



DEGREE PROJECT, IN MATHEMATICS , SECOND LEVEL  
*STOCKHOLM, SWEDEN 2015*

# Investigation of Outflow Boundary Conditions for Convection-Dominated Incompressible Fluid Flows in a Spectral Element Framework

ERIK BOSTRÖM

KTH ROYAL INSTITUTE OF TECHNOLOGY

SCI SCHOOL OF ENGINEERING SCIENCES



# Investigation of Outflow Boundary Conditions for Convection-Dominated Incompressible Fluid Flows in a Spectral Element Framework

ERIK BOSTRÖM

Master's Thesis in Scientific Computing (30 ECTS credits)  
Master Programme in Mathematics (120 credits)  
Royal Institute of Technology year 2015  
Supervisor at Argonne National Laboratory was Oana Marin  
Supervisor at KTH was Philip Schlatter  
Examiner was Michael Hanke

TRITA-MAT-E 2015: 11  
ISRN-KTH/MAT/E--15/11--SE

Royal Institute of Technology  
*School of Engineering Sciences*

**KTH SCI**  
SE-100 44 Stockholm, Sweden

URL: [www.kth.se/sci](http://www.kth.se/sci)



# Abstract

In this thesis we implement and study the effects of different convective outflow boundary conditions for the high order spectral element solver Nek5000 in the context of solving convective-dominated fluid flow problems. By numerical testing we show that the convective boundary conditions preserve the spatial and temporal convergence rates of the solver. We also study highly convective test cases such as a single vortex propagating through the outflow boundary, and the typical Kármán vortex shedding problem to analyze the accuracy and stability. A detailed comparison with the natural boundary condition that corresponds to the variational form of the incompressible Navier–Stokes equations (the Nek5000 “O” condition), and a stabilized version of it (by Dong et al. (2014)), are also presented.

Our results show a major advantage of using the convective boundary conditions over the natural counterpart in solving convective problems, both according to stability and accuracy. Analytic and numerical results show that the natural condition has big stability problems for high Reynolds numbers, which make the use of stabilization methods or damping regions crucial. But, the (Dong) stabilized natural condition does not improve accuracy, and damping regions are computationally expensive. The convective conditions show very good accuracy if its convection speed is approximated accurately, and our results indicate that it can be used without damping regions efficiently. Our results also show that the magnitude of reflections significantly depends on the amplitude of the disturbances that move through the boundary. The convective boundary condition can handle large disturbances without producing significant reflections, while the natural one or a stabilized version of it in general can not.



# Referat

## Undersökning av Utflödesrandvillkor för Konvektiv-Dominanta Inkompressibla Flöden i ett Spektral-Element Ramverk

I det här examensarbetet har vi implementerat och studerat effekterna av olika konvektiva randvillkor för spektral-element lösaren Nek5000 vid beräkningar av konvektivt dominanta flödesproblem. Med hjälp av numeriska tester bevisar vi att de nya implementationerna bevarar lösarens konvergens i både rum och tid. Vi studerar noggrannheten hos de konvektiva randvillkoren genom konvektivt dominanta testfall i form av en ensam virvel som propagerar genom utflödet, samt det klassiska Kármán-virvel-gata problemet. En detaljerad jämförelse med det naturliga randvillkoret tillhörande den svaga formuleringen av de inkompressibla Navier-Stokes ekvationerna (Nek5000 "O") och en stabiliserad version av denna är också presenterade.

Våra resultat visar tydliga fördelar med att använda de konvektiva randvillkoren mot det naturliga vid lösningar av konvektiva problem, både stabilitetsmässigt och noggrannhetsmässigt. Analytiska och numeriska resultat visar att det naturliga randvillkoret har stora stabilitetsproblem vid höga Reynolds-tal, vilket medför att specifika stabilitetsversioner eller dämpningsregioner måste användas. Men stabiliserande naturliga randvillkor (Dong) förbättrar inte noggrannheten och dämpningsregioner är dyra beräkningsmässigt. De konvektiva randvillkoren har uppvisat en väldigt god noggrannhet om konvektionshastigheten i villkoren är noggranna approximationer. Analyser av amplituden hos reflektioner har också undersökts. Våra resultat visar ett signifikant linjärt förhållande mellan storleken på störningar som genomborrar utflödes-randen och de reflektioner dessa störningar skapar. De konvektiva randvillkoren visar sig klara starka störningar bra, vilket det naturliga och den stabiliserade versionen av det naturliga randvillkoret generellt sett inte gör.





# Acknowledgements

There are several persons that have contributed to this project which I would like to mention. First of all I want to devote special thanks to Philipp Schlatter. I am extremely grateful that you introduced me to this project. It has helped me to develop in many areas, and strengthened my interests in computational fluid dynamics. I also want to thank you for all interesting discussions during this period. Special thanks I also want to devote to Oana Marin for all tips, all comments about the writing and coding, but also for your thoughtfulness. Other persons I want to thank for their contribution are Adam Peplinski for all interesting discussions regarding Nek5000, Ekaterina Ezhova for commenting the report, and Ricardo Vinuesa for borrowing me your video clip for the oral presentation.

In addition to the people above I also feel I want to name some other persons that have meant a lot to me during the master-years. I want to thank Babak Maboudi-Afgham partly for commenting the report, but also for all support you have given me, both regarding the studies and personal. You have with your great attitude made me believe that hard work pays off in the long run. I also want to thank Tahar Nabil and Jerker Nilsson for all support you have given me during lab-work and exam preparations. I wish you all the best luck in the future.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation and background . . . . .	1
1.2	Fluid dynamics – Analytic and numerical issues . . . . .	3
1.2.1	Numerical considerations . . . . .	4
1.3	Outline . . . . .	6
<b>2</b>	<b>Navier–Stokes discretization</b>	<b>7</b>
2.1	Introduction . . . . .	7
2.2	Domain decomposition . . . . .	8
2.3	Spatial discretization . . . . .	11
2.3.1	Functional spaces . . . . .	11
2.3.2	Weak formulation . . . . .	12
2.3.3	The $\mathbb{P}_N \times \mathbb{P}_N$ method . . . . .	14
2.3.4	The $\mathbb{P}_N \times \mathbb{P}_{N-2}$ method . . . . .	15
2.3.5	Semi-discrete matrix form . . . . .	15
2.4	Time discretization . . . . .	18
2.4.1	The BDFk/EXTk scheme . . . . .	18
2.4.2	The operator integration factor splitting scheme (OIFS) . . .	20
2.5	Operator splitting – The fractional step method . . . . .	22
<b>3</b>	<b>Natural outflow boundary conditions</b>	<b>25</b>
3.1	Introduction . . . . .	25
3.2	Natural outflow conditions for the incompressible Navier–Stokes equations . . . . .	27
3.3	Energy balance . . . . .	28
3.4	A stabilized natural boundary condition . . . . .	29
3.5	Problems with natural boundary conditions in pipe flow computations	30
3.5.1	A hidden pressure condition . . . . .	30
3.5.2	Non-uniqueness . . . . .	30
<b>4</b>	<b>Convective outflow boundary conditions – Theory and implementation</b>	<b>33</b>
4.1	Introduction . . . . .	33

## CONTENTS

4.2	Exact "Non-reflecting" boundary conditions . . . . .	33
4.2.1	The one dimensional case . . . . .	33
4.2.2	Higher dimensions . . . . .	34
4.3	The Sommerfeld radiation condition . . . . .	38
4.4	Estimating the convection speed . . . . .	39
4.4.1	The Orlanski scheme . . . . .	39
4.4.2	Exponential weighted moving average . . . . .	40
4.5	Nek5000 implementations . . . . .	42
4.5.1	BDFk/EXTk . . . . .	42
4.5.2	Linear characteristic method . . . . .	43
4.5.3	Curved characteristics method (OIFS) . . . . .	44
<b>5</b>	<b>Numerical results</b>	<b>45</b>
5.1	Introduction . . . . .	45
5.2	Temporal order estimation, unsteady flow . . . . .	46
5.3	Spatial convergence, Kovasznay flow . . . . .	49
5.3.1	Convergence rates . . . . .	50
5.3.2	Accuracy . . . . .	52
5.4	Convecting vortex . . . . .	56
5.4.1	$\mathbb{P}_N \times \mathbb{P}_N$ vs. $\mathbb{P}_N \times \mathbb{P}_{N-2}$ . . . . .	65
5.4.2	Robustness analysis . . . . .	66
5.4.3	Stability and reflections . . . . .	68
5.5	Flow past a circular cylinder . . . . .	70
5.5.1	$Re = 100$ . . . . .	72
5.5.2	$Re = 1000$ . . . . .	77
<b>6</b>	<b>Discussion and conclusions</b>	<b>83</b>
6.1	Outlook . . . . .	85
	<b>Appendices</b>	<b>85</b>
<b>A</b>	<b>Mathematical concepts and formulas</b>	<b>87</b>
A.1	Tensor products . . . . .	87
A.2	Orthogonal Lagrange polynomials . . . . .	88
A.2.1	Legendre polynomials . . . . .	88
A.2.2	GLL polynomials . . . . .	89
A.2.3	GL polynomials . . . . .	89
A.3	The inf-sup condition . . . . .	90
A.4	Backward differentiating scheme (BDF) . . . . .	90
A.5	Extrapolation scheme (EXT) . . . . .	91
A.6	Alternate forms of the Navier–Stokes equations . . . . .	92
A.6.1	Alternate forms of the convective term . . . . .	92
A.6.2	Alternate forms of the viscous term . . . . .	93

## *CONTENTS*

<b>B</b>	<b>Sponge layers</b>	<b>95</b>
<b>C</b>	<b>Nek5000 spectral element solver</b>	<b>97</b>
C.1	Features . . . . .	97
C.2	Basic setup . . . . .	98
	<b>Bibliography</b>	<b>101</b>



# Chapter 1

## Introduction

### 1.1 Motivation and background

Mathematical modeling of fluid flow problems often deals with unbounded domains. To be able to compute a numerical solution to such problems, the domain must be artificially truncated. However, truncated domains result in regions where the solution along the truncation surface is unknown a priori. And without knowledge about boundary conditions it is generally impossible to obtain a well posed solution to a partial differential equation. The difficulty of prescribing outflow boundary conditions arises from the impossibility to predict the flow behavior as it exits the domain. Therefore, the best one can achieve is to perform a qualitative approximation of those boundary conditions. And one must have in mind that the choice of approximation will be a significant source of errors of the whole numerical simulation. Boundary conditions of this type are commonly called “open” boundary conditions.

The quality of an open boundary condition highly depends on the complexity of the problem, and the insight one has on the solution. It does not matter if one is concerned with an accurate and stable solution method if the open boundary condition is approximated poorly; it will destroy the solution anyway.

Another problem regarding the practical use of these boundary conditions regards the discretization. Even if a complete mathematical understanding of the continuous boundary condition can be obtained, it must still be discretized to be usable. A numerical boundary condition depends significantly on the numerical discretization of the problem, not only the continuous counterpart. Thus, the choice of open boundary conditions in practice is often made as easy as possible. For instance, homogeneous Neumann boundary conditions are often a first choice [27].

In fluid dynamics we talk about inflows, where fluid enters a domain and outflows where fluids leaves a domain. The boundary conditions corresponding to a specific inflow are usually unproblematic and can generally be chosen by the user. The outflow boundary is typically an open boundary. Henceforth we use the term “outflow” boundary conditions through this thesis. Typical simulations concerned

with outflow boundary conditions are: flow simulations in wakes, jets, spatially developing pipes and boundary layers [38], all types of aerodynamic simulations, blood flow simulations in the arterial trees [15], atmospheric and ocean modeling [31] etc. The problem of open boundary conditions is interdisciplinary; the same theory is also applied in many other fields such as electrodynamics [4].

The difficulty arising from outflow boundary conditions is well known. The outflow and the region around it is usually thought to be unimportant to the characteristics of the solution, and is thus required to provide no spurious effects in the upstream direction. A common observation with outflows is that, especially at high and moderate Reynolds numbers<sup>1</sup> the computations would become unstable when strong vortices penetrate the outflow boundaries. Hence, the domain is usually elongated such as the outflow boundary condition does not influence the important part of the solution. In a simulation the flow dissipates with time, where the amount of dissipation depends on the Reynolds number. Hence in a laminar flow (low  $Re$ ) an elongated domain will imply that disturbances are smaller when they reach the outflow boundary. However, in high Reynolds number flows, especially for direct numerical simulation<sup>2</sup> (DNS) the computational grid must often be very dense to yield stable solutions and an elongated domain will be computationally very expensive. Also, the dissipation of the flow is low, and elongating the domain gives no guarantee to sufficiently damp out the disturbances anymore. The typical thing to do in this situation is to use “ad hoc” techniques; in particular some artificial damping (PML<sup>3</sup>, absorbing layers etc.) in the region close to the outflow boundary, with the purpose of damping out the disturbances [4, 7, 26]. Obviously the region where the technique is imposed must be excluded from the simulation, since the damping indeed destroys the solution, but in a favorable way. Drawbacks of damping regions are added computational time, but they may also imply non-physical reflections [26]. Thus, it is not clear beforehand that damping regions will work efficiently. The setup is mainly performed by “trial and error”, which makes the techniques problem dependent.

Typical examples of outflow boundary conditions in use are the natural boundary conditions arising from the variational form (finite element methods, spectral methods etc.) [17, 35], homogeneous Neumann boundary conditions [27], and convective boundary conditions [36, 44].

For the current project we have studied outflow boundary conditions in the context of the high-order spectral element method (SEM)[32]. The code we have used, which is highly parallel is Nek5000 [16], developed and maintained by Paul Fischer (Argonne National Laboratory, and University of Illinois, Champaign-Urbana). The code is in heavy use for various types of internal and some external flows in many research groups, including KTH Mechanics. To date, the only available choice of

---

<sup>1</sup>The Reynolds number is defined in §1.2.

<sup>2</sup>A direct numerical simulation (DNS) is a simulation in computational fluid dynamics where the flow equations are numerically solved by resolving even the smallest scales.

<sup>3</sup>Perfectly Matched Layer, see e.g. [4]



## 1.2. FLUID DYNAMICS – ANALYTIC AND NUMERICAL ISSUES

outflow boundary condition in Nek5000 is to use a natural boundary condition, also known as “O” inside the code. But the “O” condition has shown to be the source to several problems such as: poor accuracy; instability, and obvious reflections. In addition to the “O” condition a number of different treatments of the flow inside the domain are commonly used, e.g. a sponge layer (see §B) or the so called “nozzle condition” (injection of mass to accelerate and thus damp disturbances). The purpose of the work in this thesis has therefore been to study alternatives to the “O” condition; where the alternatives in particular have been convective boundary conditions and a stabilized version of the natural condition.

Due to the importance of outflow boundary conditions in incompressible flow problems, a lot of research has been devoted to it during the years. Sani and Gresho [36] gave a detailed review of what has been done in the area up to mid-1990s. Colionius [7] also gave a review of the topic, mainly focused on an exact treatment of outflow boundaries. Another relevant review regarding open boundary conditions for the shallow water equations is [31]. Some recent work regarding outflow boundary conditions in the spectral element framework has been performed by Dong et al. [10], Poux et al. [34], and Xu and Lin [44]. Detailed analysis of the natural outflow condition has been performed by Heywood et al. [17] and Rannacher [35]. The largest part of the research however use the finite differences method. Some relevant references among those are: the standard reference by Orlanski [29], Miyauchi et al. [27], and Poinso and Lele [33]. A general “non-reflective” form of an outflow boundary condition has also been widely studied; the original reference here is the famous papers by Engquist and Majda [12, 11], another well cited one is the paper by Trefethen and Halpern [42]. A short review of those “exact” boundary conditions is presented in §4.2.

## 1.2 Fluid dynamics – Analytic and numerical issues

The field of fluid dynamics deals with problems governed by partial differential equations that represents the conservation laws of mass, momentum and energy. The main equations in this topic, based on the Newton’s second law of motion is the famous Navier–Stokes equations derived by Claude–Louis Navier and George Gabriel Stokes in the early 1800’s. The incompressible Navier–Stokes equations (non-dimensionalized) are written as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{Re} \Delta \mathbf{u} + \mathbf{f}, \quad (1.1a)$$

$$\nabla \cdot \mathbf{u} = 0. \quad (1.1b)$$

where  $Re$  is the so called Reynolds number and  $\mathbf{f}$  an external force. The Reynolds number is a dimensionless quantity that governs the state of the flow (laminar, transitional or turbulent) and is defined as  $Re := UL/\nu$ , where  $U$  is the characteristic velocity [m/s] (mean velocity of the fluid),  $L$  the characteristic linear dimension [m], and  $\nu$  the kinematic viscosity [m<sup>2</sup>/s].

The Navier–Stokes equations have been a broadly studied subject in several different areas due to their physical relevance in hydrodynamics etc. but also purely mathematically the complicated non-linear equations are of great interest. The Navier–Stokes equations are indeed so complicated that even the most fundamental solutions are not yet found analytically; it is well known that one of the millennium prize problems<sup>4</sup> in mathematics is to prove that solutions of the three dimensional incompressible Navier–Stokes equations always exist. The problem is still open, even though it has been approached lately by some of the most skilled mathematicians in the world.

In two dimensions the global regularity was proven by Jean Leray in 1933 [22]. So, why is it so hard to prove existence and uniqueness of the Navier–Stokes equations in three dimensions, when it is often relatively easy for many other equations, or at least shown to be achievable? The simple, and most general answer for that is that the Navier–Stokes equations often include the problematic phenomenon of turbulence. The non-linearity of the Navier–Stokes equations makes them chaotic under certain conditions. A change in the initial conditions will change the solution completely, and hence the solution becomes extremely sensitive to disturbances; no matter how small the disturbances are, the solution will change after a finite time. Another problem is the non-locality of the equations. The pressure is a functional of the field of velocity derivatives. Hence, the pressure in one point depends on the whole velocity field. Which is of course both very costly computationally in numerical simulations and also implies problems discretizing the equations.

As with many other non-linear partial differential equations, numerical analysis is an advantageous tool to get a meaningful result. The numerical treatment of fluid dynamical problems mainly goes under the topic of “computational fluid dynamics” (CFD).

### 1.2.1 Numerical considerations

In this subsection we explain the complications in numerical solutions of fluid flow problems, i.e. why they need so high spatial resolution, and hence the need of a highly parallel solver like Nek5000.

At large Reynolds number regime, the dynamics of the flow tend to become more and more non-linear and time-dependent, which can be explained by that the non-linear term in (1.1) becomes dominant. Due to the non-linearity, the problems become extremely hard to solve analytically, especially in cause of turbulent flow. There are indeed some rare cases which have solutions in a three dimensional setting, but they are few and restricted. This non-linearity together with the complex physical geometry that often exists in fluid flow problems, cause the numerical treatments very challenging and computational costly. Even though there are numerical tools to use, the physical domains are often complex and high Reynolds

---

<sup>4</sup>The Millennium Prize Problems are seven problems in mathematics that were stated by the Clay Mathematics Institute in 2000. At the time of 2014 six of them remains unsolved, where the Navier–Stokes problem is one of them.

## 1.2. FLUID DYNAMICS – ANALYTIC AND NUMERICAL ISSUES

number computations especially in three dimensions become not only difficult but also computationally expensive to solve using DNS, where the turbulence has to be computed by the numerical method for all spatial and time scales. It can be shown that the number of floating point operations indeed are proportional to the spatial resolution of the discretization (see e.g. [30]). Hence, the number of operations grows with  $Re^3$  in three dimensions. This will also affect the memory consumption. Given the large amount of memory needed, the need of a parallel environment is crucial. Just at the super computer center PDC at the Royal Institute of Technology (KTH) it was shown that around 60% of all running jobs are related to solving something related to the Navier–Stokes equations [37].

The complications of solving the Navier–Stokes equations numerically can be divided up into three main difficulties:

1. Discretization errors are easily amplified and lead to inaccurate results, or blown up solutions. The non-linear term can potentially produce smaller and smaller scales, which can lead to the need of very high spatial resolution.
2. Parameters such as the Reynolds number implies difficulties, due to instability problems. For a very high Reynolds number there will be a very small term in front of the diffusion term that physically cannot be approximated to be zero. Indeed, approximating  $Re^{-1} \approx 0$  will drastically change the number of boundary conditions needed in the regular setting, and hence the dynamics of the system is completely changed. Letting the Reynolds number be zero gives us indeed another set of equations, namely the Euler equations. But the Euler equations are purely hyperbolic, which is not the case for the Navier–Stokes equations.
3. The incompressible Navier–Stokes equations are of mixed parabolic-hyperbolic type, which means that we cannot use efficient parabolic methods such as explicit time marching methods, neither efficient hyperbolic methods such as characteristic methods to solve it.

The first two of these are the major challenges for direct DNS, because of the need of very high resolution in high Reynolds number flows. This explains the great need of parallel computations.

There are other ways to treat turbulence simulations, i.e. large eddy simulation (LES) and Reynolds averaged Navier–Stokes equations (RANS) [13]. These work in ways that reduce the computational cost of the simulation significantly, but they have of course drawbacks: LES is for instance not universal, and RANS raise questions of the validity of the results. The limitations of our computers to date, restrict the use of DNS, which make the use of LES and RANS unavoidable. Karniadakis and Orszag estimated in 1993 that an exaFLOPS ( $10^{18}$ FLOPS) is required to carry out a DNS of a complete aeroplane [18]. The record to date on a supercomputer was estimated to  $33.86 \times 10^{15}$  FLOPS by NUDT Tianhe-2 at Guangzhou in China [1]. This shows the difficulty in computational power for DNS. In fact, the founder

of Intel corporation Gordon Moore predicted in 1965 the density of components on chips to double every 18 months [28]. This law was shown to predict the fact very well over the preceding 30 years or so. Over the recent years the clock frequency of the computers has somewhat stabilized, and the problem with cooling has led the evolution in the direction that the processors have multiple cores instead of a high clock frequency. This makes the Moore’s law a bit outdated, and maybe also irrelevant. By following the Moore’s law however, the magic limit of reaching one exaFLOPS ( $10^{18}$ FLOPS) should be reached in around 2020, which is indeed the limit Karniadakis and Orszag estimated the DNS of the aeroplane to.

### 1.3 Outline

The thesis is structured as follows. In chapter two the background theory of the numerical technique used is explained; the spectral element method is introduced and the time discretization techniques of Nek5000 are presented. In chapter three the theory of natural boundary conditions is explained, and the natural boundary condition of the incompressible Navier–Stokes equations is derived. We also review some issues that have been encountered from these natural boundary conditions. In chapter four the theory and implementation of convective boundary conditions in Nek5000 is examined. This includes also some theory of “exact” boundary conditions and wave reflections. In chapter five we analyze the outflow boundary conditions of chapter three and four numerically by several test cases: an unsteady flow to test temporal convergence rates, a stable flow problem (Kovasznay flow) to analyze spatial convergence rates, a case of a single convecting disturbance (vortex) in a base flow, and the classical case of flow past a circular cylinder.

## Chapter 2

# Navier–Stokes discretization

### 2.1 Introduction

The incompressible Navier–Stokes equations can be solved numerically in several ways, where the choice of method is highly problem dependent. Roughly speaking, the main numerical methods to approach a numerical solution to the incompressible Navier–Stokes equations are: the finite difference method (FDM), the finite element method (FEM), the finite volume method (FVM), and spectral methods (SM). FDM is easy to implement, but cannot generally be used on very complex grids. FEM is advantageous for its generality; it can handle complex geometries and is therefore applicable on a wide range of problems. SM are derived similarly as FEM from a variational formulation, and give high spatial accuracy, due to the choice of high order basis functions. But, the drawback with SM is that similarly to FDM, they have problems with complex geometries. An obvious approach is to combine the generality of FEM with the accuracy of SM. Such a combination was first presented by Patera in 1984 [32]. The hybrid method Patera presented was to be called the “spectral element method” (SEM).

In this chapter we present the spatial and temporal discretization of the incompressible Navier–Stokes equations (1.1), the way implemented in the Nek5000 code. The section should serve as a ground to understand the implementation of the

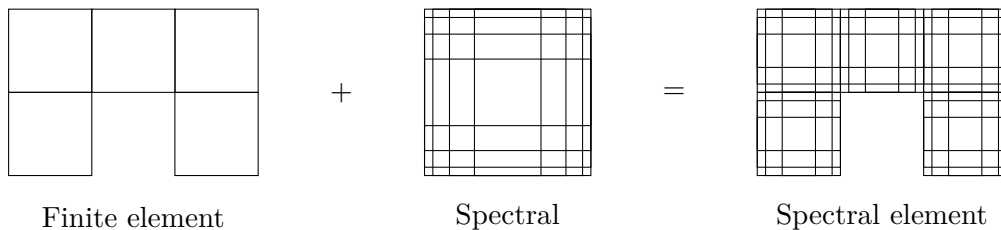


Figure 2.1: A spectral element method is a finite element method with a spectral method applied on each individual element.

boundary conditions presented in §§3,4, and give the tools needed to understand the numerical results we present in §5.

## 2.2 Domain decomposition

Let  $\Omega$  define a finite  $d = 2$  or  $3$  dimensional domain. In a SEM discretization of  $\Omega$ , the data is represented on sets of non-overlapping sub-domains (elements)  $\Omega^e$ ,  $e = 1, 2, \dots, E$  such that  $\Omega = \cup_{e=1}^E \Omega^e$ . One can consider every element in the grid individually where for each a spectral method is applied (see Figure 2.1). The spectral elements share points along the element boundaries, hence the elements are often of similar resolution. Such setup is called a conformal grid [13]. If on the other hand, hanging nodes are accepted one gets a non-conformal grid that is more complicated to deal with. The resolution of the elements depends on the order of the polynomial basis functions (“Lagrange polynomials”) of a Lagrangian interpolation over the element. We hereby shortly describe the theory. Consider a finite sequence of distinct points  $\{\xi_i\}_{i=0}^N \subset [-1, 1]$ , where  $\xi_0 = -1$  and  $\xi_N = 1$ , and let the function  $u : [-1, 1] \mapsto \mathbb{R}$  be a one dimensional representation of a solution. The Lagrangian interpolation of  $u$  reads

$$u(\xi_i) := \sum_{j=0}^N u_j \phi_j(\xi_i) = u_i, \quad i = 0, 1, \dots, N. \quad (2.1)$$

where  $\{u_j\}_{j=0}^N$  is the basis coefficients corresponding to a set of orthogonal nodal basis functions  $\{\phi_j\}_{j=0}^N$ . Nodal basis functions are known as “Lagrangian interpolants” and have the property that the basis coefficients  $\{u_i\}_{i=0}^N$  are also function values at the distinct points  $\{\xi_i\}_{i=0}^N$ , i.e.  $\phi_j(\xi_i) = \delta_{ij}$  holds, where  $\delta_{ij}$  is the Kronecker delta function, which is one if  $i = j$  and zero else. In SEM, one way to choose the Lagrange basis functions is to use Legendre polynomials  $P_N(x)$  (see §A.2), where  $N$  is the polynomial order. The discrete points  $\xi_i$  are then located in the polynomial roots of the polynomials  $(1 - x^2)P'_N(x)$ . These points are called the Gauss Lobatto Legendre (GLL) points, and the corresponding Lagrange polynomials can be defined as “GLL polynomials”. Thus we get a non-uniform grid that is more dense close to the element boundaries. One benefit of this is that the spectral elements avoid the Runge’s phenomenon<sup>1</sup>. The orthogonality of the Lagrange polynomials also leads to diagonal or block diagonal “mass matrix”<sup>2</sup>, which are favorable in time domain computations due to that the inversion of such matrices are trivial. Consider the first five GLL polynomials over non uniform GLL points in Figure 2.2.

<sup>1</sup>Runge’s phenomenon is a problem of oscillation at the edges of an interval that occurs when using polynomial interpolation with polynomials of high degree over a set of uniformly distributed interpolation points.

<sup>2</sup>The mass matrix is defined in §2.3.5. See Equation (2.23).

## 2.2. DOMAIN DECOMPOSITION

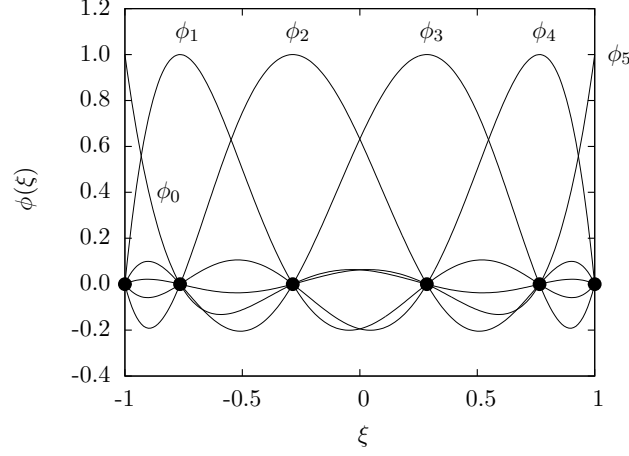


Figure 2.2: Lagrange polynomials over GLL points ( $N = 5$ ).

The GLL points have a nice property regarding the computation of numerical quadrature. For any GLL polynomial  $p$  of degree  $\leq 2N - 1$  it is given that

$$\int_{-1}^1 p(x) dx \equiv \sum_{j=0}^N \rho_j p(\xi_j), \quad \xi_j \in [-1, 1], \quad (2.2)$$

where  $\{\rho_j\}_{j=0}^N$  are the quadrature weights given by

$$\rho_j = \frac{2}{N(N+1)} \frac{1}{[P_N(\xi_k)]^2}, \quad (2.3)$$

(see also §A.2.2).

Since we are considering domains  $\Omega$  of dimension two or three, we need to extend the interpolation dimension of (2.1). This is easily performed due to the presence of tensor products (see also §A.1). The choice of GLL points gives us the reference element  $\hat{\Omega} := [-1, 1]^d$ , ( $d = 2, 3$ ) which is the corresponding element using the GLL polynomials in higher dimensions. If we know the deformation (the shape) of our domain, it is possible to define a conformal one-to-one mapping from the reference element to each element in the domain, i.e.  $\mathbf{x}^e : \hat{\Omega} \mapsto \Omega^e$ ,  $\forall e$ . If the domain is rectangular, the transformation is affine, and the Jacobian becomes a constant value. If the domain is for instance a cylinder then a polar coordinate substitution is handy to use etc. In SEM it is suitable to have elements with four corner-points in each dimension. The element shapes are therefore conveniently in forms of quadrilateral in two dimensions and hexahedral in three dimensions (Other forms are also possible, but less common; see e.g. Sherwin and Karniadakis [39] for triangular and tetrahedral SEM).

The followings are two examples of multi-dimensional discretizations, a two dimensional, and a three dimensional, with non-deformed elements, i.e. the mappings are affine.

**Example 2.1** (2D affine transformation). Assume we have a two dimensional physical domain  $\Omega = [0, L_1] \times [0, L_2]$  which is partitioned into two spectral elements:  $\Omega^1 = [0, L_1/2] \times [0, L_2]$ , and  $\Omega^2 = [L_1/2, L_1] \times [0, L_2]$ . The derivation of an affine coordinate mapping  $\mathbf{x} : \hat{\Omega} = [-1, 1]^2 \mapsto \Omega^e$ ,  $(r_1, r_2) \mapsto \mathbf{x}(r_1, r_2) = (x_1^e, x_2^e)$  is straight forward; only linear scaling needed to be applied to the elements. The transformation becomes  $\mathbf{x}(r_1, r_2) = (\min\{x_1^e\} + (L_1/4)(r_1 + 1), \min\{x_2^e\} + (L_2/2)(r_2 + 1))$ . Then for a function  $u|_{\Omega^e}$  the Lagrange interpolation expressed in reference coordinates is

$$u(\mathbf{x}(r_1, r_2))|_{\Omega^e} = \sum_{i=0}^N \sum_{j=0}^N u_{ij} \phi_i(r_1) \phi_j(r_2), \quad (r_1, r_2) \in \hat{\Omega}.$$

The  $x$  and  $y$  derivatives are also easy to express in reference coordinates. They are

$$\begin{aligned} \frac{\partial u}{\partial x}(\mathbf{x}(r_1, r_2))|_{\Omega^e} &= \sum_{i=0}^N \sum_{j=0}^N u_{ij} \frac{4}{L_1} \phi'_i(r_1) \phi_j(r_2), \quad (r_1, r_2) \in \hat{\Omega}, \\ \frac{\partial u}{\partial y}(\mathbf{x}(r_1, r_2))|_{\Omega^e} &= \sum_{i=0}^N \sum_{j=0}^N u_{ij} \phi_i(r_1) \frac{2}{L_2} \phi'_j(r_2), \quad (r_1, r_2) \in \hat{\Omega}. \end{aligned}$$

where  $\{\phi_j\}$  is the (GLL) Lagrange basis functions.

**Example 2.2** (3D affine transformation). Assume we have a three dimensional physical domain  $\Omega = [0, L_1] \times [0, L_2] \times [0, L_3]$ . In three dimensions, the reference element is  $\hat{\Omega} = [-1, 1]^3$ . Similarly to the previous example we partition  $\Omega$  into two spectral elements:  $\Omega^1 = [0, L_1/2] \times [0, L_2] \times [0, L_3]$ , and  $\Omega^2 = [L_1/2, L_1] \times [0, L_2] \times [0, L_3]$ . The affine transformation of this example becomes  $\mathbf{x}(r_1, r_2, r_3) = (\min\{x_1^e\} + (L_1/4)(r_1 + 1), \min\{x_2^e\} + (L_2/2)(r_2 + 1), \min\{x_3^e\} + (L_3/2)(r_3 + 1))$ . And for a function  $u|_{\Omega}$  we get the Lagrange interpolation expressed in reference coordinates

$$u(\mathbf{x}(r_1, r_2, r_3))|_{\Omega^e} = \sum_{i=0}^N \sum_{j=0}^N \sum_{k=0}^N u_{ijk} \phi_i(r_1) \phi_j(r_2) \phi_k(r_3), \quad (r_1, r_2, r_3) \in \hat{\Omega}.$$

Differentiating with respect to  $x$  expressed in reference coordinates yields

$$\begin{aligned} \frac{\partial u}{\partial x}(\mathbf{x}(r_1, r_2, r_3))|_{\Omega^e} &= \sum_{i=0}^N \sum_{j=0}^N \sum_{k=0}^N u_{ijk} \frac{4}{L_1} \phi'_i(r_1) \phi_j(r_2) \phi_k(r_3), \quad (r_1, r_2, r_3) \in \hat{\Omega}. \\ \frac{\partial u}{\partial y}(\mathbf{x}(r_1, r_2, r_3))|_{\Omega^e} &= \sum_{i=0}^N \sum_{j=0}^N \sum_{k=0}^N u_{ijk} \phi_i(r_1) \frac{2}{L_2} \phi'_j(r_2) \phi_k(r_3), \quad (r_1, r_2, r_3) \in \hat{\Omega}. \\ \frac{\partial u}{\partial z}(\mathbf{x}(r_1, r_2, r_3))|_{\Omega^e} &= \sum_{i=0}^N \sum_{j=0}^N \sum_{k=0}^N u_{ijk} \phi_i(r_1) \phi_j(r_2) \frac{2}{L_3} \phi'_k(r_3), \quad (r_1, r_2, r_3) \in \hat{\Omega}. \end{aligned}$$

where  $\{\phi_j\}$  is the (GLL) Lagrange basis functions.



### 2.3. SPATIAL DISCRETIZATION

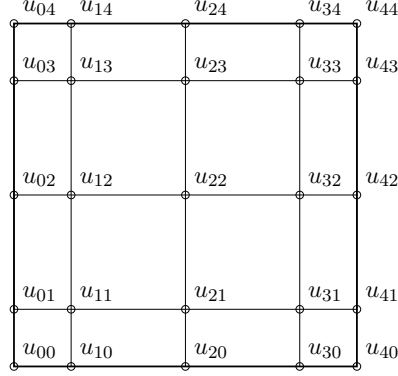


Figure 2.3: Ordering of the points in a two dimensional spectral element ( $N = 4$ ).

The ordering of the nodal points is crucial to be able to use a matrix form. The points in the grid is stored in a lexicographic order, which is a result of the tensor product representation. In two and three dimensional computations each element is represented as follows

$$(d = 2), \quad \underline{u} := (u_1, u_2, \dots, u_{(N+1)^2})^\top := (u_{00}, u_{10}, \dots, u_{ij}, \dots, u_{NN})^\top, \quad (2.4)$$

$$(d = 3), \quad \underline{u} := (u_1, u_2, \dots, u_{(N+1)^3})^\top := (u_{000}, u_{100}, \dots, u_{ijk}, \dots, u_{NNN})^\top. \quad (2.5)$$

Consider the location of the elements in the two dimensional setting (Eq. (2.4)) in Figure 2.3.

The discretization process over each element  $\Omega^e \subset \Omega$  can be divided into two separate parts. The spatial discretization, resulting in a semi-discretization of the equations, and the temporal discretization which discretizes the equations in time using the obtained spatial discretization. In the following section we explain the spatial part, which finally results in a semi-discrete matrix form, that is convenient to work with in the temporal part which follows in the section after.

## 2.3 Spatial discretization

### 2.3.1 Functional spaces

Through this section, we use several functional spaces. For convenience they are all explained here.

The  $L^2(\Omega)$  space of square integrable functions with corresponding norm:

$$L^2(\Omega)^d := \{\mathbf{u} : \Omega \mapsto \mathbb{R}^d : \|\mathbf{u}\|_{L^2(\Omega)^d} < \infty\}, \quad (2.6)$$

$$\|\mathbf{u}\|_{L^2(\Omega)^d} := \left( \int_{\Omega} |\mathbf{u}|^2 d\Omega \right)^{1/2}. \quad (2.7)$$

The  $L_0^2(\Omega)$  space is the space of square integrable functions that has a mean value zero:

$$L_0^2(\Omega) := \{q \in L^2(\Omega) : \int_{\Omega} q \, d\Omega = 0\}. \quad (2.8)$$

Define the boundary as  $\Gamma := \partial\Omega$ , and partition it in the following way:  $\Gamma = \Gamma_n \cup \Gamma_e$ ,  $\Gamma_n \cap \Gamma_e = \emptyset$ , where  $\Gamma_n$  correspond to boundaries with a natural boundary condition imposed, and  $\Gamma_e$  boundaries corresponding to essential (Dirichlet) boundary conditions. The solutions of weak formulations correspond to specific Sobolev spaces, which are spaces of functions in  $L^2$  with smoothness up to a specific differentiating order:

$$H^1(\Omega)^d := \{\mathbf{u} \in L^2(\Omega)^d : |\alpha| \leq 1, D^\alpha \mathbf{u} \in L^2(\Omega)^d\}, \quad (2.9)$$

$$H_0^1(\Omega)^d := \{\mathbf{u} \in H^1(\Omega)^d : \mathbf{u}|_{\Gamma_e} = \mathbf{0}\}, \quad (2.10)$$

where  $\alpha := (\alpha_1, \dots, \alpha_d)$ , and  $|\alpha| = \alpha_1 + \dots + \alpha_d$ ,  $D^\alpha := \frac{\partial^{|\alpha|}}{\partial x_1 \dots \partial x_d^{\alpha_d}}$  is the distributional derivative.

### 2.3.2 Weak formulation

SEM is based on the weak form of a partial differential equation. The weak form allows us to use Lagrangian basis functions together with the Galerkin method. That gives a spatial accuracy that is consistent with the order of the basis functions.

To obtain the weak formulation we follow the well known procedure from the finite element community; we multiply the incompressible Navier-Stokes equations (1.1) with test functions  $\mathbf{v} \in H^1(\Omega)^d$  and  $q \in L^2(\Omega)$ ; then we integrate over  $\Omega$  to obtain

$$\begin{aligned} \int_{\Omega} \mathbf{v} \cdot \frac{\partial \mathbf{u}}{\partial t} \, d\Omega + \int_{\Omega} \mathbf{v} \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) \, d\Omega \\ = - \int_{\Omega} \mathbf{v} \cdot \nabla p \, d\Omega + \frac{1}{Re} \int_{\Omega} \mathbf{v} \cdot \nabla^2 \mathbf{u} \, d\Omega + \int_{\Omega} \mathbf{v} \cdot \mathbf{f} \, d\Omega. \end{aligned} \quad (2.11a)$$

$$\int_{\Omega} q(\nabla \cdot \mathbf{u}) \, d\Omega = 0. \quad (2.11b)$$

Integration by parts on the viscous and pressure terms then yields

$$\begin{aligned} \int_{\Omega} \mathbf{v} \cdot \frac{\partial \mathbf{u}}{\partial t} \, d\Omega + \int_{\Omega} \mathbf{v} \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) \, d\Omega \\ = - \oint_{\Gamma_e} \mathbf{v} \cdot p \mathbf{n} \, d\Gamma_e - \oint_{\Gamma_n} \mathbf{v} \cdot p \mathbf{n} \, d\Gamma_n + \int_{\Omega} p(\nabla \cdot \mathbf{v}) \, d\Omega \\ + \frac{1}{Re} \oint_{\Gamma_e} \mathbf{v} \cdot \nabla \mathbf{u} \cdot \mathbf{n} \, d\Gamma_e + \frac{1}{Re} \oint_{\Gamma_n} \mathbf{v} \cdot \nabla \mathbf{u} \cdot \mathbf{n} \, d\Gamma_n - \frac{1}{Re} \int_{\Omega} \nabla \mathbf{u} \cdot \nabla \mathbf{v} \, d\Omega \\ + \int_{\Omega} \mathbf{v} \cdot \mathbf{f} \, d\Omega. \end{aligned} \quad (2.12a)$$

$$\int_{\Omega} q(\nabla \cdot \mathbf{u}) \, d\Omega = 0. \quad (2.12b)$$

### 2.3. SPATIAL DISCRETIZATION

The next step is to choose functional spaces  $(V \times Z) \subset (H^1(\Omega)^d \times L^2(\Omega))$  for  $(\mathbf{u}, p)$  (trial functions) such that the solution is sufficiently smooth and includes natural (derived in §3.2) and essential (Dirichlet and periodic) boundary conditions; and choose spaces  $\mathbf{v} \in V_0 := H_0^1(\Omega)^d$  and  $q \in Z := L_0^2(\Omega)$  for the test functions such that the boundary integrals that correspond to essential boundary conditions cancel. Natural boundary conditions are assumed to cancel the boundary integral  $(\oint_{\Gamma_n})$  in the case they are used and the test functions are assumed to cancel the boundary integrals  $(\oint_{\Gamma_e})$  for the essential boundary conditions (note that convective boundary conditions (§4) will be treated as Dirichlet conditions and are therefore of essential type too). Taking all this into account, then the “weak formulation” of the problem can be stated as:

*Find  $(\mathbf{u}, p) \in V \times Z$  such that*

$$\frac{d}{dt}(\mathbf{v}, \mathbf{u}) + \mathcal{A}(\mathbf{v}, \mathbf{u}, \mathbf{u}) = \mathcal{B}(\mathbf{v}, p) + \frac{1}{Re} \mathcal{C}(\mathbf{v}, \mathbf{u}) + \mathcal{F}(\mathbf{v}), \quad \forall \mathbf{v} \in V_0, \quad (2.13a)$$

$$\mathcal{B}(\mathbf{u}, q) = 0, \quad \forall q \in Z. \quad (2.13b)$$

where

$$\begin{aligned} \frac{d}{dt}(\mathbf{v}, \mathbf{u}) &= \int_{\Omega} \mathbf{v} \cdot \frac{\partial \mathbf{u}}{\partial t} d\Omega = \frac{d}{dt} \int_{\Omega} \mathbf{v} \cdot \mathbf{u} d\Omega \\ \mathcal{A}(\mathbf{v}, \mathbf{u}, \mathbf{u}) &= (\mathbf{v}, \mathbf{u} \cdot \nabla \mathbf{u}) = \int_{\Omega} \mathbf{v} \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) d\Omega, \\ \mathcal{B}(\mathbf{v}, w) &= -(\nabla \cdot \mathbf{v}, w) = - \int_{\Omega} (\nabla \cdot \mathbf{v}) w d\Omega, \\ \mathcal{C}(\mathbf{v}, \mathbf{u}) &= -(\nabla \mathbf{v}, \nabla \mathbf{u}) = - \int_{\Omega} \nabla \mathbf{u} \cdot \nabla \mathbf{v} d\Omega, \\ \mathcal{F}(\mathbf{v}) &= (\mathbf{v}, \mathbf{f}) = \int_{\Omega} \mathbf{v} \cdot \mathbf{f} d\Omega. \end{aligned}$$

Note that the regularity of the pressure solution is different than that of the velocity. Since  $\mathbf{u} \in H^1(\Omega)^d$  the interpolation of the velocity is continuous across element boundaries, whereas  $p \in L_0^2(\Omega)$  means that the pressure interpolation can be discontinuous.

A “discrete weak form” follows by the Galerkin method, where we choose trial and test functions in the same Sobolev space that is built up by of Lagrangian interpolants. It is suitable to use homogeneous Dirichlet boundary conditions for all essential boundary conditions here and add the inhomogeneity to the forcing function  $\mathbf{f}$  instead of including it in the Sobolev spaces. Thus, we choose discrete sub-spaces in the following way  $(\mathbf{u}_N, p_N) \in V_{0,N} \times Z_N \subset V_0 \times Z$ ,  $\mathbf{v} \in V_{0,N}$ , and  $q \in Z_N$  (where  $V_{0,N}$ , and  $Z_0$  are soon to be defined), and the discrete weak form reads:

Find  $(\mathbf{u}_N, p_N) \in V_{0,N} \times Z_N$  such that

$$\begin{aligned} \frac{d}{dt}(\mathbf{v}_N, \mathbf{u}_N) + \mathcal{A}_N(\mathbf{v}_N, \mathbf{u}_N, \mathbf{u}_N) &= \mathcal{B}_N(\mathbf{v}_N, p_N) + \frac{1}{Re} \mathcal{C}_N(\mathbf{v}_N, \mathbf{u}_N) \\ &\quad + \mathcal{F}_N(\mathbf{v}_N), \quad \forall \mathbf{v}_N \in V_{0,N}, \end{aligned} \quad (2.14a)$$

$$\mathcal{B}_N(\mathbf{v}_N, \mathbf{u}_N) = 0, \quad \forall q_N \in Z_N. \quad (2.14b)$$

where

$$\begin{aligned} \mathcal{A}_N(\mathbf{v}_N, \mathbf{u}_N, \mathbf{u}_N) &= (\mathbf{v}_N, \mathbf{u}_N \cdot \nabla \mathbf{u}_N)_{GLL}, \\ \mathcal{B}_N(\mathbf{v}_N, w_N) &= -(\nabla \cdot \mathbf{v}_N, w_N)_{GLL \text{ (or GL)}}, \\ \mathcal{C}_N(\mathbf{v}_N, \mathbf{u}_N) &= -(\nabla \mathbf{v}_N, \nabla \mathbf{u}_N)_{GLL}, \\ \mathcal{F}_N(\mathbf{v}_N) &= (\mathbf{v}_N, \mathbf{f}_N)_{GLL}, \\ (\mathbf{v}_N, \mathbf{u}_N) &= (\mathbf{v}_N, \mathbf{u}_N)_{GLL}, \end{aligned}$$

Here the inner products on the form  $(\cdot, \cdot)_{GLL}$  are computed with the GLL-quadrature rule<sup>3</sup>, which includes all quadrature computations that regard velocity. The inner products that include pressure are either computed with the GLL-quadrature rule (collocated setting), or the GL-quadrature<sup>4</sup>  $(\cdot, \cdot)_{GL}$  (staggered setting). In the following two subsections we present the theory of the collocated and staggered implementations respectively and the choices of approximation spaces  $V_{0,N}$  and  $Z_N$ . Since a discrete solution of SEM is defined by polynomial basis functions, the discrete subspaces must be polynomial spaces. A general definition we will use to define polynomial spaces in the following subsections is

$$\mathbb{P}_N(\Omega)^d := \{\phi : \phi \in L^2(\Omega)^d, \phi \text{ is a polynomial of degree } \leq N\}. \quad (2.15)$$

The choice of staggered against collocated grids has been discussed for a long time in the field of computational fluid dynamics. The staggered grid was introduced in the mid 1960s by Harlow and Welsh (before that the collocated grid were the only possible choice), and from that time until early 1980s the staggered grids were considered the only way to go. In the 1980s, improvements in the coupling algorithms were found, and the popularity of the collocated arrangement began to rise again [13].

### 2.3.3 The $\mathbb{P}_N \times \mathbb{P}_N$ method

In a spectral element framework the collocated approach is commonly named the  $\mathbb{P}_N \times \mathbb{P}_N$  method. In this setting the discrete spaces in the discrete variational formulation (2.14) becomes

$$V_{0,N} := V_0 \cap \mathbb{P}_N(\Omega)^d, \quad (2.16)$$

$$Z_N := Z \cap \mathbb{P}_N(\Omega). \quad (2.17)$$

<sup>3</sup>Quadrature on GLL grid (see §A.2.2).

<sup>4</sup>Quadrature on GL grid (see §A.2.3).

### 2.3. SPATIAL DISCRETIZATION

The collocated approach assumes that the GLL nodal basis functions are used for both velocity and pressure.

Collocating the velocity and the pressure points has drawbacks. Since the pressure appear as a gradient, it is not uniquely defined. The pressure solutions satisfies the weak continuous relation

$$(\nabla p, \mathbf{v}) = 0, \quad \forall \mathbf{v} \in V_0. \quad (2.18)$$

Thus, any pair  $(\mathbf{u}, p + \text{constant})$  is a solution to (2.18) as long as  $(\mathbf{u}, p)$  is a solution. In the original setting the  $\mathbb{P}_N \times \mathbb{P}_N$  method does not satisfy the inf-sup condition (see §A.3), which implies a set of spurious pressure modes (see e.g. [9]). The spurious pressure modes is a well known phenomena. However, often the pressure is not the interesting quantity; so this problem seems fine as long as the velocity field give us the right result.

In the Nek5000 code the collocated method is available mostly thanks to prof. Tomboulides. He introduced a high order splitting routine, that has shown high order accuracy in time and minimal mass conservation errors [41].

#### 2.3.4 The $\mathbb{P}_N \times \mathbb{P}_{N-2}$ method

It was proven by Maday and Patera [24] that choosing the basis functions of the pressure field two polynomial orders below the velocity field indeed eliminates the problem of spurious pressure modes and therefore enforce the inf-sup condition. In this setting the functional spaces for the discrete weak formulation (2.14) therefore become

$$V_{0,N} := V_0 \cap \mathbb{P}_N(\Omega)^d, \quad (2.19)$$

$$Z_N := Z \cap \mathbb{P}_{N-2}(\Omega). \quad (2.20)$$

Instead of using the GLL points for both velocity and pressure, the pressure is now chosen in GL (Gauss Lobatto) nodes (see §A.2). Hence different quadrature rules are performed for different terms in the discrete weak formulation.

#### 2.3.5 Semi-discrete matrix form

The spatial discretization of the spectral element method is easiest described in matrix form. We concentrate on a general case of a non-deformed conformal grid to describe the context. In the computations we mainly restrict ourself to two dimensions to save space and keep the notations simple. The matrix operators can then be extended by the tensor product form (see A.1). More complicated implementations and more details can be considered in the literature by e.g. Fischer, Deville and Mund [9].

First some assumptions and definitions. Let  $\{\phi_j\}_{j=0}^N$  represent a set of Lagrangian interpolants. Assume that the domain is rectangular  $\Omega := [0, L_1] \times \dots \times$

$[0, L_d]$  with only one spectral element for simplicity, and define the reference domain as  $\hat{\Omega} := [-1, 1]^d$ . Then an affine transformation  $\mathbf{x} : \hat{\Omega} \mapsto \Omega$ ,  $(r_1, \dots, r_d) \mapsto (x_1, \dots, x_d)$  is given by  $\mathbf{x}(r_1, \dots, r_d) = ((L_1/2)(r_1 + 1), \dots, (L_d/2)(r_d + 1))$ . For a single coordinate direction the sequences  $\{\xi_l\}_{l=0}^N \subset [-1, 1]$  and  $\{\rho_l\}_{l=0}^N$  are denoting respectively the quadrature nodes and weights of an integral approximation. Also let  $\text{diag}(A_1, \dots, A_m)$  represents a diagonal block matrix with the matrices  $A_1, \dots, A_m$  through the diagonal, and similarly  $\text{diag}(\underline{a})$  a diagonal matrix with the elements of the vector  $\underline{a}$ , through the diagonal.

The inner products of (2.14) are computed using (2.1) as

$$(u, v)_N = \int_{\Omega} uv \, d\Omega = \sum_{\hat{i}\hat{j}} \sum_{ij} v_{\hat{i}\hat{j}} \left( \int_{\Omega} \phi_{\hat{i}} \phi_{\hat{j}} \phi_i \phi_j \, dx dy \right) u_{ij}, \quad (2.21)$$

where  $\underline{v} = v_{\hat{i}\hat{j}}$  and  $\underline{u} = u_{ij}$  are vectors of basis coefficients ordered as in Equation (2.4). A much simpler form of the inner product (2.21) is

$$(u, v)_N = \underline{v}^\top M \underline{u}, \quad (2.22)$$

where  $M$  is the so called “mass matrix” with the entries

$$\begin{aligned} M_{\hat{k}\hat{k}} &= \int_{-1}^1 \int_{-1}^1 \phi_i(r_1) \phi_{\hat{i}}(r_1) \phi_j(r_2) \phi_{\hat{j}}(r_2) \frac{L_1}{2} dr_1 \frac{L_2}{2} dr_2 \\ &= \frac{L_1 L_2}{4} \left( \int_{-1}^1 \phi_{\hat{i}}(r_1) \phi_i(r_1) \, dr_1 \right) \left( \int_{-1}^1 \phi_{\hat{j}}(r_2) \phi_j(r_2) \, dr_2 \right) \\ &\approx \frac{L_1 L_2}{4} \left( \sum_{l=0}^N \rho_l \phi_{\hat{i}}(\xi_l) \phi_i(\xi_l) \right) \left( \sum_{l=0}^N \rho_l \phi_{\hat{j}}(\xi_l) \phi_j(\xi_l) \right) \\ &\quad \left\{ \phi_i(\xi_j) = \delta_{ij}, \quad \hat{M} := \text{diag}(\{\rho_0, \rho_1, \dots, \rho_N\}) \right\} \\ &= \frac{L_1 L_2}{4} \hat{M}_{\hat{i}\hat{i}} \hat{M}_{\hat{j}\hat{j}}. \end{aligned} \quad (2.23)$$

where  $\hat{k} = 1 + \hat{i} + (N + 1)\hat{j}$ ,  $k = 1 + i + (N + 1)j$  is the natural ordering of the elements (see Figure 2.3), and  $L_1 L_2 / 4$  is the constant Jacobian that corresponds to the affine mapping. Note, in a deformed element configuration the Jacobian will obviously not be constant which make the expressions much more complicated.

Equation (2.23) written in tensor product form yields

$$M = \frac{L_1 L_2}{4} (\hat{M} \otimes \hat{M}). \quad (2.24)$$

Where  $\hat{M}$  is the one dimensional mass matrix. The mass matrix in three dimensions becomes

$$M = \frac{L_1 L_2 L_3}{8} (\hat{M} \otimes \hat{M} \otimes \hat{M}). \quad (2.25)$$

### 2.3. SPATIAL DISCRETIZATION

Derivatives are represented by the “differential matrix”, which in one dimension is given by  $\hat{D}$ , with corresponding matrix elements

$$\hat{D}_{ij} := \left. \frac{d\phi_j}{dr} \right|_{r=\xi_i}, \quad i, j \in \{0, 1, \dots, N\}^2 \quad (2.27)$$

The derivative operators for two dimensions reads

$$D_1 := (I \otimes \hat{D}) \frac{2}{L_1}, \quad D_2 := (\hat{D} \otimes I) \frac{2}{L_2}. \quad (2.28)$$

The diffusive part of the solution is represented by the so called “stiffness matrix”. Where the one-dimensional version is defined as

$$\hat{K}_{ij} := \int_{-1}^1 \left. \frac{d\phi_i}{dr} \right|_{r=\xi_j} \left. \frac{d\phi_j}{dr} \right|_{r=\xi_i} dr \approx \sum_{l=0}^N \hat{D}_{li} \rho_l \hat{D}_{lj}, \quad i, j \in \{0, 1, \dots, N\}^2. \quad (2.29)$$

The two dimensional operator for the stiffness matrix becomes

$$K = \frac{L_2}{L_1} (\hat{M} \otimes \hat{K}) + \frac{L_1}{L_2} (\hat{K} \otimes \hat{M}) \quad (2.30)$$

Now define

$$\mathbf{M} := \text{diag}(M, \dots, M), \quad \mathbf{K} := \text{diag}(K, \dots, K), \quad \mathbf{D} := \begin{bmatrix} D_1 & \dots & D_d \end{bmatrix}, \quad (2.31)$$

i.e. we construct general matrices, that represents all flow fields. Then the inner products of the discrete variational form (2.14) can be evaluated as follow

$$\begin{aligned} \mathcal{A}_N(\mathbf{v}, \mathbf{u}, \mathbf{u}) &= (\mathbf{v}, \mathbf{u} \cdot \nabla \mathbf{u}) \\ &= \sum_{\hat{i}\hat{j}} v_{i,\hat{j}} \begin{bmatrix} u_{1,i\hat{j}} \\ u_{2,i\hat{j}} \end{bmatrix} \left( \int_{\Omega} \phi_i \phi_j \sum_{ij} (\phi'_i \phi_j u_{1,i\hat{j}} + \phi_i \phi'_j u_{2,i\hat{j}}) d\Omega \right) \\ &= \underline{\mathbf{v}}^\top \begin{bmatrix} M \text{diag}(\underline{u}_1) & M \text{diag}(\underline{u}_2) \end{bmatrix} (D_1 \underline{u}_1 + D_2 \underline{u}_2) \\ &= \underline{\mathbf{v}}^\top \underbrace{\begin{bmatrix} M \text{diag}(\underline{u}_1) D_1 & \mathbf{0} \\ \mathbf{0} & M \text{diag}(\underline{u}_2) D_2 \end{bmatrix}}_{:= \mathbf{C}(\underline{\mathbf{u}})} \underline{\mathbf{u}} \\ &= \underline{\mathbf{v}}^\top \mathbf{C}(\underline{\mathbf{u}}) \underline{\mathbf{u}}. \\ \mathcal{B}_N(\mathbf{v}, p) &= -(\nabla \cdot \mathbf{v}, p) \\ &= -\sum_{\hat{i}\hat{j}} p_{i,\hat{j}} \left( \int_{\Omega} \phi_i \phi_j \sum_{ij} (\phi'_i \phi_j v_{1,i\hat{j}} + \phi_i \phi'_j v_{2,i\hat{j}}) d\Omega \right) \\ &= -\underline{p}^\top (D_1 \underline{v}_1 + D_2 \underline{v}_2) = -\underline{\mathbf{v}}^\top \mathbf{D}^\top \underline{p} \\ \mathcal{C}_N(\mathbf{v}, \mathbf{u}) &= -(\nabla \mathbf{v}, \nabla \mathbf{u}) = -\underline{\mathbf{v}}^\top \mathbf{K} \underline{\mathbf{u}} \\ \mathcal{F}_N(\mathbf{v}) &= (\mathbf{v}, \mathbf{f}) = \underline{\mathbf{v}}^\top \mathbf{M} \underline{\mathbf{f}}. \end{aligned}$$

This results in the following semi-discrete matrix form of the Navier-Stokes equations

$$\mathbf{M} \frac{d\mathbf{u}}{dt} + \mathbf{C}(\mathbf{u})\mathbf{u} + \frac{1}{Re} \mathbf{K}\mathbf{u} - \mathbf{D}^\top \underline{p} = \mathbf{M}\mathbf{f}, \quad (2.32a)$$

$$-\mathbf{D}\mathbf{u} = \mathbf{0}. \quad (2.32b)$$

## 2.4 Time discretization

The spatial discretization is where SEM is applied. For the temporal discretization there are several time-stepping schemes that can be used. In Nek5000 there are two different choices available. The BDFk/EXTk scheme based on backward differentiating and extrapolation, and the OIFS/Characteristic method which is a sub-cycling method where the advective part is integrated more often than the diffusive part. We explain both of the methods in this section. Our numerical results are mainly based on the BDFk/EXTk formulation, but we also explain OIFS/Characteristics for the discussion of convective boundary conditions based on curved characteristics (§4.5.3).

The order of time discretization is important for accuracy and stability. Generally, a higher order temporal discretization makes it possible to choose larger time-steps and still preserves a relatively high accuracy provided the method is stable. A higher order method reaches in general a certain level of accuracy faster than a low order method. And, the computational cost of a low order method is not always lower than for a higher order method, which makes the higher order often to prefer. The time step size also relates to the stability of the method. A small time step size must be chosen to obtain stability, which makes high order methods advantageous in this manner too. Implicit time stepping is in general to be preferred over the explicit analog since implicit methods have a larger stability region. But, a drawback of them is that they can be very complex to compute. The Navier-Stokes equations are complex, and therefore require rather complex time-stepping methods. Since the equations include both velocity and pressure, the time advancement can hardly be done in a single equation. Instead so called “splitting” methods are used, that decouple the velocity and pressure and solve the system in two steps. This procedure can be performed in several ways. In this thesis we restrict ourselves to shortly explain the methods included in the Nek5000 code, which are based on the fractional step method.

### 2.4.1 The BDFk/EXTk scheme

The backward differentiating scheme of order  $k$  (BDFk) is an implicit formulation that is based on Taylor expansion. A time derivative on the form  $dy/dt = f(y)$  is discretized as follows with the BDFk scheme

$$\frac{1}{\Delta t} \sum_{i=0}^k \beta_k y^{n+1-i} \approx f(y^{n+1}) \quad (2.33)$$



## 2.4. TIME DISCRETIZATION

where  $\{\beta_i\}_{i=0}^k$  are BDF coefficients (see §A.4) and  $\Delta t$  the time-step size. Note that the number of implicit relations to be solved are the same for different orders  $k$ . Hence, the computational cost does not depend on the choice of order, i.e. there is no reason to pick  $k = 2$  in front of  $k = 3$  to save computational cost.

In the Navier–Stokes discretization (2.32) the only way to avoid the iterative form of a non-symmetric, non-linear system, that occurs if the advection terms are treated implicitly is obviously to treat them explicitly. A simple explicit method that satisfies this proposed by Karniadakis et al. [19] is to use high order extrapolation on the non-linear terms including the advection part. The  $k$ th order extrapolation (EXTk) of a general non linear term  $f(y^{n+1})$  is given by

$$f(y^{n+1}) \approx \sum_{i=1}^k \alpha_i f(y^{n+1-i}). \quad (2.34)$$

where  $\{\alpha_i\}_{i=1}^k$  is a set of extrapolation coefficients corresponding to the specific order of the scheme (see §A.5). Adding the BDFk and EXTk schemes together results in the BDFk/EXTk scheme

$$\frac{1}{\Delta t} \sum_{i=0}^k \beta_i y^{n+1-i} = \sum_{i=1}^k \alpha_i f(y^{n+1-i}). \quad (2.35)$$

The stability regions of the second and third order BDFk/EXTk schemes are considered by solving the scalar equation  $dy/dt = \lambda y$  and are depicted in Figure 2.4. We can note that the BDF3/EXT3 scheme does include more of the imaginary axis eigenvalues than the BDF2/EXT2 scheme.

The BDFk/EXTk discretization applied to the semi-discrete equation (2.32) yields

$$\mathbf{M} \frac{1}{\Delta t} \sum_{i=0}^k \beta_i \underline{\mathbf{u}}^{n+1-i} - \mathbf{D}^\top \underline{\mathbf{p}}^{n+1} + \frac{1}{Re} \mathbf{K} \underline{\mathbf{u}}^{n+1} = - \sum_{i=1}^k \alpha_i \mathbf{C}(\underline{\mathbf{u}}^{n+1-i}) \underline{\mathbf{u}}^{n+1-i} + \mathbf{M} \underline{\mathbf{f}}^{n+1} \quad (2.36a)$$

$$-\mathbf{D} \underline{\mathbf{u}}^{n+1} = \mathbf{0}. \quad (2.36b)$$

Which is transformed into

$$\mathbf{H} \underline{\mathbf{u}}^{n+1} - \mathbf{D}^\top \underline{\mathbf{p}}^{n+1} = \mathbf{M}(\underline{\mathbf{f}}^{n+1} - \frac{1}{\Delta t} \sum_{i=1}^k \beta_i \underline{\mathbf{u}}^{n+1-i}) - \frac{1}{\Delta t} \sum_{i=1}^k \alpha_i \mathbf{C}(\underline{\mathbf{u}}^{n+1-i}) \underline{\mathbf{u}}^{n+1-i} \quad (2.37a)$$

$$-\mathbf{D} \underline{\mathbf{u}}^{n+1} = \mathbf{0}. \quad (2.37b)$$

where  $\mathbf{H} = \text{diag}(H, \dots, H)$  is the discrete Helmholtz operator defined as  $H = \frac{1}{Re} K + \frac{\beta_0}{\Delta t} M$ .

It is convenient to write (2.37) as a linear system. We get:

$$\begin{bmatrix} \mathbf{H} & -\mathbf{D}^\top \\ -\mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{u}}^{n+1} \\ \underline{\mathbf{p}}^{n+1} \end{bmatrix} = \begin{bmatrix} \underline{\tilde{\mathbf{f}}}^{n+1} \\ \mathbf{0} \end{bmatrix}, \quad (2.38)$$

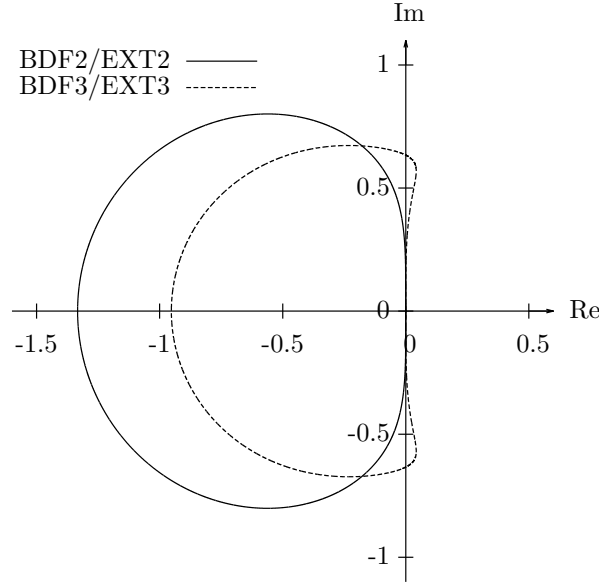


Figure 2.4: Stability regions of BDFk/EXTk methods.

where

$$\tilde{\mathbf{f}}^{n+1} = \mathbf{M} \left( \mathbf{f}^n - \frac{1}{\Delta t} \sum_{i=1}^k \beta_i \mathbf{u}^{n+1-i} \right) - \frac{1}{\Delta t} \sum_{i=1}^k \alpha_i \mathbf{C}(\mathbf{u}^{n+1-i}) \mathbf{u}^{n+1-i}. \quad (2.39)$$

This system can be solved with the fractional step method (see §2.5).

### 2.4.2 The operator integration factor splitting scheme (OIFS)

Operator integration factor splitting (OIFS), due to Maday, Rønquist and Patera [25] helps producing an efficient and more stable numerical scheme. The method separates the linear viscous from the non-linear advection term in the incompressible Navier–Stokes equations. In the OIFS method an explicit term is used for the advection part and treats the implicit Stokes part of the equations implicitly; this is similar to the BDFk/EXTk scheme. However, the OIFS method differs a lot from the BDFk/EXTk scheme. OIFS uses the Lagrangian formulation to compute the material derivative which makes it more complex than the BDFk/EXTk scheme to implement.

The first step in the derivation of the OIFS is to introduce an integrating factor  $\mathcal{Q}_C^{t*}$  which indeed never need to be explicitly defined. By multiplying both sides of Equation (2.32) with this integrating factor we obtain

$$\frac{d}{dt} \left[ \mathcal{Q}_C^{t*}(t) \mathbf{M} \mathbf{u} \right] + Re \mathcal{Q}_C^{t*}(t) \mathbf{C}(\mathbf{u}) = \mathcal{Q}_C^{t*}(t) (-\mathbf{K} \mathbf{u} + \mathbf{D}^\top \underline{p} + \mathbf{M} \mathbf{f}), \quad (2.40)$$

## 2.4. TIME DISCRETIZATION

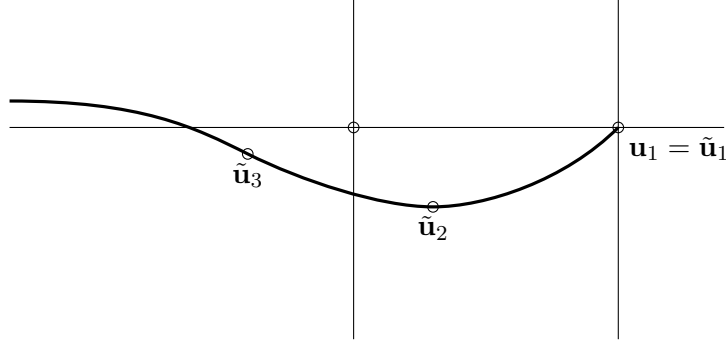


Figure 2.5: In the OIFS method the Lagrangian formulation is used to compute the material derivative. Consider a typical streamline backward in time from the value  $\underline{\mathbf{u}}_1$ , and the solutions  $\tilde{\underline{\mathbf{u}}}_2$ , and  $\tilde{\underline{\mathbf{u}}}_3$  at times  $t - \Delta t$  and  $t - 2\Delta t$ .

where the integrating factor is constructed such that

$$\frac{d}{dt} [\mathcal{Q}_C^{t*}(t) \mathbf{M} \underline{\mathbf{u}}] = \mathcal{Q}_C^{t*}(t) (-\mathbf{K} \underline{\mathbf{u}} + \mathbf{D}^\top \underline{\mathbf{p}} + \mathbf{M} \underline{\mathbf{f}}), \quad (2.41)$$

and

$$\frac{d}{dt} [\mathcal{Q}_C^{t*}(t) \mathbf{M} \underline{\mathbf{u}}] = Re \mathcal{Q}_C^{t*}(t) \mathbf{C}(\underline{\mathbf{u}}), \quad \mathcal{Q}_C^{t*}(t^*) = I, \quad (2.42)$$

holds. Then the BDFk scheme is applied to the time derivative, resulting in

$$\begin{aligned} \mathbf{M} \beta_0 \underline{\mathbf{u}}^{n+1} + \sum_{j=1}^k \beta_{k-j} S \left( \mathcal{Q}_C^{t^{n+1}}(t^{n+1-j}) \mathbf{M} \underline{\mathbf{u}}^{n+1-j} \right) \\ = \Delta t (-\mathbf{K} \underline{\mathbf{u}}^{n+1} + \mathbf{D}^\top \underline{\mathbf{p}}^{n+1} + \mathbf{M} \underline{\mathbf{f}}^{n+1}), \end{aligned} \quad (2.43)$$

where the term

$$S \left( \mathcal{Q}_C^{t^{n+1}}(t^{n+1-j}) \mathbf{M} \underline{\mathbf{u}}^{n+1-j} \right) = \mathbf{M} \tilde{\underline{\mathbf{u}}}_j^{n+1}, \quad j = 1, \dots, k, \quad (2.44)$$

represents a solution computed with fourth order Runge Kutta (RK4) sub-stepping. Here  $\{\tilde{\underline{\mathbf{u}}}_j^{n+1}\}_{j=1}^k$  are the velocities at off grid locations on the characteristic curve backward in time from  $\underline{\mathbf{u}}^{n+1}$  (see Figure 2.4.2). The off grid values are obtained by solving a separate initial value problem

$$\mathbf{M} \frac{d}{ds} \tilde{\underline{\mathbf{u}}}_j(s) = -Re C(\tilde{\underline{\mathbf{u}}}_j(s)) \tilde{\underline{\mathbf{u}}}_j(s), \quad t^{n+1-j} \leq s \leq t^{n+1}, \quad (2.45a)$$

$$\tilde{\underline{\mathbf{u}}}_j(t^{n+1-j}) = \underline{\mathbf{u}}_j(s). \quad (2.45b)$$

where the time steps of the sub-problem are of size  $\Delta s = \Delta t / \gamma$ , where  $\gamma$  is the number of sub-steps of the RK4 scheme that are used to solve (2.45). The number of sub-steps shall be chosen such that the CFL stability condition of the problem is

satisfied. The RK4 scheme is an appropriate choice since it can handle eigenvalues on the imaginary axis [9]. The final form of the full Navier-Stokes discretization can now be stated as

$$\mathbf{H}\underline{\mathbf{u}}^{n+1} - \mathbf{D}^\top \underline{p}^{n+1} = -\frac{\mathbf{M}}{\Delta t} \left( \sum_{j=1}^k \beta_j \tilde{\underline{\mathbf{u}}}^{n+1-j} \right) + \mathbf{M}\underline{\mathbf{f}}^{n+1} \quad (2.46a)$$

$$-\mathbf{D}\underline{\mathbf{u}}^{n+1} = 0, \quad (2.46b)$$

which yields the OIFS analogue to the linear system (2.38):

$$\begin{bmatrix} \mathbf{H} & -\mathbf{D}^\top \\ -\mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{u}}^{n+1} \\ \underline{p}^{n+1} \end{bmatrix} = \begin{bmatrix} \tilde{\underline{\mathbf{f}}}^n \\ \mathbf{0} \end{bmatrix}, \quad (2.47)$$

where

$$\tilde{\underline{\mathbf{f}}}^{n+1} = \mathbf{M} \left( \underline{\mathbf{f}}^{n+1} - \frac{1}{\Delta t} \sum_{i=1}^k \beta_i \tilde{\underline{\mathbf{u}}}^{n+1-i} \right). \quad (2.48)$$

## 2.5 Operator splitting – The fractional step method

The solution procedures of the linear systems (2.38) and (2.47) are not trivial tasks. Much effort has been devoted to decoupling the pressure from the velocity field calculation, to eliminate so called splitting errors that occur in this process [9]. The simplest, most popular decoupling method is the fractional-step method, which has a first order time error that can be improved to second order by pressure correction.

Recall, we are given a system on the form:

$$\begin{bmatrix} \mathbf{H} & -\mathbf{D}^\top \\ -\mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{u}}^{n+1} \\ \underline{p}^{n+1} \end{bmatrix} = \begin{bmatrix} \tilde{\underline{\mathbf{f}}}^{n+1} \\ \mathbf{0} \end{bmatrix}, \quad (2.49)$$

The solution is computed in two steps. First one solves

$$\mathbf{H}\underline{\mathbf{u}}^* = \tilde{\underline{\mathbf{f}}}^n + \mathbf{D}^\top \underline{p}^n \quad (2.50)$$

by approximating the inverse  $\mathbf{H}$ ; resulting in

$$\underline{\mathbf{u}}^* = \mathbf{H}^{-1}(\tilde{\underline{\mathbf{f}}}^n + \mathbf{D}^\top \underline{p}^n), \quad (2.51)$$

which is followed by a pressure correction step

$$\mathbf{E}\nabla^2 \underline{p}^{n+1} = -\frac{1}{\Delta t} \mathbf{D}\underline{\mathbf{u}}^*, \quad (2.52)$$

$$\underline{\mathbf{u}}^{n+1} = \underline{\mathbf{u}}^* + \Delta t \mathbf{M}^{-1} \mathbf{D}^\top \nabla^2 \underline{p}^{n+1}, \quad (2.53)$$

$$\underline{p}^{n+1} = \underline{p}^n + \nabla^2 \underline{p}^{n+1}, \quad (2.54)$$

## 2.5. OPERATOR SPLITTING – THE FRACTIONAL STEP METHOD

where  $\mathbf{E} := \mathbf{D}\mathbf{M}^{-1}\mathbf{D}^\top$  is a consistent Poisson operator for the pressure.

The solution procedure differs a bit depending on whether we use  $\mathbb{P}_N \times \mathbb{P}_N$  or  $\mathbb{P}_N \times \mathbb{P}_{N-2}$ , since the  $\mathbb{P}_N \times \mathbb{P}_N$  must set a boundary condition for the pressure (see [9], p.312),

$$\mathbf{n} \cdot \nabla p = \mathbf{n} \cdot [-\nabla \times (\nabla \times \mathbf{u}) + \mathbf{f}], \quad \text{on } \partial\Omega, \quad (2.55)$$

while the  $\mathbb{P}_N \times \mathbb{P}_{N-2}$  does not. Roughly speaking the solution procedures for two different methods can be summarized as follow:

$\mathbb{P}_N \times \mathbb{P}_N$ , **solution procedure:**

1. Discretize in time;
2. Take continuous divergence of momentum equation to arrive at a Poisson equation for pressure, with special boundary conditions.

$\mathbb{P}_N \times \mathbb{P}_{N-2}$ , **solution procedure:**

1. Discretize in space using compatible approximation spaces;
2. Solve coupled system for pressure/velocity.



## Chapter 3

# Natural outflow boundary conditions

### 3.1 Introduction

Recall that, to obtain an unique solution to a partial differential equation suitable boundary conditions must be specified on the boundaries. However, it is not necessary that all the boundary conditions are explicitly set. “Natural” boundary conditions are such boundary conditions. In the weak form of a partial differential equation natural boundary conditions are incorporated implicitly. And since SEM uses the weak form for the spatial discretization, natural boundary conditions exist in our incompressible Navier–Stokes discretization.

Due to the implicit nature of natural boundary conditions they often go under the name “do-nothing” boundary conditions. Also the name “traction boundary condition” frequently appears, which refers to a specific form where traction is imposed on the boundary; thus the boundary condition has in this setting a specific physical meaning. However, we will mainly use the name “natural” boundary condition through the remaining of this thesis, to refer to the general form.

Before moving on to the incompressible Navier–Stokes equations, let us first present a short introduction of the background theory of natural boundary conditions. A typical example problem presented in the book [23] (p.181) is this:

*“A river with parallel straight banks  $b$  units apart has stream velocity given by  $v(x, y) = v(x)j$ , where  $j$  is the unit vector in the  $y$  direction. Assuming that one of the banks is the  $y$  axis and that the point  $(0, 0)$  is the point of departure, what path should a boat take to reach the opposite bank in the shortest possible time? Assume that the speed of the boat in still water is  $c$ , where  $c^2 > v^2$ .”*

In this problem the right end–point at  $x = b$  is not specified. But it must be specified as a part of the solution to the problem. Indeed, if  $v(x)$  is a critical function (i.e. min or max), then a certain condition must hold at  $x = b$ . This condition is a natural boundary condition. The existence of a solution to this kind of problem is clear. If we start at a point  $a$ , and the path we want to follow is continuous, then there certainly exist shortest paths to point  $b$ . It does not matter if we know or not know the derivative or exact height at the  $b$  location. Thus, it is sufficient to add a

boundary condition at  $a$  and the solution will be unique.

Let us give an example to explain the mathematical framework.

**Example 3.1.** Assume that we want to find the minimal function  $y^*(x)$  to the functional

$$J = \int_{x_1}^{x_2} f(x, y, y') dx, \quad (3.1)$$

where one or both of the boundary points  $y(x_1)$  and  $y(x_2)$  are unknown. (Let us for instance assume that the problem is the problem with the boat above, where  $y(x_2)$  is unknown.) Define a set of comparison functions as  $S = \{g(x) : g(x) = y^* + \epsilon\eta(x), \epsilon > 0, \eta(x) \in \mathbb{R}\}$ . Here  $\eta(x)$  is a function that prescribes the direction in which the derivative is computed. Then substitute  $y(x, \epsilon, \eta) \in S$  in Equation (3.1) and obtain the new functional

$$J(\epsilon) = \int_{x_1}^{x_2} f(x, y^* + \epsilon\eta, (y^*)' + \epsilon\eta') dx. \quad (3.2)$$

Now, apply the standard technique of finding stationary points, i.e. derive the functions and compute the roots:

$$\left. \frac{dJ}{d\epsilon} \right|_{\epsilon=0} = \left. \frac{\partial f}{\partial y'} \eta \right|_{x_1}^{x_2} + \int_{x_1}^{x_2} \left( \frac{\partial f}{\partial y} - \frac{d}{dx} \left( \frac{\partial f}{\partial y'} \right) \eta \right) dx = 0. \quad (3.3)$$

Since  $\frac{\partial f}{\partial y} - \frac{d}{dx} \left( \frac{\partial f}{\partial y'} \right) = 0$  is the well known Euler–Lagrange equation, the minimization problem exist if the boundary term is equal to zero. i.e.

$$\left. \frac{\partial f}{\partial y'} \eta \right|_{x_1}^{x_2} = \left. \frac{\partial f}{\partial y'} \right|_{x=x_2} \eta(x_2) - \left. \frac{\partial f}{\partial y'} \right|_{x=x_1} \eta(x_1) = 0. \quad (3.4)$$

These conditions are satisfied if  $\eta(x_1) = 0$  and  $\eta(x_2) = 0$ . But in the case the boundary points are variable, i.e.  $\eta(x_1)$  and  $\eta(x_2)$  are arbitrary, then the only way to ensure that the condition (3.4) is satisfied is to choose

$$\left. \frac{\partial f}{\partial y'} \right|_{x=x_1} = 0, \quad \text{and} \quad \left. \frac{\partial f}{\partial y'} \right|_{x=x_2} = 0. \quad (3.5)$$

This conditions are exactly what we call the natural boundary conditions of the problem. Boundary conditions where  $\eta = 0$  are called essential boundary conditions.

Mathematically, an intuition of natural boundary conditions for a variational form of a partial differential equation is basically a requirement for the boundary integrals at a non-Dirichlet boundary to cancel after integration by parts has been applied in the derivation of the weak form of a PDE. The minimization problem explained above is another way to explain the concept, which may give a more physical intuition. The minimization problem of a PDE is typically a measure of the energy stored in a system.



### 3.2. NATURAL OUTFLOW CONDITIONS FOR THE INCOMPRESSIBLE NAVIER–STOKES EQUATIONS

## 3.2 Natural outflow conditions for the incompressible Navier–Stokes equations

To derive the natural boundary conditions for the incompressible Navier–Stokes equations (1.1) we choose a domain  $\Omega$  of dimension two or three with the boundary  $\Gamma$  that is partitioned into the disjoint sets of points corresponding to the outflow boundary  $\Gamma_O$  where the natural outflow boundary conditions is imposed, and  $\Gamma \setminus \Gamma_O$  which are boundaries with essential boundary conditions.

Recall from (1.1) the dimensionless form of the incompressible Navier–Stokes equations

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} + \mathbf{f}, \quad (3.6a)$$

$$\nabla \cdot \mathbf{u} = 0. \quad (3.6b)$$

A variational formulation was derived in §2.3.2. Let us quickly re-do the calculations: multiply (3.6a) with a test function  $\mathbf{v} \in H_0^1(\Omega)^d$  (see Equation (2.10)), and integrate over the domain  $\Omega$

$$\begin{aligned} \int_{\Omega} \mathbf{v} \cdot \frac{\partial \mathbf{u}}{\partial t} d\Omega + \int_{\Omega} \mathbf{v} \cdot \mathbf{u} \cdot \nabla \mathbf{u} d\Omega \\ = - \int_{\Omega} \mathbf{v} \cdot \nabla p d\Omega + \int_{\Omega} \mathbf{v} \cdot \nabla^2 \mathbf{u} d\Omega + \int_{\Omega} \mathbf{v} \cdot \mathbf{f} d\Omega, \end{aligned} \quad (3.7)$$

Then apply integration by parts on the pressure and viscous terms of (3.7) to obtain

$$\begin{aligned} \int_{\Omega} \mathbf{v} \cdot \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \mathbf{f} \right) d\Omega \\ = \int_{\Omega} (\nabla \cdot \mathbf{v}) p d\Omega - \frac{1}{Re} \int_{\Omega} \nabla \mathbf{v} \cdot \nabla \mathbf{u} d\Omega + \oint_{\Gamma} \mathbf{v} \cdot \left( -p + \frac{1}{Re} \nabla \mathbf{u} \right) \cdot \mathbf{n} d\Gamma. \end{aligned} \quad (3.8)$$

By the choice  $\mathbf{v} \in H_0^1(\Omega)^d$ , the surface integrals corresponding to essential boundary conditions cancel, and we end up with

$$\begin{aligned} \int_{\Omega} \mathbf{v} \cdot \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \mathbf{f} \right) d\Omega \\ = \int_{\Omega} (\nabla \cdot \mathbf{v}) p d\Omega - \frac{1}{Re} \int_{\Omega} \nabla \mathbf{v} \cdot \nabla \mathbf{u} d\Omega + \oint_{\Gamma_O} \mathbf{v} \cdot \left( -p + \frac{1}{Re} \nabla \mathbf{u} \right) \cdot \mathbf{n} d\Gamma. \end{aligned} \quad (3.9)$$

From which the following natural boundary condition then follows from the surface integral over  $\Gamma_O$

$$-p\mathbf{n} + \frac{1}{Re}(\mathbf{n} \cdot \nabla \mathbf{u}) = 0, \quad \text{on } \Gamma_O. \quad (3.10)$$

### 3.3 Energy balance

An energy estimate can be used to study e.g. the stability of the system (1.1). Multiply (1.1a) with  $\mathbf{u}$ , and integrate over  $\Omega$ , then we obtain

$$\frac{d}{dt}(\mathbf{u}, \mathbf{u}) + (\mathbf{u}, \mathbf{u} \cdot \nabla \mathbf{u}) = (\nabla \cdot \mathbf{u}, p) - \frac{1}{Re}(\nabla \mathbf{u}, \nabla \mathbf{u}) + \oint_{\Gamma} \mathbf{u} \cdot \left( -p + \frac{1}{Re} \nabla \mathbf{u} \right) \cdot \mathbf{n} d\Gamma. \quad (3.11)$$

Now we evaluate each terms in (3.11) separately (note that some terms cancel by Equation (1.1b))

$$\begin{aligned} \frac{d}{dt}(\mathbf{u}, \mathbf{u}) &= \frac{d}{dt} \int_{\Omega} \mathbf{u} \mathbf{u} \, d\Omega \\ &= \frac{1}{2} \frac{d}{dt} \int_{\Omega} |\mathbf{u}|^2 \, d\Omega, \\ &= \frac{1}{2} \frac{d}{dt} \|\mathbf{u}\|_{L^2(\Omega)}^2, \\ (\mathbf{u}, \mathbf{u} \cdot \nabla \mathbf{u}) &= \int_{\Omega} \mathbf{u} \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) \, d\Omega \\ &= \int_{\Omega} \mathbf{u} \cdot \frac{1}{2} \nabla |\mathbf{u}|^2 \, d\Omega \\ &= \oint_{\Gamma} \frac{1}{2} |\mathbf{u}|^2 \mathbf{u} \cdot \mathbf{n} d\Gamma - \int_{\Omega} \frac{1}{2} |\mathbf{u}|^2 \underbrace{(\nabla \cdot \mathbf{u})}_{=0} \, d\Omega \\ &= \oint_{\Gamma} \frac{1}{2} |\mathbf{u}|^2 \mathbf{u} \cdot \mathbf{n} d\Gamma, \\ (\nabla \cdot \mathbf{u}, p) &= \int_{\Omega} \underbrace{(\nabla \cdot \mathbf{u})}_{=0} \cdot p \, d\Omega = 0, \\ \frac{1}{Re}(\nabla \mathbf{u}, \nabla \mathbf{u}) &= \frac{1}{Re} \int_{\Omega} |\nabla \mathbf{u}|^2 \, d\Omega \\ &= \frac{1}{Re} \|\nabla \mathbf{u}\|_{L^2(\Omega)}^2. \end{aligned}$$

Then,

$$\begin{aligned} \frac{1}{2} \frac{d}{dt} \|\mathbf{u}\|_{L^2(\Omega)}^2 &= -\frac{1}{Re} \|\nabla \mathbf{u}\|_{L^2(\Omega)}^2 + \oint_{\Gamma \setminus \Gamma_O} \mathbf{u} \cdot \left( -p + \frac{1}{Re} \nabla \mathbf{u} - \frac{1}{2} |\mathbf{u}|^2 \right) \cdot \mathbf{n} d\{\Gamma \setminus \Gamma_O\} \\ &\quad + \oint_{\Gamma_O} \mathbf{u} \cdot \left( -p + \frac{1}{Re} \nabla \mathbf{u} - \frac{1}{2} |\mathbf{u}|^2 \right) \cdot \mathbf{n} d\Gamma_O. \end{aligned} \quad (3.12)$$

From (3.12) we can notice that the energy decay depends entirely on the boundary terms for high Reynolds numbers. If those terms are negative then the energy change over time is also negative; but if they are positive then a growth of energy

### 3.4. A STABILIZED NATURAL BOUNDARY CONDITION

is possible. Hence, we obtain the following inequality to ensure that the energy is non-positive

$$\mathbf{u} \cdot \left( -p + \frac{1}{Re} \nabla \mathbf{u} - \frac{1}{2} |\mathbf{u}|^2 \right) \cdot \mathbf{n} \Big|_{\Gamma \setminus \Gamma_O} + \mathbf{u} \cdot \left( -p + \frac{1}{Re} \nabla \mathbf{u} - \frac{1}{2} |\mathbf{u}|^2 \right) \cdot \mathbf{n} \Big|_{\Gamma_O} \leq 0. \quad (3.13)$$

If we assume that the non-outflow part of the boundary is either of Dirichlet or periodic type, then the total energy change over these boundaries is controlled. This is true due to the fact that Dirichlet boundary conditions are pre-defined and the total energy change over the periodic boundaries cancel due to the periodicity. Hence, the only boundary term that may cause uncontrolled energy growth is the outflow boundary. We get the stability constraint

$$\mathbf{u} \cdot \left( -p + \frac{1}{Re} \nabla \mathbf{u} - \frac{1}{2} |\mathbf{u}|^2 \right) \cdot \mathbf{n} \leq 0, \quad \text{on } \Gamma_O. \quad (3.14)$$

Where the two first terms cancel due to that the natural boundary condition (3.10) holds at  $\Gamma_O$ . Thus, to restrict blow ups we have

$$-\frac{1}{2} |\mathbf{u}|^2 (\mathbf{u} \cdot \mathbf{n}) \leq 0, \quad \text{on } \Gamma_O. \quad (3.15)$$

which does not hold for  $(\mathbf{u} \cdot \mathbf{n})|_{\Gamma_O} < 0$ . Let us state this result as a theorem:

**Theorem 3.1.** *If  $\Omega^d$ ,  $d = 2$  (or  $3$ ) is a closed domain with boundary  $\Gamma = \Gamma_1 \cup \Gamma_O$ ,  $\Gamma_1 \cap \Gamma_O = \emptyset$ , where  $\Gamma_1$  is of either Dirichlet or periodic type and  $\Gamma_O$  is an outflow boundary with the natural boundary condition (3.10) imposed. Then,*

$$Re \rightarrow \infty \implies \begin{cases} \frac{d}{dt} \|\mathbf{u}\|_{L^2(\Omega)}^2 \leq 0, & \mathbf{u} \cdot \mathbf{n}|_{\Gamma_O} \geq 0, \\ \frac{d}{dt} \|\mathbf{u}\|_{L^2(\Omega)}^2 > 0, & \mathbf{u} \cdot \mathbf{n}|_{\Gamma_O} < 0. \end{cases} \quad (3.16)$$

### 3.4 A stabilized natural boundary condition

A way to stabilize the natural condition (3.10) was presented in the paper by Dong, et. al (2014) [10]. It is based on the results of the previous subsection. Adding the condition  $\mathbf{u} \cdot \left( -p + \frac{1}{Re} \nabla \mathbf{u} - \frac{1}{2} |\mathbf{u}|^2 \right) \cdot \mathbf{n} = 0$  if an energy influx on  $\Gamma_O$  is present, the energy growth as  $Re \rightarrow \infty$  is eliminated. To remove the discontinuity that appears when fluxes turn from negative to positive from one time step to another a smooth step function was used

$$\Theta(\mathbf{n}, \mathbf{u}) := \frac{1}{2} \left[ 1 - \tanh \left( \frac{\mathbf{n} \cdot \mathbf{u}}{U\delta} \right) \right], \quad (3.17)$$

where  $U$  is the characteristic velocity scale, and  $\delta > 0$  is a chosen non-dimensional positive constant that regulates the amount of smoothness, i.e. as  $\delta \rightarrow 0$ ,  $\Theta(\mathbf{n}, \mathbf{u})$  approaches a step function. The new stabilized outflow boundary condition reads

$$-p\mathbf{n} + \frac{1}{Re} (\mathbf{n} \cdot \nabla \mathbf{u}) - \left[ \frac{1}{2} |\mathbf{u}|^2 \Theta(\mathbf{n}, \mathbf{u}) \right] \mathbf{n} = \mathbf{0}, \quad \text{on } \Gamma_O. \quad (3.18)$$

### 3.5 Problems with natural boundary conditions in pipe flow computations

Some interesting remarks about natural boundary conditions in pipe flow computations were proposed by Heywood et al. [17], and Rannacher [35]. These are reviewed in this section.

#### 3.5.1 A hidden pressure condition

Consider a two dimensional cross section of a pipe  $\Omega = [0, L_1] \times [-L_2, L_2]$ , and with flow in the  $x$  direction. On the upper and lower boundaries “no slip” boundary conditions are imposed. If we compute the integral over the outflow boundary were the natural boundary condition (3.10) is imposed we get

$$0 = \oint_{\Gamma_O} \left( -p + \frac{1}{Re} \frac{\partial u}{\partial x} \right) dx = - \int_{-L_2}^{L_2} p dx + \frac{1}{Re} \int_{-L_2}^{L_2} \frac{\partial u}{\partial x} dx = - \int_{-L_2}^{L_2} p dx. \quad (3.19)$$

where the fundamental theorem of calculus was applied to cancel the  $\frac{1}{Re} \int_{-L_2}^{L_2} \frac{\partial u}{\partial x} dx$  term. Indeed, this means that the mean pressure is zero over the outflow boundary. Hence in a bifurcated pipe with two (or more) outlets where natural boundary conditions are imposed on each, the mean pressure must be zero at all the outlets. If the outlets are placed asymmetrically, this is certainly wrong, and gives a physically incorrect solution. This result is interesting since it appear because of the pressure dependence, which is the main difference between the natural and the convective boundary conditions that we present in the next chapter.

#### 3.5.2 Non-uniqueness

Due to the incompressibility constraint (1.1b) it is possible to change the variational formulation of the incompressible Navier–Stokes equations using different modifications of the transport and diffusion terms and still preserve the same continuous solution. It is possible to rewrite the convection term of the variational form (2.13) in the following way

$$(\mathbf{u} \cdot \nabla \mathbf{u}, \mathbf{v}) = \frac{1}{2}(\mathbf{u} \cdot \nabla \mathbf{u}, \mathbf{v}) - \frac{1}{2}(\mathbf{u} \cdot \nabla \mathbf{v}, \mathbf{u}) \quad (3.20)$$

and similarly for the viscous term

$$\frac{1}{Re}(\nabla \mathbf{u}, \nabla \mathbf{v}) = \frac{1}{Re} \left( \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^\top), \frac{1}{2}(\nabla \mathbf{v} + \nabla \mathbf{v}^\top) \right). \quad (3.21)$$

That (3.20) and (3.21) hold are proven in §A.6. These two modifications result in two new natural boundary conditions; see the theorems below. The first one is indeed the same boundary condition as the stabilized natural boundary condition (3.18) when the stability fix is activated, which means that the stabilized condition itself also is a true natural boundary condition of the discrete variational form (2.14).

### 3.5. PROBLEMS WITH NATURAL BOUNDARY CONDITIONS IN PIPE FLOW COMPUTATIONS

**Theorem 3.2.** *If the (3.20) modification is applied, then we obtain the following natural boundary condition*

$$\left(-p + \frac{1}{Re} \nabla \mathbf{u} - \frac{1}{2} |\mathbf{u}|^2\right) \cdot \mathbf{n} = \mathbf{0}, \quad \text{on } \Gamma_O. \quad (3.22)$$

*Proof.* Integration by parts of (3.20) yields

$$\begin{aligned} (\mathbf{u} \cdot \nabla \mathbf{u}, \mathbf{v}) &= \frac{1}{2} (\mathbf{u} \cdot \nabla \mathbf{u}, \mathbf{v}) - \frac{1}{2} (\mathbf{u} \cdot \nabla \mathbf{v}, \mathbf{u}) \\ &= \frac{1}{2} \int_{\Omega} \mathbf{v} \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) \, d\Omega + \frac{1}{2} \int_{\Omega} \mathbf{v} \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) \, d\Omega - \frac{1}{2} \oint_{\Gamma} \mathbf{v} \cdot |\mathbf{u}|^2 \cdot \mathbf{n} \, d\Gamma \\ &= \int_{\Omega} \mathbf{v} \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) \, d\Omega - \frac{1}{2} \oint_{\Gamma} \mathbf{v} \cdot |\mathbf{u}|^2 \cdot \mathbf{n} \, d\Gamma. \end{aligned}$$

Substitute this into the variational form (3.7) and we get the boundary term

$$\oint_{\Gamma} \mathbf{v} \cdot \left(-p + \frac{1}{Re} \nabla \mathbf{u} - \frac{1}{2} |\mathbf{u}|^2\right) \cdot \mathbf{n} \, d\Gamma$$

From which the natural boundary condition (3.22) follow.  $\square$

**Theorem 3.3.** *If the (3.21) modification is applied, then we obtain the following natural boundary condition*

$$\left(-p + \frac{1}{2Re} [\nabla \mathbf{u} + \nabla \mathbf{u}^T]\right) \cdot \mathbf{n} = \mathbf{0}, \quad \text{on } \Gamma_O. \quad (3.23)$$

*Proof.* This follows directly from the linearity of inner products after substituting (3.21) into (3.7).  $\square$

Heywood et al. applied the original natural boundary condition (3.10) and the two different ones (3.22), and (3.23), on a Poiseuille flow problem. The original natural boundary condition showed expected streamlines, but the two other conditions implied non-physical behavior at the outflow, with streamlines either pointing inwards or outwards instead of straight. Since the boundary condition (3.22) indeed is similar to the stabilized boundary condition (3.18), the natural and the stabilized natural boundary conditions are not expected to yield similar results.



## Chapter 4

# Convective outflow boundary conditions – Theory and implementation

### 4.1 Introduction

A convective boundary condition is a specific artificial boundary condition that neglects the diffusive effects near the boundary and assumes that the flow is purely advective. The general form of the convective boundary condition yields

$$\frac{\partial \phi}{\partial t} + \mathbf{c} \cdot \nabla \phi = 0, \quad \text{on } \Gamma_O, \quad (4.1)$$

where  $\phi$  is some physical quantity to be convected and  $\mathbf{c}$  the convecting speed. Convective boundary conditions often go under the names “radiation”, “wave-permeable” or “non-reflective” boundary conditions. Other common names that relate to specific convective outflow boundary conditions are the “Sommerfeld radiation condition” and the “Orlanski boundary condition”.

The chapter is structured as follows: first we explain the background theory, then we present several ways to implement the convective condition in the spectral element framework.

### 4.2 Exact "Non-reflecting" boundary conditions

In the following section we review the fact of constructing “non-reflective” boundary conditions of the wave equation. For highly convective problems the dissipation is negligible, and the flow therefore becomes close to hyperbolic. By the superposition principle of waves [14] it is possible to explain such flows in terms of wave modes.

#### 4.2.1 The one dimensional case

The one dimensional wave equation reads

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}, \quad (4.2)$$

where  $c$  is the wave speed. By the linearity of the differential operators in (4.2) it is possible to separate (4.2) into

$$\left(\frac{\partial}{\partial t} + c\frac{\partial}{\partial x}\right)\left(\frac{\partial}{\partial t} - c\frac{\partial}{\partial x}\right)u = 0. \quad (4.3)$$

Equation (4.3) implies the existence of two types of solutions: right going solutions  $u^\rightarrow(x, t)$  that satisfy

$$\frac{\partial u^\rightarrow}{\partial t} + c\frac{\partial u^\rightarrow}{\partial x} = 0, \quad (4.4)$$

and left going solutions  $u^\leftarrow(x, t)$  that satisfy

$$\frac{\partial u^\leftarrow}{\partial t} - c\frac{\partial u^\leftarrow}{\partial x} = 0. \quad (4.5)$$

By the superposition principle, the solution to (4.2) can then be expressed as a sum of the left and right going solutions

$$u(x, t) = u^\rightarrow(x, t) + u^\leftarrow(x, t). \quad (4.6)$$

An exact continuous solution to the convective equation  $u_t + cu_x = 0$  is therefore the solution of the wave equation where the right going solutions are the solutions  $u = u^\rightarrow$ , and where there are no left going solutions, i.e.  $u^\leftarrow = 0$ . This is exactly what the convective boundary condition tries to achieve; and such boundary conditions are commonly called “non-reflecting” boundary conditions. If the flow is purely hyperbolic, then the convective boundary condition is exact in the continuous setting. But if the convective boundary condition is an approximation, then the errors do certainly imply wave solutions in the opposite direction; wave solutions that satisfy (4.5). This is always present in numerical simulations in some amount, due to round off errors and discretization errors of the approximation  $q \approx 0$ , which can only be satisfied down to machine precision.

### 4.2.2 Higher dimensions

In two and higher dimensions everything becomes much more complicated instantly. Now waves are allowed to travel in infinitely many directions. Consider the wave equation in two dimensions

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right). \quad (4.7)$$

Let us for simplicity assume that  $c^2 = 1$ . Then, in order to find “non-reflective” boundary conditions, (4.7) is rewritten to

$$\left( \frac{\partial^2}{\partial x^2} - \left( \frac{\partial^2}{\partial t^2} - \frac{\partial}{\partial y^2} \right) \right) u = 0. \quad (4.8)$$



#### 4.2. EXACT "NON-REFLECTING" BOUNDARY CONDITIONS

Splitting of the differential operator gives

$$\left( \frac{\partial}{\partial x} - \sqrt{\frac{\partial^2}{\partial t^2} - \frac{\partial^2}{\partial y^2}} \right) \left( \frac{\partial}{\partial x} + \sqrt{\frac{\partial^2}{\partial t^2} - \frac{\partial^2}{\partial y^2}} \right) u = 0. \quad (4.9)$$

Hence, a two dimensional analogue to (4.4) yields

$$\left( \frac{\partial}{\partial x} + \sqrt{\frac{\partial^2}{\partial t^2} - \frac{\partial^2}{\partial y^2}} \right) u^{\rightarrow} = 0. \quad (4.10)$$

However, the differential operator  $\sqrt{\frac{\partial^2}{\partial t^2} - \frac{\partial^2}{\partial y^2}}$  is non-local<sup>1</sup> in space and time, and does therefore become an integral equation that cannot be approximated locally by a numerical method. Approximations are therefore unavoidable, even in the continuous case.

A convenient way to consider traveling waves is to study wave modes of the form

$$u = z\kappa = e^{i(\xi x + \eta y + \omega t)} \quad (4.11)$$

where  $z = e^{i\omega t}$  is the “temporal discretization factor” and  $\kappa = e^{i\xi x + i\eta y}$  the “spatial amplification factor”. Such a solution satisfies the wave equation (4.7) exactly, but not for all  $(\xi, \eta)$  and  $\omega$ . By substituting the wave mode into the wave equation (with  $c = 1$ ) the continuous dispersion relation  $\omega(\xi/\eta)$  of the wave equation is obtained

$$\omega^2 = \xi^2 + \eta^2. \quad (4.12)$$

Dividing Equation (4.12) with  $\omega^2$  results in the relation of  $x$  and  $y$  velocities with constant velocity one, i.e.

$$\left( -\frac{\xi}{\omega}, -\frac{\eta}{\omega} \right) = (-\cos \theta, -\sin \theta), \quad \theta \in [0, 2\pi]. \quad (4.13)$$

where the angle  $\theta$  represents the angle of incidence for a wave traveling in the positive  $x$  direction (see Figure 4.1). Thus, the solution to (4.7) admits plane wave solutions traveling with constant speed one in all possible directions. If the angle  $\theta$  lies in the interval  $[-\pi, \pi]$  then the group velocity is positive and we have a right moving solution, and for the angle  $\theta \in [\pi, 3\pi/2]$  the solution is left moving. See Figure 4.1a. Waves in the  $x$  direction are represented by

$$\xi = \pm \omega \sqrt{1 - \left( \frac{\eta}{\omega} \right)^2}, \quad (4.14)$$

---

<sup>1</sup>In order to check whether a differential equation holds at a certain point one only need to know values in a neighborhood of the point if the differential operator is local. For a non-local differential operator one needs in particular information from points far away from the point considered. This is for instance the case if the differential operator involves integrals.

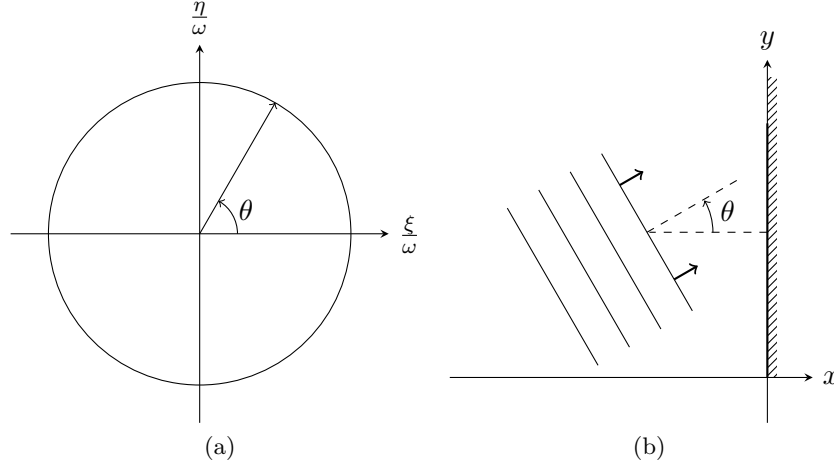


Figure 4.1: (a) Dispersion relation of the two dimensional wave equation in Fourier space. (b) Angle of incidence for a plane wave packet.

with  $\frac{\eta}{\omega} = \sin \theta \in [-1, 1]$ , and  $\theta \in [-\pi, \pi]$ . Due to the square root, the relation (4.14) has two solutions for each  $\xi$ :

$$\xi = \omega \sqrt{1 - \left(\frac{\eta}{\omega}\right)^2}, \quad (4.15)$$

and

$$\xi = -\omega \sqrt{1 - \left(\frac{\eta}{\omega}\right)^2}, \quad (4.16)$$

where (4.15) is the dispersion relation of a left moving solution, and (4.16) the dispersion relation of a right moving solution. Let us study the positive equation (4.15). This dispersion relation cannot be a dispersion relation of a true differential equation because of the square root (see the splitting (4.10)). The best one can achieve is to approximate the square root with a polynomial expansion (e.g. Taylor series). Then one get a so called “pseudo-differential operator” instead. But as a consequence the normal derivative  $\partial_x u$  at any outflow boundary point cannot be computed locally, either in space or time. To get a local boundary condition it is instead better to approximate the pseudo-differential, by a real differential operator. But this also means that one must accept some spurious errors in the solution. This idea was proposed by Engquist and Majda [12]. Their approximation of the square root yields

$$\xi = \omega \sqrt{1 - \left(\frac{\eta}{\omega}\right)^2} \approx \omega r(\eta/\omega), \quad r(\eta/\omega) \in \mathbb{Q}. \quad (4.17)$$

#### 4.2. EXACT "NON-REFLECTING" BOUNDARY CONDITIONS

Since  $r(\eta/\omega)$  is chosen as a rational function we can write

$$r(s) = \frac{p_m(s)}{q_n(s)}, \quad s := \frac{\eta}{\omega}, \quad (4.18)$$

where  $p_m(s), q_n(s)$  are polynomials of orders  $m$ , and  $n$ . Recall that, a side of the basic requirement that an outflow boundary condition must yield a well-posed problem, the main goal is non-reflectivity. Reflections may actually be a bigger issue than using a physically incorrect boundary condition since they influence the whole solution. Approximating the rational function  $r(s)$  in (4.18) exactly yields a non-reflective boundary condition in the continuous setting. Hence, partial differential equations that has a dispersion relation that satisfies  $r(s)$  as exact as possible is the main goal.

A natural choice of estimating  $r(s)$  is by Taylor expansion. But, it can be shown that Taylor expansions is unstable for order greater than two and cannot therefore be used generally [12]. Engquist and Majda proposed ways to approximate  $r(s)$  by Padé approximants instead, which showed to be a better alternative. It is worthy to mention that the Padé expansion is not proven to be the optimal. Other ways have been considered and proven to yield well-posed results too (Trefethen and Halpern [42]). For the easiest case:  $p_0$ , and  $q_0$  in (4.18) the Padé approximant is simply one, resulting in  $\xi = \omega$ , which is the dispersion relation of the first order advection equation (the convective condition); which is obviously only true if  $\eta \rightarrow 0$  (see Figure 4.1). Higher order approximations can be considered in the Table 4.1. As we can notice, the boundary conditions become very complex already at the third order approximation. Also the Padé approximants are very accurate if  $s \rightarrow 0$ , but not if  $s \pm 1$  ( $\theta \approx \pi/2, 3\pi/2$ ); see Figure 4.2. A lot of studies have been made for the well-posedness of these boundary conditions. For example Trefethen and Halpern [42] found that if  $r(s)$  is approximated by sufficiently many points in  $(-1, 1)$ , then well-posedness is assured. They also found that absorbing boundary conditions of this kind are well-posed if and only if  $(m, n)$  lies in one of two distinct diagonals in the Padé table, where the Padé table is the table that is obtained by write out all possible choices of  $(m, n)$ ,  $m, n \geq 0$ .

The Padé approximants and the exact boundary conditions presented by Engquist and Majda are exact in the continuous case. They make it possible to specify outflow boundary conditions in forms of a differential equation, but since the expressions become very complicated already at lower orders they are not easy to discretize. The convective boundary condition is the first Engquist-Majda boundary condition, but certainly the less accurate. All angles of incidence of incoming waves to the boundary other than the correct will imply a poor approximation of the dispersion relation of the wave, which implies reflections (waves in the opposite direction).

$r(s)$	$f_i(u)$
1	$f_1(u) = \left( \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} \right) \Big _{x=x_O} = 0$
$1 - \frac{1}{2}s^2$	$f_2(u) = \left( \frac{\partial^2 u}{\partial t^2} + \frac{\partial^2 u}{\partial t \partial x} - \frac{1}{2} \frac{\partial^2 u}{\partial y^2} \right) \Big _{x=x_O} = 0$
$\frac{1 - \frac{3}{4}s^2}{1 - \frac{1}{4}s^2}$	$f_3(u) = \left( \frac{\partial^3 u}{\partial t^3 \partial x} - \frac{1}{4} \frac{\partial^3 u}{\partial x \partial y^2} - \frac{\partial^3 u}{\partial^3 t} + \frac{3}{4} \frac{\partial^3 u}{\partial t \partial y^2} \right) \Big _{x=x_O} = 0$
$\vdots$	$\vdots$
	$f_{N+1}(u) = \left( \frac{\partial}{\partial t} f_N(u) - \frac{1}{4} \frac{\partial u}{\partial y} f_N(u) \right) \Big _{x=x_O} = 0$

Table 4.1: The Engquist-Majda boundary conditions.

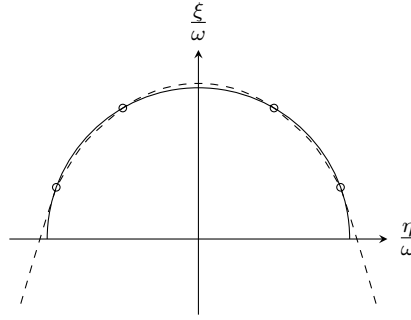


Figure 4.2: The approximation problem. The solid line half circle is the square root solution  $\sqrt{1-s^2}$  to be approximated, and the dashed line a high order polynomial approximation.

### 4.3 The Sommerfeld radiation condition

Named after the theoretical physicist Arnold Sommerfeld (1868-1951); the Sommerfeld radiation boundary condition is a classical simple form of the convective condition (4.1), by choosing the convection speed  $\mathbf{c}$  constant. The Sommerfeld radiation condition was first used for electromagnetic waves to eliminate incoming waves in the solution generated from an oscillating dipole antenna. The term “radiation boundary condition” was found by Sommerfeld, who defined it as the boundary condition that “the sources must be sources, not sinks, of energy. The energy which is radiated from the sources must scatter to infinity; no energy may be radiated from infinity into the field.” ([40], p.189).

The problem of determining  $\mathbf{c}$  is certainly crucial, but not always possible. The Sommerfeld radiation condition is a great choice if the convection speed can be accurately approximated. But if the approximation is hard to perform, the Sommerfeld condition is certainly not a great choice. Instead, one may look at other choices to estimate the convection speed iteratively.

## 4.4 Estimating the convection speed

If the Sommerfeld condition does not give appropriate results one may test to estimate the convection speed. It can be a question of estimating the speed in all directions, or to vary the speed in the normal direction to the boundary.

The choice of convection speed for the convective condition is often a non trivial task, especially if the flow contains vortices etc. If we want to accurately compute the convective boundary condition the choice of convecting speed may indeed be the most important thing in the whole implementation. Wrong convection speeds lead to reflections; this is confirmed by our numerical results in §5.4.2.

### 4.4.1 The Orlanski scheme

The classical method to estimate a varying convection speed is to use a “Orlanski type” boundary condition, which is a commonly used name for computing the convection speed locally. The “Orlanski boundary condition” refers to the well cited work by Orlanski (1976) [29]. Orlanski proposed a method of estimating the convection speed from the solution at the previous time step. The basic idea stems from rewriting the convective condition  $\phi_t + C\phi_x = 0$  as

$$C = \frac{\partial\phi/\partial t}{\partial\phi/\partial x} \quad (4.19)$$

and then discretize the derivatives using the Leap-Frog finite differences scheme. The method has shown some success, but does not directly relate to the CFL<sup>2</sup> number which directly relates to the stability of the scheme. Orlanski proposed ways to get rid of the CFL related problem by limit the value of  $C$  to the interval

$$0 \leq C \leq \frac{\Delta x}{\Delta t}. \quad (4.20)$$

Then one can freely choose  $\Delta x$  and  $\Delta t$  such that stability is ensured. Several tests have been made regarding both the Orlanski scheme and the estimation of  $C$  in the finite differences community; the general conclusion is that the Orlanski scheme results in a  $C$  that is often oscillating with time, either being over or under-predicted.

In a spectral element framework, the implementation of the Orlanski scheme is much more complicated than the original finite differences approach. Now we have a non-uniform grid, that means a varying  $\Delta x$ . The  $\Delta x$  between a boundary point and its neighbor is often very small. Hence,  $\Delta t$  must be chosen very small to ensure the relation (4.20).

The Orlanski method can in many situations be a nice tool to use. However, if the flow is dominated with strong disturbances with large local velocity derivatives in space, then it will be very hard to gain stable results. Since we are mainly interested in such flows, we do not use the Orlanski scheme in our implementations.

---

<sup>2</sup>The CFL condition (Courant–Friedrich–Levy, 1928) is a non-dimensional number that is a necessary condition for stability. For the BDF3/EXT3 discretization, the CFL bound is  $|\mathbf{u}|\Delta t/\Delta x \leq 0.6339$ .

#### 4.4.2 Exponential weighted moving average

As was just mentioned; often in high Reynolds number flows, and in particular when we are concerned with turbulence, the flow is dominated by flow instabilities such as eddies and vortices. Using a local estimate of the convection speed from the velocity fields will certainly not give the right group velocity of the flow over the boundary. Vortices and eddies may indeed give a negative local velocity, even though they are superimposed on top of a base flow.

One way to obtain a smooth estimate of a solution over time is to use a low-pass filter. The basic idea is to disregard the disturbances in the flow and therefore obtain a sufficiently good estimation of the base flow.

The exponentially weighted moving average (EWMA) filter (Robert Goodell Brown in 1956) is a good choice, since it unlike some other averaging methods does not require any minimum number of observations to be made before it begins to produce results. For a continuous function  $q(t)$  EWMA is identical to the discrete first-order causal low-pass filter

$$\bar{q} = \int_{-\infty}^t T(\tau - t, \Delta) q(\tau) d\tau, \quad (4.21)$$

where  $\bar{q}$  is the temporally filtered quantity,  $T$  is the parameterized filter kernel, and  $\Delta$  is its associated filter width. There are basically two requirements the kernel  $T$  must satisfy: (i) it must be positive, and (ii) if  $\Delta \rightarrow 0$  then  $T \rightarrow \delta$ , where  $\delta$  is the dirac delta function [20]. One simple kernel that fulfills (i)-(ii) is the exponential kernel

$$T(\tau - t, \Delta) = \frac{1}{\Delta} \exp\left(-\frac{\tau - t}{\Delta}\right). \quad (4.22)$$

By substitution into (4.21) one obtains

$$\bar{q} = \frac{1}{\Delta} \int_{-\infty}^t \exp\left(-\frac{\tau - t}{\Delta}\right) q(\tau) d\tau. \quad (4.23)$$

The integral formulation of the filter (4.23) is not practical since it requires the storage of the complete time history of the signal  $q$ . Instead the differential form is preferred, which is obtained by derivating (4.23) with respect to  $t$

$$\frac{d\bar{q}}{dt} = \frac{q - \bar{q}}{\Delta}, \quad (4.24)$$

which can easily be discretized in time. Applying the BDFk scheme (§2.4.1) yields

$$\frac{1}{\Delta t} \sum_{i=0}^k \beta_i \bar{q}^{n+1-i} = \frac{q^{n+1} - \bar{q}^{n+1}}{\Delta}. \quad (4.25)$$

Rearrangement gives

$$\bar{q}^{n+1} = \frac{\left(\frac{\Delta t q^{n+1}}{\Delta} - \sum_{i=1}^k \beta_i \bar{q}^{n+1-i}\right)}{\beta_0 + \Delta t / \Delta}. \quad (4.26)$$

#### 4.4. ESTIMATING THE CONVECTION SPEED

The most common form of this discretization is to use the first order BDF1 (Euler implicit) version, which yields

$$\bar{q}^{n+1} = \alpha q^{n+1} + (1 - \alpha) \bar{q}^n, \quad (4.27)$$

where

$$\alpha = \left( \frac{\Delta}{\Delta + \Delta t} \right). \quad (4.28)$$

The parameter  $\alpha \in (0, 1)$  controls the strength of the filter. The practical explanation of the variables simply are:  $q^n$  is the solution at time-step  $n$ , and  $\bar{q}^n$  is the filtered solution at time-step  $n$ . Some examples of the exponential filter (4.27) can be seen in Figure 4.3.

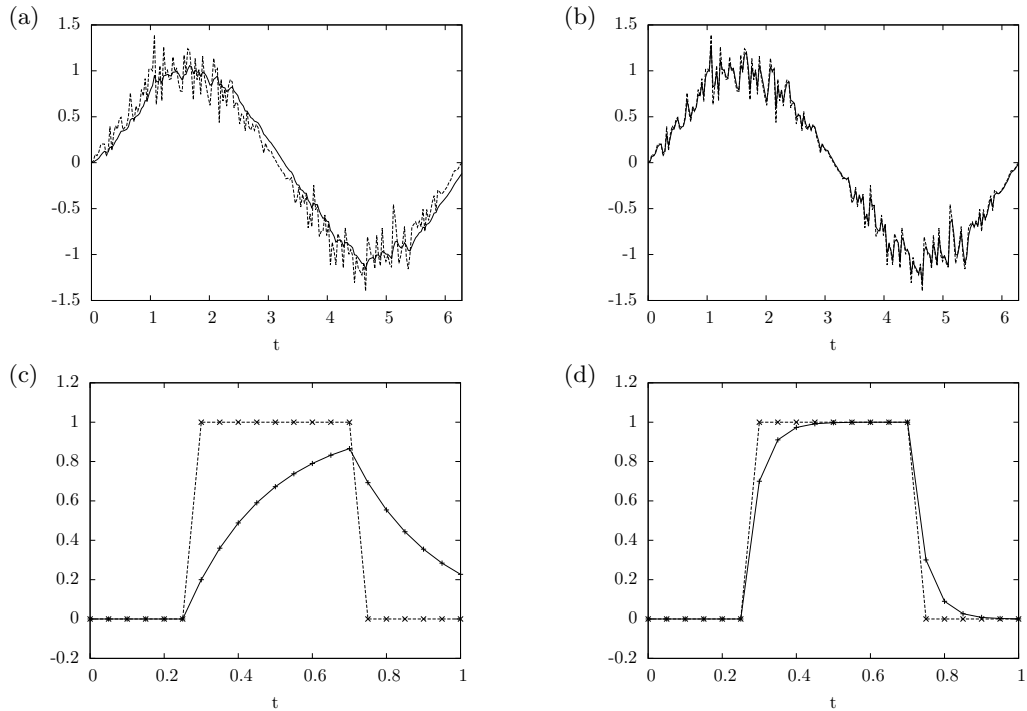


Figure 4.3: Exponential filter applied on a noisy sinus curve: (a)  $\alpha = 0.2$ , (b)  $\alpha = 0.7$ ; and the filter applied to a typical discontinuity: (c)  $\alpha = 0.2$ , (d)  $\alpha = 0.7$ .

## 4.5 Nek5000 implementations

In this section we present different discretization techniques of the convective condition (4.1) to use with the Nek5000 solver. In a first implementation we apply the BDFk/EXTk scheme on the convective condition. In a second implementation we apply the exact solution of the convective equation that satisfies linear characteristic lines in time. And the third implementation considered, that is still under development, is based on the same principle as the second but now non-linear characteristic curves (based on the OIFS/Characteristic implementation) are considered instead of linear ones.

### 4.5.1 BDFk/EXTk

The discrete variational form of (4.1) yields:

Find  $\mathbf{u} \in V_N$  such that

$$\frac{d}{dt}(\mathbf{u}, \mathbf{v})_{GLL} = -(\mathbf{c} \cdot \nabla \mathbf{u}, \mathbf{v})_{GLL}, \quad \forall \mathbf{v} \in V_{0,N}, \quad (4.29)$$

where the quadrature  $(\cdot, \cdot)_{GLL}$  is computed as in §A.2.2. Proceeding as in §2.3.2 we end up with the following semi-discrete form

$$\mathbf{M} \frac{d\mathbf{u}}{dt} = -\hat{\mathbf{C}}\mathbf{u} \quad (4.30)$$

where  $\mathbf{M}$  is the stiffness matrix defined in (2.23) and where  $\hat{\mathbf{C}}$  is given by (shall not be confused with  $\mathbf{C}$  in §2.3.5)

$$\hat{\mathbf{C}} = \mathbf{M} \left( \sum_{i=1}^d c_i D_i \right) \quad (4.31)$$

where  $\{c_i\}_{i=1}^d$  are the convection speeds, and  $D_i$  the differentiation matrices (2.27)-(2.28). Note that  $\hat{\mathbf{C}}$  may be time-dependent if e.g. a exponential moving average technique (§4.4.2) is used.

Time discretization is applied to the left hand side of (4.30) by the BDFk scheme

$$\frac{d}{dt} \mathbf{M}\mathbf{u}^{n+1} \approx \frac{1}{\Delta t} \mathbf{M} \sum_{i=0}^k \beta_i \mathbf{u}^{n+1-i}, \quad (4.32)$$

where  $\{\beta_i\}_{i=0}^k$  is the set of backward differentiating coefficients (see table A.2). For the convective term of (4.30) the explicit extrapolation scheme EXTk is applied

$$\hat{\mathbf{C}}^{n+1} \mathbf{u}^{n+1} \approx \sum_{i=1}^k \alpha_i \hat{\mathbf{C}}^{n+1-i} \mathbf{u}^{n+1-i}, \quad (4.33)$$



#### 4.5. NEK5000 IMPLEMENTATIONS

where  $\{\alpha\}_{i=1}^k$  are the extrapolation coefficients (see table A.3). Equations (4.32) and (4.33) together yields

$$\underline{\mathbf{u}}^{n+1} = \left[ -\sum_{i=1}^k \beta_i \underline{\mathbf{u}}^{n+1-i} - \Delta t \mathbf{M}^{-1} \left( \sum_{i=1}^k \alpha_i \hat{\mathbf{C}}^{n+1-i} \underline{\mathbf{u}}^{n+1-i} \right) \right] / \beta_0. \quad (4.34)$$

This expression is valid for all points in  $\Omega$ . To get a matrix form that treats the boundary points only, we add a mask  $\mathcal{M}_{OBC}$  which is one for the outflow boundary points and zero elsewhere (like the subroutine `userbc()` do in the Nek5000 code; see, §C). The final form of the boundary condition can now be stated as

$$\underline{\mathbf{u}}^{n+1} = \left[ -\sum_{i=1}^k \beta_i \underline{\mathbf{u}}^{n+1-i} - \Delta t \mathcal{M}_{OBC} \mathbf{M}^{-1} \left( \sum_{i=1}^k \alpha_i \hat{\mathbf{C}}^{n+1-i} \underline{\mathbf{u}}^{n+1-i} \right) \right] / \beta_0. \quad (4.35)$$

##### 4.5.2 Linear characteristic method

It is well known that the convective equation (4.1) has an exact solution for suitable initial conditions. Any initial condition  $\phi_0(\mathbf{x}) := \phi(\mathbf{x}, t = 0)$  simply moves with constant shape in space. A specific point travels along a linear trajectory in time; after a so called characteristic line  $\frac{d\mathbf{x}}{dt} = \mathbf{c}$ , resulting in the continuous solution  $\phi(\mathbf{x}, t) = \phi_0(\mathbf{x} + \mathbf{c}t)$ . The discrete solution follows as

$$\phi(\mathbf{x}, t^{n+1}) = \phi(\mathbf{x} - \mathbf{c}\Delta t, t^n), \quad (4.36)$$

for an arbitrary initial condition  $\phi_0(\mathbf{x})$ . Thus, if the condition (4.36) is imposed as an outflow boundary condition over  $\mathbf{u}^n$  at  $\mathbf{x} = \mathbf{x}_O$  (outflow boundary points), then

$$\mathbf{u}^{n+1}(\mathbf{x}_O) = \mathbf{u}^n(\mathbf{x}_O - \mathbf{c}\Delta t). \quad (4.37)$$

This implementation in comparison to the BDF3/EXT3 discretization is purely local. For each boundary point we only need to pick the solutions at the locations  $\mathbf{x}_O - \mathbf{c}\Delta t$ . Figure 4.4 shows a one-dimensional case.

One problem we are concerned with in this implementation is that the locations  $\mathbf{x}_O - \mathbf{c}\Delta t$  are significantly not points of the grid, and must therefore be interpolated. We use linear interpolation for this purpose,

$$\mathbf{u}^n(\mathbf{x}_O - \mathbf{c}\Delta t) = \mathbf{u}^n(\mathbf{x}_{(O-1)}) + \mathbf{u}(\mathbf{x}_O - \mathbf{c}\Delta t - \mathbf{x}_{(O-1)}) \frac{\mathbf{u}^n(\mathbf{x}_O) - \mathbf{u}^n(\mathbf{x}_{(O-1)})}{\mathbf{x}_O - \mathbf{x}_{(O-1)}}, \quad (4.38)$$

where  $\mathbf{x}_{(O-1)}$  is the location of the point one  $\mathbf{x}$ -step into the domain from the outflow boundary. More sophisticated (higher order) interpolation methods can certainly be used too, and are also to prefer for deformed grids etc. However, the interpolation may itself be the most computational expensive part of this implementation.

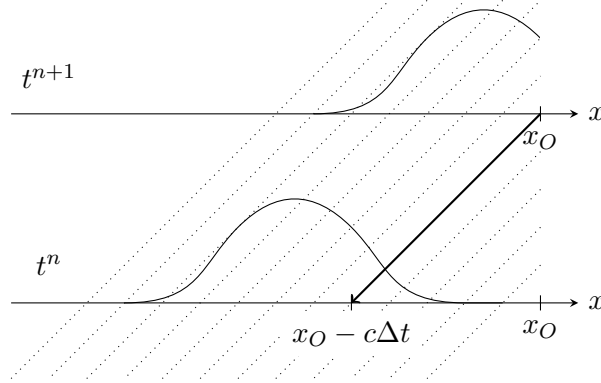


Figure 4.4: The solution to the one dimensional advection equation at an outflow boundary can be written as  $u(x_O, t^{n+1}) = u(x_O - c\Delta t, t^n)$ . The solution is constant along its characteristic lines (dotted). Here the initial condition is a Gaussian pulse.

### 4.5.3 Curved characteristics method (OIFS)

If we assume that the convection speed of the convective equation (4.1) is a time-dependent smooth function, then the characteristic lines are curved, i.e. the solution to  $\phi_t + c(t)\phi_x$  is  $x'(t) = c(t)$ . In the discrete setting where the convection speed must be approximated, this can lead to inaccurate approximations, especially if drastic changes of the derivative of  $c(t)$  appear in between two time steps.

Since curved characteristics are part of the Nek5000 code in the OIFS /Characteristic time discretization (§2.4.2), the same sub-routines can be used. The curved characteristics are then treated by Runge Kutta sub-stepping. Even though we through this implementation can use more accurate convection speeds, the convective condition is still a rather rough approximation to the incompressible Navier Stokes equations. The usefulness of this implementation regarding accuracy is therefore open for further discussion. For instance, if the convection speed cannot be estimated with a great accuracy, it is no reason to use this implementation over the linear analog. However, if the OIFS/Characteristics time-stepping is used to ensure stability, the Runge–Kutta sub-stepping may be used also for the boundary points.

## Chapter 5

# Numerical results

### 5.1 Introduction

Numerical simulations are performed to test several properties of the outflow boundary conditions presented in §§3,4. Convergence rates in both time and space are investigated, and a detailed comparison of natural against convective outflow boundary conditions is performed in two highly convective test cases: a single disturbance (vortex) moving across the outflow boundary, and the classical flow past a circular cylinder case (Kàrmàn vortex street). Accuracy is analyzed by comparing the solution to a reference solution, computed either from an exact solution or from the solution for an elongated domain. Stability is considered by studying the effects as the strength of disturbances is increased.

Four outflow boundary conditions are considered; see below. The function  $\mathbf{f}_b$  is a forcing function that is added to all the conditions (which is equal to the zero vector in the original setting).

**NBC1.** The natural boundary condition of the variational form derived in §3.2, i.e. the Nek5000 “O” condition:

$$-p\mathbf{n} + \frac{1}{Re}(\mathbf{n} \cdot \nabla \mathbf{u}) = \mathbf{f}_b, \quad \text{on } \Gamma_O. \quad (5.1)$$

**NBC2.** The stabilized natural boundary condition proposed by Dong et al. [10], presented in §3.4

$$-p\mathbf{n} + \frac{1}{Re}(\mathbf{n} \cdot \nabla \mathbf{u}) - \left[ \frac{1}{2} |\mathbf{u}|^2 \Theta(\mathbf{n} \cdot \mathbf{u}) \right] \mathbf{n} = \mathbf{f}_b, \quad \text{on } \Gamma_O, \quad (5.2)$$

where negative flux is controlled by the smooth step function

$$\Theta(\mathbf{n} \cdot \mathbf{u}) = \frac{1}{2} \left[ 1 - \tanh \left( \frac{\mathbf{n} \cdot \mathbf{u}}{U\delta} \right) \right], \quad (5.3)$$

where the smoothness is regulated by the parameter  $\delta$ , and where  $U$  is base flow magnitude in the stream-wise direction.

**CBC1.** The BDFk/EXTk ( $k = 2, 3$ ) discretized convective boundary condition derived in §4.5.1,

$$\underline{\mathbf{u}}^{n+1} = \left[ -\sum_{i=1}^k \beta_i \underline{\mathbf{u}}^{n+1-i} - \Delta t \mathcal{M}_{OBC} \mathbf{M}^{-1} \left( \sum_{i=1}^k \alpha_i \hat{\mathbf{C}}^{n+1-i} \underline{\mathbf{u}}^{n+1-i} \right) + \underline{\mathbf{f}}_b^n \right] / \beta_0. \quad (5.4)$$

where  $\alpha_i$  and  $\beta_i$  are the EXT3 respective BDF3 coefficients (see §§A.4,A.5).

**CBC2.** The linear characteristic boundary condition derived in §4.5.2,

$$\mathbf{u}(\mathbf{x}, t^{n+1}) = \mathbf{u}(\mathbf{x} - \Delta t \mathbf{c}^n \mathbf{n}, t^n) + \mathbf{f}_b^n, \quad \text{on } \Gamma_O. \quad (5.5)$$

For all tests in this chapter the BDFk/EXTk time discretization of Nek5000 is used. If nothing else is mentioned the time-step size is chosen as  $\Delta t = 0.001$ , and the standard element order is  $N = 12$ . For NBC1, CBC1, and CBC2 we mainly use the collocated setting  $\mathbb{P}_N \times \mathbb{P}_N$ , since we experienced it less noisy than the staggered counterpart  $\mathbb{P}_N \times \mathbb{P}_{N-2}$  (more about that in §5.4.1). The implementation of NBC2 was performed by editing already existing subroutines in the Nek5000 code corresponding to the free-stress formulation (logical `IFSTRS`, see §C, and [16]), which is only working in the  $\mathbb{P}_N \times \mathbb{P}_{N-2}$  setting at date.

## 5.2 Temporal order estimation, unsteady flow

In this section we consider an unsteady flow with a known analytic solution that satisfies the convective outflow boundary condition exactly. The goal is to study if the temporal convergence order of the solver is preserved when it is used together with the new convective conditions CBC1 and CBC2. The choice of temporal order in Nek5000 is given by the parameter `TORDER` (in the `.rea` file; see §C); which equals the order the BDFk/EXTk  $k = 2, 3$  discretization. It is easy to test that the code is consistent with `TORDER` by using exact boundary conditions. This fact has also been tested numerically before. See e.g. the example problem "eddy" in the Nek5000 repository [16].

An exact unsteady flow solution over the domain  $[0, 2\pi]^2$ , which is adapted from [44] is given by

$$u = \sin(t - x) \sin(y), \quad (5.6a)$$

$$v = -\cos(t - x) \cos(y), \quad (5.6b)$$

$$p = \frac{1}{4}(\cos(2(t - x)) - \cos(2y)). \quad (5.6c)$$

where  $t \geq 0$ . It is straight forward to test by substitution that this is a solution to the convective boundary condition  $\mathbf{u}_t + \mathbf{c} \cdot \nabla \mathbf{u} = 0$  with the constant convection speed

## 5.2. TEMPORAL ORDER ESTIMATION, UNSTEADY FLOW

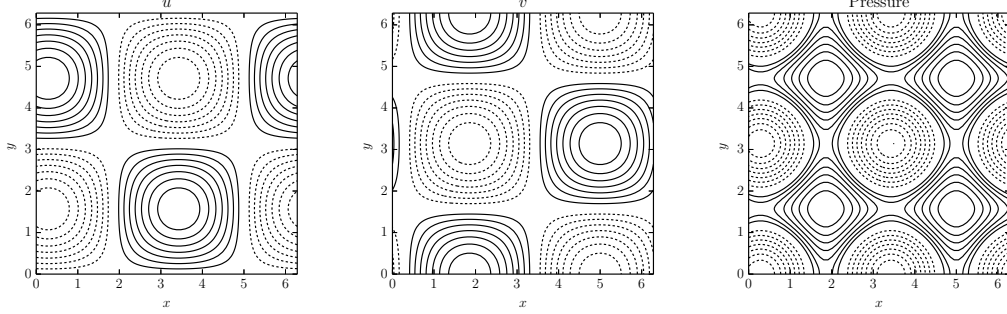


Figure 5.1: The exact unsteady flow fields (5.6a)-(5.6c) at time  $t = 1$ . Solid lines represents positive- and dashed lines negative- values.

$\mathbf{c} = (1, 0)^\top$ . The fact that the solution exactly satisfies the convective condition implies that no forcing ( $\mathbf{f}_b$ ) has to be added to CBC1 and CBC2 in order to obtain an expression of the solution at the outflow boundary. The flow fields at time  $t = 1$  can be considered in Figure 5.1. The flow moves in the positive  $x$  direction.

The domain was selected such that the solution undergoes a full period in the  $y$  direction and periodic boundary conditions can therefore be used. Hence, the domain  $[0, 2\pi] \times [0, 2\pi]$  was chosen. For the time advancement errors to dominate the spatial counterparts the domain was discretized into a sufficiently dense grid;  $8 \times 8$  spectral elements of order  $N = 16$  was chosen. At the left boundary we used the given exact solution (5.6a)-(5.6b) as a Dirichlet boundary condition. At the upper and lower boundaries, periodic boundary conditions were set, and at the right boundary we imposed one of the convective boundary conditions.

To observe the ratio of the temporal discretization errors, the  $x$ -velocity solution was systematically computed at  $t = 1$  for varying time step-sizes resulting in a sequence of numerical solutions

$$\{\tilde{u}_{h/2^i}\}_{i=0}^n, \quad n \in \mathbb{N}. \quad (5.7)$$

Here, the subscript represents a constant temporal step-size  $h/2^i = \Delta t$ . The largest step-size was chosen as  $h = 0.004$ , and four different solutions were computed ( $n = 3$ ).

The temporal order can be estimated from a log-log diagram. Let us recall the procedure: Since the error is a finite number, it follows that; if the temporal order is  $\mathcal{N}$ , there exist a constant  $C \in \mathbb{R}$  independent of  $h$  such that

$$|\tilde{u}_{\Delta t} - u| \leq C\Delta t^{\mathcal{N}}. \quad (5.8)$$

It is also possible to write

$$\tilde{u}_{\Delta t} - u = C\Delta t^{\mathcal{N}} + \mathcal{O}(\Delta t^{\mathcal{N}+1}). \quad (5.9)$$

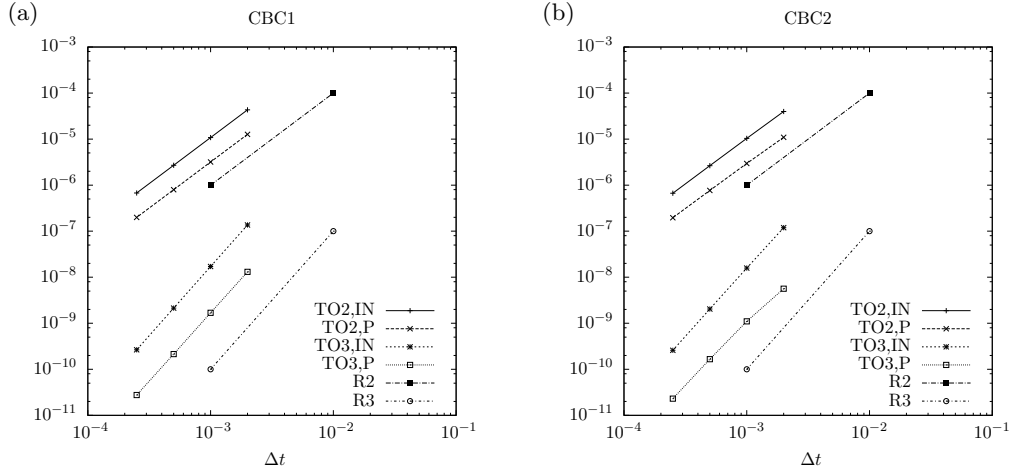


Figure 5.2: Temporal order estimation. TO2 - TORDER=2, TO3 - TORDER=3, IN -  $\|\cdot\|_\infty$ , P - error in the point  $(x = x_O, y = 3.92)$ , R2 - Reference solution of order 2, R3 - Reference solution of order 3.

Then using the logarithm laws we get

$$\log |\tilde{u}_{\Delta t} - u| = \log |C| + \mathcal{N} \log(\Delta t) + \mathcal{O}(\Delta t). \quad (5.10)$$

from which it is easy to determine  $\mathcal{N}$ . Since (5.10) is a linear function of the form  $y = ax + b$  that maps the step-size  $\Delta t$  to the error  $|\tilde{u}_{\Delta t} - u|$ , the order  $\mathcal{N}$  is easily determined as the slope of the curve  $\mathcal{N} \log(\Delta t)$  if (5.10) is plotted in a log-log plot diagram assuming that the order is the same for all  $\Delta t$ . Consider the log-log plot of the solutions corresponding to CBC1 and CBC2 in Figure 5.2a-b.

In Figure 5.2a we observe that CBC1 has temporal orders that are very close to the expected ( $k = 2, 3$ ), and that the slopes of the error curves are close to constant. In Figure 5.2b we see the order estimation of the CBC2 condition. Here the temporal orders are close to 2 and 3 for small  $\Delta t$ , but smaller if  $\Delta t$  is increased. This may also be expected since the implementation of CBC2 assumes that the solution has constant shape, which is a better approximation for small  $\Delta t$ .

Numerical values for the temporal orders can be computed from the error ratios of the sequence of numerical solutions (5.7). An approximation of the order  $p$  follows from

$$\left| \frac{\tilde{u}_{\Delta t} - u}{\tilde{u}_{\Delta t/2} - u} \right| = \frac{C\Delta t^{\mathcal{N}} + \mathcal{O}(\Delta t^{\mathcal{N}+1})}{C(\Delta t/2)^{\mathcal{N}} + \mathcal{O}((\Delta t/2)^{\mathcal{N}+1})} = 2^{\mathcal{N}} + \mathcal{O}(\Delta t) \quad (5.11)$$

Thus,

$$\log_2 \left| \frac{\tilde{u}_{\Delta t} - u}{\tilde{u}_{\Delta t/2} - u} \right| = \mathcal{N} + \mathcal{O}(\Delta t). \quad (5.12)$$

### 5.3. SPATIAL CONVERGENCE, KOVASZNAVY FLOW

The values computed this way can be considered in Table 5.1. Since we have computed the solution using four different  $\Delta t$ , the min and max ratios (5.12) were considered. We can see that the orders of CBC2 are pretty much below the correct orders in the outflow boundary point. Indeed, the minimal order was closer to 2 (2.36) than 3 in that case. The orders for CBC1 however, are very close to the correct orders.

	TO2, $e(x_O, 3.92)$	TO2, $\ e\ _\infty$	TO3, $e(x_O, 3.92)$	TO3, $\ e\ _\infty$
CBC1 min	2.00	2.00	3.00	3.00
CBC1 max	2.00	2.00	3.00	3.00
CBC2 min	1.89	1.94	2.36	2.91
CBC2 max	1.97	1.98	2.85	2.98

Table 5.1: Time order computation. TO2 : Time order = 2; TO3 Time order = 3;  $\|e\|_\infty$  : max norm of the error;  $e(x_O, 3.92)$  : error in the outflow boundary point  $(x_O, 3.92)$ .

### 5.3 Spatial convergence, Kovasznay flow

To estimate numerical convergence rates in space of the new implementations and analyze the accuracy of outflow conditions in a steady flow situation we consider Kovasznay flow [21], which is a simple exact steady state solution to the incompressible Navier–Stokes equations.

The velocity and pressure fields of Kovasznay flow are given by

$$u = 1 - e^{\lambda x} \cos(2\pi y), \quad (5.13a)$$

$$v = \frac{\lambda}{2\pi} e^{\lambda x} \sin(2\pi y), \quad (5.13b)$$

$$p = \frac{1}{2} \left( 1 - e^{2\lambda x} \right), \quad (5.13c)$$

where  $\lambda$  is a parameter that contains the Reynolds number contribution to the flow:

$$\lambda = \frac{Re}{2} - \sqrt{\frac{Re^2}{4} + 4\pi^2}. \quad (5.14)$$

Physically, the Kovasznay flow is the low-speed flow of a viscous fluid past an periodic array of cylinders. Consider streamlines and contour plots of the flow fields (5.13a)-(5.13c) over the domain  $[-0.5, 1] \times [-0.5, 1.5]$  in Figure 5.3. The Reynolds number was chosen as  $Re = 40$ .

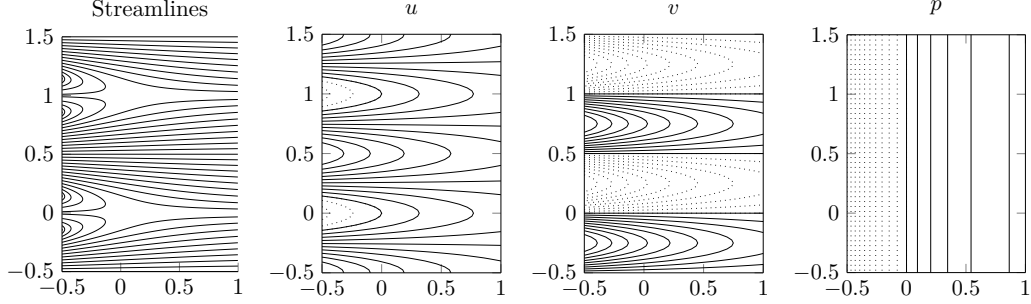


Figure 5.3: Kovaszny flow solutions. Solid lines represent positive and dotted lines negative values. The axes are:  $x \rightarrow$ ,  $y \uparrow$ . The flow moves in the positive  $x$  direction. The min and max values of the respective fields are given by:  $\min(u) = -0.6191$ ,  $\max(u) = 2.6191$ ,  $\min(v) = -0.2483$ ,  $\max(v) = 0.2483$ ,  $\min(p) = -0.8107$ , and  $\max(p) = 0.4272$ .

### 5.3.1 Convergence rates

In a first test we analyze if the convective boundary conditions preserve the exponential convergence of the spectral element discretization. To compute the spatial convergence rate the spatial errors must be dominating over all other possible errors in the solution. To test the outflow boundary conditions, that do not satisfy the Navier–Stokes equations exactly, we need to add necessary forcing ( $\mathbf{f}_b$ ) to the outflow conditions. Since we know the exact solution beforehand, we simply compute the remainder as (5.13a)-(5.13b) are substituted into the expressions of the boundary conditions.

We consider the steady state solutions that were computed from the initial condition  $(1, 0)^\top$  on grids with a fixed number of spectral elements ( $4 \times 4$ ) with monotonically varying element orders in the interval 3 to 16. For the computational domain, we used  $[-0.5, 1] \times [-0.5, 1.5]$ , similar to the solutions depicted in Figure 5.3. At the inflow the exact solutions (5.13a)-(5.13b) were imposed as Dirichlet conditions. At the upper and lower boundaries periodic boundary conditions was set, and at the outflow one of the convective boundary conditions was imposed.

The forcing that is required for CBC1 (with  $\mathbf{c} = (1, 0)^\top$ ) to satisfy the Kovaszny flow is computed as:

$$f_{x,\text{CBC1}} = (u_t + 1u_x)|_{x=x_O} = -\lambda e^{\lambda x_O} \cos(2\pi y), \quad (5.15a)$$

$$f_{y,\text{CBC1}} = (v_t + 1v_x)|_{x=x_O} = \frac{\lambda^2}{2\pi} e^{\lambda x_O} \sin(2\pi y). \quad (5.15b)$$

And in the case of CBC2, (with  $\mathbf{c} = (1, 0)^\top$ ) similarly

$$f_{x,\text{CBC2}} = u(x_O, y) - u(x_O - 1\Delta t, y) = e^{\lambda x} \cos(2\pi y) (e^{-\lambda 1\Delta t} - 1), \quad (5.16a)$$

$$f_{y,\text{CBC2}} = v(x_O, y) - v(x_O - 1\Delta t, y) = -e^{\lambda x} \sin(2\pi y) (e^{-\lambda 1\Delta t} - 1). \quad (5.16b)$$



### 5.3. SPATIAL CONVERGENCE, KOVASZNAV FLOW

Since this test is about spatial convergence, sufficiently small time-step sizes were chosen such that the temporal discretization errors did not affect the spatial counterparts;  $\Delta t = 0.001$  was tested to be small enough for this purpose. After a steady state was reached the errors between the numerical solution and the exact solution (5.13a)-(5.13b) in both the  $L^2$  and  $L^\infty$  norms were computed. In Figure 5.4 the norms are plotted against element orders for the CBC1 and CBC2 cases with the forcing functions (5.15)-(5.16) added. For comparison we also computed the errors using exact boundary conditions (denoted as E in the Figures). From the plots we can observe that exponential convergence is obtained down to the given tolerance levels ( $\approx 10^{-12}$ ) for all outflow boundary conditions considered; in both the  $\mathbb{P}_N \times \mathbb{P}_N$  and the  $\mathbb{P}_N \times \mathbb{P}_{N-2}$  formulations.

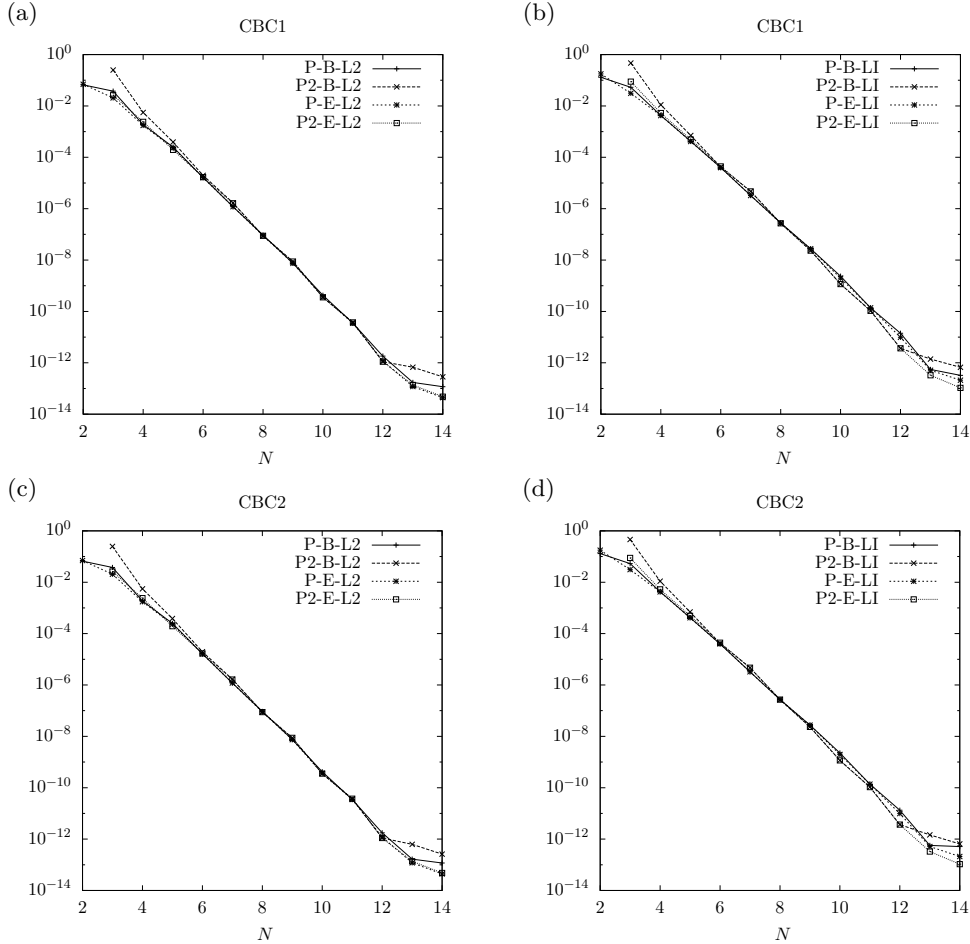


Figure 5.4: Spatial convergence in  $L^2$  and  $L^\infty$  norms. P:  $\mathbb{P}_N \times \mathbb{P}_N$ , P2:  $\mathbb{P}_N \times \mathbb{P}_{N-2}$ , B : BDF3/EXT3, E : Exact, L2:  $L^2$ -norm, LI:  $L^\infty$ -norm.

### 5.3.2 Accuracy

As  $x \rightarrow \infty$  it directly follows from (5.13a)-(5.13c) that  $u \rightarrow 1$ ,  $v \rightarrow 0$  and  $p \rightarrow 0$ . This means that the shape of the streamlines converge to straight lines, and that the base flow at the given  $x$  converge to  $(1, 0)$ . Hence for the convective boundary conditions the convecting speed  $\mathbf{c} = (1, 0)^\top$  will be correct for large  $x$ , but not as  $x \rightarrow -0.5$ .

The accuracy of the boundary conditions depends on how well they approximates the Navier–Stokes equations for the specific problem; the errors are computed equally to the forcing function expressions derived in (5.15)-(5.16) if the convection speed is specified. In a complex flow where the exact solution is not known, no specific forcing can generally be derived for the outflow boundary conditions. For the convective boundary conditions certainly the choice of  $\mathbf{c}$  becomes the only user-dependent source of errors.

For comparison we now also consider the error NBC1 gives us. The forcing functions (errors) of NBC1 are computed as

$$f_{x,\text{NBC1}} = \left( -p + \frac{1}{Re} u_x \right) \Big|_{x=x_O} = -\frac{1}{2}(1 - e^{\lambda x_O}) - \frac{1}{Re} \lambda e^{\lambda x_O} \cos(2\pi y), \quad (5.17a)$$

$$f_{y,\text{NBC1}} = \frac{1}{Re} v_x \Big|_{x=x_O} = \frac{\lambda^2}{2\pi Re} e^{\lambda x_O} \sin(2\pi y). \quad (5.17b)$$

Consider in Figure 5.5 the forcing functions (errors) (5.15)-(5.17) of the boundary conditions plotted against the location of the outflow boundary for all three outflow boundary conditions (NBC1, CNC1 and CBC2). NBC2 does not make sense here since the flux over the outflow boundary is positive for both the  $x$  and  $y$  velocity fields. It is noticeable both from the equations and the plots that, as  $x_O \rightarrow \infty$  the errors tend to zero exponentially for all three outflow conditions. CBC1 with unit convection speed gives largest- and CBC2 with unit convection speed the smallest-errors of the three. However, both the convective conditions do converge to homogeneous Neumann conditions as the flow approaches a steady state (since  $\mathbf{u}_t \rightarrow \mathbf{0}$ ); see Figure 5.5.

The errors of the convective conditions are easy to deduce. If we disregard errors related to pressure, the errors can directly be described by the diffusion term  $\frac{1}{Re} \nabla^2 \mathbf{u}$  of the incompressible Navier–Stokes, which do converge to zero if  $Re \rightarrow \infty$  (which imply that the convective condition equals the incompressible Navier–Stokes equation as  $\mathbf{c} = \mathbf{u}$ ,  $\nabla p = 0$  and  $\mathbf{u}_t = \mathbf{0}$ ), but is a problem for low Reynolds numbers.

It is of course possible to add the diffusion term to the convective boundary condition. Then, the convective condition becomes the advection-diffusion equation. This condition may work better for the Kovasznay flow case, but one cannot guarantee that it will be a better condition generally; this requires more analysis. But for high Reynolds numbers the diffusion term is of no interest anyway since it is small; we have restricted ourselves to the convective conditions and highly convective problems beforehand. Outflow boundary conditions for laminar flow may be approached differently, which is worthy to mention. For the interested reader, tests

### 5.3. SPATIAL CONVERGENCE, KOVASZNAY FLOW

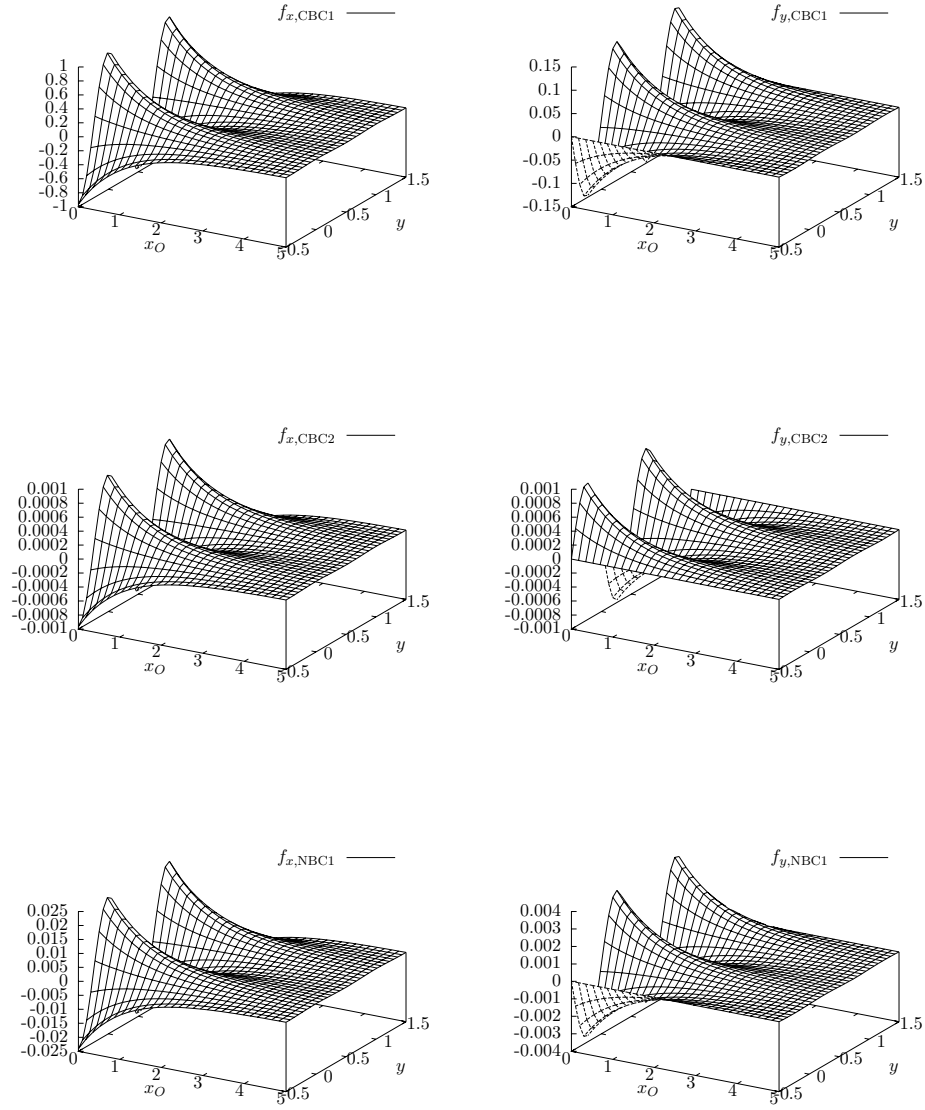


Figure 5.5: The forcing functions (5.15)-(5.17) plotted for varying outflow locations.

with the advection-diffusion treated outflow boundary condition can be considered in the paper by Miyauchi et al. [27].

Another interesting aspect that relates to the choice of convection speed for the convective boundary conditions. Since Kovasznay flow is a steady flow, the solution is equivalent to the base flow. Hence a correct estimation of the convection speed will be the same as the solution (5.13a)-(5.13b). But, if the flow has non-zero diffusion (which it certainly has for  $Re = 40$ ), then the solution will not be accurate. The flow field for CBC1 can be considered in Figure 5.4d. As we can see, there is a significant amount of errors in the contour plots close to the outflow boundary especially in the  $v$  field. A “correct” convection speed does not give appropriate results here.

In Figure 5.6 we note that the natural condition shows better results for this test case. The streamlines are consistent with the streamlines of the exact solution (compare to Figure 5.3). There are some direct possible reasons to this. The natural condition does not depend on time; it does not converge to a homogeneous Neumann condition in a steady state because of the pressure term; and the spatial derivative has the  $1/Re$  factor in front, which makes it smaller than the counterpart of the convective condition. The accuracy of the pressure field is also better in the NBC1 results compared to the results for the convective conditions.

As a final remark; we might point out that the purpose of the convective condition is not to use it in steady flows and in particular not for low Reynolds numbers. But, one may bear in mind, that if the convective condition is used for lower Reynolds number flows, even though a correctly estimated convection speed is available, the boundary condition may not produce streamlines that are consistent with the true solution.

### 5.3. SPATIAL CONVERGENCE, KOVASZNAY FLOW

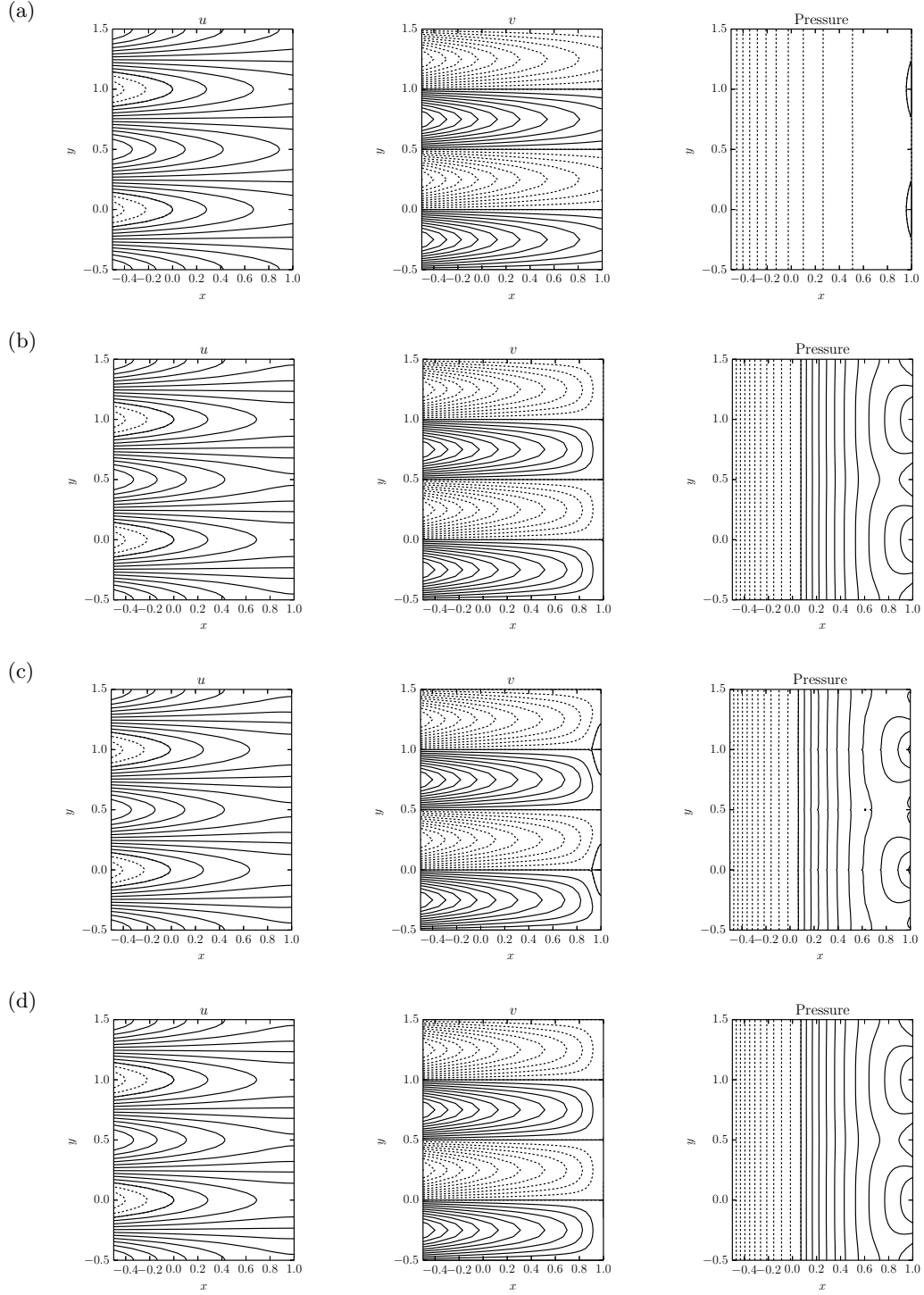


Figure 5.6: (a) NBC1, (b) CBC1 with  $\mathbf{c} = (1, 0)^\top$  (c) CBC1 with  $\mathbf{c} = \mathbf{u}_{\text{exact}}$ , (d) CBC2 with  $\mathbf{c} = (1, 0)^\top$ .

## 5.4 Convecting vortex

In this section we consider the problem of a single vortex that is superimposed onto a two dimensional flow. This is a typical convective test problem with a controllable disturbance. The flow domain is chosen as  $[0, L_1] \times [0, L_2]$ , and the base flow an unit velocity in the  $x$  direction (a so called plug flow). At the inflow  $\Gamma_I$  a Dirichlet boundary condition is imposed equal to the base flow  $\mathbf{u}|_{\Gamma_I} = (1, 0)^\top$ ; and on the upper and lower boundaries periodic boundary conditions were set; see Figure 5.7a. The vorticity distribution of the vortex at the initial state ( $t = 0$ ) was chosen as

$$\omega(r) = \frac{A}{R^2} e^{-\frac{r^2}{2R^2}} \left( \frac{r^2}{R^2} - 2 \right), \quad (5.18)$$

where  $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ ,  $A$  is the vortex strength and  $R$  is the vortex radius, see Figure 5.7b. This configuration is an useful test case since the total circulation of the vortex is zero and therefore no corrections of the initial boundary conditions are needed [33]. In our standard configuration the initial placement of the vortex was chosen in the middle of the domain  $x_0 = L_1/2$ ,  $y_0 = L_2/2$ . Due to the fact that the flow is purely laminar, the viscosity can freely be chosen without producing any disturbances (like vortices) to the flow. The only thing that the viscosity will influence is the amount of dissipation of the vortex. According to this, the Reynolds number was picked large enough to restrict the vortex to dissipate significantly; the size of the vortex was assumed to be invariant with time. A sufficiently large Reynolds number was tested to be  $Re = 1000$ . For higher Reynolds numbers stability problems occurred if the natural condition (NBC1) was used. A more detailed stability analysis is performed later on in this section. We used  $u_{ex}(\mathbf{x}) = u(\mathbf{x}_0 + \mathbf{c}t)$  as the exact solution to compute the errors in the domain. Small errors due to dissipation of the vortex (since  $Re = 1000$ ) were ignored in the analysis. Those errors are of less importance here since the exact solution after the

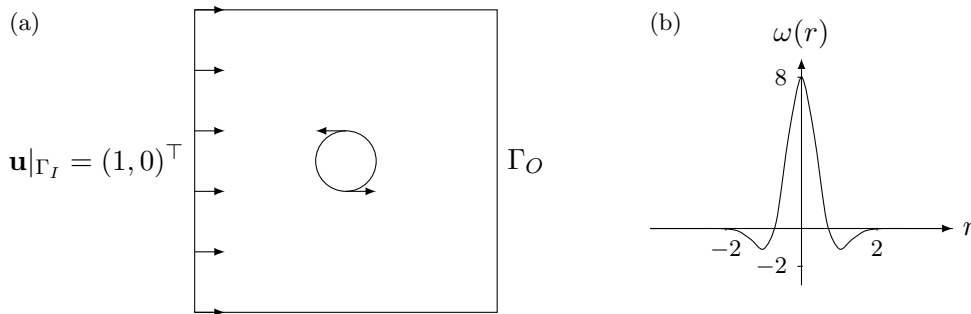


Figure 5.7: (a) Configuration of the single vortex problem. The vortex has the initial position  $x_0 = L_1/2$ ,  $y_0 = L_2/2$ , and moves with the base flow  $(1, 0)$ . (b) The vorticity distribution of the vortex. Equation (5.18) with  $A = R = 1$ .

#### 5.4. CONVECTING VORTEX

vortex has left domain will be correct anyway; i.e. equal to the base flow  $(1, 0)^\top$ . The errors of interest are measured either at states when the vortex travels over the boundary, or after the vortex has left the domain. In the latter case all errors (larger than machine precision) that is still left in the domain corresponds to numerical errors.

As a first test case, the NBC1 was imposed at the outflow boundary. The flow fields of the times  $t = 4, 5$ , and  $6$  are depicted in Figure 5.8. The first row shows the  $x$  velocity field scaled such that the vortex has equally large positive and negative parts. At time  $t = 5$  when the vortex has penetrated the outflow boundary, the shape of the vortex is disturbed. In the  $x$  velocity field we see that the contour lines are rotated clockwise, and in the  $y$  velocity field the contour lines are rotated counter-clockwise. One may also note that the minimal  $x$  velocity is lower when the vortex cross the boundary than it is when the vortex is inside. Another problem is observed in the vorticity plot at  $t = 6$ , where the part of the vortex with positive  $x$ -velocity ( $y < 5$ ) has close to circular vorticity contours, but the upper half with negative velocity has serious problems. This phenomena can also be considered in Figure 5.12, where the vorticity in the  $y$  direction are plotted at the boundary.

Figure 5.9 shows states when the vortex has left the domain. Now, everything on top of the base flow corresponds to errors. At the same time as the vortex touches the outflow boundary, numerical reflections that travel upstream are generated. Those are small, travel fast, and when they reach the inflow boundary a new physical reflection that now travels with the base flow occur [33, 43]. The physical reflections that move with the base flow can be considered in the (a) figures. In the same plots we can also see that the NBC1 case does not maintain the base flow on the side of the reflections close to the outflow boundary. There is still a lot of energy left that does not advect out, but rather dissipates with time. When a reflection reaches the outflow, new numerical reflections appear and the same procedure happens again. However, now the reflections are smaller. Indeed the amplitude of the reflections seems to be proportional to the amplitude of the reflected flow (see §5.4.3). In Figure 5.9b, we can also note that there are some noticeable reflections that correspond to the vortex wake.

The Figures 5.10-5.11 show the same single vortex problem computed with the CBC1 condition. Now, the vortex leaves the domain without any shape variances. The reflections in the velocity fields are easier to localize than in the NBC1 case since no significant amount of energy remains on the side of the reflection close to the outflow boundary now. But, reflections that correspond to the vortex wake are still present for later times (see Figure 5.11b).

## CHAPTER 5. NUMERICAL RESULTS

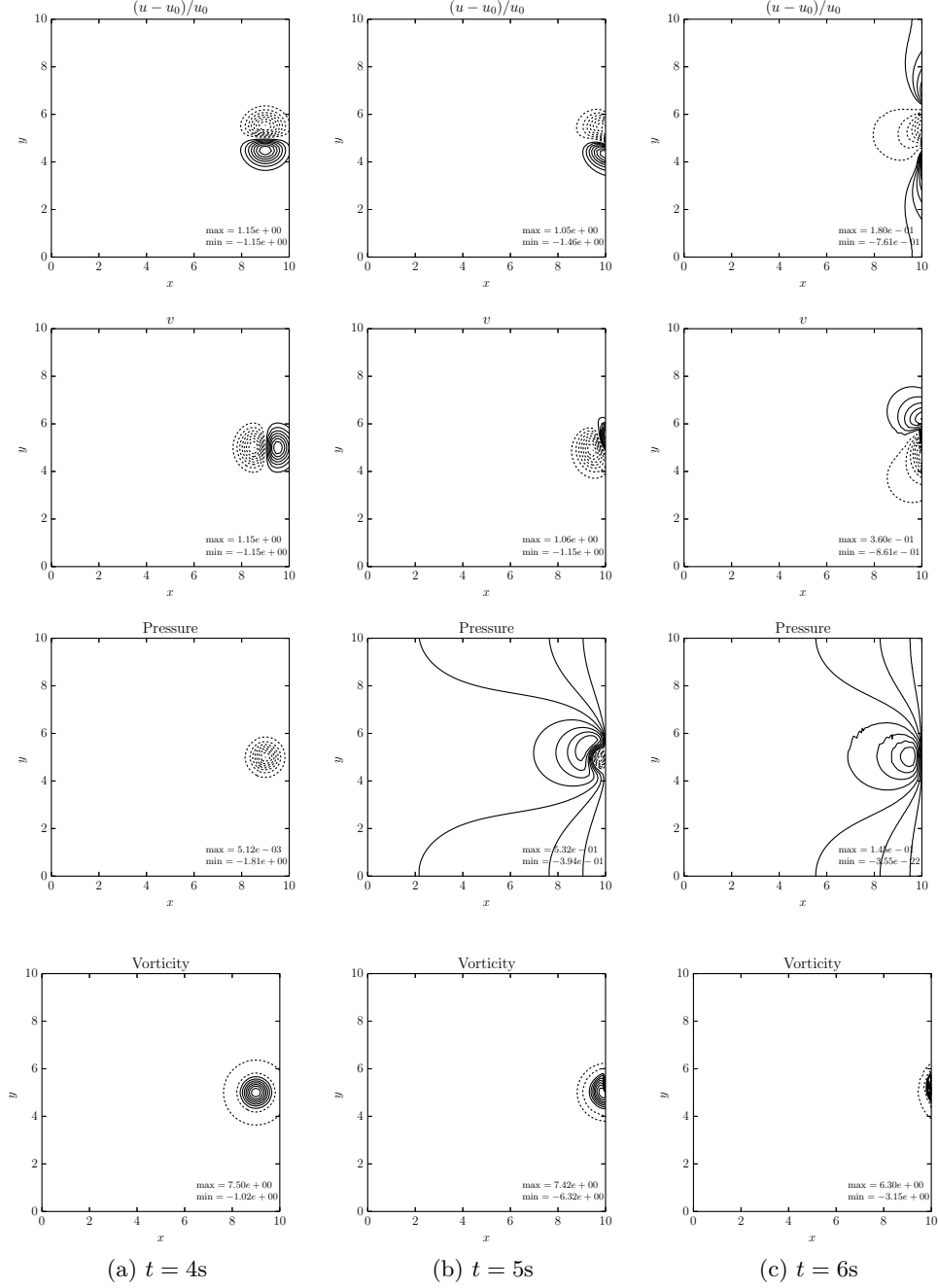


Figure 5.8: NBC1. Snapshots of the flow fields at the times  $t = 4, 5,$  and  $6$ s. Dashed lines show the negative part of the solution and solid lines the positive part.



#### 5.4. CONVECTING VORTEX

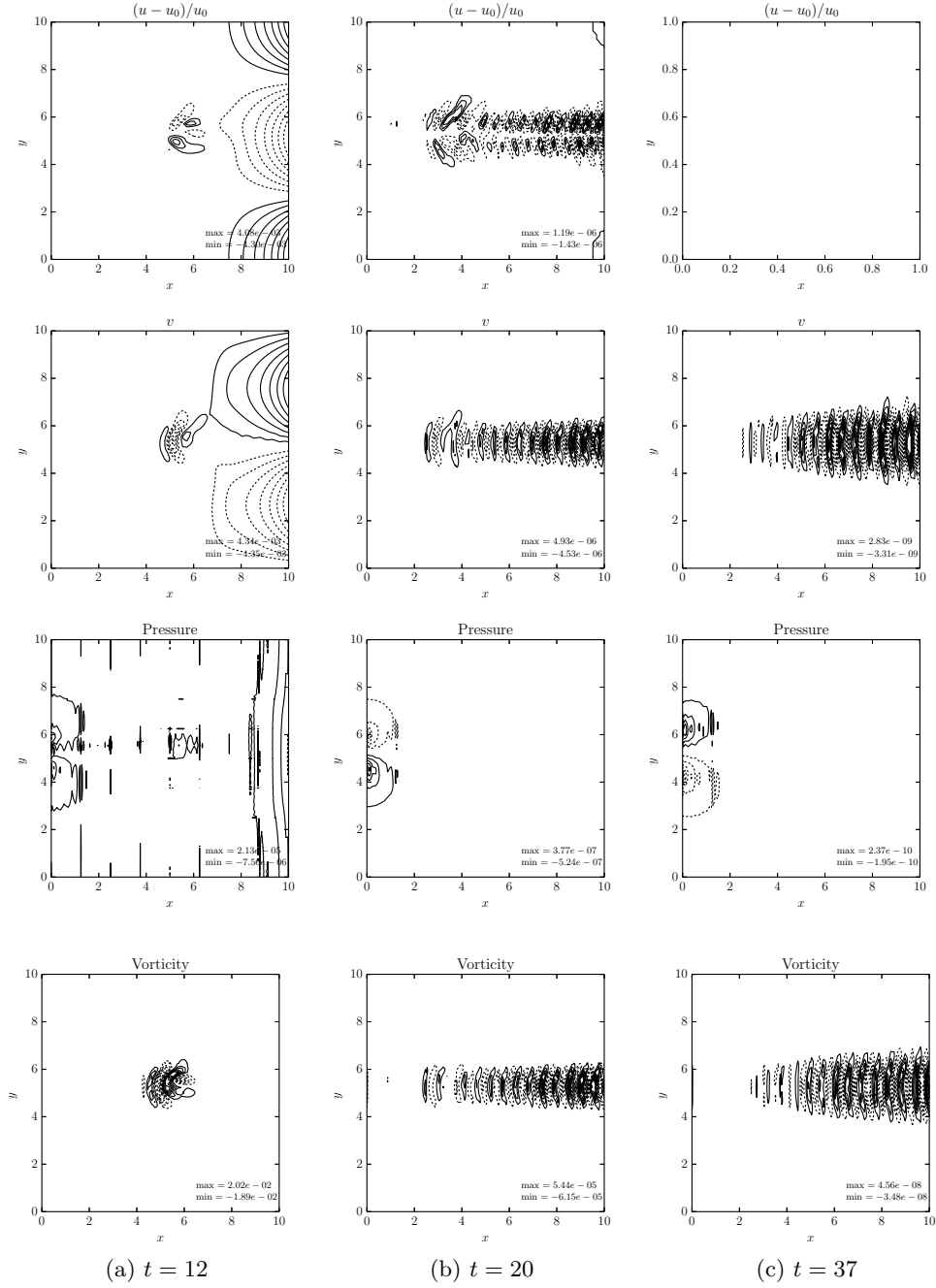


Figure 5.9: NBC1. Snapshots of the flow fields at the times  $t = 12, 20$ , and  $37$ s. Dashed lines show the negative part of the solution and solid lines the positive part.

## CHAPTER 5. NUMERICAL RESULTS

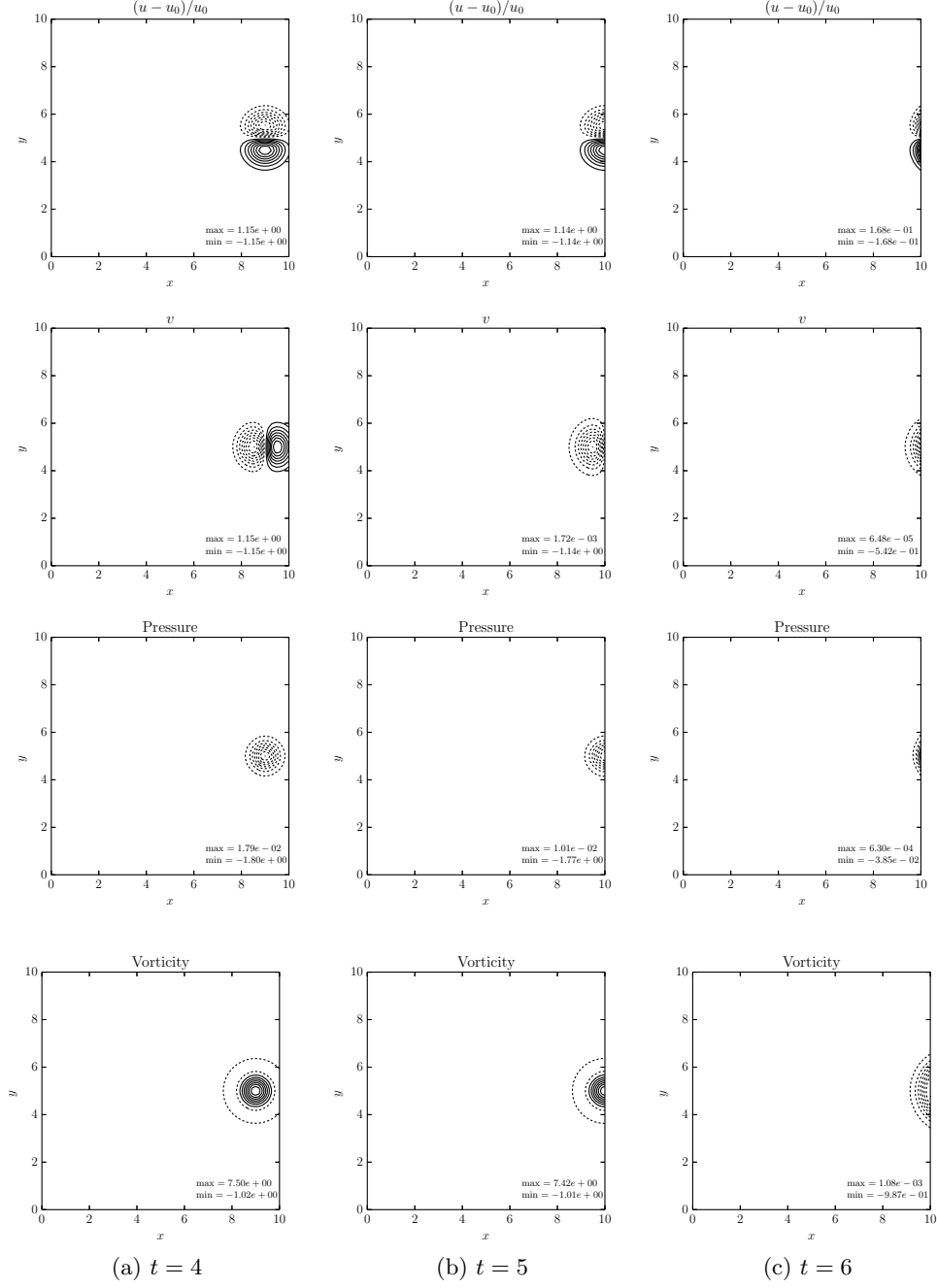


Figure 5.10: CBC1. Snapshots of the flow fields at the times  $t = 4, 5,$  and  $6$ s. Dashed lines show the negative part of the solution and solid lines the positive part.

#### 5.4. CONVECTING VORTEX

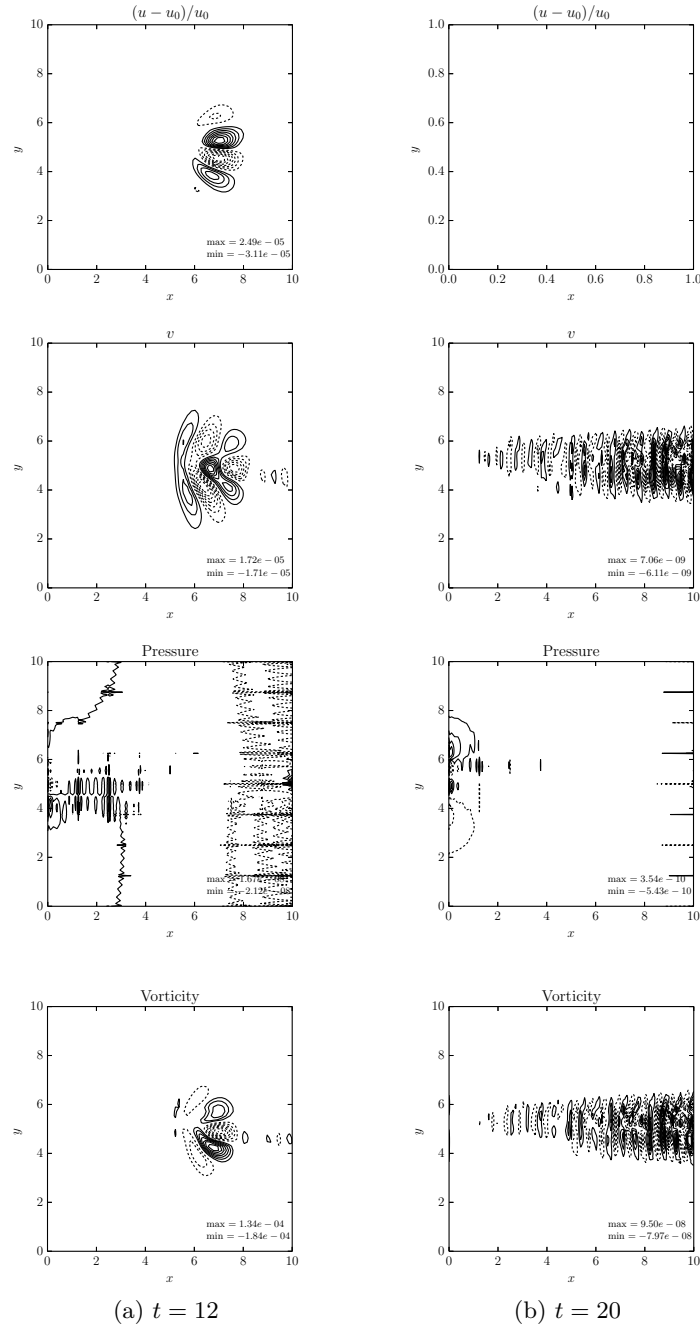


Figure 5.11: CBC1. Snapshots of the flow fields at the times  $t = 12$  and 20s. Dashed lines show the negative part of the solution and solid lines the positive part.

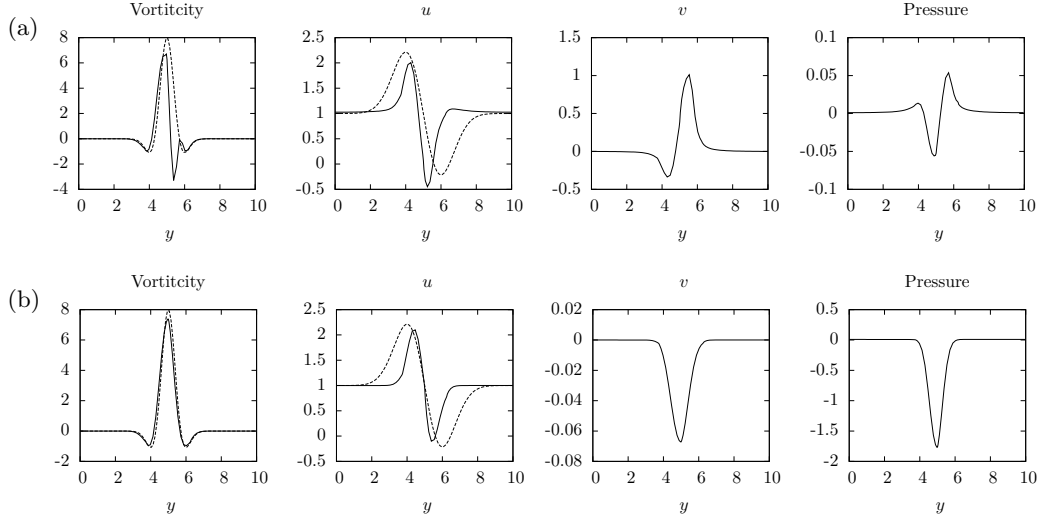


Figure 5.12: Solutions at the outflow at time  $t = 5$ . (---) Exact solution of the vortex given from (5.18) with  $A = R = 1$ . (—) Numerical solution. (a) NBC1, (b) CBC1.

The error over a region is preferably studied by considering the  $L^2$  norm. We now present an error analysis that makes it easy to measure and compare the size of reflections in the domain. Figure 5.13 shows the  $L^2$  error vs time of the  $u$  field for both CBC1 and NBC1. Four different element orders were considered:  $N = 8, 12, 16, 20$ . We can observe the times the vortex reaches the boundary, and when reflections appear and leave the domain. In the NBC1 plot (see Figure 5.13a), we can notice a peak in the error curve (after  $t \approx 5$ s) in the time interval when the vortex travels across the boundary. This peak can be explained as the error of the asymmetric behavior of the vortex that we saw in Figure 5.8. Another significant observation is the steps in the error curves. These typically correspond to reflections that travel over the domain. Consider for instance the  $N = 12$  curve and compare to the contour plots. The reflection that we clearly identify at  $t = 12$  (Figure 5.11) corresponds to the first step in the error time plot (Figure 5.13b). An illustrative explanation to these plots can be considered in Figure 5.15.

In Figure 5.14a the NBC1 and NBC2 solutions are compared against each other for  $N = 12$ . We can note a small difference between the two. This small difference corresponds to the fact that the NBC2 does have smaller errors in the parts of the boundary where it is negative flux present.

In Figure 5.14b similarly the CBC1 and CBC2 solutions are compared against each other for  $N = 12$ . Here the results almost coincide.

#### 5.4. CONVECTING VORTEX

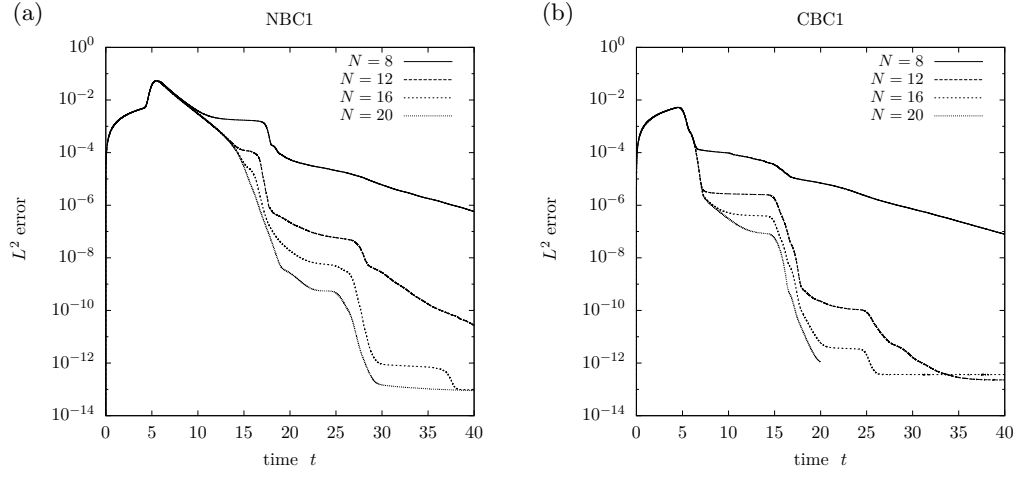


Figure 5.13: Error vs time plots of the  $x$  velocity errors in the  $L^2$  norm.

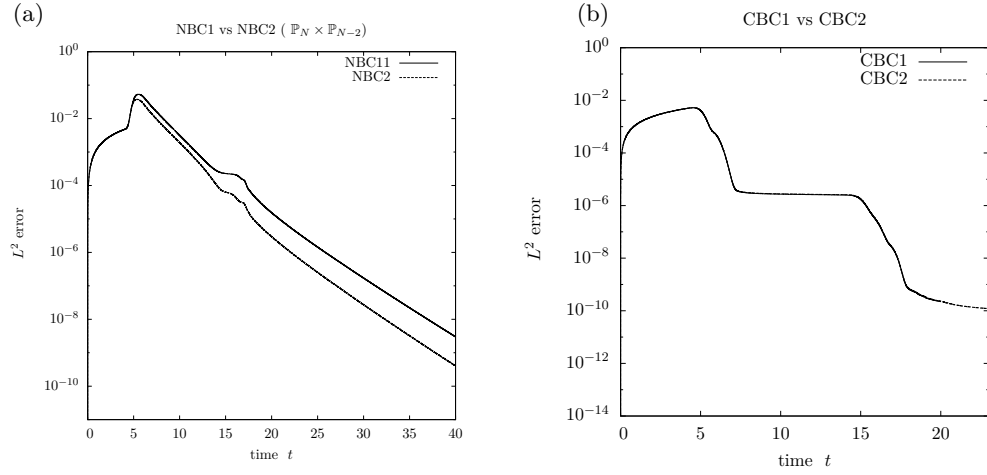
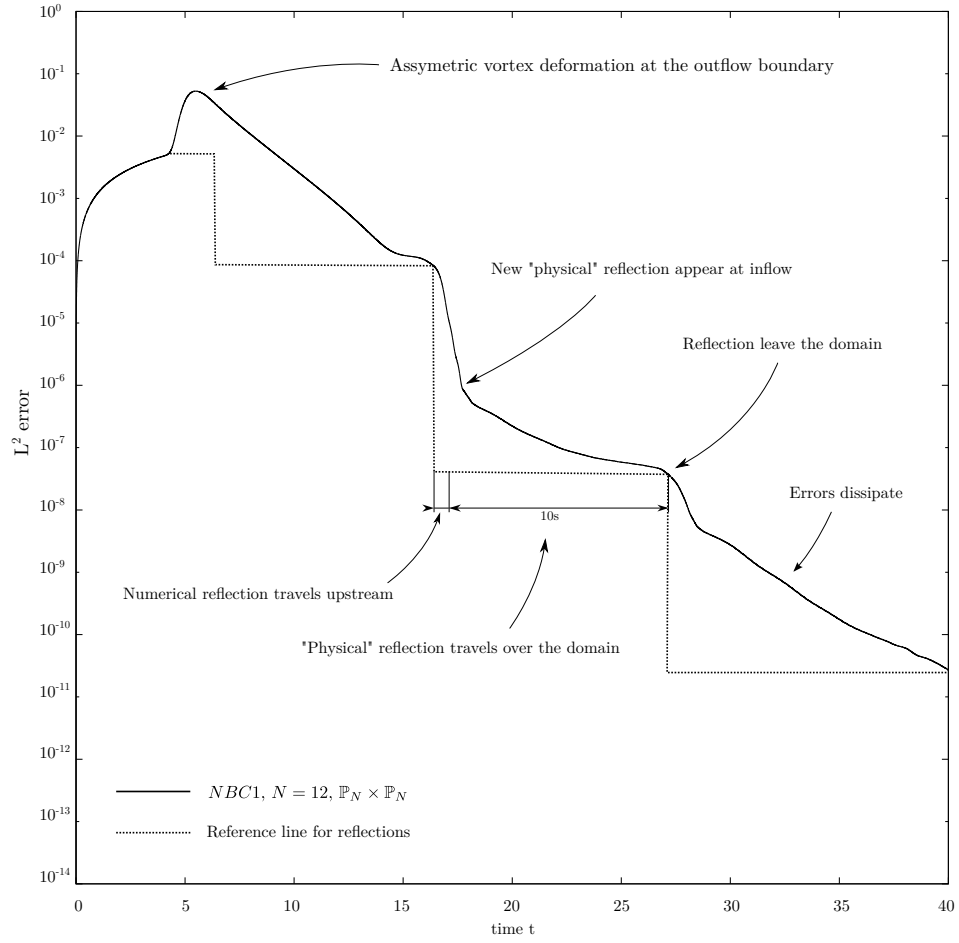


Figure 5.14: Comparison of: (a) the NBC1 and NBC2 results, (b) the CBC1 and CBC2 results.  $N = 12$


 Figure 5.15: Explanation.  $L^2$  error vs time plots.  $NBC1, \mathbb{P}_N \times \mathbb{P}_N$ .

## 5.4. CONVECTING VORTEX

### 5.4.1 $\mathbb{P}_N \times \mathbb{P}_N$ vs. $\mathbb{P}_N \times \mathbb{P}_{N-2}$

We have noticed some differences in the results obtained using  $\mathbb{P}_N \times \mathbb{P}_N$  and  $\mathbb{P}_N \times \mathbb{P}_{N-2}$  formulations. For both the natural boundary conditions (NBC1 and NBC2) and the convective (CBC1 and CBC2) these problems occur. The errors in the velocity fields in the  $\mathbb{P}_N \times \mathbb{P}_{N-2}$  setting are covered by noise (wiggles). The wiggles are located close to the element boundaries. As the element order is increased, the wiggles also decrease in size due to the faster dissipation of the higher resolution. Consider in Figures 5.16(a and d), the error-time plots of both the  $\mathbb{P}_N \times \mathbb{P}_N$  and the  $\mathbb{P}_N \times \mathbb{P}_{N-2}$  results plotted side by side for the NBC1 and CBC1 solutions, and in Figures 5.16(b and d) the  $u$  and  $v$  velocity fields at  $t = 20$ . The wiggles are most evident in the CBC1,  $u$  field (Figure 5.16d).

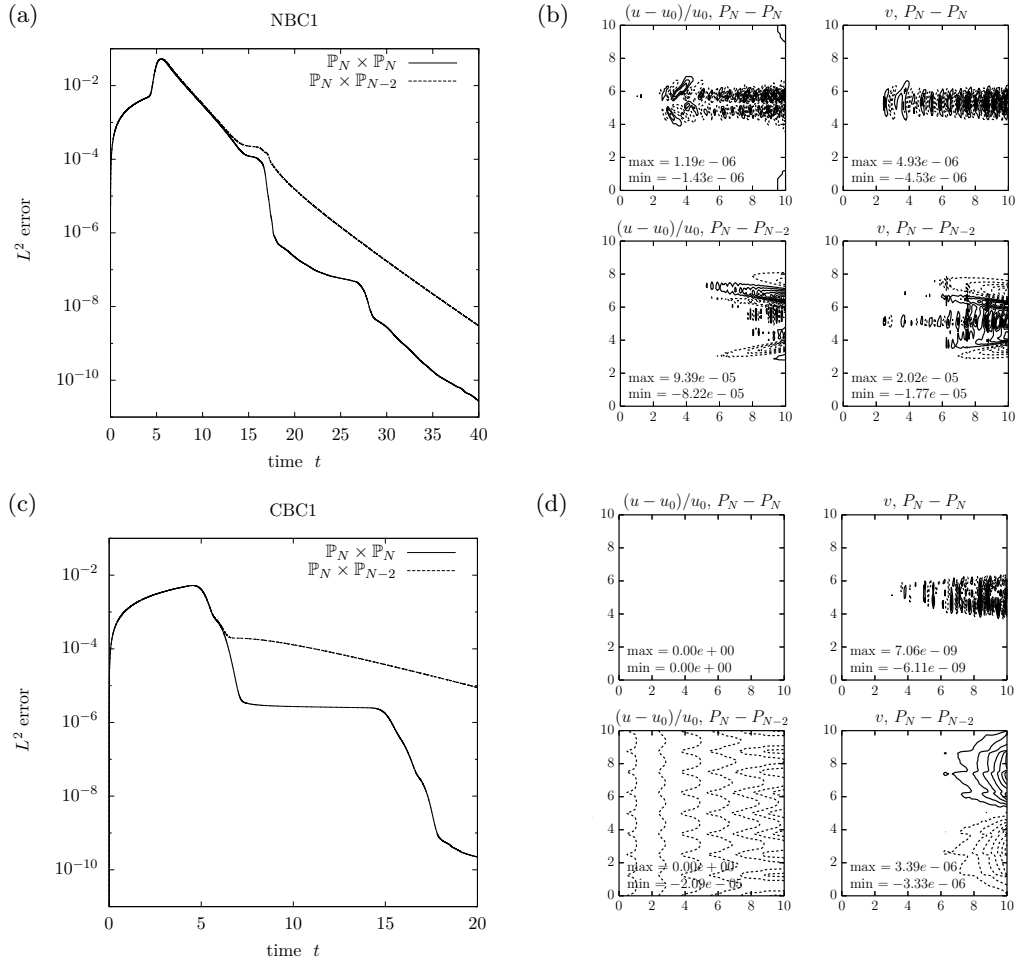


Figure 5.16: Comparison of  $\mathbb{P}_N \times \mathbb{P}_N$  and  $\mathbb{P}_N \times \mathbb{P}_{N-2}$  solutions. (a) CBC1 – error–time plots. (b) CBC1 – velocity fields at  $t = 20$ . (c) NBC1 – error–time plots. (d) NBC1 – velocity fields at  $t = 20$ .

### 5.4.2 Robustness analysis

The convecting vortex problem is a perfect test case to test the susceptibility of disturbances of the convective boundary conditions. In this subsection we perform several tests to examine the robustness, i.e. how disturbances of the system (the convective boundary condition) affect the solution.

#### Convection speed

As has been pointed out in §4, for the convective boundary conditions a perfectly estimated convection speed is seldom available in more complex simulations. An approximation of the base flow must be performed, which certainly has rather bad accuracy. We hereby perform a test where the convection speed is chosen other than the base flow  $(1, 0)^\top$ . Let  $\mathbf{c} = (c_x, 0)^\top$ ; both  $c_x$  values below and above one were tested, with the main focus on large  $c_x$ . All tests were performed in the same domain as before, and with the polynomial order fixed to  $N = 12$ . The results can be considered in Figure 5.17. As we can see, similar to the error-time plots of NBC1 in the previous section (Figure 5.8-5.9), there is now a peak of the error curve as the vortex crosses the boundary. This does not actually mean we have the same kind of errors as for NBC1. Indeed, the vortex still leaves almost symmetrically over the boundary, but the wrong convection speed of the boundary condition changes the magnitude of the vortex. As the condition is not a correct solution to the flow anymore, reflections do occur. Indeed, the size of the reflections seems to converge to a specific magnitude as  $c_x \rightarrow \infty$ .

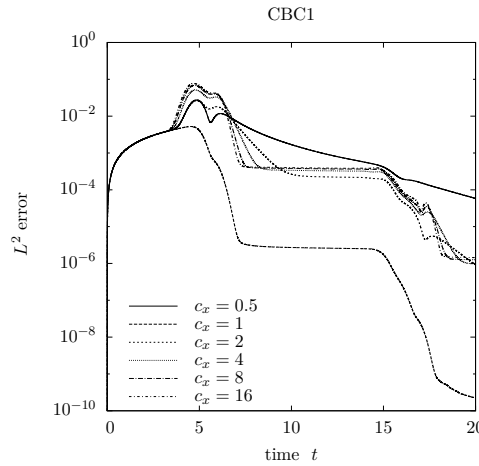


Figure 5.17: Robustness analysis of the convective outflow boundary conditions. Error time plots of different convection speeds  $c_x$ .



#### 5.4. CONVECTING VORTEX

##### Angles of incidence

Another interesting test case regarding robustness is to analyze the magnitude of the errors as disturbances approach the boundary at angles of incidence other than zero. We let the vortex start from several locations in the domain (see Figure 5.18) and change the base flow direction such that the vortex reaches the boundary condition at the same time, with the same speed ( $U_\infty = 1$ ), and at the same location ( $x = x_O, y = L_y/2$ ) for all tests. The setting of the convective boundary condition was fixed to the original one, i.e. the case with angle of incidence is zero. The angles considered were  $0^\circ, 15^\circ, 30^\circ$  and  $45^\circ$ . In Figure 5.19a we show the error-time graphs for the CBC1 condition, and in Figure 5.19b we present for comparison the similar results for the stabilized natural condition (NBC2). We also tested the NBC1 condition, but it displays stability problems for the higher angles ( $30^\circ$  and  $45^\circ$ ). This happens due to the fact that the magnitude of negative flux over the boundary is increased with the higher angles and therefore implies instability by Theorem 3.1 (also see next section). As long as stability is preserved the natural conditions are perfectly robust against different angles of incidence of disturbances as expected (since it is an implicit condition); see Figure 5.19a. This is obviously not the case for the convective boundary condition, which also has been confirmed analytically in §4.2. The convective condition only satisfies traveling characteristics in one given direction, which needs to be determined a priori. As can be seen in Figure 5.19, both errors and reflections increase with an increased incident angle of the vortex. However, the accuracy as the vortex travels across the boundary is not much worse for CBC1 than for NBC1 or NBC2; even for the worst angle considered ( $45^\circ$ ). Tests with the CBC2 condition have also been performed, and the plots look similar to the plots for CBC1.

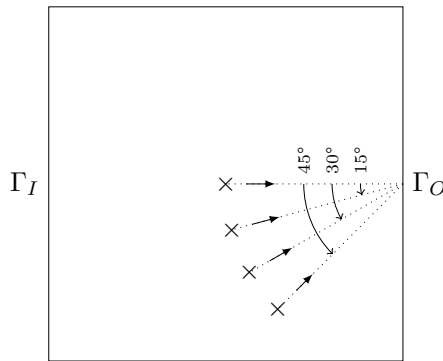


Figure 5.18: Initial locations ( $\times$ ) of the vortex and the angles of incidence of the different cases.

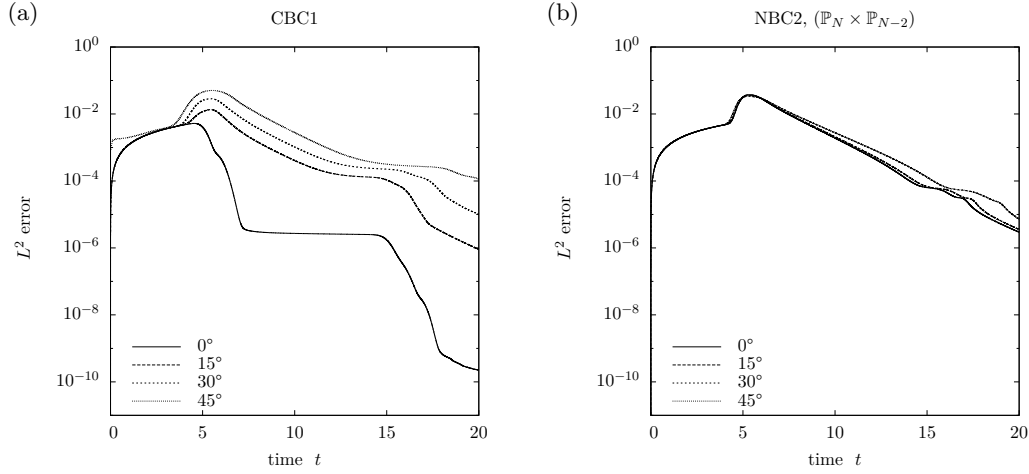


Figure 5.19: Error-time plots corresponding of the  $u$  field for different angles of incidence of the vortex. (a) CBC1, (a) NBC2.

### 5.4.3 Stability and reflections

We have already experienced stability issues for the NBC1 condition in the previous subsection as the shape of the vortex was slightly changed. A simple way to test stability is to regulate the strength of the vortex (parameter  $A$  in (5.18)) and see if the solution remains stable or not as the vortex travels trough the outflow boundary. Indeed, we do not need to change  $A$  much until the solution blows up for the NBC1 boundary condition (see Figure 5.20a); when the strength of the vortex is changed, then both the amount of negative flux over the outflow boundary and the CFL number will be affected. It suffices to change the amplitude ( $A$ ) to approximately 1.05 and the NBC1 condition blows up rapidly as the vortex approaches the boundary. Using the stabilized natural condition NBC2 instead, we obtained a stable result all the way up to  $A \approx 16$ , when simulations systematically were performed by doubling  $A$  from  $A = 0.125$  (0.125, 0.25, 0.5, 1, 2, 4, 8, 16) ; see Figure 5.20b. Both the convective conditions (CBC1, and CBC2) also remain stable up to the same limit; see Figure 5.21, and the plots for CBC2 coincide. The reason for instability here is that the strong vortex implies a large CFL number, and the solution blows up instantly, which has nothing to do with the boundary condition itself.

Considering the same results we can also observe the size of the first reflection that appears at the inflow boundary when numerical spurious errors are reflected into “physical” reflections. We picked the error at  $t = 14$ s for all separate solutions computed with CBC1; then the errors were plotted against the vortex amplitude (see Figure 5.21a). If we assume that the size of the reflections are equal to the magnitude of the steps we have determined in the error-time plots, then we can conclude that the reflections are proportional to the strength of the vortex ( $A$ ), by

#### 5.4. CONVECTING VORTEX

observation. The reflections in the results of the natural boundary conditions are harder to localize, but are also close to proportional to the vortex strength from what one can observe from Figure 5.20b. There are several other aspects one must have in mind due to reflections. Consider for instance the figures in the robustness subsection (§5.4.2) again. In Figure 5.17, we can note that for  $c_x \geq 4$  the size of the

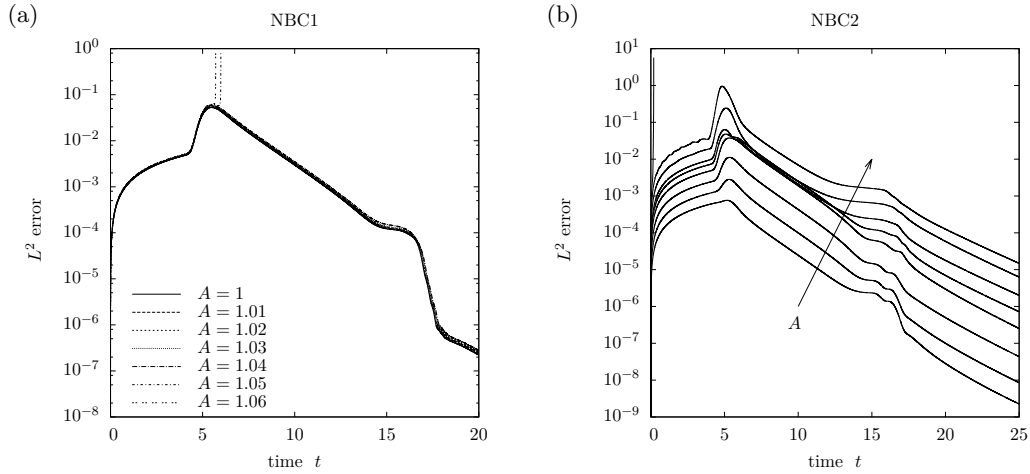


Figure 5.20: Error time plots of the  $u$  solutions computed with different  $A$ . (a) Instability occur for NBC1. (b) Error-time plots corresponding to the vortex strengths:  $A \in \{0.125, 0, 25, 0.5, 1, 2, 4, 8, 16\}$  (where  $A = 16$  is the one that blows up instantly).

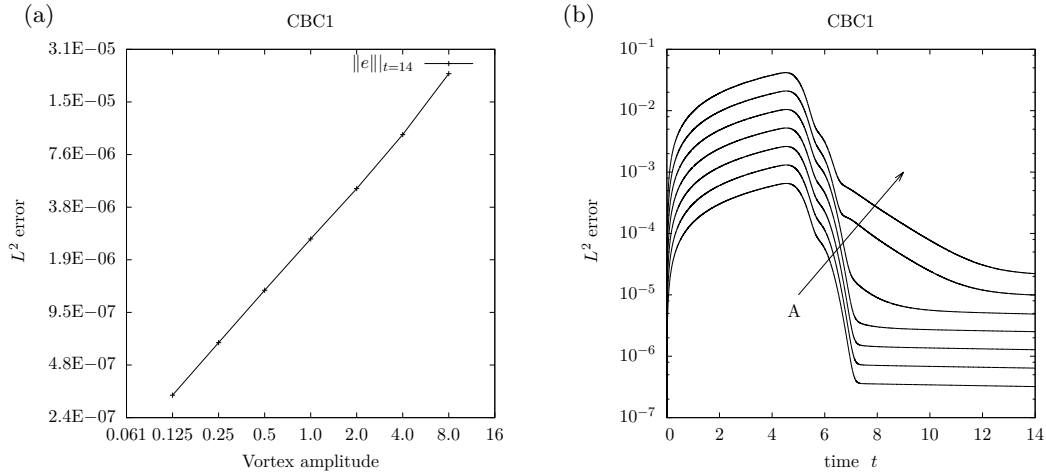


Figure 5.21: CBC1, (a) Size of the first downstream reflection in the  $x$  velocity field for varying vortex amplitudes. (b) Error-time plots corresponding to the vortex strengths:  $A \in \{0.125, 0, 25, 0.5, 1, 2, 4, 8\}$ .

reflections does not vary significantly; there are almost equally large reflections. But those reflections are larger than the reflections that occur if the convection speed is chosen exactly ( $c_x = 1$ ). There is obviously no proportional relation here. The same thing can be observed from the angle disturbance analysis (Figure 5.19). However, what can be noted is that reflections are proportional to the size of the error peak that corresponds to asymmetry as the vortex crosses the outflow boundary.

## 5.5 Flow past a circular cylinder

In this section we consider the flow past a circular cylinder, i.e. the typical Kàrmàn vortex street problem. This test case has been intensively investigated due to its relevance to practical applications. Depending on the Reynolds number, the flow in the cylinder wake is either steady ( $Re \lesssim 50$ ), laminar with vortex shedding ( $50 \lesssim Re \lesssim 150$ ), transitional ( $150 \lesssim Re \lesssim 300$ ), or turbulent ( $Re \gtrsim 300$ ) [2]. As the flow is unsteady ( $Re \gtrsim 50$ ), then the cylinder wake usually shapes into a regular pattern of two parallel slightly shifted rows of vortices. The vortices are formed at the two points of separation on the cylinder. The purpose of this test case is to consider errors in the solutions computed with different domain sizes and consider stability at high Reynolds numbers.

The flow configuration is depicted in Figure 5.22a. We consider a rectangular two dimensional domain of size  $[-L_1, L_2] \times [-L_3, L_3]$  where the cylinder is of circular type with radius  $r$ , and placed with its centrum in  $(x, y) = (0, 0)$ . We choose a rather dense spectral element grid in this problem since the boundary of the cylinder must be densely meshed to provide accurate results. The meshing around the cylinder is illustrated in Figure 5.22b.

The boundary conditions of the problem are chosen in the following way: at the left boundary  $\Gamma_I := \{-L_1\} \times [-L_3, L_3]$  the constant inflow  $\mathbf{u} = (1, 0)^\top$  is imposed; at the upper and lower boundaries ( $[-L_1, L_2] \times \{L_3\}$  and  $[-L_1, L_2] \times \{-L_3\}$ ) periodic boundary conditions are set due to symmetry; on the cylinder boundary

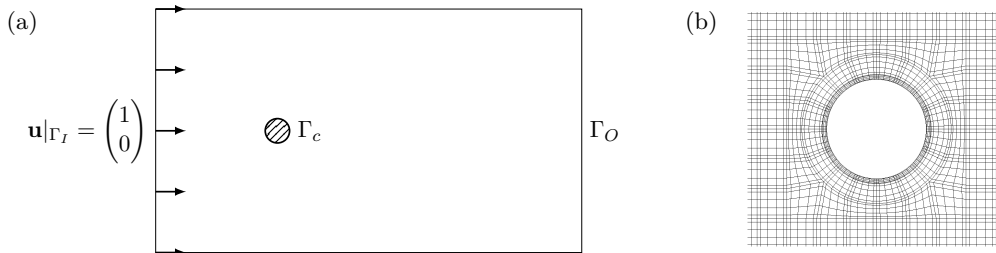


Figure 5.22: (a) Domain configuration. Flow enters from the left boundary with constant unit velocity, pass past a circular cylinder, with boundary  $\Gamma_c$  and leave the domain at the outflow  $\Gamma_O$ . (b) Meshing around the cylinder. Here the element order is  $N = 5$ .

## 5.5. FLOW PAST A CIRCULAR CYLINDER

a homogeneous Dirichlet (“No-slip”) conditions is used; and the right boundary  $\Gamma_O := \{L_2\} \times [-L_3, L_3]$  is the outflow boundary, where one of the outflow boundary condition is imposed. The distance from the cylinder to the periodic boundaries was chosen large enough such that the periodic boundaries do not contain any important information of the cylinder wake. Hence, a larger domain in the  $y$  direction may be needed for lower Reynolds numbers. We may mention that the purpose of our simulations is not to analyze the physical behavior of an accurate modeled vortex shedding. It is possible that the periodic boundary conditions may give rise to some blockage effect [2] etc. which we need to disregard, our results may therefore differ from experimental ones.

We observed that with the initial condition set to  $u = 1, v = 0$ , vortex shedding appeared after a rather long time ( $> 10$ s). This implies expensive computations. To save time, we therefore added a disturbance to the initial condition to enforce the shedding to start. This “fix” was adapted from [10] and simply adds an initial condition with a slightly higher  $x$  velocity at the upper part of the cylinder than the lower. For the initial base flow  $(1, 0)$  we therefore used

$$u = 1 + 0.1 \tanh\left(\frac{y}{0.2\sqrt{2}}\right), \quad (5.19a)$$

$$v = 0. \quad (5.19b)$$

which is actually a flow field where the upper half of the domain has  $u = 1.1$  and the lower half has  $u = 0.9$ .

We present two test configurations; first a flow of  $Re = 100$  (§5.5.1), where vortex shedding is present but with rather weak vortices; secondly we consider the case of  $Re = 1000$  (§5.5.2), where the flow is more complicated and the vortices are stronger. In both cases the convection speed of the convective boundary conditions was chosen as  $\mathbf{c} = (1, 0)^\top$ . The same tests were performed for both configurations: several domains were considered, where the distance to the outflow boundary was varied. The purpose was to test the accuracy and stability of the boundary conditions, and eventually how much the domain must be elongated for the flow to be accurate close to the cylinder.

The drag and lift coefficients (dimensionless quantities of the forces) that act on the cylinder are considered. They are computed as

$$C_D = \frac{2\tilde{F}_x}{\rho U^2}, \quad C_L = \frac{2\tilde{F}_y}{\rho U^2}, \quad (5.20, 5.21)$$

where  $\tilde{F}_x$  is the time averaged drag force,  $\tilde{F}_y$  is the root mean square (r. m. s.) lift force, and  $U$  is the characteristic velocity. Also the frequency of the vortex shedding is of great interest. A dimensionless form related to the frequency is the Strouhal number. It is given by

$$St = \frac{fD}{U}, \quad (5.22)$$

where  $f$  is the frequency in hertz, and  $D$  is the diameter of the cylinder.

### 5.5.1 $Re = 100$

In the  $Re = 100$  case, the element order was chosen as  $N = 5$  in all meshes, and the time-stepping was overall performed with the constant step-size  $\Delta t = 0.001$ . Five different domains were considered; all with the same inflow and cylinder placement, but with a varying location of the outflow boundary. The inflow boundary was defined as  $\Gamma_I = \{(x, y) : x = -15, -15 \leq y \leq 15\}$ , and the outflow boundary conditions was imposed on  $x \in X_O = \{5, 10, 15, 20, 35\}$  over the boundaries  $\{\Gamma_{O,x=i}\}_{i \in X_O}$ , where  $\Gamma_{O,x=i} := \{(x, y) : x = i, -15 \leq y \leq 15\}$ .

In Figure 5.23 we consider the drag and lift coefficients plotted over the time interval  $[0, 80]$  for the NBC1 case. The oscillations mean there is shedding in the cylinder wake. For the shortest domain where the outflow is  $\Gamma_{O,x=5}$  it is noticeable that the phase of the oscillations is significantly larger than for the other cases, which have close to similar curves. The Strouhal number is in this case  $St = 0.1602$  which can be compared to the Strouhal number of the reference solution ( $\Gamma_{O,x=35}$ ) that is  $St = 0.1675$ .

Figure 5.24 shows the drag and lift forces in the CBC1 case. Here the Strouhal number of the  $\Gamma_{O,x=5}$  result is  $St = 0.1639$ , which is closer to the reference solution than NBC1 result of the same domain. From the drag and lift results we noted times where the lift forces of the different solutions coincided. Even though the frequencies of the shedding were not exactly equal for the different cases we wanted to pick times where the solutions close to the cylinder were equal. Therefore we picked the time of the second last peak for all results in Figure 5.23 and Figure 5.24 respectively. In the NBC1 case, the times observed were  $\{75.18, 72.97, 72.67, 72.59, 72.56\}$  corresponding to  $\{\Gamma_{O,x=i}\}_{i \in X_O}$ . For the CBC1 case we got similarly the sequence of times:  $\{73.37, 72.51, 72.53, 72.55, 72.56\}$ . Then the vorticity fields were plotted at that times, side by side for comparison. See in Figure 5.25 and 5.26 contour plots of the various cases. It is easy to see that the frequency of the shedding is slightly different in the  $\Gamma_{O,x=5}$  case (solutions are shifted in the  $x$  direction), and that the convective condition performs overall better than the natural.

For a more detailed analysis of the accuracy related to the placement of the outflow we also considered the cross sections  $\{(x, y) : -15 \leq x \leq x_i, y = 0, i \in X_O\}$  over the vorticity fields, and the several cross sections  $\{(x, y) : x = x_i, -15 \leq y \leq 15, i \in X_O\}$  over the  $x$  velocity fields. The result with the outflow boundary at  $x_O = 35$ , was used as a reference solution. Results corresponding to NBC1 and CBC1 can be considered in Figures 5.27-5.30. For the NBC1 case we can observe that the solutions close to and at the outflow boundary are not consistent with the reference solution. But otherwise the curves almost coincide. The results confirm the observations from Figure 5.25, and also the results from the convective vortex case, i.e. the natural boundary condition NBC1 has accuracy problems as vortices cross the boundary. In the case of CBC1 however, the cross sections of the vorticity coincide with the reference solution for all solutions, even for the short

## 5.5. FLOW PAST A CIRCULAR CYLINDER

domain. We can note some dissimilarities to the reference solution at the boundary for the solutions in domains with outflows at  $x_O = 5$ , and  $x_O = 10$ . But those are small. And for the longer domains the solutions are accurate even over the outflow boundary.

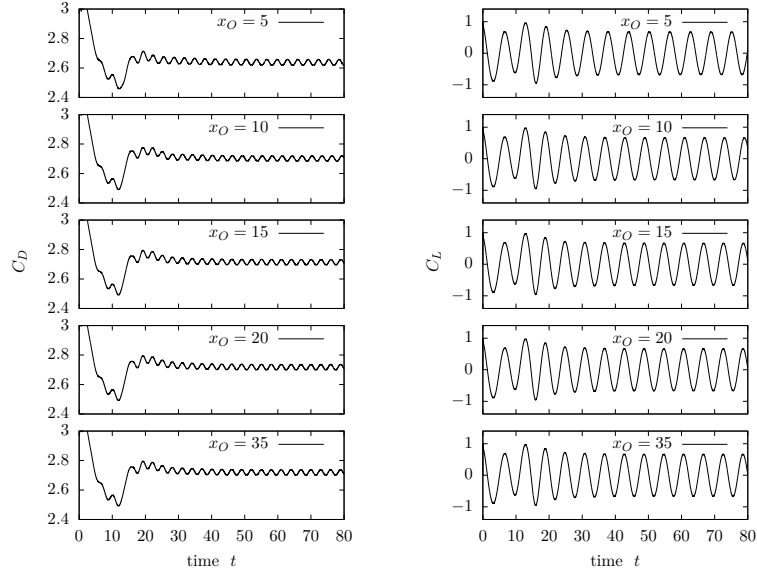


Figure 5.23: NBC1,  $Re = 100$ . Drag and lift coefficients of the cylinder.

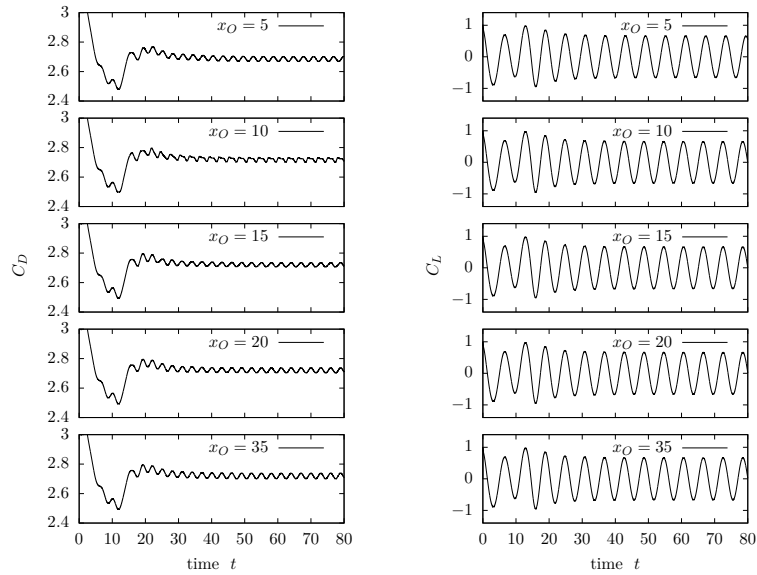


Figure 5.24: CBC1,  $Re = 100$ . Drag and lift coefficients of the cylinder.



Figure 5.25: NBC1,  $Re = 100$ . Comparison of vorticity contours. (a)  $x_O = 5$ , (b)  $x_O = 10$ , (c)  $x_O = 15$ , (d)  $x_O = 20$ , (e)  $x_O = 35$ .

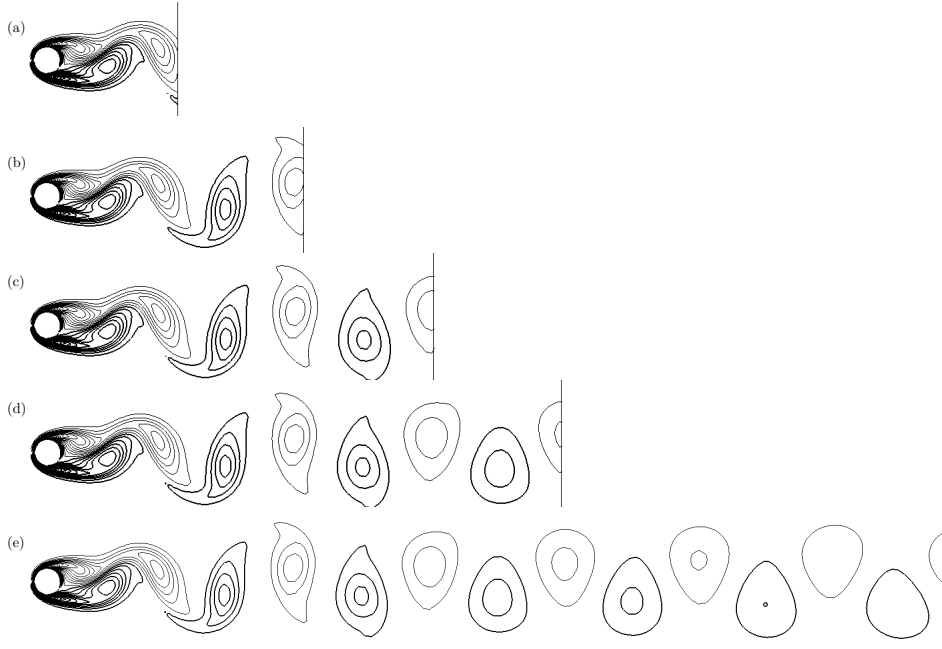


Figure 5.26: CBC1,  $Re = 100$ . Comparison of vorticity contours. (a)  $x_O = 5$ , (b)  $x_O = 10$ , (c)  $x_O = 15$ , (d)  $x_O = 20$ , (e)  $x_O = 35$ .



### 5.5. FLOW PAST A CIRCULAR CYLINDER

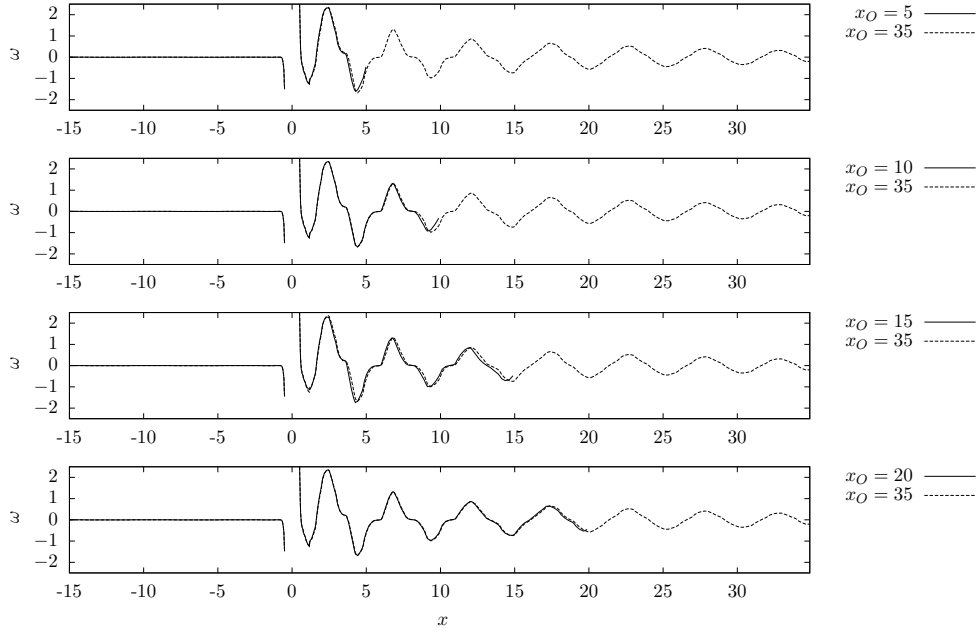


Figure 5.27: NBC1,  $Re = 100$ . Cross sections for  $y = 0$ .

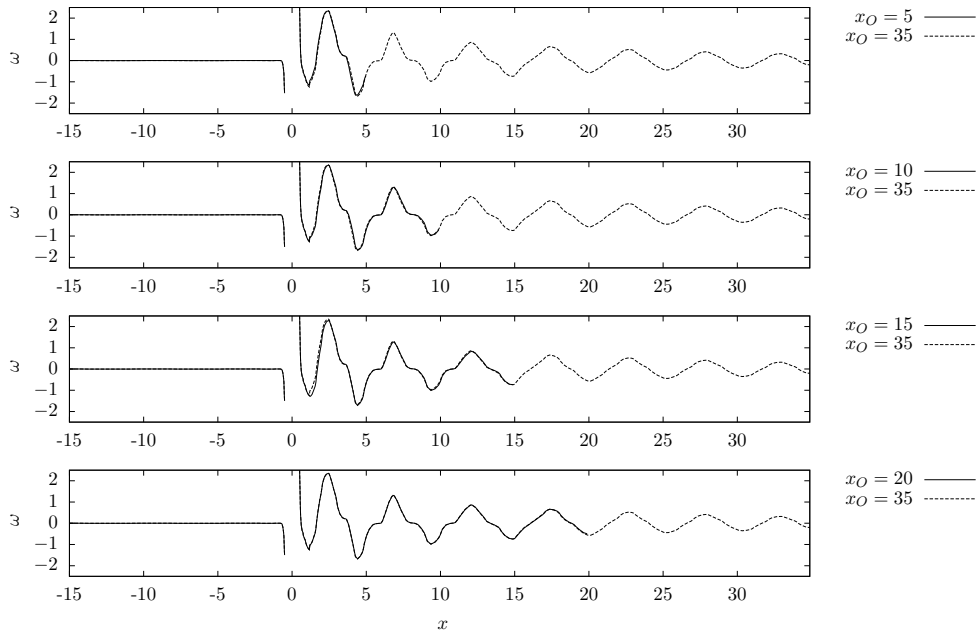


Figure 5.28: CBC1,  $Re = 100$ . Cross sections for  $y = 0$ .

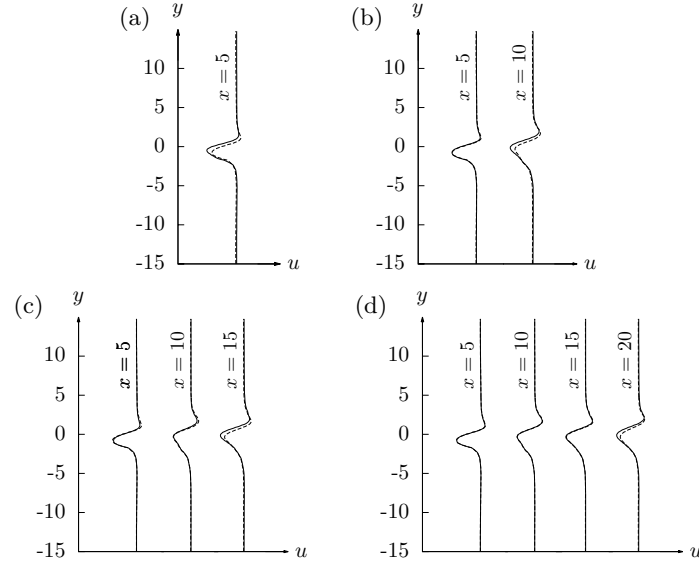


Figure 5.29: NBC1,  $Re = 100$ . Cross sections of the  $u$  field along the  $x$  axis. Dashed line: reference solution  $x_O = 35$ . Solid lines: (a)  $x_O = 5$ , (b)  $x_O = 10$ , (c)  $x_O = 15$ , (d)  $x_O = 20$ .

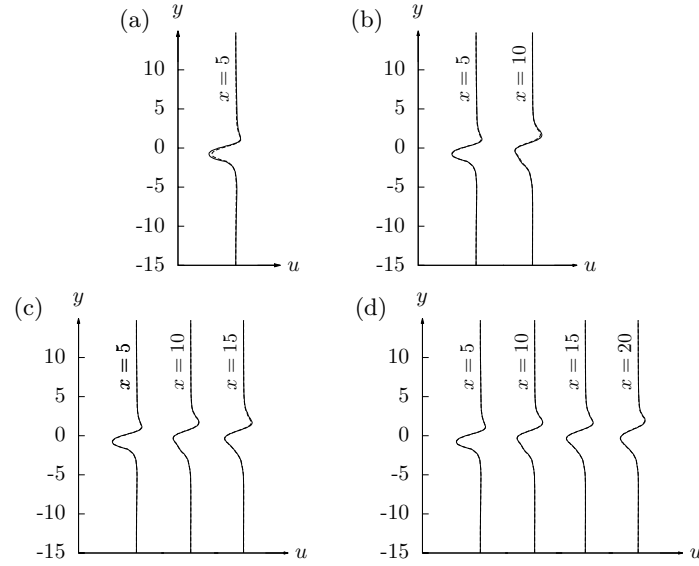


Figure 5.30: CBC1,  $Re = 100$ . Cross sections of the  $u$  field along the  $x$  axis. Dashed line: reference solution  $x_O = 35$ . Solid lines: (a)  $x_O = 5$ , (b)  $x_O = 10$ , (c)  $x_O = 15$ , (d)  $x_O = 20$ .

## 5.5. FLOW PAST A CIRCULAR CYLINDER

### 5.5.2 $Re = 1000$

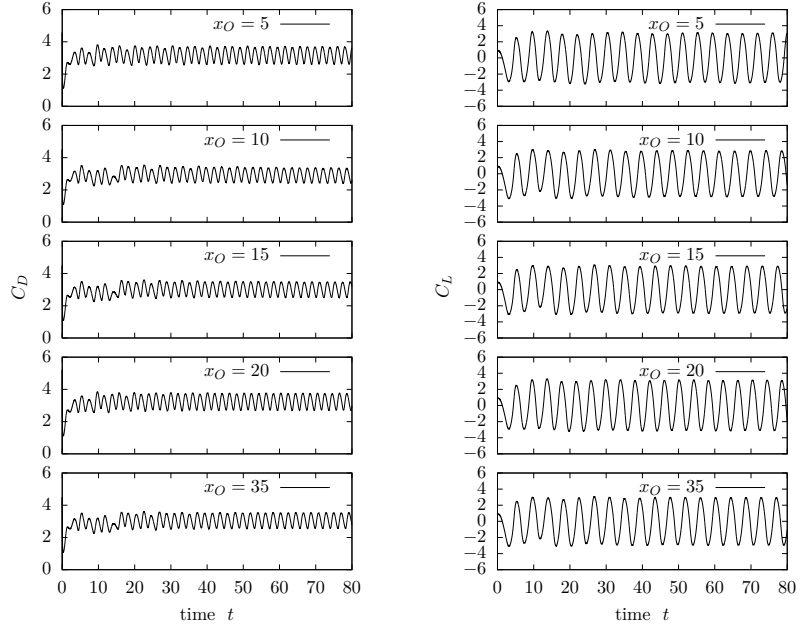
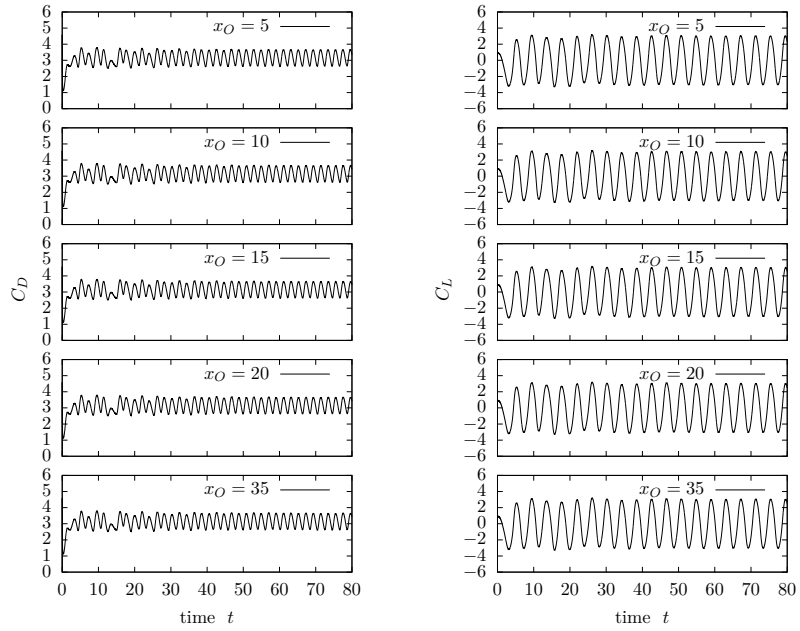
We follow on with the more convective dominated flow of  $Re = 1000$ . Now the cylinder wake becomes turbulent in the three dimensional setting, and the vortices of the wake are stronger than in the  $Re = 100$  case due to less dissipation. Indeed, now the vortices that are produced by the cylinder have similar properties as the single vortex in the previous section.

Since the problem now is more sensible (Due to the higher Reynolds number. See e.g. §1.2.) we choose to increase the resolution of the mesh; the order of each elements is now chosen as  $N = 12$ . Also the domain is shrunk down in the  $y$  direction to speed up the computations; the domain size in the  $y$  direction was changed from  $[-15, 15]$  to  $[-5, 5]$ . We assume that it is sufficient with a smaller domain here because of the less amount of dissipation.

The standard natural boundary condition (NBC1) becomes unstable in this setting. As the first vortex of the shedding hits the boundary the problem blows-up. This confirms the analytic results of §3. Half of a vortex has negative flux, and the vortices now have a strength that is higher than  $A = 1.05$ , which was shown to be the approximate limit of instability for  $Re = 1000$ . Hence, this case is a great possibility to compare the stabilized natural boundary condition with the convective. We do not show any results of the CBC2 case here, but testing have shown similar results as for the CBC1 conditions also for CBC2.

The tests were chosen similar as for the  $Re = 100$  case. In Figure 5.31 consider the drag and lift coefficients on the cylinder in the time interval  $t \in [0, 80]$  of the NBC2 solutions. A first general observation is that there is more oscillations compared to the  $Re = 100$  results (higher Strouhal number), and for NBC2 the drag and lift curves do not coincide for any of the solutions. Similar to before we noted down times in the lift coefficient plots to obtain states for comparison. Now we picked times at the last peaks. Figures 5.33-5.34 show the vorticity fields at the chosen times, and the Figures 5.35 and 5.37 show cross sections of the vorticity fields and the  $x$  velocity fields. We can observe that for all domain sizes the solution over the outflow is very inaccurate, but also the solution some distance into the domains are rather erroneous for NBC2. Especially for the domains with outflows  $x_O \leq 15$  this is evident, where the only solution that may be used in an accurate simulation is the solution close to the cylinder (at  $x = 5$ ). For the case with outflow at  $x_O = 20$  however, the solutions at  $x = 5, 10$ , and  $15$  are still accurate.

Considering the results of the convective boundary condition CBC1 we can see improvements over the NBC1 counterparts in all tests. The vorticity plots now almost coincide even for the shortest domain considered (see Figure 5.34), the Strouhal numbers are accurate even for the computations on the shortest domain (the frequency of the drag and lift plots coincide, see Figure 5.32), and the cross section plots (Figures 5.36 and 5.38) strengthen the conclusions even more.


 Figure 5.31: NBC2,  $Re = 1000$ . Drag and lift coefficients on the cylinder.

 Figure 5.32: CBC1,  $Re = 1000$ . Drag and lift coefficients on the cylinder.

## 5.5. FLOW PAST A CIRCULAR CYLINDER

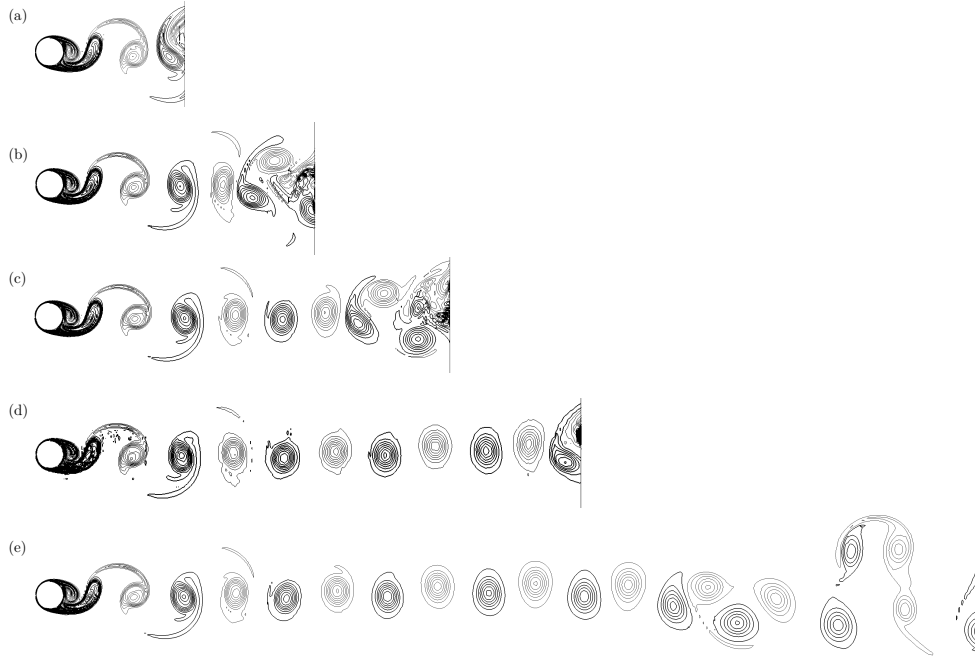


Figure 5.33: NBC2,  $Re = 1000$ . Comparison of vorticity contours. (a)  $x_O = 5$ , (b)  $x_O = 10$ , (c)  $x_O = 15$ , (d)  $x_O = 20$ , (e)  $x_O = 35$ .

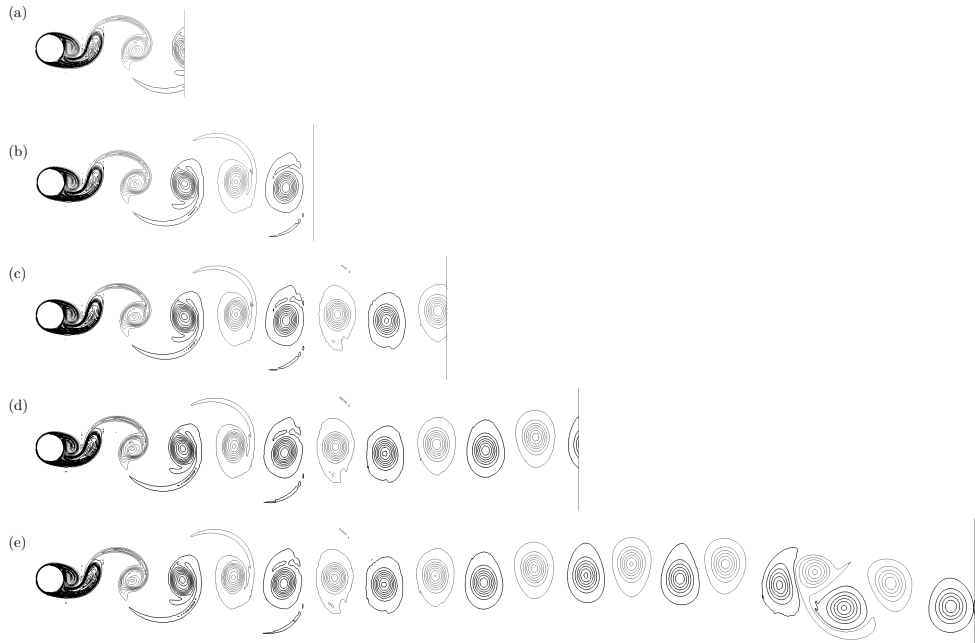


Figure 5.34: CBC1,  $Re = 1000$ . Comparison of vorticity contours. (a)  $x_O = 5$ , (b)  $x_O = 10$ , (c)  $x_O = 15$ , (d)  $x_O = 20$ , (e)  $x_O = 35$ .

## CHAPTER 5. NUMERICAL RESULTS

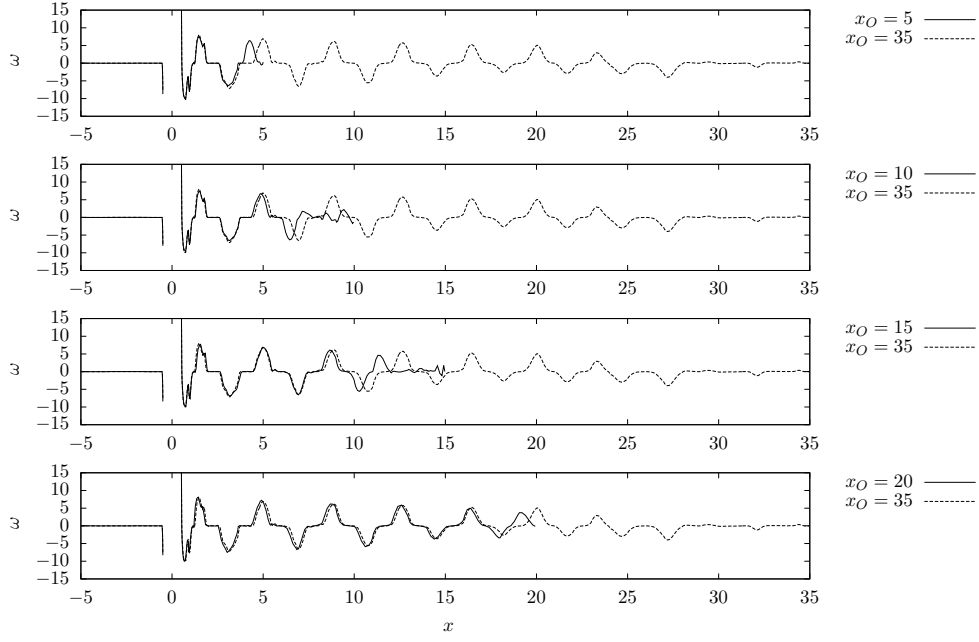


Figure 5.35: NBC2,  $Re = 1000$ . Cross sections for  $y = 0$  of the vorticity field.

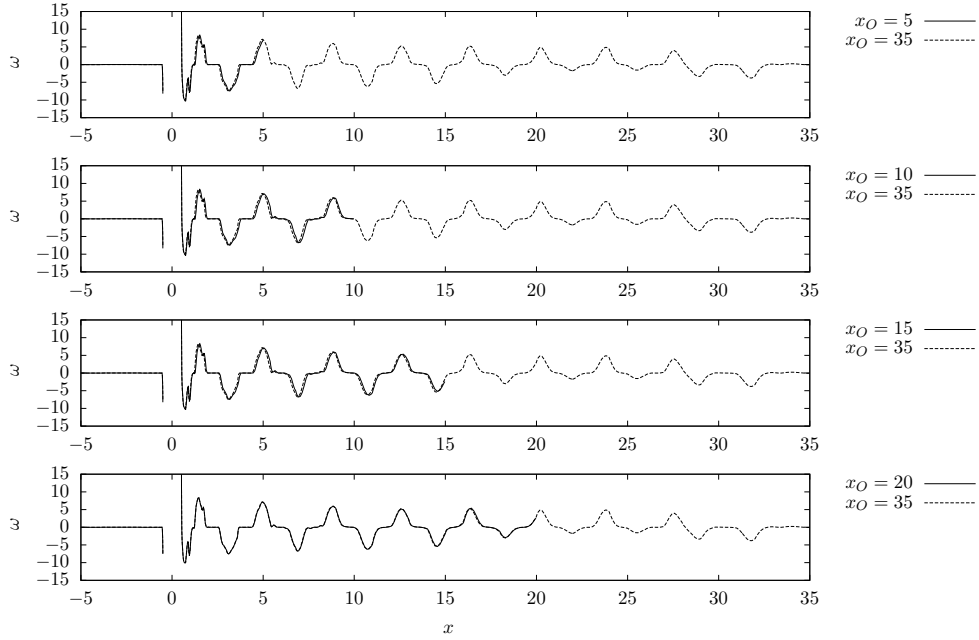


Figure 5.36: CBC1,  $Re = 1000$ . Cross sections for  $y = 0$  of the vorticity field.

## 5.5. FLOW PAST A CIRCULAR CYLINDER

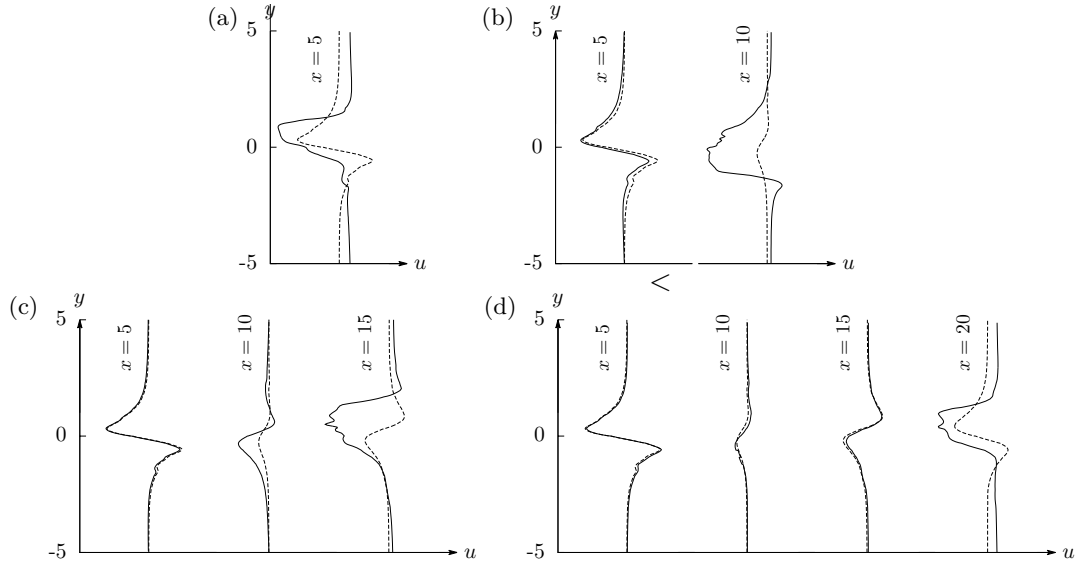


Figure 5.37: NBC2,  $Re = 1000$ . Cross sections of the  $u$  field along the  $x$  axis. Dashed line: reference solution  $x_O = 35$ . Solid lines: (a)  $x_O = 5$ , (b)  $x_O = 10$ , (c)  $x_O = 15$ , (d)  $x_O = 20$ .

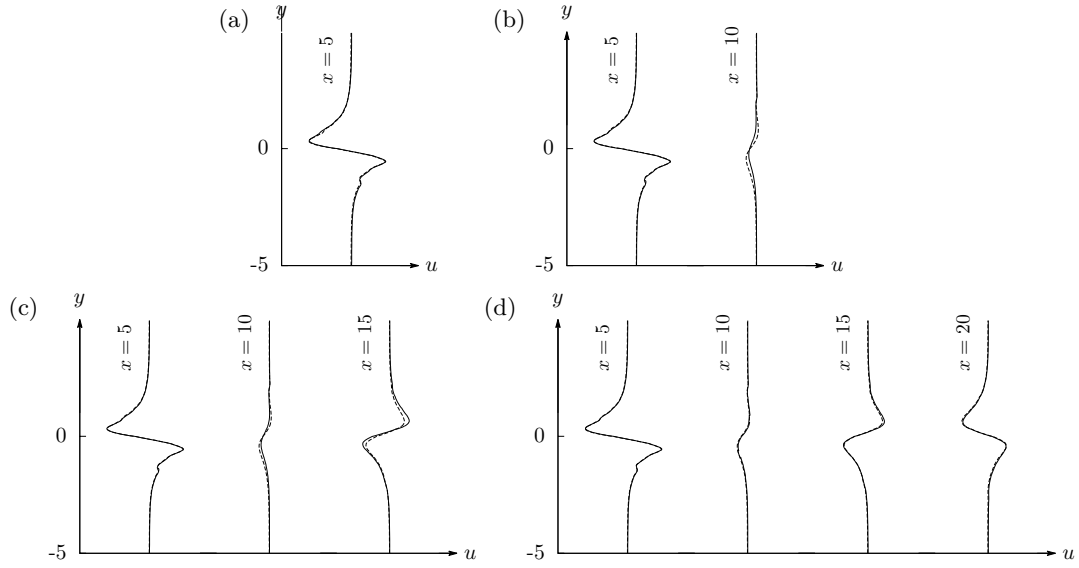


Figure 5.38: CBC1,  $Re = 1000$ . Cross sections of the  $u$  field along the  $x$  axis. Dashed line: reference solution  $x_O = 35$ . Solid lines: (a)  $x_O = 5$ , (b)  $x_O = 10$ , (c)  $x_O = 15$ , (d)  $x_O = 20$ .





## Chapter 6

# Discussion and conclusions

In this thesis we have presented the implementation and analysis of convective boundary conditions for incompressible flows in a spectral element framework. We have compared new implementations with the already available natural boundary condition (the so called “O” condition) of the Nek5000 solver, and a stabilized version of it (by Dong et al. [10]).

Different discretization alternatives of the convective condition  $\phi_t + \mathbf{c} \cdot \nabla \phi = 0$  were investigated. The first and main idea was to use the same discretization technique for the convective condition as for the inner points, i.e. SEM in space, and the BDFk/EXTk scheme in time. Another idea was to impose the discretized exact solution of the convective equation directly, i.e. simply pick the solution from the former time step by following the characteristic lines of the solution. A third technique was also discussed, where the accuracy of the second implementation was studied even further by considering an implementation that allowed curved characteristics. The curved characteristic implementation was examined due to the accessibility to the OIFS/Characteristic time stepping technique in Nek5000. The usefulness of this implementation may be significant if the OIFS/Characteristic time-stepping is used for stability purposes, which make the need of sub-stepping also required for the boundary points. More analysis are needed for the curved characteristic method.

Tests of spatial and temporal discretization orders, indicate that the BDF3/EXT3 discretized convective boundary condition preserves the order of the solver, both in space and in time. The linear characteristic implementation preserves the spatial convergence, but the temporal order depends on the time step size, where a smaller step size is better. This means that both can be used successfully, but the characteristic version requires sufficiently small step sizes in time ( $\Delta t = 0.001$  shall be sufficient here) to provide accurate results.

As long as the user has a sufficient insight in the physical solution (leads to an accurate estimation of the convection speed), and that the flow is convection-

## CHAPTER 6. DISCUSSION AND CONCLUSIONS

dominant ( $Re$  is large), then our numerical tests show remarkable good results for the convective boundary conditions. The convective conditions handle strong vortices without significant reflections, and in comparison to the natural boundary conditions they showed better results in all unsteady flow tests. Even in situations where the convection speed was relatively poorly estimated the amount of reflections and accuracy errors were still not much worse than what the natural boundary condition provided us with. In the classical Kármán vortex street problem the convective boundary conditions gave accurate results, even for highly truncated domains.

A steady state problem was also considered (Kovasznay flow; same test as for spatial convergence), where the Reynolds number was picked low ( $Re = 40$ ). For this test case the convective conditions had problems, both because of the fact that a steady state solution transforms the convective equation into a homogeneous Neumann condition, but also that the convective conditions do not be an accurate approximation of the Navier–Stokes equations for this specific problem ( $Re$  small). We can conclude that for the convective boundary conditions to work well the diffusion term of the Navier–Stokes equations must not be dominant. In case of low Reynolds number flows, the natural boundary condition indeed works better.

Regarding stability, our tests show that the convective conditions are a great improvement over the natural condition. The natural condition indeed implies blown up solutions for sufficiently large Reynolds numbers together with negative flux over the boundary. For instance, if a vortex travels across the boundary, the solution may blow up since half the vortex has negative flux. The stability fix of Dong et al. [10] was shown to cure the stability issue, but did still imply as bad accuracy as the natural condition.

One must keep in mind that the convective boundary condition is not always a good general choice. The main problem regarding the implementation of the convective condition  $\mathbf{u}_t + \mathbf{c} \cdot \nabla \mathbf{u} = 0$  is the approximation of the convection speed  $\mathbf{c}$ , which showed rather bad robustness properties. Our results indicate that the convective condition does only become an accurate outflow boundary condition if the flow is highly convective (large  $Re$ ) and if the convection speed is close to the group velocity of the flow. Our tests show that a wrongly estimated convection speed implies reflections.

Another problem that is also related to the convection speed is the importance of correct angle of incidence of the streamlines over the boundary. The convective condition does only give minimal amount of reflections if it is transporting the flow in the correct direction. Our results show that also a wrong angle of incidence of a disturbance leads to reflections.

Due to time restrictions etcetera there are some analysis that were not included in this thesis. Let us describe some of these observations and ideas shortly.

A method of computing the convection speed from the flow fields using exponential filtering was considered (§4.4.2). The tests we have performed this far show shared results, and more testing is needed. Our results indicate that the filter seems

## 6.1. OUTLOOK

to work better for small disturbances than large. Small disturbances can often be close to completely disregarded by the filter, while larger cannot. The more time-steps a disturbance need to travel across the boundary, the harder it is for the exponential filter to completely disregard it.

Damping regions such as sponge layers (see §B) were also tested together with the different outflow boundary conditions. More analysis are needed also here. The results this far show that sponge regions produce reflections if the amount of damping is large. If we for instance try to damp out rather strong vortices, the length of the sponge region must be chosen very long to restrict significant reflections. If the aim of the addition of a sponge region is to damp out numerical reflections, then our testing shows that the sponge region does not work well without a great amount of modification; it may indeed produce larger reflections itself than the numerical reflections it is supposed to damp out.

## 6.1 Outlook

The BDFk/EXTk and the linear characteristic discretizations of the convective boundary condition have shown advantages over the natural Nek5000 “O” condition in convection-dominated problems, both according to accuracy and stability. Therefore it is relevant for researchers solving any kind of unsteady flow problems to use convective conditions instead of the “O” condition. We therefore intend to include the convective boundary conditions in the Nek5000 repository. The first step might be to add some example case. Since the estimation of the convection speed is crucial, general estimation approaches might be further investigated (filtering etc.). For the convective boundary conditions to work optimally, there might be several types of approximation alternatives available such as constant, local, and filtered types.

Also we like to add the (Dong et al.) stabilized version of the “O” condition to the Nek5000 repository since it is generally to prefer over the standard “O” condition; and may be to prefer over the convective conditions too when a significant amount of diffusion is present.

Other possible continuations of this thesis are: further testing of the convective conditions in more complex flows; analysis of optimal sponge regions to damp out upstream reflections; new ways of estimating the convection speed (such as filtering techniques); further analytic analysis of the size of numerical reflections for different outflow boundary conditions; and investigation in mass conservation errors of the different outflow boundary conditions studied.



## Appendix A

# Mathematical concepts and formulas

### A.1 Tensor products

A tensor product, defined as  $\otimes$  is the most general bilinear operation. It can be applied to vectors, matrices, tensors, vector spaces, algebras, topological vector spaces, and modules etc.

The tensor product of two vector spaces  $V$  and  $W$  is defined as the vector space over  $\mathbb{F}$ , given by the map

$$\phi : V \times W \mapsto V \otimes W, \quad (\text{A.1})$$

equipped with the following fundamental properties:

- i. The map  $\phi$  is bilinear. i.e. for the scalar  $\alpha$ , and the vectors  $\mathbf{v}, \mathbf{v}^{(1)}, \mathbf{v}^{(2)} \in V$  and  $\mathbf{w}, \mathbf{w}^{(1)}, \mathbf{w}^{(2)} \in W$  we have

$$(\mathbf{v}^{(1)} + \mathbf{v}^{(2)}) \otimes \mathbf{w} = \mathbf{v}^{(1)} \otimes \mathbf{w} + \mathbf{v}^{(2)} \otimes \mathbf{w}, \quad (\text{A.2})$$

$$\mathbf{v} \otimes (\mathbf{w}^{(1)} + \mathbf{w}^{(2)}) = \mathbf{v} \otimes \mathbf{w}^{(1)} + \mathbf{v} \otimes \mathbf{w}^{(2)} \quad (\text{A.3})$$

$$\alpha(\mathbf{v} \otimes \mathbf{w}) = (\alpha\mathbf{v}) \otimes \mathbf{w} = \mathbf{v} \otimes (\alpha\mathbf{w}) \quad (\text{A.4})$$

$$\mathbf{0} \otimes \mathbf{w} = \mathbf{v} \otimes \mathbf{0} = \mathbf{0}. \quad (\text{A.5})$$

- ii. If  $\{\mathbf{v}_i\}_{i=1}^n$  is a basis of  $V$  and  $\{\mathbf{w}_j\}_{j=1}^m$  a basis of  $W$ , then  $\{\mathbf{v}_i \otimes \mathbf{w}_j, i = 1, 2, \dots, n, j = 1, 2, \dots, m\}$  is a basis of  $V \otimes W$ . The dimension of  $V \otimes W$  will therefore be given by the product of the dimensions of  $V$  and  $W$ .

- iii. Every element in a vector space can be written as a linear combination of its basis: given scalar coefficients  $a_i$  and  $b_i$ , we can write  $V \ni \mathbf{v} = a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \dots + a_n\mathbf{v}_n$  and  $W \ni \mathbf{w} = b_1\mathbf{w}_1 + b_2\mathbf{w}_2 + \dots + b_m\mathbf{w}_m$ . Hence we might write

$$\mathbf{v} \otimes \mathbf{w} = \sum_{i=1}^n \sum_{j=1}^m a_i \cdot b_j (\mathbf{v}_i \otimes \mathbf{w}_j). \quad (\text{A.6})$$

Thus, by defining  $c_{ij} := a_i \cdot b_j$ , we get

$$\mathbf{v} \otimes \mathbf{w} = \sum_{i=1}^n \sum_{j=1}^m c_{ij} (\mathbf{v}_i \otimes \mathbf{w}_j). \quad (\text{A.7})$$

This is the form of a general element in  $V \otimes W$ . However, every polynomial in two variables  $x$  and  $y$  can not be written as a product  $(a_0 + a_1x \dots) \cdot (b_0 + b_1y + \dots)$  (e.g.  $xy + 1$  can not), so the form (A.6) is not always possible in general. However, if  $V$  and  $W$  are one dimensional this problem is not present, which is the case in e.g. the SEM implementation. Hence, for SEM, every element are represented in the form of (A.7).

Note that, adding one more vector space to the problem  $U$ , with elements  $\mathbf{u}$  represented in the basis  $\{\mathbf{u}_k\}_{k=1}^l$ , results in

$$\mathbf{v} \otimes \mathbf{w} \otimes \mathbf{u} = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l c_{ijk} (\mathbf{v}_i \otimes \mathbf{w}_j \otimes \mathbf{u}_k). \quad (\text{A.8})$$

## A.2 Orthogonal Lagrange polynomials

### A.2.1 Legendre polynomials

“Legendre functions” are the solutions to the “Legendre’s differential equation”

$$\frac{d}{dx} \left[ (1 - x^2) \frac{d}{dx} P_N(x) \right] + N(N + 1) P_N(x) = 0. \quad (\text{A.9})$$

These solutions for  $N = 0, 1, 2, \dots$  (with the normalization  $P_N(1) = 1$ ) form a polynomial sequence of orthogonal polynomials called the “Legendre polynomials”. The first few Legendre polynomials are given in table A.1.

$N$	$P_N(x)$
0	1
1	$x$
2	$\frac{1}{2}(3x^2 - 1)$
3	$\frac{1}{2}(5x^3 - 3x)$
4	$\frac{1}{8}(35x^4 - 30x^2 + 3)$
5	$\frac{1}{8}(63x^5 - 70x^3 + 15x)$
6	$\frac{1}{16}(231x^6 - 315x^4 + 105x^2 - 5)$
7	$\frac{1}{16}(429x^7 - 693x^5 + 315x^3 - 35x)$
8	$\frac{1}{128}(6435x^8 - 12012x^6 + 6930x^4 - 1260x^2 + 35)$
9	$\frac{1}{128}(12155x^9 - 25740x^7 + 18018x^5 - 4620x^3 + 315x)$
10	$\frac{1}{256}(46189x^{10} - 109395x^8 + 90090x^6 - 30030x^4 + 3465x^2 - 63)$ .

Table A.1: Legendre polynomials.

## A.2. ORTHOGONAL LAGRANGE POLYNOMIALS

### A.2.2 GLL polynomials

Gauss–Lobatto–Legendre (GLL) Lagrangian basis functions are given by:

$$\phi_j(\xi) = \frac{-1}{N(N+1)} \frac{(1-\xi^2)P'_N(\xi)}{(\xi-\xi_j)P_N(\xi_j)}, \quad 0 \leq j \leq N, \quad \xi \in [-1, 1]. \quad (\text{A.10})$$

where  $\{\xi_j\}_{j=0}^N$  are the roots of the polynomial  $(1-\xi^2)P'_N(\xi)$  (GLL points).

The GLL quadrature rule is given by

$$\int_{-1}^1 u(\xi) dx \equiv \sum_{k=0}^N \rho_k u(\xi_k), \quad (\text{A.11})$$

where

$$\xi_k : \xi_0 = -1, \xi_N = 1, \text{ zeros of } P'_N(\xi), \quad 1 \leq k \leq N-1 \quad (\text{A.12})$$

$$\rho_k = \frac{2}{N(N+1)} \frac{1}{[P_N(\xi_k)]^2}, \quad 0 \leq k \leq N. \quad (\text{A.13})$$

The GLL quadrature is exact for polynomials of orders up to  $2N-1$ , see [9].

### A.2.3 GL polynomials

Gauss–Legendre (GL) Lagrangian basis functions are given by:

$$\phi_j(\zeta) = \frac{P_{N-1}(\zeta)}{(\zeta-\zeta_j)P'_{N-1}(\zeta_j)}, \quad 1 \leq j \leq N-1, \quad \zeta \in [-1, 1]. \quad (\text{A.14})$$

where  $\{\zeta_j\}_{j=1}^{N-1}$  are the roots of the polynomial  $P_{N-1}(\zeta)$  (GL points).

The GL quadrature rule is given by

$$\int_{-1}^1 u(\zeta) dx \equiv \sum_{k=1}^{N-1} \omega_k u(\zeta_k), \quad (\text{A.15})$$

where

$$\zeta_k : \text{zeros of } P_{N-1}(\zeta), \quad 1 \leq k \leq N-1 \quad (\text{A.16})$$

$$\omega_k = \frac{2}{(1-\zeta_k^2)[P'_{N-1}(\zeta_k)]^2}, \quad 1 \leq k \leq N-1. \quad (\text{A.17})$$

The GL quadrature is exact for polynomials of orders up to  $2N-3$ , see [9].

### A.3 The inf-sup condition

The so called “inf-sup” condition by Brezzi [5] and Babuška [3] guarantees existence and uniqueness of a solution to a Stokes problem (e.g. the semi-discrete system of §2.3), i.e. it ensures the absence of spurious pressure modes and non-physical wiggles.

Consider the Steady Stokes problem over the domain  $\Omega$

$$-\frac{1}{Re}\nabla^2 \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega, \quad (\text{A.18a})$$

$$-(\nabla \cdot \mathbf{u}) = 0 \quad \text{in } \Omega, \quad (\text{A.18b})$$

The discrete variational form of this problem reads:

Find  $(\mathbf{u}_N, p_N) \in (V_N^d \subset H_0^1(\Omega)^d) \times (Z_N \subset L_0^1(\Omega))$  such that

$$\frac{1}{Re}(\nabla \mathbf{u}_N, \nabla \mathbf{v}_N)_N - (p_N, \nabla \cdot \mathbf{v}_N)_N = (\mathbf{f}, \mathbf{v}_N)_N, \quad \forall \mathbf{u} \in V_N^d \quad (\text{A.19a})$$

$$-(\nabla \cdot \mathbf{u}_N, q_N)_N = 0, \quad \forall q \in Z_N. \quad (\text{A.19b})$$

where  $(\cdot, \cdot)_N$  is the discrete inner product to be defined. A function  $q_N \in Z_N$  is called a spurious mode of the Stokes problem (A.18) if  $\forall \mathbf{v}_N \in V_N^d$

$$(\nabla \cdot \mathbf{v}_N, q_N)_N = 0. \quad (\text{A.20})$$

Now suppose that there exist a solution  $(\mathbf{u}_N, p_N)$  to (A.19), and that  $q_N$  satisfies (A.20) then  $p_N + \alpha q_N$ ,  $\alpha \in \mathbb{R}$  is also a solution to (A.19), i.e. uniqueness is no longer ensured.

The inf-sup condition reads: Problem (A.18) is well posed, if there exist a real  $\beta_N > 0$  such that

$$\beta_N = \inf_{q_N \in Z_N, q_N \neq 0} \sup_{\mathbf{v}_N \in X_N} \frac{(\nabla \cdot \mathbf{v}_N, q_N)_N}{\|\mathbf{v}_N\|_{H_0^1(\Omega)^d} \|q_N\|_{L_0^2(\Omega)}}. \quad (\text{A.21})$$

This condition is obviously not satisfied if a spurious mode (A.20) exist in  $Z_N$ , which is the case for the standard setting of  $\mathbb{P}_N \times \mathbb{P}_N$ .

### A.4 Backward differentiating scheme (BDF)

A backward differentiating formula (BDF) is a family of implicit methods used to solve the initial value problem

$$\frac{\partial y}{\partial t} = f(t, y), \quad y(t_0) = y_0. \quad (\text{A.22})$$



#### A.5. EXTRAPOLATION SCHEME (EXT)

The general formula for a BDF is written as

$$\frac{1}{\Delta t} \sum_{i=0}^k \beta_i y^{n+1-i} \approx f(y^{n+1}) \quad (\text{A.23})$$

where  $\Delta t$  denotes the time step-size. The coefficients  $\{\beta_i\}_{i=0}^k$  are chosen so that the method achieves order  $k$ , which is the maximum possible.

The BDF coefficients  $\{\beta_i\}_{i=0}^k$  of order 1 to 5 are given by:

$k$	$\beta_0$	$\beta_1$	$\beta_2$	$\beta_3$	$\beta_4$	$\beta_5$
1	1	-1				
2	3/2	-2	1/2			
3	11/6	-3	3/2	-1/3		
4	25/12	-4	3	-4/3	1/4	
5	137/60	-5	5	-10/3	5/4	-1/5

Table A.2: BDFk coefficients.

### A.5 Extrapolation scheme (EXT)

The  $k$ :th order extrapolation (EXTk) of a general non linear term  $f(y^{n+1})$  is given by

$$f(y^{n+1}) \approx \sum_{i=1}^k \alpha_i f(y^{n+1-i}), \quad (\text{A.24})$$

where the coefficeints  $\{\alpha_i\}_{i=1}^k$  are given by:

$k$	$\alpha_1$	$\alpha_2$	$\alpha_3$
1	1		
2	2	-1	
3	3	-3	1

Table A.3: EXTk coefficients.

## A.6 Alternate forms of the Navier–Stokes equations

### A.6.1 Alternate forms of the convective term

The convection operator can be written in two different ways

$$\nabla \cdot \mathbf{u}\mathbf{u} = 0, \quad \text{conservative form} \quad (\text{A.25})$$

$$(\mathbf{u} \cdot \nabla)\mathbf{u} = 0, \quad \text{convective form} \quad (\text{A.26})$$

$$\frac{1}{2}(\mathbf{u} \cdot \nabla)\mathbf{u} + \frac{1}{2}\nabla \cdot \mathbf{u}\mathbf{u}, \quad \text{skew-symmetric form.} \quad (\text{A.27})$$

**Theorem A.1.** (A.25)-(A.27) are equivalent if  $\nabla \cdot \mathbf{u} = 0$ .

*Proof.* (A.25)  $\implies$  (A.26):

$$\nabla \cdot \mathbf{u}\mathbf{u} = \mathbf{u}(\nabla \cdot \mathbf{u}) + \mathbf{u} \cdot \nabla \mathbf{u} = \mathbf{u} \cdot \nabla \mathbf{u}.$$

(A.27)  $\implies$  (A.26): Use the result above to rewrite the second term, then

$$\frac{1}{2}(\mathbf{u} \cdot \nabla)\mathbf{u} + \frac{1}{2}\nabla \cdot \mathbf{u}\mathbf{u} = \frac{1}{2}(\mathbf{u} \cdot \nabla)\mathbf{u} + \frac{1}{2}(\mathbf{u} \cdot \nabla)\mathbf{u} = (\mathbf{u} \cdot \nabla)\mathbf{u}.$$

□

Although (A.25)-(A.27) are equivalent in the continuous setting, differences do appear in the discrete case, due to the non-exact representation of  $\mathbf{u}$ ,  $\nabla \cdot \mathbf{u}$  and  $\nabla \mathbf{u}$ . Numerical tests has shown that for spectral element discretizations based on the convective term yields more accurate results than discretizations based on the skew symmetric analogue, with very small differences [8]. Thus, accuracy is hardly the reason to choose one of these versions over another. However, the distribution of the eigenvalues varies for the three. Couzy computed the eigenvalues numerically by discretizing the following expressions with using the spectral element method

$$\frac{\partial \theta}{\partial t} = -\nabla \cdot \mathbf{u}\theta, \quad -\nabla \cdot \mathbf{u} = 0, \quad (\text{A.28})$$

$$\frac{\partial \theta}{\partial t} = -\mathbf{u} \cdot \nabla \theta, \quad -\nabla \cdot \mathbf{u} = 0, \quad (\text{A.29})$$

$$\frac{\partial \theta}{\partial t} = -\frac{1}{2}(\mathbf{u} \cdot \nabla \theta + \nabla \cdot \mathbf{u}\theta), \quad -\nabla \cdot \mathbf{u} = 0, \quad (\text{A.30})$$

With homogeneous Dirichlet boundary conditions on  $\theta$ . The results he obtained was that the largest imaginary eigenvalue grows like  $\mathcal{O}(N^2)$  and there is no difference in the conservative, convective, and skew-symmetric form for the case of  $\mathbf{u} = \mathbf{1}$ . But if the velocity cannot be represented exactly, the skew-symmetric form is preferable, since the real parts of the eigenvalues was shown to be always close to machine epsilon. The conservative and convection forms gave rise to real components with positive and negative signs.

## A.6. ALTERNATE FORMS OF THE NAVIER-STOKES EQUATIONS

In another study by Wilhelm and Kleiser ?? it was noticed that an unstable behavior occurred for staggered grids for the conservative and skew-symmetric form but stable for the convective.

Recall the trilinear convection term of the variational form (??)

$$c(\mathbf{w}, \mathbf{u}, \mathbf{v}) = \int_{\Omega} \mathbf{w} \cdot \nabla \mathbf{u} \cdot \mathbf{v} \, d\Omega, \quad \forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in H^1(\Omega). \quad (\text{A.31})$$

The skew-symmetrized form of (A.31) reads

$$\tilde{c}(\mathbf{w}, \mathbf{u}, \mathbf{v}) = \frac{1}{2}c(\mathbf{w}, \mathbf{u}, \mathbf{v}) - \frac{1}{2}c(\mathbf{w}, \mathbf{v}, \mathbf{u}) \quad (\text{A.32})$$

**Theorem A.2.** *Whenever  $\nabla \cdot \mathbf{u} = 0$  in  $\Omega$ , if  $\mathbf{u} \cdot \mathbf{n} = 0$ , or  $\mathbf{v} = \mathbf{0}$  on  $\Gamma$ , then*

$$\tilde{c}(\mathbf{u}, \mathbf{u}, \mathbf{v}) = c(\mathbf{u}, \mathbf{u}, \mathbf{v}). \quad (\text{A.33})$$

*Proof.* Since  $\nabla \cdot \mathbf{u} = 0$  we get

$$\nabla \cdot (\mathbf{u}\mathbf{v}) = \mathbf{v}(\nabla \cdot \mathbf{u}) + \mathbf{u} \cdot \nabla \mathbf{v} = \mathbf{u} \cdot \nabla \mathbf{v}.$$

Thus,

$$\begin{aligned} \tilde{c}(\mathbf{u}, \mathbf{u}, \mathbf{v}) &= \frac{1}{2}c(\mathbf{u}, \mathbf{u}, \mathbf{v}) - \frac{1}{2}c(\mathbf{v}, \mathbf{u}, \mathbf{u}) \\ &= \frac{1}{2} \int_{\Omega} \mathbf{u} \cdot \nabla \mathbf{u} \cdot \mathbf{v} \, d\Omega - \frac{1}{2} \int_{\Omega} \mathbf{v} \cdot \nabla \mathbf{u} \cdot \mathbf{u} \, d\Omega \\ &= \frac{1}{2} \int_{\Omega} \mathbf{u} \cdot \nabla \mathbf{u} \cdot \mathbf{v} \, d\Omega - \frac{1}{2} \int_{\Omega} (\nabla \cdot (\mathbf{v}\mathbf{u})) \cdot \mathbf{u} \, d\Omega \\ &= \frac{1}{2} \int_{\Omega} \mathbf{u} \cdot \nabla \mathbf{u} \cdot \mathbf{v} \, d\Omega + \frac{1}{2} \int_{\Omega} \mathbf{u} \cdot \nabla \mathbf{u} \cdot \mathbf{v} \, d\Omega - \oint_{\partial\Omega} \mathbf{v}\mathbf{u}\mathbf{u} \cdot \mathbf{n} \, dS \\ &= \int_{\Omega} \mathbf{u} \cdot \nabla \mathbf{u} \cdot \mathbf{v} \, d\Omega \\ &= c(\mathbf{u}, \mathbf{u}, \mathbf{v}). \end{aligned}$$

□

### A.6.2 Alternate forms of the viscous term

$$\frac{1}{Re} \nabla^2 \mathbf{u}, \quad (\text{A.34})$$

$$\nabla \cdot \frac{1}{Re} \left( \nabla \mathbf{u} + (\nabla \mathbf{u})^\top \right), \quad (\text{A.35})$$

$$-\frac{1}{Re} \nabla \times \nabla \times \mathbf{u}, \quad (\text{A.36})$$

$$-\frac{1}{Re} (\nabla(\nabla \cdot \mathbf{u}) - \nabla \times \nabla \times \mathbf{u}), \quad (\text{A.37})$$

**Theorem A.3.** *If  $\nabla \cdot \mathbf{u} = 0$ , then (A.34)-(A.37) are equivalent.*

Note that the only form that is admissible to use if  $\nabla \cdot \mathbf{u} \neq 0$  is (A.34). The other three are only for incompressible flows.



## Appendix B

### Sponge layers

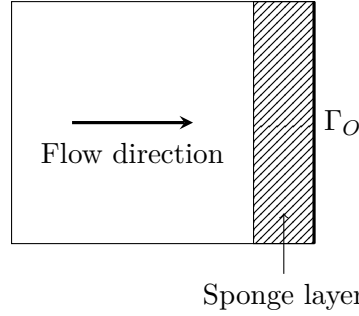


Figure B.1: A sponge layer can be added to the domain to damp out unwanted reflections of an outflow boundary condition.

In a sponge region (see Figure B.1) the idea is to damp out all unwanted reflections by adding extra forcing to the flow. Hence, the sponge region itself will destroy the physical correctness of the region it is applied to. The consequence of this is obviously that the solution in the sponge region cannot be used and this extra resolution must be removed from the solution. Thus, a sponge region results in extra computational costs. The sponge region treatment has been widely adopted because of its simplicity, robustness, non-stiff nature, and flexibility to handle complex geometries and unstructured grids [26].

The idea of a sponge region can be applied in many ways. We consider the way presented by in the manual by Chavallier et al. [6]. The implementation procedure follows: an additional volume force is added to the flow of the form

$$\mathbf{F} = \lambda(x)(\mathbf{u}_{\text{ref}} - \mathbf{u}), \quad (\text{B.1})$$

where  $\mathbf{u}_{\text{ref}}$  a known reference solution, and where  $\lambda(x)$  is a non-negative damping

function which is non-zero only within the sponge region (see Figure B.2),

$$\lambda(x) = \lambda_{\max} \left[ S \left( \frac{x - x_{\text{start}}}{\Delta_{\text{rise}}} \right) - S \left( \frac{x - x_{\text{end}}}{\Delta_{\text{fall}}} + 1 \right) \right], \quad (\text{B.2})$$

where  $\lambda_{\max}$  is the maximum strength of the damping,  $s_{\text{start}}$  to  $x_{\text{end}}$  the spatial length of the region and  $\Delta_{\text{rise}}$  and  $\Delta_{\text{fall}}$  are the rise and fall distances of the damping function, and the  $S$  function is a smooth step function given by

$$S(x) = \begin{cases} 0, & x \leq 0, \\ 1 / \left( 1 + \exp \left( \frac{1}{x-1} + \frac{1}{x} \right) \right), & 0 < x < 1, \\ 1, & x \geq 1. \end{cases} \quad (\text{B.3})$$

The aim is to construct the sponge such that  $\mathbf{u} \rightarrow \mathbf{u}_{\text{ref}}$  as the flow travels through the region. The damping function characterize the strength of the sponge by its length and amplitude. Larger sponges perform better than smaller sponges with the same strength since they damp flow less intense, but are more computationally expensive to use. Sponge design is a highly “trial-and-error” procedure, hence cost-optimal sponges are seldom used.

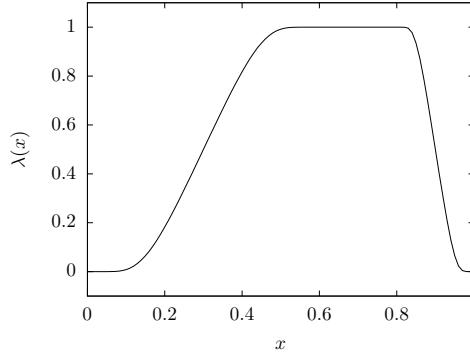


Figure B.2: A typical damping function with  $\lambda_{\max} = 1$ ,  $\Delta_{\text{rise}} = 0.6$  and  $\Delta_{\text{fall}} = 0.2$ .

## Appendix C

# Nek5000 spectral element solver

Nek5000 is a spectral element code based on the Nekton 2.0 project that was developed by Paul Fischer, Lee Ho, and Einar Rønquist in 1986-1991, in collaboration with two of the most known developers of the spectral element theory, Anthony Patera and Yvone Maday. The Nekton 2.0 code was the first spectral element code designed that could handle three dimensions and one of the first commercially available codes for distributed-memory parallel processors. Nek5000 is used by researchers all over the world, several people at KTH Mechanics included. The aim of this section is to give a short introduction.

### C.1 Features

- scales to over a million processes
- high-order spatial discretization using spectral elements
- high-order semi-implicit time-stepping
- incompressible + low Mach number (variable density) flows
- efficient preconditioners (multigrid + scalable coarse grid solves)
- highly optimized computational kernels (e.g. matrix-matrix multiply)
- low memory footprint and scalable memory design
- high performance parallel I/O
- ALE/moving meshes and free surface flow
- accurate Lagrangian particle tracking
- conjugate fluid-solid heat transfer
- scalable communication kernels

- build-in profiling analysis
- interface to VisIt<sup>1</sup> for parallel data analysis and visualization
- interface to MOAB for advanced meshing capabilities

See [16] for more information.

## C.2 Basic setup

Nek5000 is written in Fortran77 and C. It uses the Message Passing Interface (MPI) for parallelization. Thus, a Fortran77 and a C compiler has to be installed on a computer system to be able to compile the code. As a user, the specification of the desired problem is written into some specific user files. Examples of the files are given in the repository of the Nek5000 code, and can easily be used to suit the specific needs. There are three main files that have to be edited. The content of them are given below:

- **.rea** file (ascii or binary) :
  - **parameters**: viscosity, conductivity ( $\kappa$ ), number of steps, time-step size, order of the time-stepping, frequency of output, iteration tolerances, flow rate, filter strength, etc. + free parameters that can be used in the **.usr** file.

```

***** PARAMETERS *****
2.610000      NEKTON VERSION
2  DIMENSIONAL RUN
  103 PARAMETERS FOLLOW
    1.000000      p01 DENSITY
    1.000000      p02 VISCOS
    0.0000000E+00
    0.0000000E+00
    0.0000000E+00
    0.0000000E+00
    1.000000      p07 RHOCp
    1.000000      p08 CONDUCT
    0.0000000E+00
    0.0000000E+00
    5000.000      p10 FINTIME
    1.0000000E-03  p11 NSTEPS
    0.0000000E+00  p12 DT
    0.0000000E+00  p13 IOCOMM
    0.0000000E+00  p14 IOTIME
    500.0000      p15 IOSTEP
    0.0000000E+00  p16 PSSOLVER
    0.0000000E+00
    0.0000000E+00
    0.0000000E+00
    0.0000000E+00
    1.0000000E-08  p21 DIVERGENCE
    1.0000000E-12  p22 HELMHOLTZ
    0.0000000E+00  p23 NPSCAL
    1.0000000E-06  p24 TOLREL
    1.0000000E-10  p25 TOLABS
    0.3500000      p26 COURANT/NTAU
    3.000000      p27 TORDER

```

<sup>1</sup>VisIt is an Open Source, interactive, scalable, visualization, animation and analysis tool.  
<https://wci.llnl.gov/simulation/computer-codes/visit/>



## C.2. BASIC SETUP

- **logicals:** Steady or unsteady solution, advection on/off etc.

T	IFFLOW	solve for fluid (velocity , pressure)
T	IFHEAT	solve for heat (temperature and/or scalars)
T	IFTRAN	solve transient equations
T	IFADVC	specify the fields with convection
T	IFTMSH	specify the field(s) defined on T mesh
F	IFAXIS	axisymmetric formulation
F	IFSTRS	stress formulation in the incompressible case
T	IFLOMACH	low Mach number compressible flow
F	IFMGRID	moving grid (free surface flow)
F	IFMVBD	moving boundary (free surface flow)
F	IFCHAR	use characteristics for convection operator

- **geometry:** In 3d xyz locations of each of the eight points for each element, or the xy locations of each of the four points for each element in 2d.

ELEMENT		1	[	1a]	GROUP	0
0.0000000E+00	6.2500000E-02	6.2500000E-02	0.0000000E+00	6.2500000E-02	6.2500000E-02	6.2500000E-02
0.0000000E+00	0.0000000E+00	6.2500000E-02	6.2500000E-02	6.2500000E-02	6.2500000E-02	6.2500000E-02
ELEMENT		2	[	1b]	GROUP	0
6.2500000E-02	1.2500000E-01	1.2500000E-01	6.2500000E-02	6.2500000E-02	6.2500000E-02	6.2500000E-02
0.0000000E+00	0.0000000E+00	6.2500000E-02	6.2500000E-02	6.2500000E-02	6.2500000E-02	6.2500000E-02
...						

- **boundary conditions:** Specified as follow: P = Periodic, V = Dirichlet (velocity), v = user specified Dirichlet (in .usr file), O = Outflow (natural), E = inner points etc.

***** FLUID BOUNDARY CONDITIONS *****							
P	1	1	241.0000	3.000000	0.0E+00	0.0E+00	0.0E+00
E	1	2	2.000000	4.000000	0.0E+00	0.0E+00	0.0E+00
E	1	3	17.00000	1.000000	0.0E+00	0.0E+00	0.0E+00
P	1	4	16.00000	2.000000	0.0E+00	0.0E+00	0.0E+00
...							

- **restart conditions:** Specify a file to use as an initial condition.

1	PRESOLVE/RESTART OPTIONS	*****
	file0.f000XX	

- **output specifications:**

***** OUTPUT FIELD SPECIFICATION *****	
6	SPECIFICATIONS FOLLOW
T	COORDINATES
T	VELOCITY
T	PRESSURE
T	TEMPERATURE
F	TEMPERATURE GRADIENT
0	PASSIVE SCALARS

- **.usr file:** Contains Fortran subroutines to be specified by the user.

- **userchk():** interrogates the solution at the end of each time-step (diagnostic purposes). Calls all points once in each time step.

SUBROUTINE USERCHK
INCLUDE 'SIZE'
INCLUDE 'TOTAL'
RETURN
END

## APPENDIX C. NEK5000 SPECTRAL ELEMENT SOLVER

- **useric()**: specify the initial condition. Note that the condition for each of the velocity and temperature field has to be set separately.

```
SUBROUTINE USERIC (IX,IY,IZ,ieg)
INCLUDE 'SIZE'
INCLUDE 'NEKUSE'

UX = ..
UY = ..
UZ = ..
TEMP = ..
RETURN
END
```

- **userbc()**: user defined boundary conditions (v in **.rea** file). Calls the boundary points individually.

```
SUBROUTINE USERBC (IX,IY,IZ,ieg)
INCLUDE 'SIZE'
INCLUDE 'NEKUSE'

UX = ..
UY = ..
UZ = ..
TEMP = ..
RETURN
END
```

- **userf()**: user defined forcing function

```
SUBROUTINE USERF (IX,IY,IZ,ieg)
INCLUDE 'SIZE'
INCLUDE 'NEKUSE'

FFX = ..
FFY = ..
FFZ = ..
RETURN
END
```

- **SIZE** file:

- **ldim**: dimension (= 2 or 3).

```
parameter (ldim=2)
```

- **lx1**, **ly1**, **lz1**: order of the basis functions ( $N$ ). **lelt**: the maximum number of elements per processor.

```
parameter (lx1=12,ly1=lx1,lz1=1,lelt=80,lev=lelt)
```

- **lxd**: polynomial order of integration for convective terms ( $l_{cd}=3*lx1/2$ ).

```
parameter (lxd=(lx1*3)/2,lyd=lxd,lzd=1)
```

- **lx2**, **ly2**:  $\mathbb{P}_N \times \mathbb{P}_N$  (=lx1,ly1) or  $\mathbb{P}_N \times \mathbb{P}_{N-2}$  (=lx1-2,ly1-2).

```
parameter (lx2=lx1-2) !Pn x Pn-2
parameter (ly2=ly1-2) !Pn x Pn-2
parameter (lz2=lz1)
```

# Bibliography

- [1] Top500, the list. <http://www.top500.org>.
- [2] *Flow Around Circular Cylinders: Volume I: Fundamentals*. Flow Around Circular Cylinders: A Comprehensive Guide Through Flow Phenomena, Experiments, Applications, Mathematical Models, and Computer Simulations. OUP Oxford, 1997.
- [3] I. Babuška. The finite element method with lagrangian multipliers. *Numerische Mathematik*, 20(3):179–192, 1973.
- [4] J.P. Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of Computational Physics*, 114(2):185 – 200, 1994.
- [5] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 8(R2):129–151, 1974.
- [6] M. Chavallier, P. Schlatter, Lundblad, and D.S. A. Henningsson. *Simpson, a pseudo-spectral solver for incompressible boundary layer flows*. 2007.
- [7] T. Colonius. Modeling artificial boundary conditions for compressible flow. *Annual Review of Fluid Mechanics*, 36(1):315–345, 2004.
- [8] W. Couzy. Spectral element discretization of the unsteady navier-stokes equations and its iterative solution on parallel computers. 1995.
- [9] M.O. Deville, P.F. Fischer, and E.H. Mund. *High-Order Methods for Incompressible Fluid Flow*, volume 9. Cambridge University Press, 2002.
- [10] S. Dong, G.E. Karniadakis, and C. Chrysosostomidis. A robust and accurate outflow boundary condition for incompressible flow simulations on severely-truncated unbounded domains. *Journal of Computational Physics*, 2014.
- [11] B. Engquist and A. Majda. Radiation boundary conditions for acoustic and elastic wave calculations. *Communications on Pure and Applied Mathematics*, 32(3):314–358, 1979. NR 20140805.

## BIBLIOGRAPHY

- [12] B. Engquist and A. Majda. Absorbing boundary conditions for the numerical simulation of waves. *Mathematics of Computation*, 1:629–651, 1997. NR 20140805.
- [13] J.H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics*. Springer Berlin Heidelberg, 2001.
- [14] R.P. Feynman, R.B. Leighton, and M.L. Sands. *The Feynman Lectures on Physics*. Number v. 1 in The Feynman Lectures on Physics. Addison-Wesley, 1963.
- [15] P.F. Fischer, F. Loth, S.W. Lee, D Smith, H.M. Tufo, and H.S. Bassiouny. Parallel simulation of high reynolds number vascular flows. 02/2006 2006.
- [16] P.F Fischer, J.W. Lottes, and S.G. Kerkemeier. nek5000 web page, 2008. <http://nek5000.mcs.anl.gov>.
- [17] J. G. Heywood, R. Rannacher, and S. Turek. Artificial boundaries and flux and pressure cconditions for the incompressible navier–stokes equations. *International Journal for Numerical Methods in Fluids*, 22(5):325–352, 1996.
- [18] G. Karniadakis and S. Orszag. Nodes, modes and flow codes. *Physics Today*, 46(3):34–42, 1993.
- [19] G.E. Karniadakis, M. Israeli, and S. Orszag. High-order splitting methods for the incompressible navier-stokes equations. *Journal of Computational Physics*, 97(2):414 – 443, 1991.
- [20] E. Åkervik, L. Brandt, D.S. Henningson, J. Hoepffner, O. Marxen, and P. Schlatter. Steady solutions of the navier-stokes equations by selective frequency damping. *Physics of Fluids (1994-present)*, 18(6):–, 2006.
- [21] L.I.G. Kovasznay. Laminar flow behind a two-dimensional grid. *Mathematical Proceedings of the Cambridge Philosophical Society*, 44:58–62, 1 1948.
- [22] L. Leray. On the motion of a viscous liquid filling space. *Acta mathematica.*, 1934.
- [23] J.D. Logan. *Applied Mathematics*. A Wiley-Interscience publication. Wiley, 1997.
- [24] Y. Maday and A.T. Patera. Spectral element methods for the incompressible Navier-Stokes equations. In *State-of-the-art surveys on computational mechanics (A90-47176 21-64)*. New York, American Society of Mechanical Engineers, 1989, p. 71-143. Research supported by DARPA., pages 71–143, 1989.
- [25] Y. Maday, A.T. Patera, and E.M. Rø nquist. An operator-integration-factor splitting method for time-dependent problems: Application to incompressible fluid flow. *Journal of Scientific Computing*, 5(4):263–292, 1990.

## BIBLIOGRAPHY

- [26] A. Mani. Analysis and optimization of numerical sponge layers as a nonreflective boundary treatment. *Journal of Computational Physics*, 231(2):704 – 716, 2012.
- [27] T. Miyauchi, M. Tanahashi, and M. Suzuki. Inflow and outflow boundary conditions for direct numerical simulations. *Transactions of the Japan Society of Mechanical Engineers Series B*, 60(571):813–821, 1994.
- [28] G. E. Moore. Cramming More Components onto Integrated Circuits. *Electronics*, 38(8):114–117, April 1965.
- [29] I. Orlanski. A simple boundary condition for unbounded hyperbolic flows. *Journal of Computational Physics*, 21(3):251 – 269, 1976.
- [30] S. Orszag. Analytical theories of turbulence. *Journal of Fluid Mechanics*, 41:363–386, 4 1970.
- [31] E.D. Palma and R.P. Matano. On the implementation of passive open boundary conditions for a general circulation model: The barotropic mode. *Journal of Geophysical Research: Oceans*, 103(C1):1319–1341, 1998.
- [32] A.T. Patera. A spectral element method for fluid dynamics: Laminar flow in a channel expansion. *Journal of Computational Physics*, 54(3):468 – 488, 1984.
- [33] T.J. Poinso and S.K. Lele. Boundary conditions for direct simulations of compressible viscous flows. *Journal of Computational Physics*, 101(1):104 – 129, 1992.
- [34] A. Poux, S. Glockner, and M. Azañez. Improvements on open and traction boundary conditions for navier–stokes time-splitting methods. *Journal of Computational Physics*, 230(10):4011 – 4027, 2011.
- [35] R. Rannacher. Methods for numerical flow simulation. In *Hemodynamical Flows*, volume 37 of *Oberwolfach Seminars*, pages 275–332. Birkhäuser Basel, 2008.
- [36] R. L. Sani and P. M. Gresho. Résumé and remarks on the open boundary condition minisymposium. *International Journal for Numerical Methods in Fluids*, 18(10):983–1008, 1994.
- [37] P. Schlatter. Personal communication.
- [38] P. Schlatter, Q. Li, G. Brethouwer, A.V. Johansson, and D.S. Henningson. Simulations of spatially evolving turbulent boundary layers up to  $Re_\theta = 4300$ . *International Journal of Heat and Fluid Flow*, 31(3):251–261, 2010.
- [39] S.J. Sherwin and G.E. Karniadakis. A triangular spectral element method; applications to the incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 123(1–4):189 – 229, 1995.

## BIBLIOGRAPHY

- [40] A. Sommerfeld. *Partial differential equations in physics*. Pure and Applied Mathematics. Elsevier Science, 1949.
- [41] A.G. Tomboulides, J.C.Y. Lee, and S. Orszag. Numerical simulation of low mach number reactive flows. *Journal of Scientific Computing*, 12(2):139–167, 1997.
- [42] L.N. Trefethen and L. Halpern. Well-posedness of one-way wave equations and absorbing boundary conditions. *Mathematics of Computation*, 47(176):pp. 421–435, 1986.
- [43] R. Vichnevetsky and E.C. Pariser. High order numerical sommerfeld boundary conditions: Theory and experiments. *Computers & Mathematics with Applications*, 11(1–3):67 – 78, 1985.
- [44] C.J. Xu and Y.M. Lin. A numerical comparison of outflow boundary conditions for spectral element simulations of incompressible flows. *Commun. Comput. Phys.*, 2:477–500, 2007.



TRITA -MAT-E 2015:11  
ISRN -KTH/MAT/E-15/11--SE