{U}}^h(x, y, t) = \mathbf{U}_M^h(t) \cdot \mathbf{p}^h(x, y) \tag{3.30}$$

$$\mathbf{U}(x, y, t) \approx \bigcup_{h=0} \tilde{\mathbf{U}}^h(x, y, t) \tag{3.31}$$

Combining equation 3.30 and 3.18 gives us an approximation of $\mathbf{S}_1$ and $\mathbf{S}_2$.

$$\tilde{\mathbf{S}}_1^h(x, y, t) \approx \mathbf{U}_M^h(t) \cdot \frac{\partial \mathbf{p}^h(x, y)}{\partial x} = \eta_x^h \left(\mathbf{U}_M^h(t) \cdot \frac{\partial \mathbf{p}(\eta, \xi)}{\partial \eta} \right) \tag{3.32}$$

$$\tilde{\mathbf{S}}_2^h(x, y, t) \approx \mathbf{U}_M^h(t) \cdot \frac{\partial \mathbf{p}^h(x, y)}{\partial y} = \xi_y^h \left(\mathbf{U}_M^h(t) \cdot \frac{\partial \mathbf{p}(\eta, \xi)}{\partial \xi} \right) \tag{3.33}$$

For a non-linear flux function, evaluating the the flux of our modal solution approximation isn't straightforward. However, using the convenient property of the Lagrange basis evaluating to unity at the chosen nodes introduced in section 3.1.3 means that we can transform our modal solution expansions to nodal ones, evaluate the weights directly, and then arrive at a nodal flux approximation employing the same polynomial space. The nodal weights are obtained by simply evaluating the approximations at the nodes.

$$\mathbf{U}_N^h(t) = \tilde{\mathbf{U}}^h(\boldsymbol{\eta}_c, \boldsymbol{\xi}_c, t) \tag{3.34}$$

$$\mathbf{S}_N^h(t) = \tilde{\mathbf{S}}^h(\boldsymbol{\eta}_c, \boldsymbol{\xi}_c, t) \tag{3.35}$$

The flux reconstruction within cell h thus becomes

$$\hat{\mathbf{F}}^h(x, y, t) = \mathbf{F}_N^h \cdot \boldsymbol{\ell}^h(x, y), \tag{3.36}$$

Where $\mathbf{F}_N^h = \mathbf{F}\left(\mathbf{U}_N^h, \mathbf{S}_N^h\right)$.The reconstruction at the cell interface is much the same. A one-dimensional nodal expansion is used.

$$\hat{\mathbf{F}}_\Gamma^h(x, y, t) = \mathbf{F}_\Gamma^h \cdot \boldsymbol{\ell}^{h,1D}(\chi \hat{\mathbf{t}}_\Gamma \cdot \mathbf{1}) \tag{3.37}$$

The tangent of the boundary $\hat{\mathbf{t}}_\Gamma$ is defined as

$$\hat{\mathbf{t}}_\Gamma = \frac{\partial \mathbf{\Gamma}}{\partial \chi} \tag{3.38}$$

Since the solution is dually defined at the cell interface, a suitable numerical flux function $\mathbf{H}$ is chosen to reconcile the case for which the left and right state differ.

$$\mathbf{F}_\Gamma^h = \mathbf{H}\left(\mathbf{U}_\Gamma^{h,L}, \mathbf{U}_\Gamma^{h,R}, \mathbf{S}_\Gamma^{h,L}, \mathbf{S}_\Gamma^{h,R}\right). \tag{3.39}$$

3.2.3 Operator construction

Now that we have suitable approximations, these can be inserted into equation 3.27. We choose the Legendre polynomials to be our trial space, and transpose[4] our approximations before inserting them.

$$\underbrace{\iint_{\Omega^*} \mathbf{p}\left(\tilde{\mathbf{U}}^h\right)_t^T d\eta d\xi}_{(i)} = \underbrace{\iint_{\Omega^*} J_h^{-1}\nabla_{\eta\xi}\mathbf{p}\cdot\left(\hat{\mathbf{F}}^h\right)^T d\eta d\xi}_{(ii)} - \underbrace{\oint_\chi \mathbf{p}\left(\hat{\mathbf{F}}_\Gamma^h\right)^T \cdot J_h^{-1} d\boldsymbol{\chi}}_{(iii)}. \tag{3.40}$$

Due to the length of equation 3.40, each term is treated separately. The first term yields the mass matrix M.

$$(i) \rightarrow \underbrace{\iint_{\Omega^*} \mathbf{p}\mathbf{p}^T d\eta d\xi}_{M}\, \mathbf{U}_M^h \tag{3.41}$$

The second term gives the stiffness matrices K_1 and K_2.

$$(ii) \rightarrow \eta_x^h \underbrace{\iint_{\Omega^*} \mathbf{p}_\eta \boldsymbol{\ell}^T d\eta d\xi}_{K_1}\, \mathbf{F}_{1,N}^h + \xi_y^h \underbrace{\iint_{\Omega^*} \mathbf{p}_\xi \boldsymbol{\ell}^T d\eta d\xi}_{K_2}\, \mathbf{F}_{2,N}^h \tag{3.42}$$

And from the third term, the boundary operators $Q_{e,w,n,s}$ are obtained.

$$\begin{aligned}(iii) \rightarrow \eta_x^h \overbrace{\int_\chi \mathbf{p}^e \left(\boldsymbol{\ell}^{1D}(\chi)\right)^T d\chi}^{Q_e}\, \mathbf{F}_e^h - \eta_x^h \overbrace{\int_\chi \mathbf{p}^w \left(\boldsymbol{\ell}^{1D}(-\chi)\right)^T d\chi}^{Q_w}\, \mathbf{F}_w^h \\ + \xi_y^h \underbrace{\int_\chi \mathbf{p}^n \left(\boldsymbol{\ell}^{1D}(\chi)\right)^T d\chi}_{Q_n}\, \mathbf{F}_n^h - \xi_y^h \underbrace{\int_\chi \mathbf{p}^s \left(\boldsymbol{\ell}^{1D}(-\chi)\right)^T d\chi}_{Q_s}\, \mathbf{F}_s^h\end{aligned} \tag{3.43}$$

[4]The action of transposing pertains only to the dimensions of the functional space, not the two spatial dimensions nor the four variable dimensions. This is in contrast to the two dot products in equation 3.40 which only operate along the spatial dimension.

Using the newly identified operators, equation 3.40 becomes[5]

$$\dot{\mathbf{U}}_M^h = M^{-1}\left[K_1^h \mathbf{F}_{1,N} + K_2^h \mathbf{F}_{2,N} + Q_w^h \mathbf{F}_w^h - Q_e^h \mathbf{F}_e^h + Q_s^h \mathbf{F}_s^h - Q_n^h \mathbf{F}_n^h\right]. \quad (3.44)$$

Ultimately, by employing the ideas presented so far in this chapter, explicit expressions for the operators are found. The mass matrix is by definition the identity matrix, and its inverse naturally so too.

$$M = I \implies M^{-1} = I \quad (3.45)$$

The stiffness matrices become

$$K_1 = \left[P^{1D} D \otimes P^{1D}\right] G_{\mathbf{b}} L^T \quad (3.46)$$

$$K_2 = \left[P^{1D} \otimes P^{1D} D\right] G_{\mathbf{b}} L^T, \quad (3.47)$$

and the boundary operators

$$Q_e = P\left[\mathbf{b}^{1D}(1) \otimes G_{\mathbf{b}}^{1D}\right] L^{1D^T} \quad (3.48)$$

$$Q_w = P\left[\mathbf{b}^{1D}(-1) \otimes G_{\mathbf{b}}^{1D}\right] L^{1D^T} \quad (3.49)$$

$$Q_n = P\left[G_{\mathbf{b}}^{1D} \otimes \mathbf{b}^{1D}(1)\right] L^{1D^T} \quad (3.50)$$

$$Q_s = P\left[G_{\mathbf{b}}^{1D} \otimes \mathbf{b}^{1D}(-1)\right] L^{1D^T} \quad (3.51)$$

3.2.4 Temporal discretization

We let the temporal discretization be relatively agnostic to the spatial discretization, and thus replace the right hand side of equation 3.44 with a residual function dependent only on the modal weights and time.

$$\dot{U}_M^h = R(U_M^h, t) \quad (3.52)$$

The classical fourth order Runge-Kutta method is used due to it's favorable stability and high order. The residual equation is used to express the method as

$$\begin{aligned}
k_1 &= R\left(U_M^h,\ t\right) \\
k_2 &= R\left(U_M^h + k_1 \Delta t/2,\ t + \Delta t/2\right) \\
k_3 &= R\left(U_M^h + k_2 \Delta t/2,\ t + \Delta t/2\right) \\
k_4 &= R\left(U_M^h + k_3 \Delta t,\ t + \Delta t\right) \\
U_M^h &= U_M^h + \Delta t\left[k_1 + 2k_2 + 2k_3 + k4\right]/6.
\end{aligned}$$

[5]The superscript h, in relation to the operators, is used as a shorthand for the operator multiplied with it's preceding scaling factor, e.g. $K_1^h = \eta_x^h K_1 = \frac{2}{\Delta x_h} K_1$.

A convective and a viscous a CFL number based on [17, 44] are used to determine the time step as follows.

$$\Delta t_c \leq \frac{C_c}{(2N_p+1)\left(\frac{|u|+c}{\Delta x}+\frac{|u|+c}{\Delta x}\right)} \tag{3.53}$$

$$\Delta t_v \leq \frac{\rho C_v}{(2N_p+1)^2\left(\sqrt{2}\mu\ max\left\{\frac{4}{3},\frac{\gamma}{Pr}\right\}\right)\left(\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}\right)} \tag{3.54}$$

$$\Delta t = min\left\{\Delta t_c, \Delta t_v\right\} \tag{3.55}$$

Numerical flux

The notation of the average $\{\!\{u\}\!\} = \frac{1}{2}\left(u^+ + u^-\right)$ and the jump $[\![u]\!] = \hat{\mathbf{n}}^- u^- + \hat{\mathbf{n}}^+ u^+$, commonly used by the DG-community, will from hereon be used to define the interfacial flux-functions. For a more thorough introduction to these operators, the reader is referred to [26]. A Rusanov flux is used for the convective fluxes at the cell interfaces.

$$\mathbf{H}^c\left(\mathbf{U}^L, \mathbf{U}^R\right) = \{\!\{\mathbf{F}^c\}\!\} - \frac{1}{2} a\, [\![\mathbf{U}]\!] \tag{3.56}$$

The largest wave speed is used to determine the directionality of the flux, i.e. $a = max\{a_L, a_R\}$, the largest wave speed of the convective flux being $a_1 = |u| + c$ and $a_2 = |v| + c$ in the x- and y direction respectively.

As pointed out in [18], greater care must be taken in the reconstruction of the interfacial diffusive fluxes than their convective counterparts. This is due to the dependence of the diffusive fluxes on the spatial derivatives of the solution. Additional terms are thus introduced to the diffusive numerical flux in order to account for the case where there's a jump in the solution variables between adjacent cells. The viscous flux described here is that of the Direct Discontinuous-Galerkin method (DDG) described in [37].

$$\mathbf{H}^v\left(\mathbf{U}^L, \mathbf{U}^R, \mathbf{S}^L, \mathbf{S}^R\right) = \mathbf{F}^v\left(\{\!\{\mathbf{U}\}\!\}, \mathbf{S}^*\left(\mathbf{U}^L, \mathbf{U}^R, \mathbf{S}^L, \mathbf{S}^R\right)\right) \tag{3.57}$$

Here S^* is a special reconstruction of the interfacial gradient given by

$$\mathbf{S}^* = \{\!\{\mathbf{S}\}\!\} + \frac{\beta_0}{\boldsymbol{\Delta}} [\![\mathbf{U}]\!]\, \hat{\mathbf{n}} + \beta_1 \boldsymbol{\Delta}\, [\![\nabla \mathbf{S} \cdot \hat{\mathbf{n}}]\!]\,. \tag{3.58}$$

Ultimately β_0 was set to 1 and β_1 to 0.

3.3 Boundaries

In this section, two types of boundary implementations will be used. Firstly, a brief description of the numerical treatment of conforming boundaries are presented. Thereafter, a thorough description of an immersed boundary method follows.

3.3.1 Conforming boundaries

The flux functions of section 3.2.4 are used to weakly enforce certain boundary conditions. This is done by prescribing the state of the conserved variables, and their derivatives, exterior to the domain. The second derivatives of $\mathbf{U}$ are to be extrapolated from the interior in all boundary discretizations, meaning that all terms resulting from $\nabla\mathbf{S}$ in equation 3.58 can be neglected in the numerical boundary treatment. Most of the ensuing boundary discretizations are taken from [38].

Far field

The discretization of a far field or inflow boundary is achieved by prescribing

$$\mathbf{U}^+ = \mathbf{U}_\infty \tag{3.59}$$

and extrapolating derivatives from the interior.

$$\mathbf{S}^+ = \mathbf{S}^- \tag{3.60}$$

Subsonic outflow

A partially non-reflecting subsonic outflow boundary is discretized by extrapolating all variables and derivatives, except the pressure, which is determined by

$$p^+ = 2p_\Gamma - p^-, \tag{3.61}$$

where p_Γ is the prescribed pressure at the outflow boundary.

Adiabatic wall

Again, all variables and derivatives are extrapolated from the interior. Then the velocity is set to zero such that the no-slip condition is enforced.

$$\mathbf{u}^+ = \mathbf{0} \tag{3.62}$$

While it is possible to enforce the adiabatic condition by requiring that there's no heat flux along the normal of the wall such as done in [18], a simpler and consistent formulation when no-slip is already assumed is to require there to be no mass, nor energy flux along the normal such as in equation 2.18. We therefore construct the aforementioned exterior gradients by subtracting the interior gradients projected onto the wall normal from interior gradient.

$$[\nabla\rho]^+ = [\nabla\rho]^- - \left([\nabla\rho]^- \cdot \hat{\mathbf{n}}\right)\hat{\mathbf{n}} \tag{3.63}$$
$$[\nabla(\rho E)]^+ = [\nabla(\rho E)]^- - \left([\nabla(\rho E)]^- \cdot \hat{\mathbf{n}}\right)\hat{\mathbf{n}} \tag{3.64}$$

Symmetry

The symmetry boundary condition is implemented by mirroring all directional quantities[6] and reflecting all scalar quantities w.r.t. the boundary. This leads to the following implementation.

$$\mathbf{u}^+ = \mathbf{u}^- - 2\left(\mathbf{u}^- \cdot \hat{\mathbf{n}}\right)\hat{\mathbf{n}} \tag{3.65}$$
$$\mathbf{S}^+ = \mathbf{S}^- - 2\left(\mathbf{S}^- \cdot \hat{\mathbf{n}}\right)\hat{\mathbf{n}} \tag{3.66}$$
$$\rho^+ = \rho^- \tag{3.67}$$
$$E^+ = E^- \tag{3.68}$$

3.3.2 Immersed Boundary Method

In this section an immersed boundary method based on the imposition through the use of a fictitious fluid field is detailed. The method presented here is loosely based on the one found in [30], although, with some significant differences. The fictitious fluid field (which shall hereafter be known as the ghost fluid), is used to simulate the presence of an obstacle in the flow. Given a domain Ω we partition it into a fluid part Ω_F, a solid part Ω_S, and the bound between these two parts Γ. On Ω_F the solution is governed by the compressible Navier-Stokes while on Ω_S solution values can be prescribed. The idea is to prescribe solution values on Ω_S such that chosen boundary conditions are fulfilled on Γ.

[6] Directional quantity here refers to any quantity that can be thought of as having a direction, i.e. be expressed as a vector.

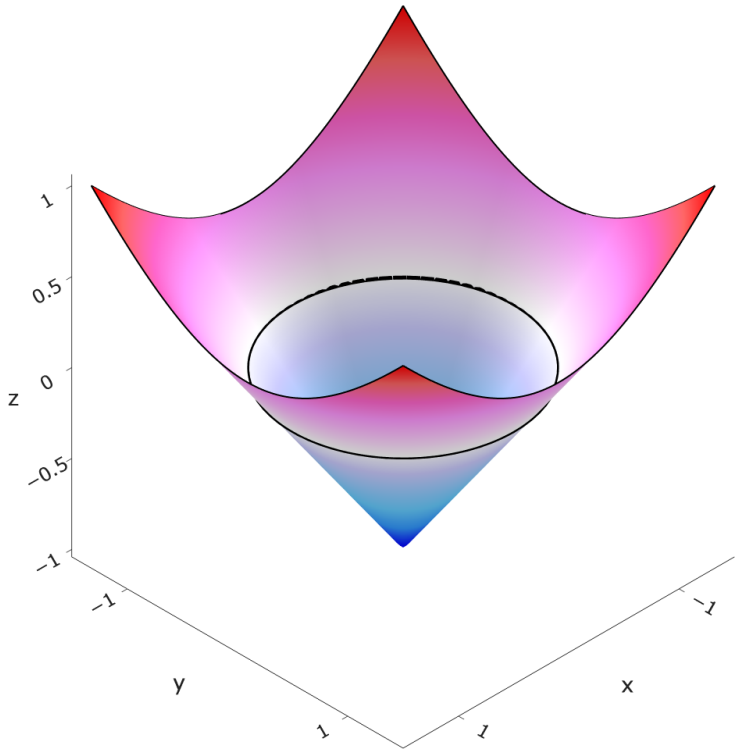


Figure 3.6: Signed distance function of a unit circle.

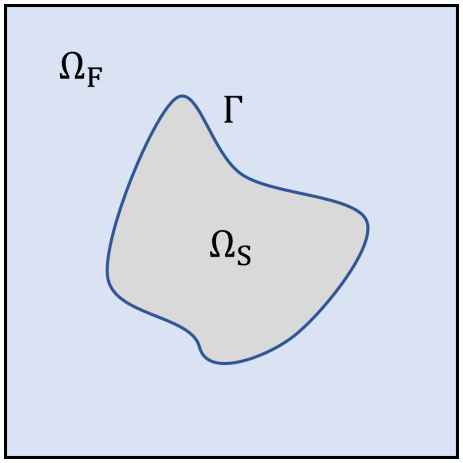


Figure 3.7: A domain partitioned into two separate domains, Ω_F and Ω_S, separated by the boundary Γ.

First, we need some way of partitioning the domain as stated. We choose to describe the geometry implicitly through the use of a signed distance

function $\phi(x, y)$. The partitioning is then defined as

$$\Omega_F = \{(x, y) \in R \mid \phi(x, y) > 0\} \tag{3.69}$$
$$\Omega_S = \{(x, y) \in R \mid \phi(x, y) < 0\} \tag{3.70}$$
$$\Gamma = \{(x, y) \in R \mid \phi(x, y) = 0\}. \tag{3.71}$$

While it would be convenient to treat the conditions on Γ generally, some choice should be made as the remainder of the implementation will depend on what conditions we want to have fulfilled. The no-slip condition of 2.2.4 is thus selected.

We'll first treat the no-slip condition, i.e. $\mathbf{u} = \mathbf{0}$ on Γ. If we do an odd reflection of u and v about the boundary Γ, then u and v must pass through zero at the boundary, and the condition is fulfilled. We use the signed distance property of ϕ to define the vector $\boldsymbol{\delta}$ from the closest point on Γ to some point $\mathbf{x}$.

$$\boldsymbol{\delta}(\mathbf{x}) = \phi(\mathbf{x})\nabla\phi(\mathbf{x}) \tag{3.72}$$

A reflection of point $\mathbf{x}$ about Γ can now be constructed as

$$\mathbf{x}_r(\mathbf{x}) = \mathbf{x} - 2\boldsymbol{\delta}(\mathbf{x}). \tag{3.73}$$

The odd extension of the velocity in the ghost domain becomes

$$\mathbf{u}(\mathbf{x}) = -\mathbf{u}(\mathbf{x}_r), \qquad \mathbf{x} \in \Omega_S. \tag{3.74}$$

Even extensions are used for the density and the energy.

$$\rho(\mathbf{x}) = \rho(\mathbf{x}_r), \qquad \mathbf{x} \in \Omega_S. \tag{3.75}$$
$$E(\mathbf{x}) = E(\mathbf{x}_r), \qquad \mathbf{x} \in \Omega_S. \tag{3.76}$$

An even extension is used for the derivatives of $\mathbf{U}$.

$$\nabla\mathbf{U}(\mathbf{x}) = \nabla\mathbf{U}(\mathbf{x}_r), \qquad \mathbf{x} \in \Omega_S. \tag{3.77}$$

The above continuous treatment of Ω_S must be discretized in a way compatible with the rest of the numerical treatment detailed in this chapter. The modal bases is transformed into its nodal representation. This is done such that the domain in which any particular node lies can be identified, and the proper treatment be applied to that node. The nodes that are within Ω_S are denoted as the ghost points $\mathbf{x}_{GP}$. Each ghost point has a corresponding image point which is the reflection of of the ghost point about the boundary.

$$\mathbf{x}_{IP} = \mathbf{x}_r(\mathbf{x}_{GP}) \tag{3.78}$$

The question then becomes; how do you recreate the solution state at $\mathbf{x}_{IP}$. The naive approach might be to use the nodal solution approximation detailed in section 3.1.3. However, since the polynomial of each node only vanish at other nodes, this reconstruction is ill suited for cells containing both ghost points and image points. It could perhaps still be possible to obtain the values at the ghost points using this reconstruction, but one would need to solve for these unknowns implicitly. Instead, the solution state at the image point is reconstructed such that it has no dependence on any ghost points. The two following criterions were used to find a suitable reconstruction.

1. The reconstructed state at any image point should be independent of what occurs within the solid domain such that the value at the ghost points can be evaluated explicitly, and in any order.

2. The reconstructed state should be consistent, i.e., if an image point coincides with a node, then the reconstruction at that image point should take on the value of the solution at the coincident node.

First the cell wherein the image point lies is identified. Then the distance between the image point and the nodes of the cell is determined.

$$d_i = \|\mathbf{x}_i - \mathbf{x}_{IP}\| \tag{3.79}$$

Then a set of weights are constructed based on the reciprocal of the squared distance.

$$w_i = \begin{cases} d_i^{-2} & \mathbf{x}_i \in \Omega_F \\ 0 & \text{else} \end{cases} \tag{3.80}$$

The weights are normalized.

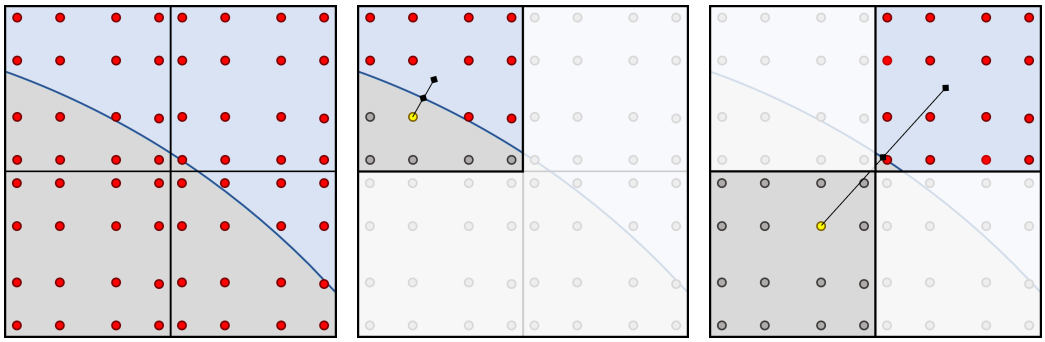
$$\hat{w}_i = \frac{w_i}{\sum w_i} \tag{3.81}$$

Using the nodal values $\mathbf{U_i^h}$, we ultimately are left with the following approximation at the image point.

$$\mathbf{U}_{IP} = \sum \mathbf{U}_i^h \hat{w}_i \tag{3.82}$$

The reconstruction satisfies criterion 1 as only fluid points are used in its construction. Additionally, it can be shown that

$$\lim_{\mathbf{x}_{IP} \to \mathbf{x}_i} \hat{w}_j = \begin{cases} 1 & j = i \\ 0 & j \neq i \end{cases} \tag{3.83}$$



(a) Four cells intersected by a boundary.

(b) Sampling at image point is done within same cell as ghost node.

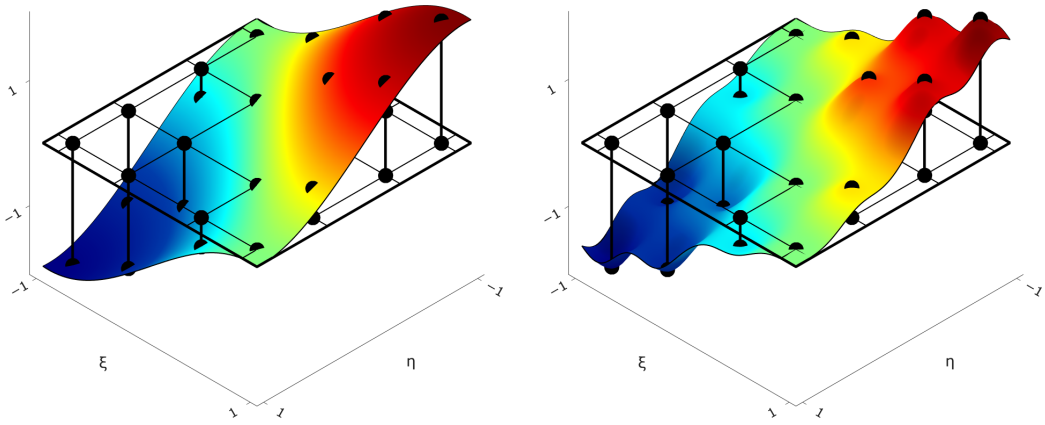
(c) Sampling at image point is done in different cell than the one containing the ghost node.

Figure 3.8: A display of distinct sampling configurations. The solid part of the domain is depicted in gray, while the fluid is depicted as blue. The yellow node is the ghost node for which a value must be sampled at its image point (black diamond). Only fluid nodes (red) are used to reconstruct the state at the image point, while other ghost nodes (gray) are ignored.

meaning that

$$\lim_{\mathbf{x}_{IP} \to \mathbf{x}_i} \mathbf{U}_{IP} = \mathbf{U}_i^h, \tag{3.84}$$

i.e., the reconstruction satisfies criterion 2 as well. In essence, since the functional basis used to reconstruct the state at the image point only depends on the distance to a particular node, the state is reconstructed using radial basis functions (RBF).



(a) Lagrange reconstruction. (b) RBF reconstruction.

Figure 3.9: The reconstruction of the continuous solution across some cell h using its nodal values $\mathbf{U}^h_{N,i}$.

Chapter 4

Test cases

The chosen test case is that of a cylinder in a flow. This test case has been abundantly documented both experimentally as well as numerically. This test case is used to asses the presented IBM.

4.1 Flow over cylinder

There are multiple reasons why the flow over a cylinder is a good benchmark to evaluate the immersed boundary method presented in this thesis. It is a relatively simple geometry, while also being sufficiently complicated as to not be trivial. It is a well documented benchmark case for which there exist a plethora of experimental and numerical results. It shows distinct flow behaviour in various flow domains and gives rise to the Kármán vortex street above some critical Reynolds number. The Kármán vortex street serves as a useful phenomena on which quantitative and qualitative assessment of a numerical solution can be based. A rectangular flow domain with a cylinder of diameter D at its origin is used. The flow domain extends symmetrically about its origin in the vertical direction and spans a total height of H. Horizontally, the domain spans the distance between an inlet and an outlet located L_1 and L_2 away from the cylinder. Similarly to the previous test case, the free stream- and outflow boundary treatment of 3.3 are used at inlet and outlet, while a symmetry boundary condition is used for the top and bottom "wall".

The flow regimes can roughly be classified according to table 4.1. Regime 3 was ultimately chosen due to it's transient, but orderly behaviour of alternating vortices. Further, a Reynolds number of 100 was chosen as it coincides with the works [1,2,3]. A low mach number was desired such that the simulations could be compared to incompressible solutions. Even so, the Mach

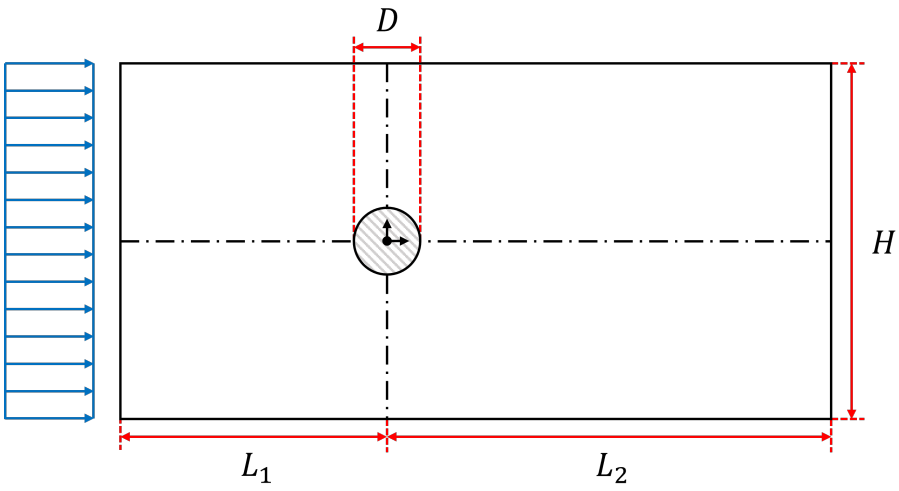


Figure 4.1: A depiction of the dimensions of the computational domain used for the a circular cylinder in a flow test case.

number still needed to be sufficiently high as to allow reasonable simulation times without violating the stringent CFL-condition. A Mach number of 0.25 was chosen as a suitable compromise between the two. However, due to a small calculation error[1], the Mach number ended up being ≈ 0.1786. Fortunately, the results of [6] and [29] indicate that this change in the Mach number should not lead to drastic changes in the resulting flow. The Prandtl number was simply held constant at 0.72, being a good fit for air at standard conditions[42]. The chosen flow state is summarized in table 4.1.

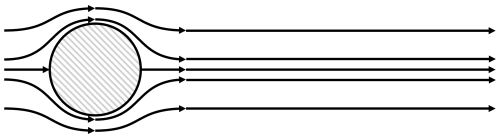
Re	Ma	Pr
100	0.1786	0.72

Table 4.1: Values of the dimensionless numbers used to characterize the flow of the test case.

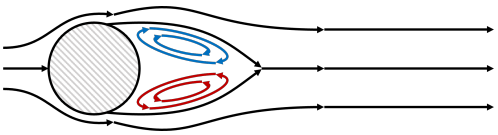
The distance function that will be used to describe the circle is given as

$$\phi(x, y) = \sqrt{x^2 + y^2} - \frac{1}{2}. \tag{4.1}$$

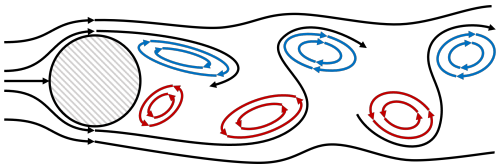
[1]The relation $p_0 = \frac{\gamma}{Ma_0^2}$ was accidentally used instead of the supposed $p_0 = \frac{1}{\gamma Ma_0^2}$ yielding a Mach number of 0.17857142857142858.



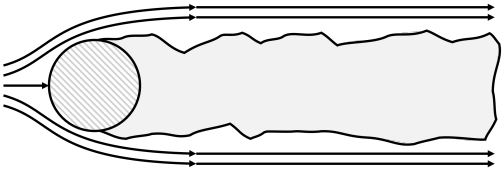
(a) $Re \in [0, 5]$ Known as stokes flow, this regime is characterized by inertial effects being negligible compared to viscous forces. This leads to symmetric streamlines upstream and downstream of the cylinder[27].



(b) $Re \in [5, 40]$ The flow separates behind the cylinder and a recirculation zone consisting of two counter-rotating vortices form[4]. The length of the recirculation zone increase with increasing Re[27].



(c) $Re \in [40, 200]$ The recirculating vortices become inherently unstable, and as such, minute disturbances are amplified until a periodic, but unsteady pattern is formed.[27] The pattern is characterized by the alternate shedding of vortices from either side of the cylinder[4]. Although unsteady, the flow is still laminar.



(d) $Re \in [200)$ The wake becomes turbulent[27]. While It is possible to identify distinct regimes above $Re = 200$, for simplicity no such distinction has been made in this text.

Figure 4.2: Classification of four distinct flow regimes for the flow around a cylinder. It should be noted that the Reynolds numbers given are estimates. In reality there will be some overlap between the regimes.

L_1	Length between inlet and center of cylinder
L_2	Length between outlet and center of cylinder
H	Height of domain
N_c	Number of elements in x-direction
M_c	Number of elements in y-direction
N_p	Order of 1D basis functions
Re_0	Reference Reynolds number
Ma_0	Reference Mach number
Pr_0	Reference Prandtl number
C_{CFL}	Convective CFL number
C_{DFL}	Diffusive CFL number

Table 4.2: Input parameters of the flow over a cylinder test case.

Chapter 5

Results

Several simulations were performed at various orders p and refinement levels r. At low Reynolds numbers (regime 3, table 4.1) a computationally expensive transient exists between symmetric initial conditions and the steady periodic behaviour described in section 4.1[33]. However, this transient can be significantly shortened by altering initial or inlet conditions. The following method ended up being effective in triggering vortex shedding.

1. Simulate cylinder using symmetric initial- and boundary conditions until a modest wake has developed.

2. Project solution onto the grid of a new simulation with a vertical offset of one cylinder radius.

3. Let solution converge to regular vortex street.

This method and initialization with previously converged solutions were used to obtain convergence towards steady periodic vortex shedding.

Four simulations were performed at various levels of grid refinement. These four simulations were all performed at the same similar flow conditions, with Re = 100, Ma = 0.1786, and Pr = 0.72. The simulations are denoted A, B, C, D in order of the lowest to the finest grid used for the simulation. Various details regarding the simulations is found in table 5 and the dimensions of the domain is described in table 5.1.

According to Newton's second law of motion, the change in momentum w.r.t. time of the fluid is equal to the force acting on the fluid.

$$\mathcal{F} = \iint_{\Omega} \frac{\partial (\rho \mathbf{u})}{\partial t} d\Omega \tag{5.1}$$

The change in momentum w.r.t. is readily found from the momentum equation of the Navier-Stokes equations. The momentum equation is expressed

L_1	L_2	H
10	20	20

Table 5.1: Dimensions of computational domain for the test case of a circular cylinder in a flow. All values given are normalized w.r.t. the diameter of the cylinder.

Table 5.2: The computational details of simulations performed on the flow over the cylinder test case.

	A	**B**	**C**	**D**
r	1	2	3	4
p	4	4	4	4
N_{cells}	48	96	144	192
M_{cells}	32	64	96	128
N_{DOFs}	24576	98304	221184	393216
C_{CFL}	0.4	0.4	0.4	0.4
C_{DFL}	0.4	0.4	0.4	0.4
t_{CPU}	2580	40080	264960	282720
t_{sim}	57.46	98.54	158.18	79.61
t_{CPU}/t_{sim}	45	407	1675	3551
$t_{CPU}/\left(t_{sim}N_{DOFs}\right)$	$1.83 \cdot 10^{-3}$	$4.14 \cdot 10^{-3}$	$7.57 \cdot 10^{-3}$	$9.03 \cdot 10^{-3}$

in terms of the momentum flux tensor $\boldsymbol{\sigma}$.

$$\left(\rho \mathbf{u}\right)_t + \nabla \cdot \boldsymbol{\sigma} = \mathbf{0} \tag{5.2}$$

The momentum flux consists of a convective term, a pressure term, and a viscous term.

$$\boldsymbol{\sigma} = \rho \mathbf{u}\mathbf{u}^T + p\boldsymbol{I} - \boldsymbol{\tau} \tag{5.3}$$

From Newton's third law, since the cylinder is at rest, the force acting on the cylinder must be of equal magnitude and opposite direction to that of the force acting on the fluid exerted by the cylinder. Therefore, by combining equation (5.1) and (5.2) and using the divergence theorem, the force acting on the cylinder is

$$\mathcal{F}_{\text{cyl}} = -\oint_S \boldsymbol{\sigma} \cdot \hat{\mathbf{n}} dS. \tag{5.4}$$

Here S is the cylinder surface and $\hat{\mathbf{n}}$ the outward facing unit normal. Since a no slip condition is supposed to apply at the boundary in question, the

convective term $\rho \mathbf{u}\mathbf{u}^T$ is neglected. Nondimensionalizing equation 5.4 then leaves us with

$$\mathbf{c} = \frac{1}{\frac{1}{2}\rho_0 u_0^2 D_0} \int_S (\boldsymbol{\tau} - p\boldsymbol{I} \cdot \hat{\mathbf{n}})\, dS, \tag{5.5}$$

where the vector $\mathbf{c}$ is used to find the drag- and the lift coefficient. The unit vectors $\hat{\mathbf{i}}$ and $\hat{\mathbf{j}}$, being parallel and orthogonal to the freestream flow direction respectively, are then used to find the drag- and lift coefficient.

$$c_D = \mathbf{c} \cdot \hat{\mathbf{i}}, \qquad c_L = \mathbf{c} \cdot \hat{\mathbf{j}} \tag{5.6}$$

The nondimensional Strouhal number is defined based on the frequency of oscillations of the lift coefficient ω, the cylinder diameter D_0, and the reference velocity u_0.

$$St = \frac{\omega D_0}{u_0} \tag{5.7}$$

In order to assess convergence, the change in the lift coefficient between consecutive extrema was used. Simulation B, C, and D initially descends rapidly. This initial decent quickly slows down and a more irregular pattern emerges. Simulation A display this irregular pattern from the beginning. It is believed that this is due to simulation A being initialized with an already converged solution of the same order and grid size. Simulation B, C, and D were all initialized from grids coarser than that of the their respective meshes. There could be multiple causes for this irregular behaviour. The sampling frequency might be to low, leading to errors in peak estimation. Alternatively, non-harmonic frequency components of the signal might be present or simply not have had sufficient time to decay. The output of the simulations were all saved as snapshots at regular intervals. In order to better estimate periodicity and peak values of drag and lift, cubic splines were used to interpolate simulation parameters in between saved snapshots. The drag coefficient c_D, lift coefficient c_L, and the Strouhal number St were all estimated based on the.

It is of interest to find some quantity that can be used to assess the efficacy of the immersed boundary. The no-slip boundary condition implies that the momentum should vanish at the boundary. One could based on this derive an error coefficient using the convective term of equation (5.3). This however, could potentially introduce situations where the error appear small due to cancellation. To avoid this, a boundary integral of the momentum magnitude is used instead. The error estimate is then formulated as

$$\hat{e} = \int_S \rho \left\| \mathbf{u} \right\| dS. \tag{5.8}$$

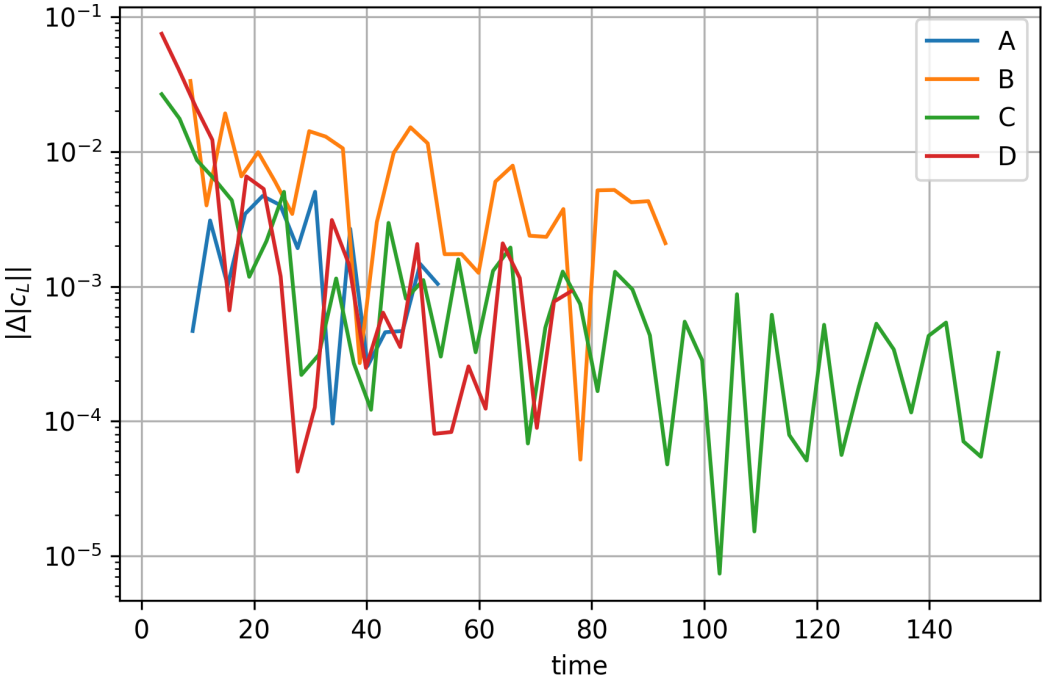


Figure 5.1: The evolution of the absolute change between consecutive peaks in the absolute lift coefficient in absolute w.r.t. time. Simulations A - D are shown. The graphs are used in assessing the convergence of the simulations towards steady periodic behaviour.

The resulting error estimates of simulations A - D can be found in table 5.4 or in figure 5.2.

Table 5.3: Resulting parameters of simulation of flow over a cylinder at $Re = 100$ for previous and current work.

	Ma	$\mathbf{c_D}$	$\mathbf{c_L}$	**St**
Liu[35]	-	1.35 ± 0.012	± 0.339	0.165
Calhoun[5]	-	1.330 ± 0.0014	± 0.298	0.175
Karagiozis[29]	0.05	1.317	± 0.320	0.168
	0.25	1.336	± 0.319	0.168
Canuto[6]	0.1	1.35 ± 0.0096	± 0.328	0.166
	0.2	1.36 ± 0.0093	± 0.332	0.165
A	0.1786	1.632 ± 0.052	± 0.922	0.161
B		1.404 ± 0.021	± 0.546	0.166
C		1.386 ± 0.014	± 0.449	0.162
D		1.254 ± 0.011	± 0.346	0.165

Table 5.4: The error estimate of simulations A - D.

A	**B**	**C**	**D**
0.122	0.051	0.061	0.045

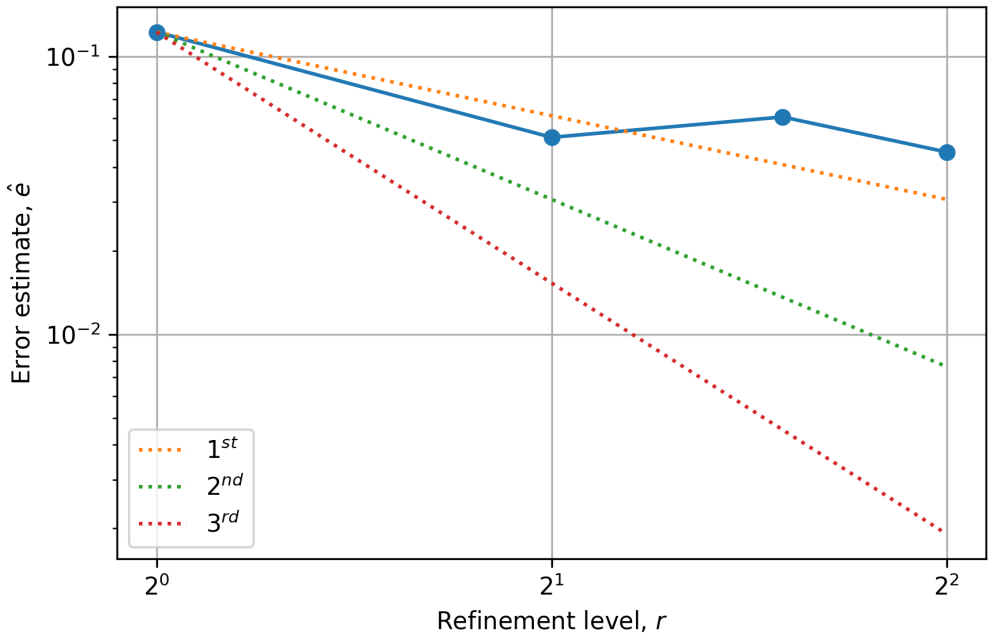


Figure 5.2: The error estimate of the imposed boundary condition w.r.t. refinement level of the grid.

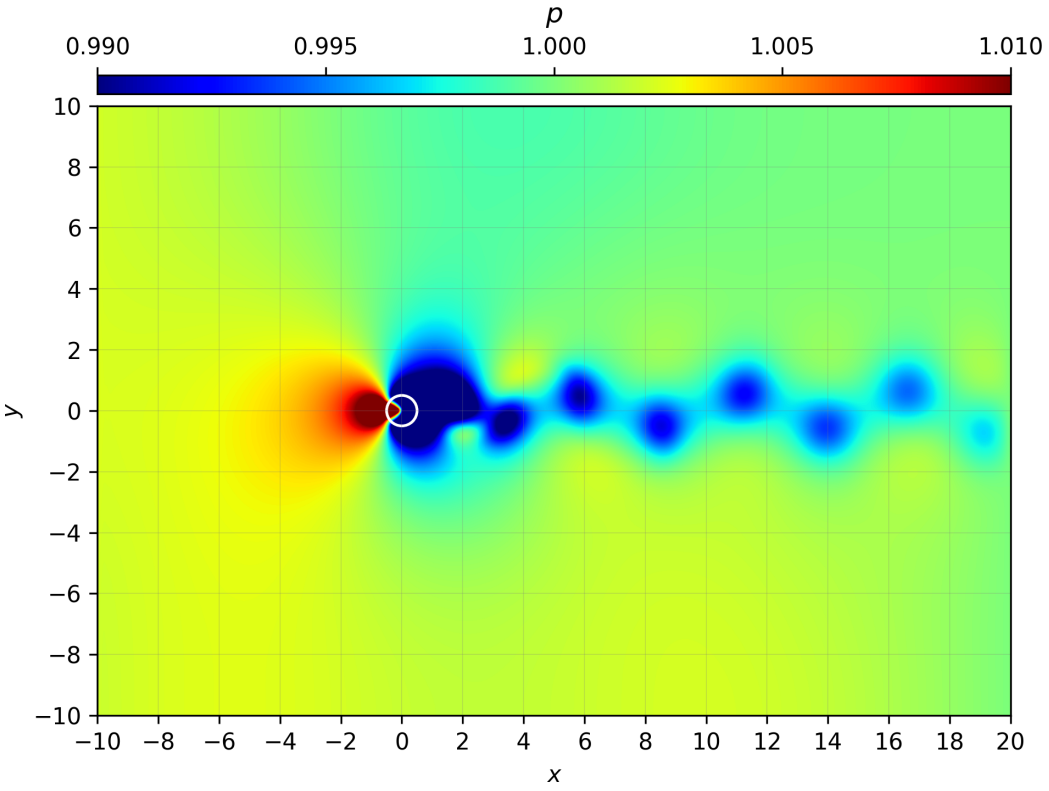


Figure 5.3: The pressure distribution of the flow in vicinity of the cylinder. As expected, a increase in pressure is observed in front of the cylinder and a pressure drop is observed in its wake. The pressure drop align with the vortices shown in 5. The cylinder is shown as a white circle. It is simulation D that is depicted and the time of the snapshot corresponds to that of 5 c.

There's a decrease in error between refinement level 1 and 2 similar to that one would expect from a first order method. Thereafter, the error surprisingly increase slightly between level 2 and 3 before ultimately, at refinement level 4, decreasing to a value slightly below that of refinement level 2. No clear pattern or trend could be inferred from the available data.

Interestingly, the drag coefficient of simulation B is substantially smaller than that of simulation A, but almost identical to simulation C. A significant decrease from simulation C to D is then observed. Further, the drag coefficient of simulation D is lower than that of all selected sources. The lift coefficient is slightly higher. However, the Strouhal number is in good agreement with [6] and [35]. Unlike [6][5][35], the variation in the drag coefficient seem to not be fully symmetric about the mean for simulation A - C. This might merely come down to the way the coefficient is reported, but it could also be due to some error introduced by the presented immersed boundary

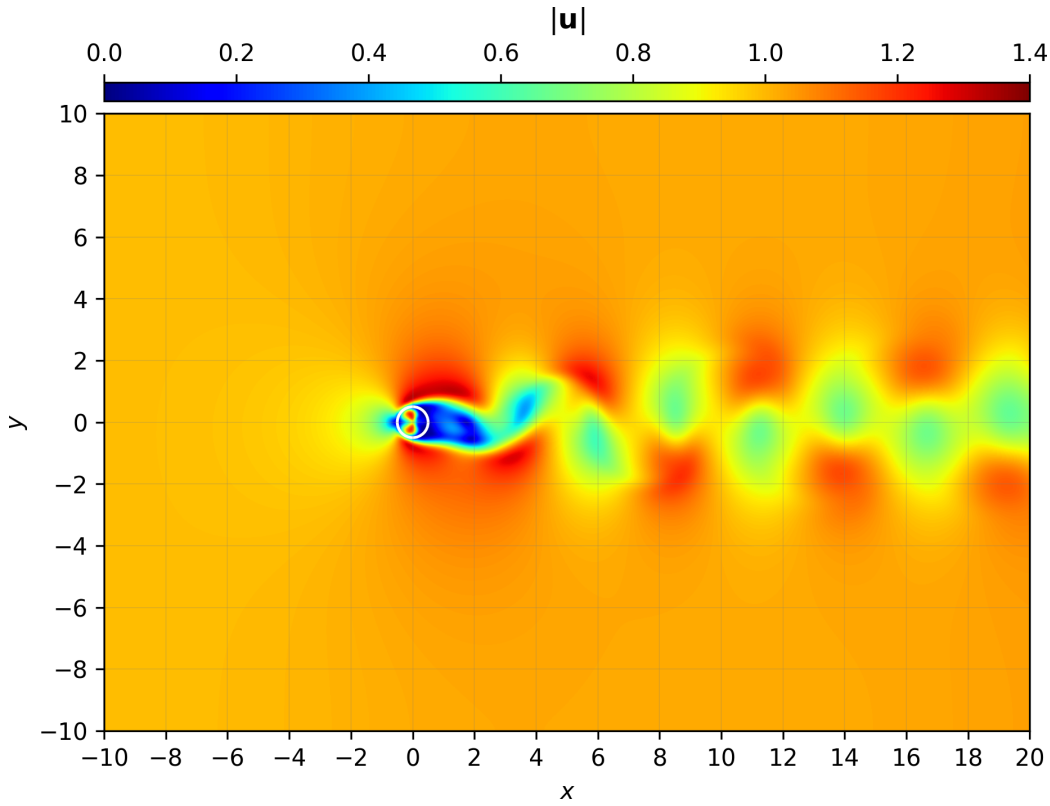


Figure 5.4: The velocity magnitude of the flow in vicinity of the cylinder. A stagnation point can be seen right in front of the cylinder and a velocity deficit is observed immediately behind the cylinder. Alternating fluctuations in the velocity magnitude can be seen further downstream. The cylinder is shown as a white circle. It is simulation D that is depicted and the time of the snapshot corresponds to that of 5 c.

method.

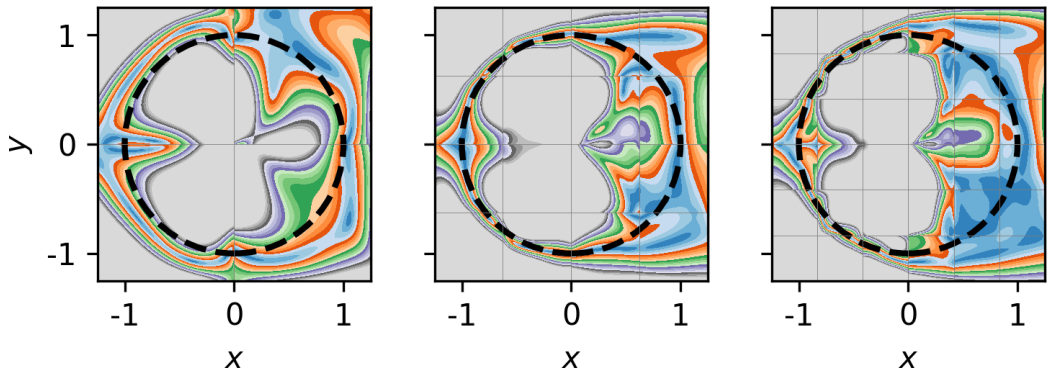
The strange behaviour of the error estimate and the drag coefficient could be indicative of some non-linear effect dependent on the proximity between nodal locations to the immersed boundary. It should be noted that the cylinder does occupy a very small portion of a uni-distant cartesian grid, meaning that quite few cells are used to simulate the immersed boundary. In fact, on the finest grid, the cylinder only occupies 44 cells. There could also be a dependence on the ratio of ghost fluid to fluid within a cell. The momentum flux of simulations A - D near the boundary is shown in figure 5. It appears that the basis functions have some difficulty in approximating the boundary condition for the coarser grids. Supporting the hypothesis of ill-conditioned cells, artifacts can be seen in those boundary cells where the ratio

of ghost fluid to fluid is small. These artifacts would hopefully vanish with further refinement of the grid. Unfortunately, without further investigation, no proper conclusion can be drawn from these quantities alone.

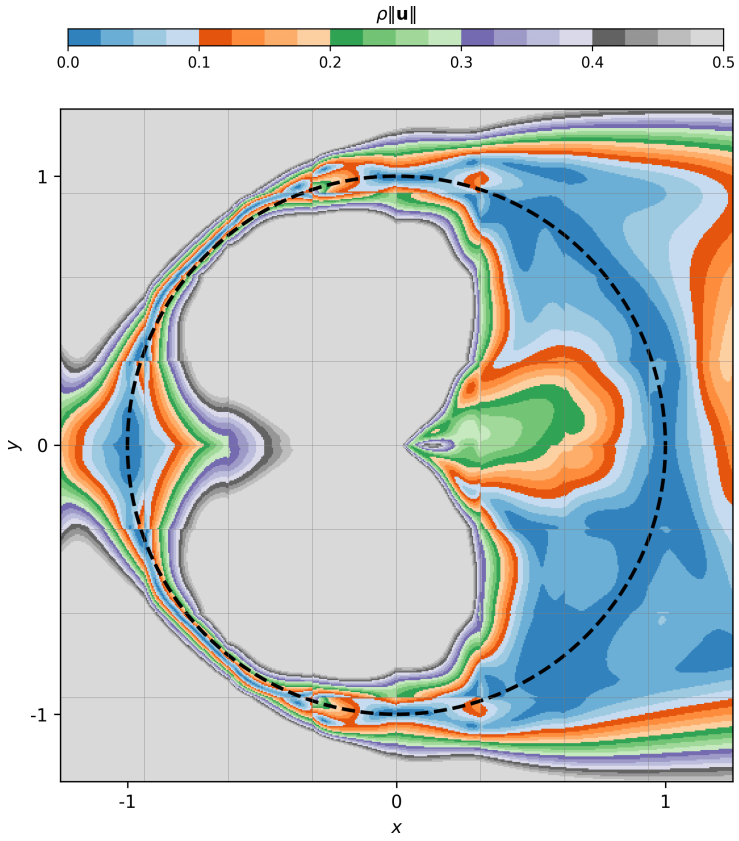
Even though the numerical quantities discussed previously in this chapter doesn't entirely align with what one would expect, the immersed boundary method is definitively able to reproduce the superficial behaviour expected. A clear vortex street of evenly spaced vortices form behind the cylinder in all simulations. The vortices being shed are of equal magnitude but alternating sign such as described in [27]. The described shedding behaviour is shown in figure 5.

Table 5.5: The specifications of the hardware on which all simulations were done.

Processor	Intel(R) Core(TM) i7-8700 CPU
Clock speed	3.20GHz
RAM	32GB
System architecture	64-bit



(a) Simulation A (b) Simulation B (c) Simulation C



(d) Simulation D

Figure 5.5: The momentum magnitude in the vicinity of the immersed boundary. Snapshots of simulations A - D are taken at points in time where no lift force is acting on the cylinder.

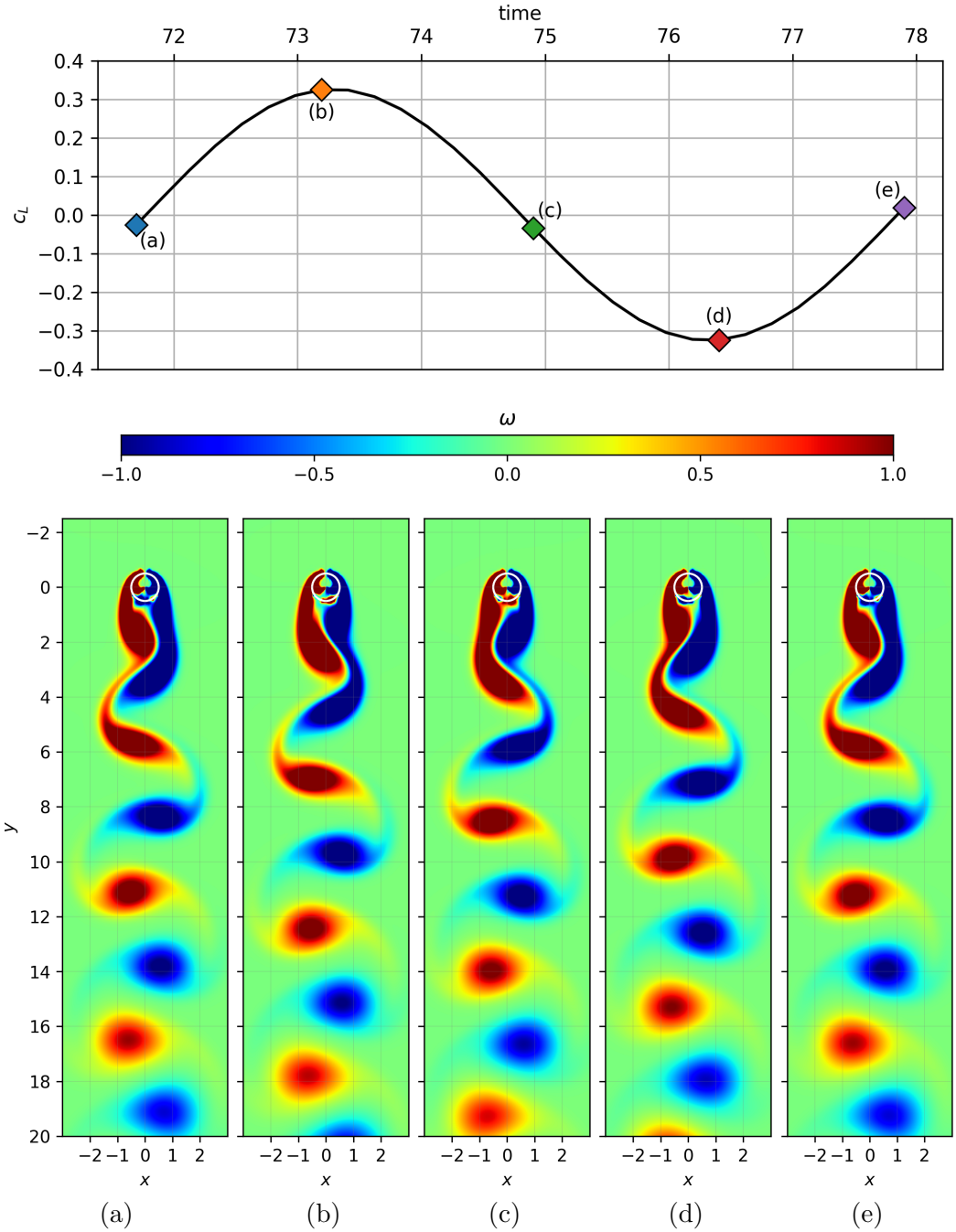
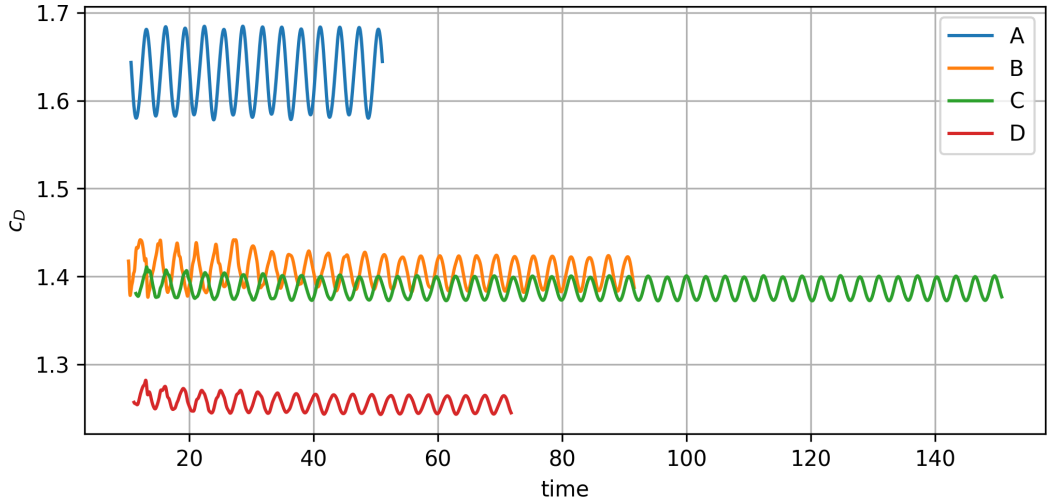
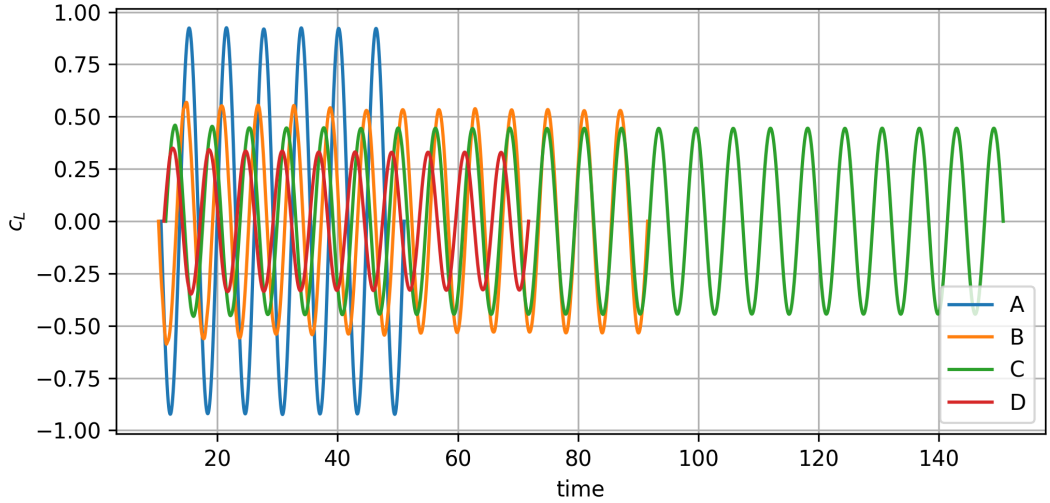


Figure 5.6: Snapshots of the vorticity field in the wake of the cylinder at different times during a single oscillation of the lift coefficient. Simulation D is shown.



(a) Drag



(b) Lift

Figure 5.7: The evolution of the drag and lift coefficient w.r.t time of simulations A - D.

Chapter 6

Conclusions

Various higher order methods, like the ENO-/WENO schemes and the DG-method, and their relation to the IBM has been reviewed. Based on the review, the DG-method coupled with the GCIBM was chosen as a starting point for the development of a high order accurate method capable of simulating complex geometries. A novel framework of basis construction was developed, and the construction of arbitrarily high order Legendre and Lagrange polynomials within this framework was shown. These basis functions were then used to construct the operators of a modal DG-method with nodal integration. An IBM method was developed for the DG-method. A no-slip Dirichlet boundary condition was enforced by setting the values of ghost nodes based on a reconstructed state sampled at corresponding image points. Radial basis functions were used to provide a reconstructed state at the image points which would only depend on the fluid nodes of the sampled cell. The task of developing a satisfactory von Neumann boundary condition was not completed due to time limitations.

The flow over a cylinder was chosen as a test case for which multiple simulations were run. Although periodic vortex shedding was achieved, The convergence history (shown in figure 5.1) leaves some to be desired. Error estimates were used to analyze the order of accuracy of the IBM, which was shown to be no higher than 1^{st} order. However, it is argued that this doesn't necessarily reflect the order of the method as the large domain in combination with equally spaced cells left very few cells to represent the boundary of the cylinder.

In spite of these issues, the results do show some promise. The drag- and lift coefficients and the Strouhal number were compared against values found in selected sources. While not in exact agreement, the drag- and lift coefficients did not deviate too far from the expected values. The Strouhal number ended up being fairly close to that found in the presented sources.

Additionally, the qualitative behaviour of the flow was in good agreement with what one would expect from the chosen test case.

In summary, the results are not entirely conclusive, but the method shows some promise and is need of further development and investigation.

Chapter 7

Further work

First and foremost, a more thorough analysis of the method needs to be completed. More simulations should be run at higher refinement levels and orders. Using equidistant cells proved to be a major bottleneck due to the size of the domain compared to the cylinder. To circumvent this, it is highly encouraged that some sort of grid refinement is implemented. The grid refinement, or h-refinement, could then easily be coupled with order refinement, or p-refinement, due to the way the basis functions are constructed. This is known as hp-refinement and is an active area of research within the DG-community[44]. A proper von Neumann boundary or a non-homogeneous Dirichlet condition must be implemented.

The use of special boundary fulfilling basis functions should also be investigated. Some additional computations might be required as it could be the case that a separate basis functions must be generated for each boundary cell. However, it is believed that the method of basis construction outlined in this thesis would significantly lessen the computational burden. More details regarding these basis functions can be found in appendix A.

After further development, some way of representing patient specific geometries should be implemented. The effects of non-ideal effects such as a slip boundary due to the patients mucous membrane, or fluid-structure-interactions between air flow and soft tissue should be investigated and implemented. Additionally, it would be highly interesting to extract and analyse sound from the simulations.

Bibliography

[1] Rémi Abgrall and Mario Ricchiuto. *High order methods for CFD - Encyclopedia of Computational Mechanics Second Edition*. Jan. 2017. DOI: 10.1002/9781119176817.ecm2112.

[2] F. Bassi and S. Rebay. "A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier–Stokes Equations". In: *Journal of Computational Physics* 131.2 (1997), pp. 267–279. ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.1996.5572.

[3] Gerd Brandstetter and Sanjay Govindjee. "A high-order immersed boundary discontinuous-Galerkin method for Poisson's equation with discontinuous coefficients and singular sources". In: *International Journal for Numerical Methods in Engineering* 101 (Dec. 2014). DOI: 10.1002/nme.4835.

[4] M. Braza, P. Chassaing, and H. Ha Minh. "Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder". In: *Journal of Fluid Mechanics* 165 (1986), pp. 79–130. DOI: 10.1017/S0022112086003014.

[5] Donna Calhoun. "A Cartesian Grid Method for Solving the Two-Dimensional Streamfunction-Vorticity Equations in Irregular Regions". In: *Journal of Computational Physics* 176.2 (2002), pp. 231–275. ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.2001.6970.

[6] Daniel Canuto and Kunihiko Taira. "Two-dimensional compressible viscous flow around a circular cylinder". In: *Journal of Fluid Mechanics* 785 (Dec. 2015), pp. 349–371. DOI: 10.1017/jfm.2015.635.

[7] G. Chavent and G. Salzano. "A finite-element method for the 1-D water flooding problem with gravity". In: *Journal of Computational Physics* 45.3 (1982), pp. 307–344. ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(82)90107-3.

[8] Guy Chavent and Bernardo Cockburn. "The local projection $P^0 - P^1$-discontinuous-Galerkin finite element method for scalar conservation laws". In: *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique* 23.4 (1989), pp. 565–592.

[9] Juan Cheng and Chi-Wang Shu. "High order schemes for CFD: A review". In: *Jisuan Wuli/Chinese Journal of Computational Physics* 26 (Sept. 2009).

[10] Shu Chi-Wang. "High-order Finite Difference and Finite Volume WENO Schemes and Discontinuous Galerkin Methods for CFD". In: *International Journal of Computational Fluid Dynamics* 17.2 (2003), pp. 107–118. DOI: 10.1080/1061856031000104851.

[11] Bernardo Cockburn, Suchung Hou, and Chi-Wang Shu. "The Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws. IV: The Multidimensional Case". In: *Mathematics of Computation* 54.190 (1990), pp. 545–581. DOI: https://doi.org/10.2307/2008501.

[12] Bernardo Cockburn, George Karniadakis, and Chi-Wang Shu, eds. *Discontinuous Galerkin methods*. en. Lecture notes in computational science and engineering. Berlin, Germany: Springer, Sept. 2011.

[13] Bernardo Cockburn, San-Yih Lin, and Chi-Wang Shu. "TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems". In: *Journal of Computational Physics* 84.1 (1989), pp. 90–113. ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(89)90183-6.

[14] Bernardo Cockburn and Chi-Wang Shu. "Runge–Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems". In: *Journal of Scientific Computing* 16 (Sept. 2001), pp. 173–261. DOI: 10.1023/A:1012873910884.

[15] Bernardo Cockburn and Chi-Wang Shu. "The Runge–Kutta Discontinuous Galerkin Method for Conservation Laws V: Multidimensional Systems". In: *Journal of Computational Physics* 141.2 (1998), pp. 199–224. ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.1998.5892.

[16] Bernardo Cockburn and Chi-Wang Shu. "TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws II: General Framework". In: *Mathematics of Computation* 52.186 (1989), pp. 411–435. ISSN: 00255718, 10886842. (Visited on 06/09/2023).

[17] Mustafa E. Danis and Jue Yan. "A new direct discontinuous Galerkin method with interface correction for two-dimensional compressible Navier-Stokes equations". In: *Journal of Computational Physics* 452 (2022), p. 110904. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2021.110904.

[18] Mustafa E. Danis and Jue Yan. "A new direct discontinuous Galerkin method with interface correction for two-dimensional compressible Navier-Stokes equations". In: *Journal of Computational Physics* 452 (2022). DOI: https://doi.org/10.1016/j.jcp.2021.110904.

[19] Michael Dumbser et al. "Central Weighted ENO Schemes for Hyperbolic Conservation Laws on Fixed and Moving Unstructured Meshes". In: *SIAM Journal on Scientific Computing* 39.6 (2017), A2564–A2591. DOI: 10.1137/17M1111036.

[20] R. Fedkiw et al. "A non-oscillatory Eulerian approach to interfaces in mul-timaterial ows". In: *Journal of Computer Physics* (Jan. 2003).

[21] Krzysztof J. Fidkowski and David L. Darmofal. "A triangular cut-cell adaptive method for high-order discretizations of the compressible Navier-Stokes equations". In: *Journal of Computational Physics* 225.2 (2007), pp. 1653–1672. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2007.02.007.

[22] Markus Geisenhofer, Florian Kummer, and Björn Müller. "A discontinuous Galerkin immersed boundary solver for compressible flows: Adaptive local time stepping for artificial viscosity-based shock-capturing on cut cells". In: (Oct. 2019).

[23] R. Ghias, R. Mittal, and H. Dong. "A sharp interface immersed boundary method for compressible viscous flows". In: *Journal of Computational Physics* 225.1 (2007), pp. 528–553. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2006.12.007.

[24] Sergei K. Godunov and I. Bohachevsky. "Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics". In: *Matematičeskij sbornik* 47(89).3 (1959), pp. 271–306. URL: https://hal.science/hal-01620642.

[25] Ami Harten et al. "Uniformly high order accurate essentially non-oscillatory schemes, III". In: *Journal of Computational Physics* 71.2 (1987), pp. 231–303. ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(87)90031-3.

[26] Jan S. Hesthaven and Tim Warburton. *Nodal Discontinuous Galerkin Methods*. Springer, 2008. ISBN: 9780387720654.

[27] E L Houghton and P W Carpenter. *Aerodynamics for Engineering Students*. 5th ed. Woburn, MA: Butterworth-Heinemann, May 2014.

[28] *ICOSAHOM 2023*. URL: https://icosahom2023.org/.

[29] K. Karagiozis, R. Kamakoti, and C. Pantano. "A low numerical dissipation immersed interface method for the compressible Navier–Stokes equations". In: *Journal of Computational Physics* 229.3 (2010), pp. 701–727. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2009.10.005.

[30] M. Khalili, Martin Larsson, and Bernhard Müller. "High order ghost-point immersed boundary method for viscous compressible flows based on summation-by-parts operators". In: *International Journal for Numerical Methods in Fluids* 89 (Nov. 2018). DOI: 10.1002/fld.4696.

[31] Jiaqing Kou et al. "Immersed boundary method for high-order flux reconstruction based on volume penalization". In: *Journal of Computational Physics* 448 (2022), p. 110721. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2021.110721.

[32] Dennis Krause and Florian Kummer. "An Incompressible Immersed Boundary Solver for Moving Body Flows using a Cut Cell Discontinuous Galerkin Method". In: *Computers & Fluids* 153 (May 2017). DOI: 10.1016/j.compfluid.2017.05.008.

[33] Mouna Laroussi, Mohamed Djebbi, and Mahmoud Moussa. "Triggering vortex shedding for flow past circular cylinder by acting on initial conditions: A numerical study". In: *Computers & Fluids* 101 (Sept. 2014), pp. 194–207. DOI: 10.1016/j.compfluid.2014.05.034.

[34] Adrián Lew and Gustavo Buscaglia. "A discontinuous-Galerkin-based immersed boundary method". In: *International Journal for Numerical Methods in Engineering* 76 (Oct. 2008), pp. 427–454. DOI: 10.1002/nme.2312.

[35] C. Liu, X. Zheng, and C.H. Sung. "Preconditioned Multigrid Methods for Unsteady Incompressible Flows". In: *Journal of Computational Physics* 139.1 (1998), pp. 35–57. ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.1997.5859.

[36] Xu-Dong Liu, Stanley Osher, and Tony Chan. "Weighted Essentially Non-oscillatory Schemes". In: *Journal of Computational Physics* 115.1 (1994), pp. 200–212. ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.1994.1187.

[37] Hailiang Liu and Jue Yan. "The Direct Discontinuous Galerkin (DDG) Methods for Diffusion Problems". In: *SIAM Journal on Numerical Analysis* 47.1 (2009), pp. 675–698. DOI: https://doi.org/10.1137/080720255.

[38] Gianmarco Mengaldo et al. "A Guide to the Implementation of Boundary Conditions in Compact High-Order Methods for Compressible Aerodynamics". In: *7th AIAA Theoretical Fluid Mechanics Conference* (2014). DOI: https://doi.org/10.2514/6.2014-2923.

[39] Mads Henrik Strand Moxness. "The Influence of the Nasal Airway in Obstructive Sleep Apnea". PhD thesis. Norwegian University of Science and Technology, 2018.

[40] Bernhard Müller and Tobias Winter. *Master's agreement.*

[41] Charles S Peskin. "Flow patterns around heart valves: A numerical method". In: *Journal of Computational Physics* 10.2 (1972), pp. 252–271. ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(72)90065-4.

[42] Bastian E Rapp. *Engineering Mathematics.* Elsevier, 2017. Chap. 9, p. 260.

[43] W H Reed and T R Hill. "Triangular mesh methods for the neutron transport equation". In: (Oct. 1973).

[44] Kevin Schaal et al. "Astrophysical hydrodynamics with a high-order discontinuous Galerkin scheme and adaptive mesh refinement". In: *Monthly Notices of the Royal Astronomical Society* 453.4 (Nov. 2015), pp. 4279–4301. DOI: 10.1093/mnras/stv1859. URL: https://doi.org/10.48550/arXiv.1506.06140.

[45] CHI-WANG SHU. *Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws.* Nov. 1997.

[46] Yu-Heng Tseng and Joel H. Ferziger. "A ghost-cell immersed boundary method for flow in complex geometry". In: *Journal of Computational Physics* 192.2 (2003), pp. 593–623. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2003.07.024.

[47] Panagiotis Tsoutsanis and Michael Dumbser. "Arbitrary high order central non-oscillatory schemes on mixed-element unstructured meshes". In: *Computers & Fluids* 225 (2021), p. 104961. ISSN: 0045-7930. DOI: https://doi.org/10.1016/j.compfluid.2021.104961.

[48] Bram van Leer. "Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method". In: *Journal of Computational Physics* 32.1 (1979), pp. 101–136. ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(79)90145-1.

[49] *VirtuOSA*. URL: https://www.sintef.no/en/projects/2020/virtuosa-virtual-surgery-in-the-upper-airways-new-solutions-to-obstructive-sleep-apnea-treatment/ (visited on 06/10/2023).

[50] Z. Wang. "Adaptive High-Order Methods in Computational Fluid Dynamics". In: *Science China Physics, Mechanics & Astronomy* 59 (Jan. 2011). DOI: 10.1007/s11433-015-5706-3.

[51] Z. Wang et al. "High-Order CFD Methods: Current Status and Perspective". In: *International Journal for Numerical Methods in Fluids* 72 (Jan. 2013), pp. 811–845. DOI: 10.1002/fld.3767.

[52] Hong Xiao et al. "An Immersed Discontinuous Galerkin Method for Compressible Navier-Stokes Equations on Unstructured Meshes". In: *International Journal for Numerical Methods in Fluids* 91 (Aug. 2019). DOI: 10.1002/fld.4765.

[53] Xu-Jiu Zhang et al. "New Immersed Boundary Method on the Adaptive Cartesian Grid Applied to the Local Discontinuous Galerkin Method". In: *Chinese Journal of Mechanical Engineering* 31 (Dec. 2018). DOI: 10.1186/s10033-018-0222-9.

Appendix A

Abstract

The methods presented in chapter 3 are also applicable to other fundamental bases other than the monomial basis. Owing to this, bases for which various boundary conditions are fulfilled exactly can be used. On encouragement from my supervisor, we wrote an abstract of a paper detailing the use of such bases. The abstract was submitted to and accepted for a contributed talk at the International Conference on Spectral and High Order Methods of 2023 (ICOSAHOM2023)[28]. The abstract is included on the next page.

Constructing a basis for enforcing immersed boundaries in discontinuous Galerkin methods

Tobias Winter[1], Bernhard Müller[2]

([1]) Department of Energy and Process Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway
tobiaswi@stud.ntnu.no
([2]) Department of Energy and Process Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway
bernhard.muller@ntnu.no

A novel approach for imposing immersed Dirichlet and Neumann boundary conditions in discontinuous Galerkin (DG) methods is presented. The approach is based on constructing a functional basis for which the linear span only includes functions satisfying the boundary condition being imposed at an immersed boundary. The basis is constructed from a starting basis multiplied by some modifier function, e.g., a signed distance function. Thereafter, a linear combination of the modified basis is found that satisfies the orthonormality constraint imposed on the mass matrix. While a unique basis must be constructed for every element intersected by an immersed boundary (IB), once constructed, those IB elements can be treated almost identically to non-IB elements. The approach is particularly interesting as it combines the convenience and quality of Cartesian grids, the compactness and the high order of DG methods, and retains this order near boundaries while imposing boundary conditions exactly. A 2D implementation of the approach is demonstrated using the heat equation, the Burgers' equation, the compressible Euler equations, and the compressible Navier-Stokes equations.

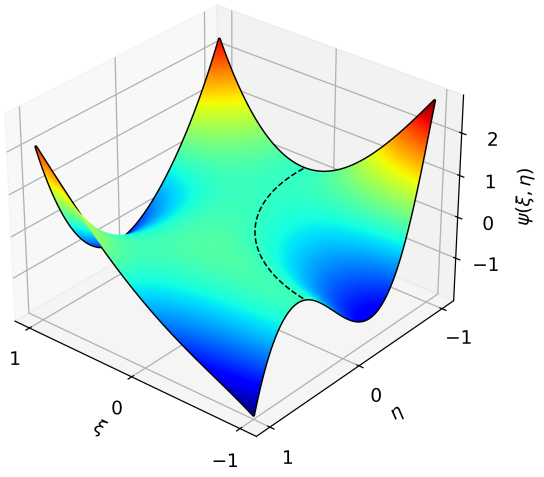


Figure 1: Graph of a function $\psi(\xi, \eta)$ from a constructed basis for an immersed boundary intersected element satisfying a Dirichlet boundary condition. The immersed boundary is shown as a dashed circle segment.

Appendix B

GUI

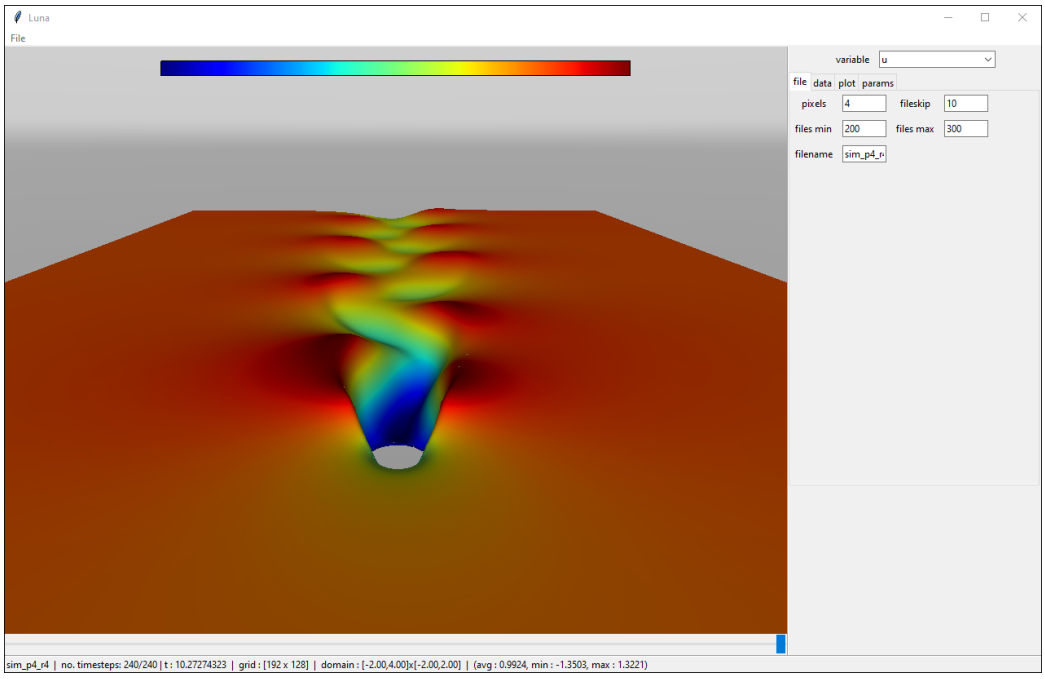


Figure B.1: A custom GUI developed to debug the DG-solver. Depicted is simulation D, with the u-component of the velocity being shown as a surface plot. The part of the domain occupied by the cylinder is hidden.

Developing a CFD-solver from the ground up is not especially hard. However, developing a CFD-solver from the ground up that actually do what you intend it to do is quite difficult. Thus, being able to analyze the output of the solver is paramount. Developing a DG-solver in particular poses some additional challenges in this regard. Firstly, using a polynomial basis means that

there exist less available software capable of properly displaying the solution than for more traditional CFD-methods like the finite volume method. Secondly, understanding the behaviour of the discontinuities at cell interfaces is not necessarily straightforward given just the traditional 2D-representation of color mapped values. Another issue that applies to CFD in general is the large amounts of data produced by transient simulations. This means that analyzing solution output with tools using software rendering, such as pythons Matplotlib, becomes tedious at best, and gruelling at worst.

For the reasons stated in the previous paragraph, a custom GUI was developed. The Tkinter python library was used for the interface elements and the OpenGL graphics API for the rendering. The program loads a time series of conserved variables and some auxiliary data like grid specifications. Vertices of the conserved- and some derived variables are generated using the proper operators. The vertices are sent to the GPU. Since the entire time series is sent to the GPU at once, changing the displayed quantity or the time can be done with virtually no latency. This makes for a pleasant debugging experience.

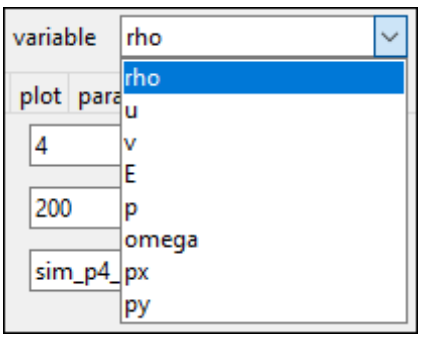
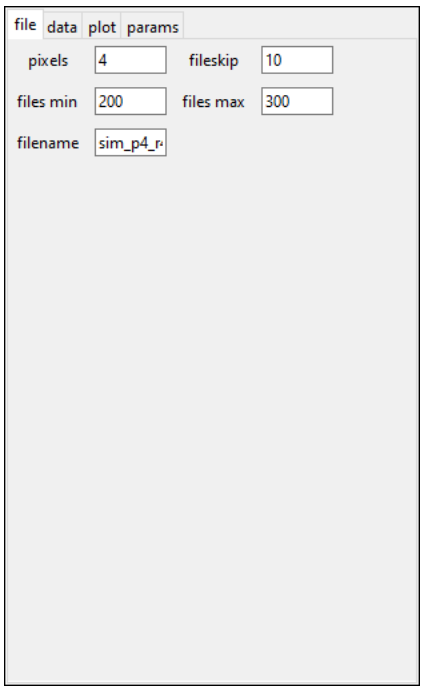
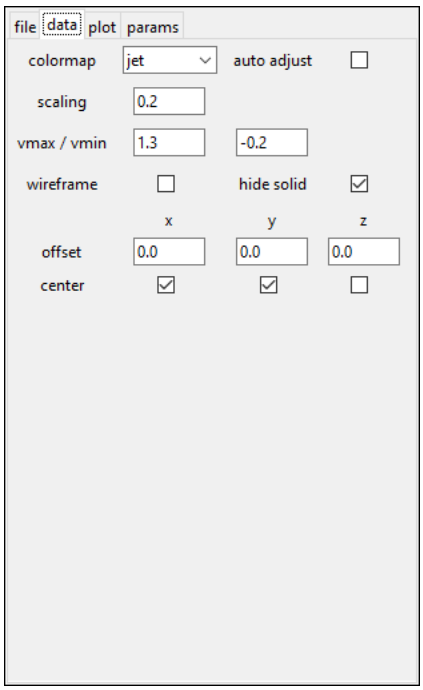


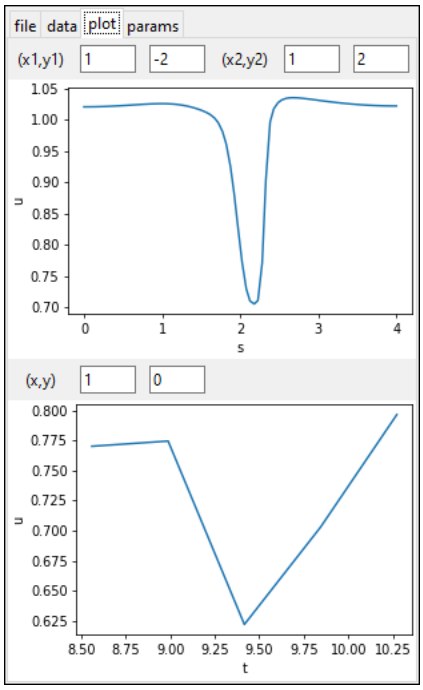
Figure B.2: Drop down menu of what variable to display.



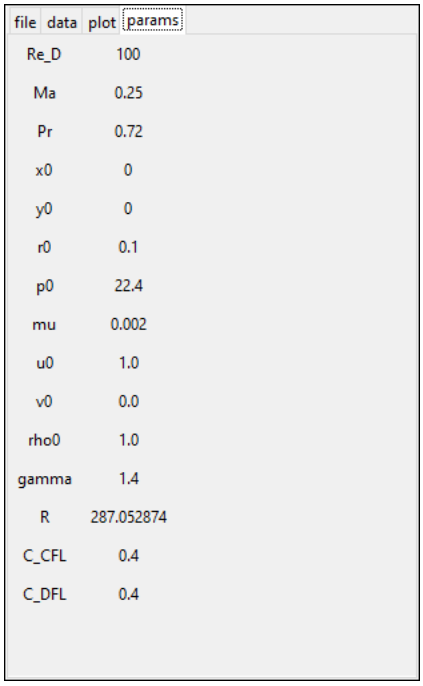
(a) File tab



(b) Data tab



(c) Plotting tab



(d) Parameter tab

Figure B.3: Various tabs of the GUI.

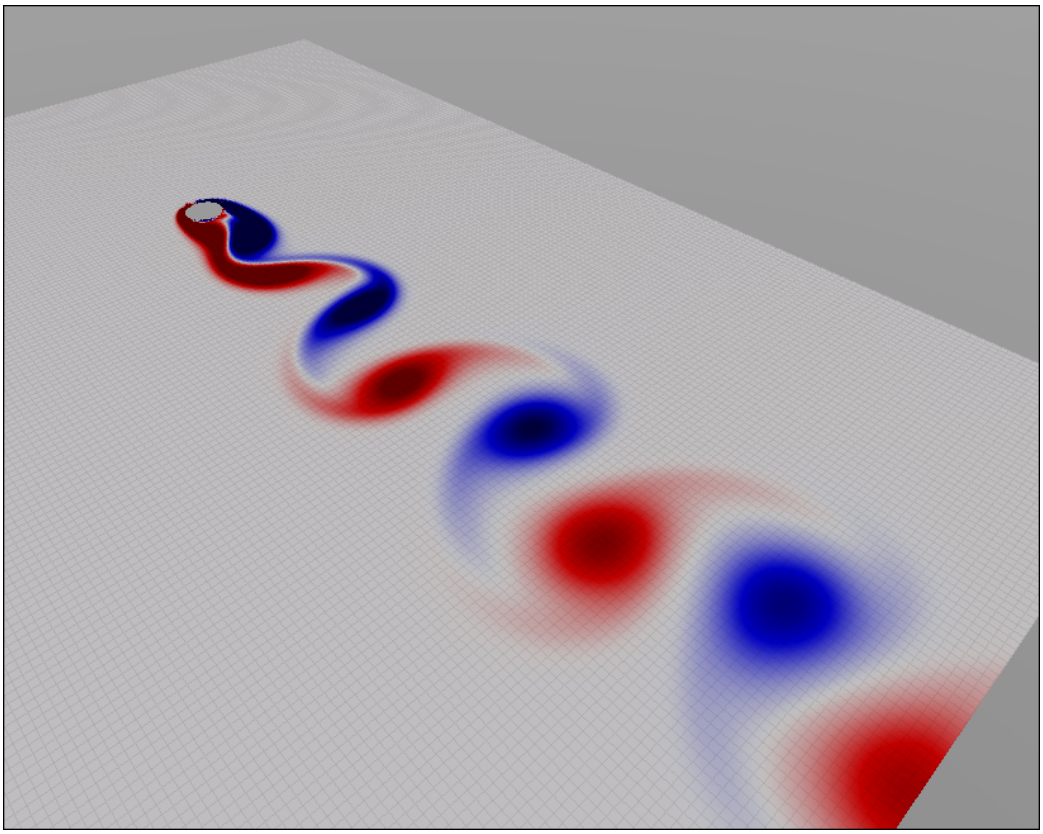


Figure B.4: Vorticity field of simulation D rendered as a part of the GUI. Wire frame is turned on such that each individual cell can be seen. The plot appears darkened due to a special shading technique meant to provide the viewer with information regarding incidence.

Appendix C

Master's agreement

The Master's agreement is an agreement between student, supervisor, and department, pertaining to the scope of the thesis to be written. The background and objectives of the Master's agreement are included on the next page.

NTNU

Norwegian University
of Science and Technology

Department of Energy
and Process Engineering

EPT-M-2023

MASTER THESIS

for

Student Tobias Winter

Spring 2023

Higher Order Method for Flow Simulations in Complex Geometries
Høyere ordens metode for strømningssimuleringer i komplekse geometrier

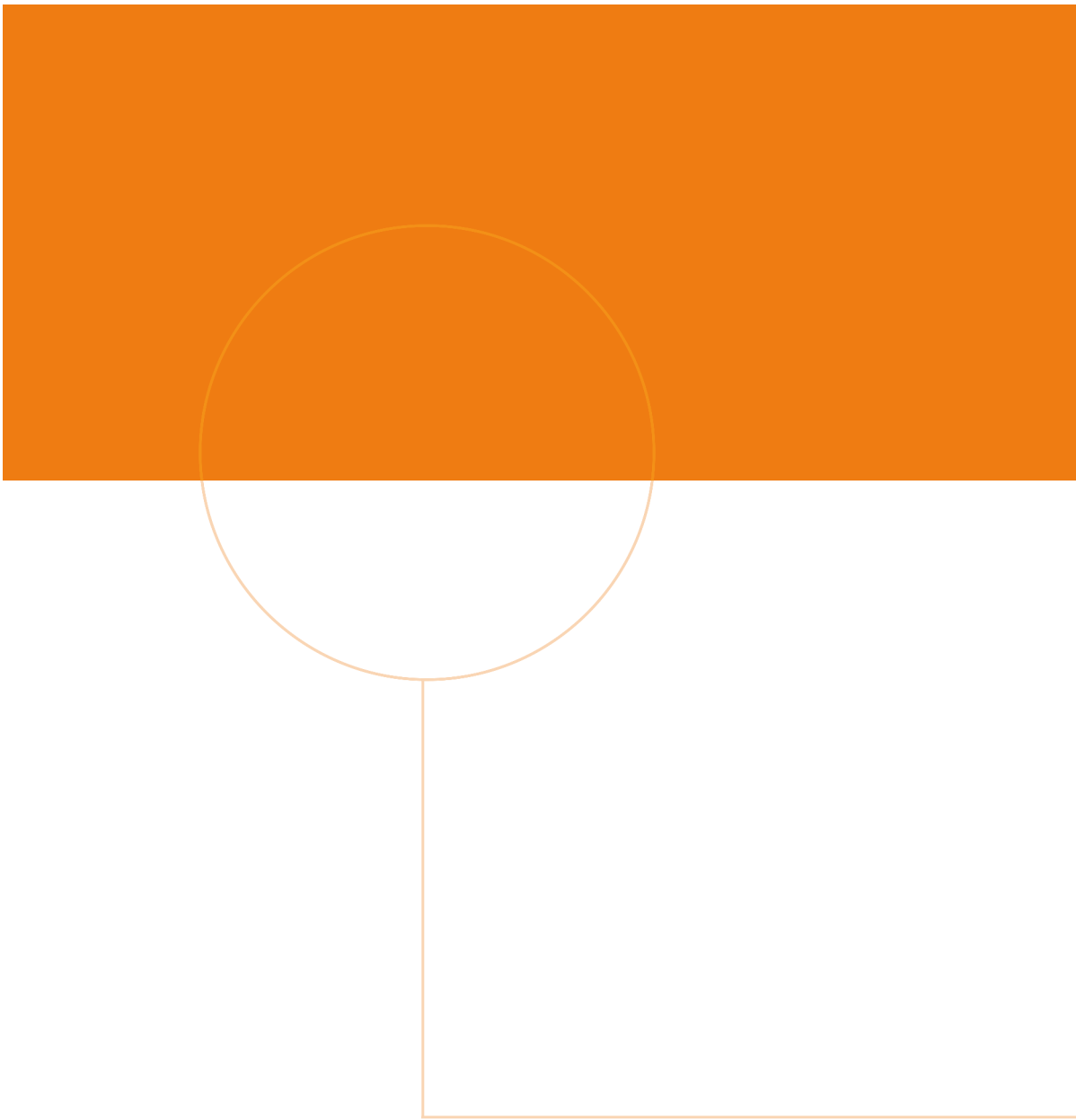
Background and objective

The master's project will be a part of the larger research project "Virtual Surgery in the Upper Airways - New Solutions to Obstructive Sleep Apnea Treatment (VirtuOSA)" funded by the Research Council of Norway. Obstructive sleep apnea (OSA) is a sleep-related breathing disorder caused by repetitive collapses of the upper airways during sleep, resulting in reduced breathing, oxygen desaturation and sleep disturbances, thus leading to cardiovascular diseases. In VirtuOSA, an interdisciplinary team with experts from St. Olavs Hospital, NTNU and SINTEF aims to develop a software tool for treatment of OSA through virtual surgery.

In the master's project, a higher order method for flow simulations based on the compressible Navier-Stokes equations will be investigated for complex geometries. General complex geometries will be considered, which are also relevant for geometries in the upper airways of OSA patients. The goal of the project is to investigate a higher order method for more accurate discretizations of the compressible Navier-Stokes equations than the frequently used second order methods. The master's project will be a part of VirtuOSA.

The following tasks are to be considered:

1. to review the literature for higher order discretization methods like the discontinuous Galerkin method and other higher order methods,
2. to investigate a higher order method for flow in complex geometries,
3. to develop, implement and test a higher order method for the 2D compressible Navier-Stokes equations in complex geometries,
4. to verify and validate the higher order method for benchmark test cases.



NTNU

Norwegian University of
Science and Technology