# Turbulent drag reduction through wall-forcing methods

A Thesis submitted to the University of Sheffield

for the degree of Doctor of Philosophy

by

Sohrab Khosh Aghdam

Department of Mechanical Engineering

The University of Sheffield

# Contents

# Acknowledgements

I hereby thank Airbus Group and the Department of Mechanical Engineering of the University of Sheffield for funding this PhD project. The resources provided by the ARCHER UK National Supercomputing Service under EPSRC Grant EP/K000225/1 and the computing facilities of N8 HPC funded by the N8 consortium, coordinated by the Universities of Leeds and Manchester are gratefully acknowledged.

I would like to warmly thank my supervisor Dr Pierre Ricco for his guidance, help and support all along this project. I am indebted to Dr Sylvain Laizet (Imperial College London) for introducing me to the Incompact3d code, for his patience, suggestions and comments. I also thank him and Professor Shuisheng He (University of Sheffield) for accepting to be part of the PhD thesis panel. A special mention goes to Dr Mehdi Seddighi (Liverpool John Moores University) for being more than a DNS expert, a true friend.

Last but not least, I am eternally grateful to my family for their unconditional love, support and help at all levels.

# Abstract

The constraints brought about by environmental and economical issues have been key elements for devising new techniques for skin-friction drag reduction in turbulent flows. Several methodologies have been applied during the last thirty years. These methods can be categorised as active, passive, closed or open-loop. In general, these techniques are mathematically modelled, then tested in experimental settings and numerical simulations. The numerical model for this study was based on the resolution of the full spatio-temporal scales through Direct Numerical Simulation (DNS). With the advent of powerful high-end computing systems endowed with several thousands of processors and relying on distributed memory programming, the performance deadlock due to highly resolved DNS is progressively being overcome. To study in a first principal basis a flow, DNS based on an efficient flow solver called Incompact3d  has been relied on more particularly focusing on the development of a large array of flow control techniques.

Motivated by extensive discussion in the literature, by experimental evidence and by recent direct numerical simulations, we study flows over hydrophobic surfaces with shear-dependent slip lengths and we report their drag-reduction properties. The laminar channel-flow and pipe-flow solutions are derived and the effects of hydrophobicity are quantified by the decrease of the streamwise pressure gradient for constant mass flow rate and by the increase of the mass flow rate for constant streamwise pressure gradient. The nonlinear Lyapunov stability analysis is employed on the three-dimensional channel flow with walls featuring shear-dependent slip lengths. The feedback law extracted through the stability analysis is recognized for the first time to coincide with the slip-length model used to represent the hydrophobic surfaces, thereby providing a precise physical interpretation for the feedback law advanced by Balogh *et al.* (2001). The theoretical framework by K. Fukagata, N. Kasagi, and P. Koumoutsakos is employed to model the drag-reduction effect engendered by the shear-dependent slip-length surfaces and the theoretical drag-reduction values are in very good agreement with our direct numerical simulation data. The turbulent drag reduction is measured as a function of the hydrophobic-surface parameters and is found to be a function of the time- and space-averaged slip length, irrespectively of the local and instantaneous slip behaviour at the wall. For slip parameters and flow conditions that could be realized in the laboratory, the maximum computed turbulent drag reduction is 50% and the drag reduction effect degrades when slip along the spanwise direction is

considered. The power spent by the turbulent flow on the hydrophobic walls is computed for the first time and is found to be a non-negligible portion of the power saved through drag reduction, thereby recognizing the hydrophobic surfaces as a passive-absorbing drag-reduction method. The turbulent flow is further investigated through flow visualizations and statistics of the relevant quantities, such as vorticity and strain rates. When rescaled in drag-reduction viscous units, the streamwise vortices over the hydrophobic surface are strongly altered, while the low-speed streaks maintain their characteristic spanwise spacing. We finally show that the reduction of vortex stretching and enstrophy production is primarily caused by the eigenvectors of the strain rate tensor orienting perpendicularly to the vorticity vector.

In a second phase, several drag-reduction techniques were implemented and benchmarked. This step was motivated by the drag-reducing potential benefits of combined active-active and active-passive techniques compared to those taken separately. With this objective in mind, three control techniques were selected and categorized as primary and secondary. The primary control method consisted in an array of steady rotating discs or rings embedded at the walls of the channel flow. The secondary methods consisting of opposition control or constant-slip hydrophobic surfaces were used to complement the primary one. It was found that the combination of the the combination of these techniques did not result in the sum of the contributions of each technique taken separately.

In addition to these studies and developments within the code, various techniques for analysing the results have been implemented and used which are presenting a novel aspect for the within the flow control area: probabilistic and eigenvalue methods. All these methods are now part of a full-fledge revised version of the code and can be used in parallel. An extensive guide has also been written for future users of the code for flow control problems.

# List of listings

# Chapter 1

# Introduction

## 1.1 Motivation

Energy consumption issues related to transportation have gained major interest in recent years due to the ever increasing demand of emerging markets and the cost in the exploitation of untapped resources. Therefore the design and prototyping of transportation, either for aircrafts, marine vessels or cars among others, have become crucial to reduce the costs in fuel consumption thereby reducing the impact of greenhouse gases (such as $CO_2$) on the environment. The 2014 report of the IPCC (IPCC, 2014) shows that based on the cumulative total anthropogenic $CO_2$ emissions from 1870, the relative change in temperature could rise from 1.8 to to $4.7^o$C depending on the considered scenario. According to the recent report by Airbus (2014) for the period 2015-2034, the air traffic will double in the next 15 years, while in the same period, the oil price is expected to get threefold. The data therein shows that in the period from 2000 to 2014 fuel consumption (in kg per 100 RPKs (Revenue Passenger Kilometres)) has decreased in average by 33%. However, fuel cost remains the dominant spending for airline companies. Therefore, the design of fuel-efficient aircrafts is of paramount importance. A wide range of possible paths can be chosen to achieve such a goal. The main rely on the formulation of more efficient fuel types, the improvement of engine cycles, the reduction of aircraft weight by means of new composite materials and the reduction of drag.

With the objective of achieving drag reduction, active and passive flow control strategies are studied by means of mathematical analysis and numerical implementation. The passive technique based on hydrophobic surfaces is tackled though Lyapunov stability analysis. Based on fairly recent progress on these surfaces, a shear-dependent slip length is modelled and its drag-reduction properties are reported. The laminar channel-flow solution is found and the effects of hydrophobicity are quantified by the decrease of streamwise pressure gradient for constant mass flow rate and by the increase of mass flow rate for constant streamwise pressure gradient. The Lyapunov stability analysis, first derived for a two-dimensional channel flow in Balogh *et al.* (2001), is applied on the three-dimensional (3D) turbulent channel flow and extended to the shear-dependent slip-length case. The feedback law extracted through the stability analysis is recognised for the first time to coincide with the simple constant-slip-length model widely used in simulations of wall-bounded turbulent flows over hydrophobic surfaces. The theoretical approach by Fukagata *et al.* (2006) is employed to model the drag-reduction effect engendered by the shear-dependent slip-length surfaces and the estimated drag-reduction.

Then, the combination of active/passive flow control techniques is assessed focusing more particularly on the drag-reduction apparatus based on rotating discs (Ricco & Hahn, 2013). The latter imposed as a primary type of control is combined either with opposition control (active) or hydrophobic surfaces (passive) techniques to evaluate their drag reduction complementary capability. The secondary control techniques both present the advantage of having large gain values, making them relevant candidates for being combined with the primary active control technique. In this way, as the drag-reduction rate is increased without requiring additional energy input, the net power obtained is increased.

Besides the aforementioned tasks to be performed in this research project, a major point was to transform a code that had never been used as a flow control tool. This involved not only several steps in terms of benchmarking the code but also to design it for a straightforward use for future research, through the possibilities offered in terms of flow control techniques but also for efficient analysis.

## 1.2 Flow control

The control of flow aims at modifying the characteristics of a flow field favourably. By following this objective, benefits could be substantial in terms of operational systems' cost. The modern progress in this area focuses on either counteracting or enhancing turbulence occurrence. Various techniques are being employed for instance to reduce turbulence effects in transportation systems thereby reducing drag. In this study, the focus is given to flow control techniques aiming at reducing drag. However, one has to keep in mind that flow control techniques can also be used to enhance mixing in combustion processes or in heat transfer systems. To successfully control the flow, the physics at play in a turbulent flow along with a robust control algorithm have to be analyzed. Nowadays, it is well-known in fluid dynamics that the skin-friction drag in turbulent boundary layers reaches higher values than in laminar boundary layers. Until the in-depth explanation of this phenomenon and the underlying physics related to regions of high skin-friction drag, very early attempts in flow control, whether in numerics or experiments, were based on trial and error. However, mathematical concepts in control theories have been used as a first principle tool and a guide in the design of the controllers. Direct numerical simulation (DNS) and large-eddy simulation (LES), although being confined to simpler configuration and reasonably low Reynolds numbers have been able to give insight in the underlying turbulent flow dynamics. These tools have allowed to reach a significant understanding and in the design of flow controllers. Therefore it is the combination of computational fluid dynamics (CFD), mathematics of control theories, along with sensor/actuator technologies that allow to fully tackle flow control problems.

The phenomenon of drag reduction was reported for the first time by Toms (1948), who noticed the effect produced by polymer addition in fluids. While performing experiments on the degradation of polymers, he observed that the addition of few ppm of long chain polymers in a turbulent flow produced a dramatic reduction of the friction drag. The dissipation of kinetic energy by turbulent flow around objects (airplanes, ships,$\cdots$) increases resistance to their motion and is what gives a very first approach to the definition of drag. It is conventionally equal to the component of the force experienced by a body, parallel to the direction of motion. Drag can be subdivided into various components which contribute to the overall force on an object as a result of fluid interactions. These components include form drag, interference drag, wave drag, and skin friction. Skin friction drag is the result of friction within the shear layer at the solid/fluid interface and represents the largest contribution to the overall drag in transportation systems. Indeed skin-friction drag accounts for about 50% of the overall drag experienced by commercial aircrafts (Gad-el Hak, 1994). This quantity increases even more for marine vessels and pipelines to respectively reach 90% and 100%. It becomes clear that even a small decrease in skin-friction drag will beneficially impact the amount of consumed fuel, thereby cutting costs the aforementioned transportation systems. For instance to reduce air resistance Lufthansa Technik carried out a two-year length test based on technically simulated sharkskin structures. The setup consisted in attaching ten 10cm × 10cm patches to the fuselage and leading edge of the wings of two Airbus A340-300. These tests aimed at assessing the durability of such surfaces under real flight conditions. From the preliminary results, it was estimated that this techniques would be able to save up to 1% kerosene per flight. Earlier experiments also provided results by covering the fuselage of airplanes (Ball, 1999). The ribbed texture parallel to the flow direction and mimicking shark skin proved to be aerodynamically efficient. The drag could be reduced up to 8% equivalent to about 1.5% in fuel savings. Scaled to the air-transport traffic of the United States based on Airbus A320, an estimated savings of $200 million per year could potentially be achieved (Ikeda, 2007).

As turbulence is intrinsically a multiscale phenomenon, the design of control devices needs to account for the coupling between macroscopic flow structures, related to the largest characteristic size of the system, and the small scales, such as the Taylor and Kolmogorov scales, generated by the nonlinear cascade linked to vortex stretching. Therefore the design of controllers requires appropriate sensors and actuators. This has been a critical issue for turbulence control, boundary-layer control in particular, because the time and length scales associated with the turbulent eddies to be controlled are extremely small at the Reynolds numbers of engineering applications, thus requiring a large number of small sensors and actuators with high-frequency response (Yoshino *et al.*, 2008). The advent of Micro-electro-mechanical systems (MEMS) starts to play a crucial role in the design of large scale arrays of sensors/actuators. A thorough review discussing the performance and trade-offs in the design of MEMS can be found in Bell *et al.* (2005). For a general review on these devices the reader can refer to Gad-el-Hak (2001).

Very early visualizations by Kline *et al.* (1967), revealed the presence of intermittent streaky structures. They noticed that some of the these streaks with low streamwise momentum would essentially lift away from the wall allowing them to interact with the outer flow. Subsequently this process was called "bursting" as it is was observed to be a rapid sequence during which a large quantity of turbulence production was involved. These bursts are transferring energy from the large to the small scales and are turbulence production

generators stepping in the sweep/ejections cycle.
Turbulent flows are a combination of organized (coherent) structures and seemingly disorganized (incoherent) structures. In the turbulent boundary layer (TBL) within the wall region, coherent structures represent about 50-80% of the energy in the turbulent fluctuations (Corino & Brodkey, 1977). During the past 60 years a fairly large amount of knowledge has been accumulated on turbulent coherent structures. Quasi-streamwise vortices (QSVs) have become a well-known key player in near-wall turbulent transport phenomena (Robinson, 1991*b*; Kline *et al.*, 1967). The peculiar features relative to these structures in the near-wall region are the so-called low- and high-speed streaks which consist of a spanwise modulation of the streamwise velocity. Jeong *et al.* (1997) have given the underlying concepts relative to these near-wall coherent structures. Thus, these velocity signals are actually generated by pairs of counter-rotating vortices aligned in the streamwise direction. These vortices are primarily found in the buffer layer (10-50 wall units) with a characteristic diameter of diameter of 20-50 wall units. It was shown as well that the tilting of QSVs in the $z$ direction is intimately related to the twisting of low-speed streaks, implying a central role in the regeneration mechanism (Hamilton *et al.*, 1995). The DNS data analysed by Kravchenko *et al.* (1993) devised the clear relationship between the QSVs and high skin-friction regions, by computing their correlation as also referenced in Bernard *et al.* (1993*b*,*a*). The overall image is as follows: slowly-moving fluid is lifted away from the wall on the ejection side of a vortex, giving rise to a low-speed streak, while a high-speed streak is being produced by a sweep on the opposite side. Further details related to the self-sustaining process of these structures can be found in Panton (2001); J. (2011).

The mechanisms in near-wall turbulence as previously described are clearly intermittent both in time and space. The great challenge in flow control resides in the fact that time and space scales are spanning over a large range. Therefore in order to obtain a potential benefit (i.e. reduction of turbulent kinetic energy production and skin friction) one has to focus on mitigating these coherent structures and the self-sustaining process of QSVs generation, by properly selecting the spatial and temporal ranges. One of the most clear technique within these lines is opposition control (Choi *et al.*, 1994). Indeed, this active cancellation scheme enables to reduce drag in a fully-developed turbulent flow by damping the effect of the near-wall vortices. By opposing the near-wall motions of the fluid, the high shear region are lifted away from the wall.

Various drag-reduction methods have been investigated. A first group includes those acting on the entirety or a portion of the flow domain. Such examples are either based on Lorentz force (Pang *et al.*, 2004), plasma generated by Dielectric Barrier Discharge (DBD) actuators (Choi *et al.*, 2014), or polymer-based solutions (Virk *et al.*, 1967; Virk, 1975). A second group of drag reduction techniques focuses on the modification of the wall-boundary conditions. For active methods, the wall actuation is such that energy is injected into the system (Choi *et al.*, 1994; Jung *et al.*, 1992). Passive methods instead do not involve any external energy input. Such techniques include riblets (Choi *et al.*, 1993), compliant walls (Hahn *et al.*, 2002), and hydrophobic surfaces (Min & Kim, 2004).

The control strategies categorisation is based on whether energy is provided to the system under control (active) or not (passive) (Gad-el Hak, 2000). One can cite the following methods as examples of active techniques: blowing and suction jets, wall-normal transpiration, external additives, geometry variation (riblets), and wall motion. For instance, macro-molecules present in surface treatment chemicals can significantly reduce the wall-shear stress in liquids. This process can be considered to be active. The molecules damp the energy in the buffer layer by acting like springs, hence reducing momentum transport.

Based on the control loops, active flow control can further be classified into open-loop (predetermined control) and closed-loop (interactive control). Predetermined control introduces the steady and unsteady energy inputs without consideration for the state of the flow field. Within interactive control framework, the controller aims at adjusting the power by a feedback sensor. The sensors are used to measure a signal to be transmitted to a controller which subsequently determines the required actuation. Such type of flow control techniques can be considered to be the most relevant as these continually react to the intermittent events occurring in the flow. However, they are generally more expensive to implement either in experiments or numerical simulations, thereby explaining the widespread use of predetermined flow control. On the other hand passive techniques present the advantage of simplicity. Such methods are usually based on the modification of the wall topology. Several techniques based on this paradigm have used patterns found in Nature as basis, e.g. shark skin for the design of riblets or the lotus leaf for superhydrophobic surfaces. Figure 1.1 sketches a summary of the different control techniques categories. As a first part of our research we have focused on the hydrophobic

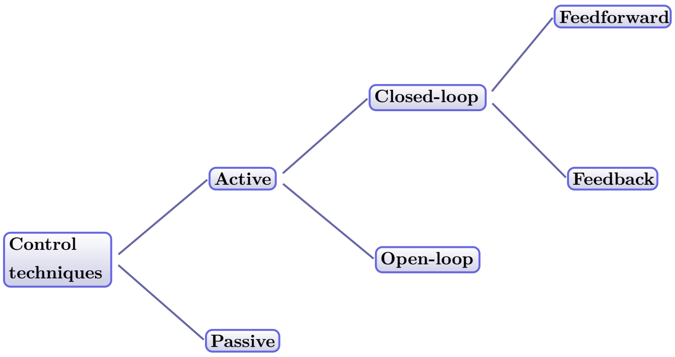


Figure 1.1: Categorization of flow control techniques.

surfaces. Their main characteristic is a finite effective slip velocity at the wall (Philip, 1972). These surfaces may achieve drag reduction for both laminar and turbulent flows (Watanabe *et al.*, 1999; Min & Kim, 2004), delay the transition to turbulence (Min & Kim, 2005), and operate over a wide range of Reynolds numbers relevant for technological applications, such as flows over marine vessels (Fukagata *et al.*, 2006). In particular, we are motivated by recent experimental and numerical research works which suggest that the characteristic slip length of the wall velocity is a function of the wall-shear stress (Lauga & Stone, 2003; Min & Kim, 2004; Choi & Kim, 2006; Busse & Sandham, 2012). The crucial observation is that this dependency is likely to be true especially for liquid turbulent flows past hydrophobic surfaces because they exert shear stresses which are much larger than in the laminar regime.
An active drag-reduction method, i.e. flush-mounted steady rotating discs, was first suggested by Keefe (1997) and numerically studied by Ricco & Hahn (2013). In this specific configuration, forcing is no longer based on velocity but on vorticity. Instead of directly involving velocity to modify the overall behaviour of the system, vorticity is injected through an array of wall flush-mounted rotating discs. Figure 1.2 shows such a system configuration.

On each plane of the channel flow, rotating discs are configured as follows: the rotation of the discs along the streamwise direction is alternatively clockwise and counter-clockwise. In this way, a series of discs in the spanwise direction have all the same rotating orientation. The combination of these rotating discs gives rise to a wave of spanwise velocity (in the centreline). The results indicate that for a fixed tip velocity (defined as $W = \Omega D/2$,

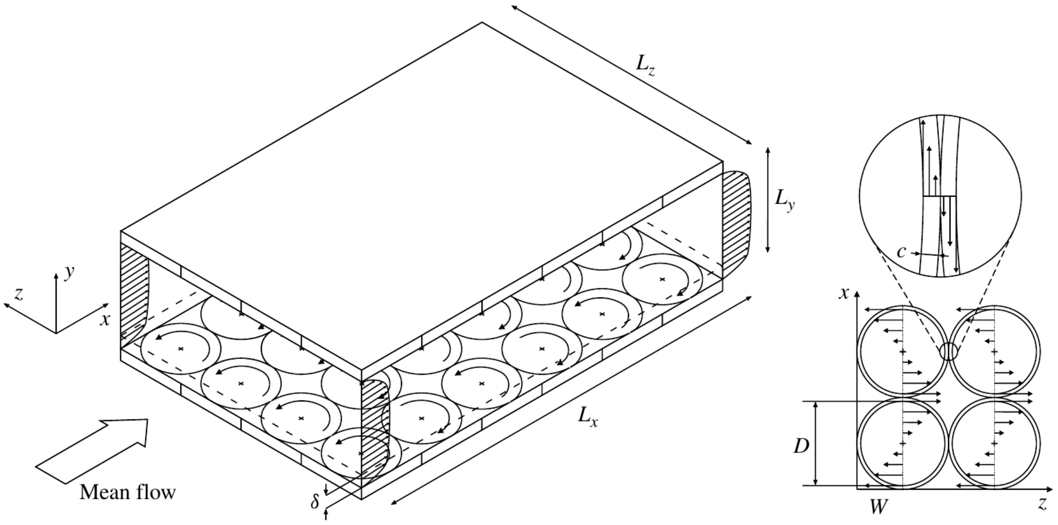


Figure 1.2: Rotating disc flow configuration (Ricco & Hahn, 2013).

where $\Omega$ is the constant angular velocity and $D$ the disc diameter), the transition from drag-increase to drag-reduction occurs at a given threshold. For fixed disc diameter, drag reduction is an increasing function of $W$ up to an optimum value. The simulations showed a significant difference between the fixed wall case and the rotating discs case. Indeed, in the latter configuration, the low-speed streaks have mostly vanished in a large region surrounding the centre of the disc. At the boundary between two discs, these streaks are still dominating. A more accurate expression for the drag reduction was derived, adapting the FIK identity Fukagata *et al.* (2002) to the particular decomposition of the flow. This new expression then accounts for the $x$-stretched structures mentioned above.

## 1.3 Numerical simulations

The flow control field greatly benefited from the advent of numerical methods and the use of DNS (Kim *et al.*, 1987*a*; Moin & Mahesh, 1998) to tackle these type of problems. Among CFD methods used, DNS provides high-fidelity solutions for turbulent flows. DNS is somehow different from other CFD techniques as it does not assume any modelling either in terms of equations closure, as is the case for Reynolds-averaged Navier-Stokes (RANS) or in terms of the captured scales as in LES and other sub-grid scale models. In this way, all the flow information is obtained at all times and scales making this tool an ideal candidate for ab-initio studies in turbulence. However, this comes at the expense of tremendous computational resource requirements for resolving turbulent three-dimensional flows which is intimately related to the Reynolds number. Indeed, as a very basic estimate, based on isotropic turbulence simulations, the number of grid points necessary in the simulations of the time-dependent three dimensional NSE scales as a power law of the Reynolds number (Piomelli & Balaras, 2002), thereby limiting its use for practical applications. However, the deadlock imposed by the Reynolds number constraint is progressively but still slowly being overcome with exascale capable supercomputers (Lee & Moser, 2015; Bernardini *et al.*, 2014) at least in very simple geometries. Indeed tackling complex flow geometries is also another challenge in addition to the Reynolds number constraint which narrows the use of DNS for practical large-scale problems.

The principal tool considered for implementing the flow control strategies relies on the DNS of a relatively low Reynolds number channel flow using the robust and efficient code called Incompact3d (Laizet & Lamballais, 2009; Laizet & Li, 2011). This code did not include any flow control capability to carry out relevant simulations within this framework. Therefore several flow control strategies had to be implemented and benchmarked to ensure the proper behaviour of the code. Although Incompact3d code is massively parallelized (as it has recently been tested up to a million cores) allowing for the use of high Reynolds number, it has been used at relatively low Reynolds numbers to quantify the potential benefits from these methods whether used on their own or combined with each other. In this manner, a compromise can be reached on the number of simulations for the testing phases, the amount of data used, the queuing period of the jobs as well as their runtime.

# Chapter 2

# Mathematical tools and conventions

In this chapter all the necessary technical tools and definitions are described. This will enable the reader to straightforwardly follow the subsequent chapters and refer to the present chapter as a guide to any technical part. First the definitions and normalizations relative to the flow are exhibited. After this step, the mathematical description of the Lyapunov stability analysis is devised.

## 2.1 Fundamental equations, flow decomposition and definitions

The equations and configuration used in this project are described in the following. The rigorous mathematical framework to study turbulence is based on the continuity and the Navier-Stokes equations (NSE) (Navier, 1823; Stokes, 1851)

$$\boldsymbol{\nabla} \cdot \mathbf{U}^* = 0, \tag{2.1}$$

$$\mathbf{U}_t^* + \mathbf{U}^* \cdot \boldsymbol{\nabla}\mathbf{U}^* = -\frac{1}{\rho^*}\boldsymbol{\nabla}P^* + \nu^*\Delta\mathbf{U}^*. \tag{2.2}$$

where $\mathbf{U}^*$ is the velocity vector which, in general, depends on all three spatial coordinates $(x^*, y^*, z^*)$, $P^*$ is the pressure and $\Delta$ is the Laplace operator. Dimensional quantities are marked by the superscript $*$. The dimensionless parameter, a.k.a. the Reynolds number (Reynolds, 1883) is denoted by $Re$ and expressed as

$$Re = \frac{\rho^*\mathcal{L}^*\mathcal{V}^*}{\mu^*}. \tag{2.3}$$

where $\mu^*$ is the dynamic viscosity of the fluid, $\mathcal{V}^*$ a characteristic velocity and $\mathcal{L}^*$ a typical length scale based on the size of the physical system e.g. the radius of a pipe through which the fluid is flowing. As in this study we will focus on the numerical investigation of

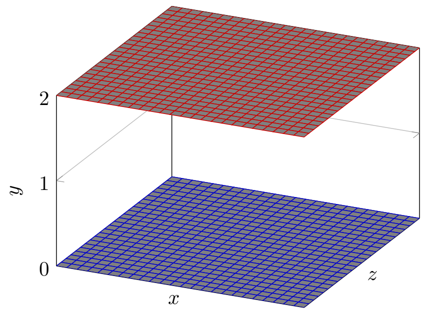


Figure 2.1: Channel flow configuration.

turbulent phenomena within a channel flow, as sketched in figure 6.2, the NSE are scaled by quantities relevant to this particular geometry. Lengths are scaled by the channel half-height $h^*$, velocities by the maximum Poiseuille velocity $U_p^*$ with rigid walls, and the time $t^*$ by $h^*/U_p^*$. Quantities non-dimensionalized through these outer units are not indicated by any symbol. The outer-flow Reynolds number is defined as $R_p = U_p^* h^*/\nu^*$, where $\nu^*$ is the kinematic viscosity of the fluid. The streamwise, wall-normal, and spanwise directions are $x$, $y$, and $z$, respectively, and $y \in [0, 2]$. Within this framework, the velocity vector field is defined as $\mathbf{W} = (U(x,y,z,t), V(x,y,z,t), W(x,y,z,t))$ and the pressure is $P(x,y,z,t)$. The velocity and the pressure thereby satisfy the incompressible continuity and NSE.

$$\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} + \frac{\partial W}{\partial z} = 0, \tag{2.4}$$

$$\frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial y} + W\frac{\partial U}{\partial z} = -\frac{\partial P}{\partial x} + \frac{1}{R_p}\left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2}\right), \tag{2.5}$$

$$\frac{\partial V}{\partial t} + U\frac{\partial V}{\partial x} + V\frac{\partial V}{\partial y} + W\frac{\partial V}{\partial z} = -\frac{\partial P}{\partial y} + \frac{1}{R_p}\left(\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2}\right), \tag{2.6}$$

$$\frac{\partial W}{\partial t} + U\frac{\partial W}{\partial x} + V\frac{\partial W}{\partial y} + W\frac{\partial W}{\partial z} = -\frac{\partial P}{\partial z} + \frac{1}{R_p}\left(\frac{\partial^2 W}{\partial x^2} + \frac{\partial^2 W}{\partial y^2} + \frac{\partial^2 W}{\partial z^2}\right). \tag{2.7}$$

For the description of the notations, the adopted configuration is based on 6.2.
The turbulent flow is decomposed into a mean and a fluctuating component.

$$(U, V, W) = (\mathcal{U}(y), 0, 0) + (u', v', w') \tag{2.8}$$

where the mean streamwise flow is

$$\mathcal{U}(y) = (L_x L_z)^{-1} \left[\overline{U}\right]_{\mathcal{I}xz} \tag{2.9}$$

with the space-average operator in the periodic directions defined as

$$[\cdot]_{\mathcal{I}xz} = \int\limits_0^{L_z} \int\limits_0^{L_x} \cdot \; \mathrm{d}x\mathrm{d}z \tag{2.10}$$

and the time averaging operator given by

$$\bar{\cdot} = \frac{1}{t_f - t_i} \int_{t_i}^{t_f} \cdot \; \mathrm{d}t \tag{2.11}$$

where $t_i$ and $t_f$ are the initial and final times defining the interval for the time averaging. Hereinafter the wall-normal coordinate $y$ has its zero at the bottom channel wall. The bulk velocity is introduced

$$\mathcal{U}_b = \frac{1}{2h} \int_0^{2h} U(y)\mathrm{d}y \tag{2.12}$$

The skin-friction coefficient is defined as usual,

$$C_f = \frac{2\nu^*}{\mathcal{U}_b^{*2}} \left.\frac{\mathrm{d}\mathcal{U}^*}{\mathrm{d}y^*}\right|_{y=0} = \frac{2}{R_p \mathcal{U}_b^2} \left.\frac{\mathrm{d}\mathcal{U}}{\mathrm{d}y}\right|_{y=0} \tag{2.13}$$

where the turbulent bulk velocity is obtained by replacing $\mathcal{U}$ for $U$ in (5.5). Unless otherwise specified, the notation $y = 0$ hereinafter indicates a quantity averaged over the two walls. The turbulent drag reduction $\mathcal{R}$ is

$$\mathcal{R}(\%) = 100 \left(1 - \frac{C_f}{C_{f,r}}\right) \tag{2.14}$$

where the subscript $r$ hereinafter denotes a quantity in the reference case of channel flow with rigid walls. The friction Reynolds number is

$$R_\tau = \frac{u_\tau^* h^*}{\nu^*} = u_\tau R_p \tag{2.15}$$

where

$$u_\tau = \sqrt{\frac{1}{R_p} \left.\frac{\mathrm{d}\mathcal{U}}{\mathrm{d}y}\right|_{y=0}} \tag{2.16}$$

is the friction velocity. Scaling in viscous units of the rigid wall, i.e. $u_{\tau,r}^*$ and $\nu^*$, is denoted by the superscript $+0$, and scaling in viscous units of the controlled wall is indicated by the superscript $+$.

The root-mean-square (rms) of a fluctuating velocity component $q'$ is defined as:

$$q_{rms} = \sqrt{(L_x L_z)^{-1} \left[\overline{q'^2}\right]_{\mathcal{I}xz}} \tag{2.17}$$

The Reynolds stresses are defined as

$$uv_{rey} = (L_x L_z)^{-1} \left[\overline{u'v'}\right]_{\mathcal{I}xz} \tag{2.18}$$

The power balance within the channel can be written as:

$$\mathcal{P}_x + \mathcal{W} + \mathcal{D} = 0 \tag{2.19}$$

where $\mathcal{P}_x$ is the power spent to pump the fluid along $x$, $\mathcal{W}$ is the power spent by the viscous action of the fluid on the hydrophobic surface, and $\mathcal{D}$ is the viscous dissipation of kinetic energy into heat. For cases for which the wall no-penetration condition is imposed on the wall-normal velocity component and slip is considered only along the streamwise direction, the three quantities in (2.19) are:

$$\mathcal{P}_x = 2\mathcal{U}_b L_x L_z \left(\frac{R_\tau}{R_p}\right)^2 \tag{2.20}$$

$$\mathcal{W} = -\frac{2}{R_p}\left[\overline{U(0)\left.\frac{\partial U}{\partial y}\right|_{y=0}}\right]_{\mathcal{I}xz} \tag{2.21}$$

and

$$\mathcal{D} = -\frac{1}{R_p}\left[\overline{\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\frac{\partial U_i}{\partial x_j}}\right]_{\mathcal{I}xyz} \tag{2.22}$$

where the Einstein summation convention of repeated indices is used. The percent power used by the fluid on the surface is $\mathcal{P}_{sp}(\%) = 100\mathcal{W}/\mathcal{P}_{x,r}$. The details the derivation of the energy terms (2.20), (2.21), and (2.22) are as given in the following.

At the wall the no-penetration condition is imposed on the wall-normal velocity component and slip is considered only along the streamwise direction. The starting point is the balance equation for the total kinetic energy (equation (1-108) in Hinze (1975)):

$$\begin{aligned}\underbrace{\frac{1}{2}\frac{\partial(U_iU_i)}{\partial t}}_{\text{I}} = \underbrace{-\frac{\partial}{\partial x_j}\left[U_j\left(P + \frac{U_iU_i}{2}\right)\right]}_{\text{II}} + \underbrace{\frac{1}{R_p}\frac{\partial}{\partial x_j}\left[U_i\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\right]}_{\text{III}} \\ \underbrace{-\frac{1}{R_p}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\frac{\partial U_i}{\partial x_j}}_{\text{IV}},\end{aligned} \tag{2.23}$$

where the Einstein summation convention of repeated indices is used and all the terms are per unit mass and time. Term I is the local change of kinetic energy and term II is the change in convective transport of the pressure and kinetic energy, which is equivalent to the work done by the total dynamic pressure $P + U_iU_i/2$. Term III is the work performed by the viscous stresses and term IV is the viscous dissipation of the kinetic energy into heat. The interest is in the time average and in the volume integral of (2.23). Term I vanishes through time averaging. The power $\mathcal{P}_x$ employed to pump the fluid along $x$ is computed by time averaging and volume integration of term II, which is first written as

$$\underbrace{-\frac{1}{2}\frac{\partial\left(U_jU_iU_i\right)}{\partial x_j}}_{\text{IIa}}\underbrace{-\frac{\partial\left(U_jP\right)}{\partial x_j}}_{\text{IIb}}. \tag{2.24}$$

Term IIa vanishes upon volume integration because of periodicity along $x$ and $z$ and because the wall-normal velocity vanishes at the walls. By introducing the time-averaged quantities and by integrating along $x$, the power $\mathcal{P}_x$ is

$$\mathcal{P}_x = -\left[\overline{\frac{\partial\left(U_jP\right)}{\partial x_j}}\right]_{\mathcal{I}xyz} = \int_0^2\int_0^{L_z}\left[\left(\overline{U}\;\overline{P}\right)\Big|_{x=0} - \left(\overline{U}\;\overline{P}\right)\Big|_{x=L_x}\right]\mathrm{d}z\mathrm{d}y, \tag{2.25}$$

where use has been made of the periodicity along $z$ and of the no-penetration condition at the walls. Due to the time-averaged pressure being independent of $y$ and $z$ and to the periodicity of the velocity along $x$, it is found

$$\mathcal{P}_x = 2L_z\mathcal{U}_b\left(\overline{P}\big|_{x=0} - \overline{P}\big|_{x=L_x}\right) = 2\mathcal{U}_bL_xL_z\left(\frac{R_\tau}{R_p}\right)^2. \tag{2.26}$$

Use has been made of (2.15), (2.16), and the time- and space-averaged $x$-momentum equation at the walls.

The volume integral of time-averaged term III is the work $\mathcal{W}$ done by the fluid on the surface through the viscous stresses:

$$\mathcal{W} = \frac{1}{R_p}\left[\overline{\frac{\partial}{\partial x_j}\left[U_i\left(\frac{\partial U_i}{\partial x_j}+\frac{\partial U_j}{\partial x_i}\right)\right]}\right]_{\mathcal{I}xyz} = -\frac{2}{R_p}\left[\overline{U(0)\left.\frac{\partial U}{\partial y}\right|_{y=0}}\right]_{\mathcal{I}xz}. \tag{2.27}$$

The final expression is obtained by using the periodicity along $x$ and $z$ and the no-penetration condition at the walls for the wall-normal velocity component.

The volume integral of time-averaged term IV is the total viscous dissipation of kinetic energy into heat:

$$\mathcal{D} = -\frac{1}{R_p}\left[\overline{\left(\frac{\partial U_i}{\partial x_j}+\frac{\partial U_j}{\partial x_i}\right)\frac{\partial U_i}{\partial x_j}}\right]_{\mathcal{I}xyz}. \tag{2.28}$$

The volume integral of the time-averaged kinetic energy equation (2.23) is therefore:

$$\mathcal{P}_x + \mathcal{W} + \mathcal{D} = 0. \tag{2.29}$$

In the case of rigid walls, (2.29) reduces to

$$\mathcal{P}_{x,r} + \mathcal{D}_r = 0. \tag{2.30}$$

By dividing each term of (2.29) by $\mathcal{P}_{x,r}$, one finds:

$$100 - \mathcal{R} + \mathcal{P}_{sp} + \frac{100\mathcal{D}}{\mathcal{P}_{x,r}} = 0, \tag{2.31}$$

where the percent power spent is $\mathcal{P}_{sp}(\%) = 100\mathcal{W}/\mathcal{P}_{x,r}$. The drag reduction $\mathcal{R}$ appears in (2.31) by use of (2.13), (2.14), (2.15), (2.16).

## 2.2 Background for the Lyapunov analysis

In this section, the mathematical formulation for the stability analysis of partial differential equations (PDEs) is introduced. The general framework for tackling PDEs, the so-called Functional Analysis is presented. A previous analysis based on the stabilization of a two-dimensional turbulent channel flow (Balogh *et al.*, 2001) is applied in a three dimensional case.

The analysis for models described by a finite number of states relies on the resolution of ordinary differential equations (ODEs). Contrary to these models, the state space of systems described by PDEs is infinite-dimensional. Indeed, a general solution of an ODE involves arbitrary constants. Obtaining a general solution for PDEs is difficult and a general solution would involve arbitrary functions. For instance, one can consider the simple PDE, $u_x = 0$. Any arbitrary function of $y$ solves the latter. This is the simplest possible linear equation of first order yet it has an infinite dimensional space solutions. This situation has to be compared with that of a linear first order ODE $d\mathbf{y}/dt = 0$ where $\mathbf{y} = (y_1, \cdots, y_p)$, whose solution space is $\mathbb{R}^p$ which is finite dimensional.

In differential equations the unknown function has the interpretation of the state of a system when the equations describe evolution of a physical system in time. For ODEs the independent variable is time and for PDEs one of the independent variables has the interpretation of time. Now the initial state (state of the system at time $t = 0$) for ODEs is prescribed as an element of $\mathbb{R}^n$ ($n$ is the length of the unknown vector $y$) while for PDEs the initial state varies in a function space. Hence, solving a PDE means finding the states of the system at different times and each of these states vary in an infinite dimensional space of function. On the other hand, solving ODEs requires to find the states of the system while remaining in a finite dimensional space. Functional analysis is dealing with these infinite dimensional spaces. One important theorem which is lacking in infinite dimensional spaces is a Heine-Borel theorem (A subset of $\mathbb{R}^N$ is compact if and only if (iff) it is closed and bounded, with $N < \infty$) concerning compactness (Rudin, 1973). Thus the topologies more suited to infinite dimensional spaces are some kind of weak topologies. Weak topology is an alternative term for initial topology. The term is commonly used for the initial topology of a topological vector space (such as a normed vector space) with respect to its continuous dual. These concepts of topological space and weak topology (Rudin, 1973) are defined to capture a particular notion of convergence of sequences for functions. This is one the building blocks of Functional Analysis as the latter deals with normed vector spaces which are all topological vector spaces. It follows that Banach and Hilbert spaces are topological spaces as well. Along with the definition of weak derivatives, these spaces are used to build new topological spaces called *Sobolev* spaces (Rudin, 1973).

The relevant framework for analysing PDEs is thus *Functional Analysis*, where functions are studied as part of normed and complete vector spaces. These specific spaces are Banach spaces in general and Hilbert spaces in particular if the latter are endowed with an inner-product. The norms defined in such spaces play a major role on the control design of closed loop systems and their stability properties. Indeed, some systems present the property of being stabilizable in a given space while not controllable or even ill-posed in another one. Therefore the choice of the right space is of paramount importance. Typical Banach spaces are $L^p$ spaces and the associated norms have some physical meaning. The space of square-integrable functions $L^2$, defines a norm that can be interpreted as a measure of the energy within the state space. Similarly, $L^\infty$ and $L^1$ norms are respectively related to the maximum value of the state and the hypervolume (if $\Omega = \mathbb{R}^n$) determined by the graph of the state.

The main issue with $L^p$ spaces resides in their being too broad. These spaces contain functions that are not well-behaved. As a consequence, alternative spaces have to be defined for analysing the PDEs' solutions. These spaces not only include the states but also the derivatives of these. The required spaces mentioned above are defining the Sobolev spaces (in the case time is involved, these are called isotropic Sobolev spaces). Physically, Sobolev spaces can be interpreted as function spaces with finite energy content. These spaces allow to solve the variational formulation of PDEs. The variational representation of a problem consists in using test functions to write the PDE formulation as an integral-type equation and uses specific properties to state existence and uniqueness of solutions Temam (1984).

To illustrate the previous statement, one can take a function belonging to $L^2(0,1)$. By definition it means that this function is square-integrable in the set (0,1). However, nothing prevents it to be discontinuous or even unbounded. In order to ensure continuity and boundedness both the function and its spatial derivative are required to be part of $L^2(0,1)$. Therefore the justification for the use of Sobolev spaces.

It has been noted that for PDEs, the space state is infinite-dimensional. Due to this specificity, it follows that there is not a stability theory based on a general Lyapunov stability analysis. Therefore, each problem has to rely on the development of specific norms, in order to prove stability. The main idea to design feedback laws, is first to choose the right norm. Then the objective is to stabilize a specified equilibrium. This process uses the chosen norm, by making it exponentially stable. As an example, one can take the function $f(t,x)$ (perturbed variable), with $x \in (0,1)$, within $L^2(0,1)$. The objective is to prove that:

$$\|f(t)\|_{L^2(0,1)} \leqslant C_1 e^{-C_2 t} \|f(0)\|_{L^2(0,1)} \tag{2.32}$$

$C_1 \geqslant 1$ is called the overshoot coefficient and $C_2 > 0$ is the decay rate. The NSE originate from the conservation of mass and momentum. Their study also needs some properties related to the divergence and curl operators. The reader should refer to reference Temam (1984), in order to have a more elaborate approach on the treatment of NSE through Functional Analysis.

## 2.3 Lyapunov stability analysis

We can now focus on the Lyapunov stability on its own. The objective is to show stability of a parabolic equilibrium profile. The NSE are by nature nonlinear and the only way global stability of the feedback system can be enforced is through nonlinear analysis. However, this doesn't necessarily mean that the stabilizing feedback control law has to be nonlinear. In the work by Balogh *et al.* (2001), global Lyapunov stability is obtained in a two-dimensional channel flow using tangential actuation and local measurement of wall shear stress. This theoretically allows to embed feedback control through MEMS. The Lyapunov stability analysis for a laminar flow is performed hereafter. The objective is to stabilize the channel flow around the chosen equilibrium point. The work by Balogh *et al.* (2001) on the stabilization of a two-dimensional channel flow is applied in the three-dimensions. At the end of the analysis, this approach allows the specification of an a-priori-unknown feedback-control boundary conditions at the wall. We find that these feedback-control boundary conditions are the same as those of the slip-length hydrophobic model as described later on in 5. Note that the following analysis considers a general Robin type boundary condition. This is justified by the fact that this type of boundary condition will be applied in 5 in order to model different type of hydrophobic surfaces.

The flow domain is $\Omega = \{(x, y, z) \in [0, L_x) \times [0, 2] \times [0, L_z)\}$. Periodic boundary conditions are applied to the homogeneous $x$ and $z$ directions. The $L^2$ norm of a vector $\boldsymbol{f}$ is defined as

$$||\boldsymbol{f}\,||_{L^2} = \sqrt{[|\boldsymbol{f}\,|^2]_{\mathcal{I}xyz}}, \tag{2.33}$$

where The perturbation velocity vector, $\mathbf{w} = (u, v, w)$, and the perturbation pressure $p$ are defined as:

$$u = U - \widehat{U}\ ,\ v = V - \widehat{V}\ ,\ w = W - \widehat{W}\ ,\ p = P - \widehat{P}, \tag{2.34}$$

where

$$\left(\widehat{\mathbf{U}}, \widehat{P}\right) = \left(\widehat{U}, \widehat{V}, \widehat{W}, \widehat{P}\right) = \left(\widehat{U}(y), 0, 0, \widehat{P}(x)\right). \tag{2.35}$$

$\widehat{U}(y)$ is given through the laminar solution and $\widehat{P}(x) = x\mathrm{d}P/\mathrm{d}x$. We operate under constant mass flow rate conditions to have bounded $\widehat{U}(y)$. Upon substitution of (2.34) in the incompressible NSE, the nonlinear perturbation equations are found,

$$\nabla \cdot \mathbf{w} = 0, \tag{2.36}$$

$$\frac{\partial \mathbf{w}}{\partial t} + (\mathbf{w} \cdot \nabla)\left(\mathbf{w} + \widehat{\mathbf{U}}\right) + \left(\widehat{\mathbf{U}} \cdot \nabla\right)\mathbf{w} = -\nabla p + \frac{1}{R_p}\nabla^2 \mathbf{w}. \tag{2.37}$$

The perturbation energy is defined through the $L^2$ norm of the perturbed velocity, i.e., $E(\mathbf{w}) = ||\mathbf{w}||^2_{L^2}$. The time derivative of $E(\mathbf{w})$ is

$$\frac{1}{2}\frac{\mathrm{d}E(\mathbf{w})}{\mathrm{d}t} = \left[u\frac{\partial u}{\partial t}\right]_{\mathcal{I}xyz} + \left[v\frac{\partial v}{\partial t}\right]_{\mathcal{I}xyz} + \left[w\frac{\partial w}{\partial t}\right]_{\mathcal{I}xyz}. \tag{2.38}$$

Each term in (2.38) is expanded separately using (2.36)-(2.37) and periodicity in the homogeneous directions. An upper bound is derived for the time derivative of the energy,

$$\frac{\mathrm{d}E(\mathbf{w})}{\mathrm{d}t} \leqslant -\frac{\alpha E(\mathbf{w})}{2} + \frac{2}{R_p}\left[u^2(x,0,z,t) + w^2(x,0,z,t)\right]_{\mathcal{I}xz} + \frac{2}{R_p}\left[\left[u\frac{\partial u}{\partial y} + w\frac{\partial w}{\partial y}\right]_0^2\right]_{\mathcal{I}xz}, \tag{2.39}$$

where

$$[\cdot]_{\mathcal{I}xz} = \int_0^{L_z}\int_0^{L_x} \cdot \,\mathrm{d}x\,\mathrm{d}z, \tag{2.40}$$

and $\alpha = R_p{}^{-1} - 4 + R_p{}^{-1}L_x^{-2} + R_p{}^{-1}L_z^{-2}$. The dimensions of the domain along the homogeneous directions, $L_x$ and $L_z$, can be taken as infinitely large, which leads to $\alpha = R_p{}^{-1} - 4$. In the uncontrolled case ($u(x,0,z,t) = w(x,0,z,t) = 0$ and $u(x,2,z,t) = w(x,2,z,t) = 0$), $E(\mathbf{w})$ decays exponentially in time if $\alpha > 0$, i.e., $R_p < 1/4$. As in Balogh *et al.* (2001), global stability is achieved not only by choosing the right range for $R_p$, but also by modifying the integral terms, which pertain to the boundaries. Following Balogh *et al.* (2001):

$$u(x,y_w,z,t) = (1-y_w)\,k\frac{\partial u}{\partial y}(x,y_w,z,t),\;\; w(x,y_w,z,t) = (1-y_w)\,k\frac{\partial w}{\partial y}(x,y_w,z,t), \tag{2.41}$$

where $y_w$=0 for the lower wall and $y_w$=2 for the upper wall. Substitution of (2.41) into (2.39) leads to

$$\begin{aligned}\frac{\mathrm{d}E(\mathbf{w})}{\mathrm{d}t} \leqslant & -\frac{\alpha E(\mathbf{w})}{2} - \frac{2}{R_p}\left(\frac{1}{k} - 1\right)\left[u^2(x,0,z) + w^2(x,0,z)\right]_{\mathcal{I}xz} \\ & - \frac{2}{kR_p}\left[u^2(x,2,z,t) + w^2(x,2,z,t)\right]_{\mathcal{I}xz}.\end{aligned} \tag{2.42}$$

By setting $k \in (0,1]$, the perturbation energy $E$ decays exponentially, thus achieving global asymptotic stabilization.

It is remarkable to note that the controller found in (2.41), i.e., based on distributed actuation that linearly relates the in-plane wall velocity to the wall-normal velocity gradient, coincides with the widely-used Navier's model of hydrophobic surfaces (Min & Kim (2004), where both streamwise and spanwise slip velocities are considered). The constant $k$ agrees with the slip length $l_s$, given in the characterization of hydrophobic surfaces (Min & Kim, 2004). To the best of our knowledge, this is the first time that this conceptual link between these two apparently unrelated areas has been advanced.

A further interesting observation can be put forward. In the stability analysis, boundary terms involving the perturbation pressure $p$, i.e., proportional to $pu$, $pv$, and $pw$, vanish either by periodicity along $x$ and $z$ or through the no-penetration condition imposed on the wall-normal velocity. If the latter condition is relaxed while the periodicity along $x$ and $z$ is maintained, a wall-based controller of the type $v = Ap$ can be designed, which has been used by Balogh *et al.* (2005) to maximize mixing in a three-dimensional pipe flow. We note here that this wall-based linear relationship between the wall-normal velocity and pressure has also been employed successfully to model the interaction between the compressible flow of air and porous surfaces (Fedorov & Khokhlov, 2001), where $A$ plays the role of the admittance. High-precision experiments of these acoustic absorbing coatings (Fedorov *et al.*, 2003; Shiplyuk *et al.*, 2004) have been shown to lead to the attenuation of the growth rate of the acoustic mode in high-Mach-number compressible laminar boundary layers. The velocity-pressure boundary condition has also been used to simulate an incompressible turbulent flow over porous surfaces (Jimenez *et al.*, 2001). This problem is obviously out of the scope of the present study, but, similar to the wall-parallel controller case, it is worthwhile to notice how a purely mathematical exercise, such as the stability analysis, helps us educe boundary conditions that synthesize controllers with precise counterparts in Nature.

We now describe the derivation of the three terms in (2.38). These are expanded and the condition for stability is derived. The terms in (2.38) are first written as:

$$\begin{aligned}2\left[u\frac{\partial u}{\partial t}\right]_{\mathcal{I}xyz} &= -\frac{2}{R_p}\left[\frac{\partial u^2}{\partial x}+\frac{\partial u^2}{\partial y}+\frac{\partial u^2}{\partial z}\right]_{\mathcal{I}xyz} - 2\left[uv\frac{\partial \widehat{U}}{\partial y}\right]_{\mathcal{I}xyz} + 2\left[p\frac{\partial u}{\partial x}\right]_{\mathcal{I}xyz} \\ &\quad + \frac{2}{R_p}\left[\left[u\frac{\partial u}{\partial y}\right]_0^2\right]_{\mathcal{I}xz} - \left[\left[u^2 v\right]_0^2\right]_{\mathcal{I}xz}, \end{aligned} \tag{2.43}$$

$$2\left[v\frac{\partial v}{\partial t}\right]_{\mathcal{I}xyz} = -\frac{2}{R_p}\left[\frac{\partial v^2}{\partial x}+\frac{\partial v^2}{\partial y}+\frac{\partial v^2}{\partial z}\right]_{\mathcal{I}xyz} + 2\left[p\frac{\partial v}{\partial y}\right]_{\mathcal{I}xyz} + \frac{2}{R_p}\left[\left[v\frac{\partial v}{\partial y}\right]_0^2\right]_{\mathcal{I}xz}, \tag{2.44}$$

$$2\left[w\frac{\partial w}{\partial t}\right]_{\mathcal{I}xyz} = -\frac{2}{R_p}\left[\frac{\partial w^2}{\partial x}+\frac{\partial w^2}{\partial y}+\frac{\partial w^2}{\partial z}\right]_{\mathcal{I}xyz} + 2\left[p\frac{\partial w}{\partial z}\right]_{\mathcal{I}xyz} + \frac{2}{R_p}\left[\left[w\frac{\partial w}{\partial y}\right]_0^2\right]_{\mathcal{I}xz}. \tag{2.45}$$

The time derivative of the energy is obtained by adding the three terms in (2.43), (2.44), and (2.45):

$$\begin{aligned}\frac{\mathrm{d}E(\mathbf{w})}{\mathrm{d}t} = &- \frac{2}{R_p}\left[\frac{\partial u^2}{\partial x}+\frac{\partial v^2}{\partial x}+\frac{\partial w^2}{\partial x}+\frac{\partial u^2}{\partial y}+\frac{\partial v^2}{\partial y}+\frac{\partial w^2}{\partial y}+\frac{\partial u^2}{\partial z}+\frac{\partial v^2}{\partial z}+\frac{\partial w^2}{\partial z}\right]_{\mathcal{I}xyz} \\ &- 2\left[uv\frac{\partial \widehat{U}}{\partial y}\right]_{\mathcal{I}xyz} + \frac{2}{R_p}\left[\left[u\frac{\partial u}{\partial y}+w\frac{\partial w}{\partial y}\right]_0^2\right]_{\mathcal{I}xz}, \end{aligned} \tag{2.46}$$

where the no-penetration condition for the wall-normal velocity component, $v(x,0,z,t) = v(x,2,z,t) = 0$, has been used. Equation (2.46) is employed to find an upper-bound estimate, to show global stability, and to evince how stability can be enhanced under specified conditions. The square of the streamwise velocity is written as:

$$u^2(x,y,z,t) = \left[ u(x,0,z,t) + \int_0^y \frac{\partial u}{\partial y}(x,\gamma,z,t)\,\mathrm{d}\gamma \right]^2 , \tag{2.47}$$

and, using the inequality $(c+d)^2 \leqslant 2(c^2+d^2)$, the following relation is found:

$$u^2(x,y,z,t) \leqslant 2u^2(x,0,z,t) + 2\left[ \int_0^y \frac{\partial u}{\partial y}(x,\gamma,z,t)\,\mathrm{d}\gamma \right]^2 . \tag{2.48}$$

Use of the Cauchy-Schwarz inequality (Rudin, 1973) on the second term of the right-hand-side of (2.48) leads to:

$$\left[ \int_0^y \frac{\partial u}{\partial y}(x,\gamma,z,t)\,\mathrm{d}\gamma \right]^2 \leqslant y \int_0^y \left[ \frac{\partial u}{\partial y}(x,\gamma,z,t) \right]^2 \,\mathrm{d}\gamma. \tag{2.49}$$

Combining (2.48) and (2.49) and integrating over the domain $\Omega$ yields:

$$\begin{aligned} \left[u^2\right]_{\mathcal{I}xyz} &\leqslant 2\left[u^2(x,0,z,t)\right]_{\mathcal{I}xyz} + 2\left[ y\int_0^2 \frac{\partial u^2}{\partial y}(x,y,z,t)\,\mathrm{d}y \right]_{\mathcal{I}xyz} \\ &\leqslant 4\left[u^2(x,0,z,t)\right]_{\mathcal{I}xz} + 4\left[\frac{\partial u^2}{\partial y}\right]_{\mathcal{I}xyz} . \end{aligned} \tag{2.50}$$

Analogous expressions are obtained for $v$ and $w$. Adding the inequalities for the three velocity components, an upper bound on the integral of the terms involving the wall-normal derivatives in (2.46) is obtained:

$$-\left[ \frac{\partial u^2}{\partial y} + \frac{\partial v^2}{\partial y} + \frac{\partial w^2}{\partial y} \right]_{\mathcal{I}xyz} \leqslant -\frac{E(\mathbf{w})}{4} + \left[u^2(x,0,z,t) + w^2(x,0,z,t)\right]_{\mathcal{I}xz} . \tag{2.51}$$

An upper bound is found for (2.46):

$$\begin{aligned} \frac{\mathrm{d}E(\mathbf{w})}{\mathrm{d}t} \leqslant &- \frac{1}{2R_p}E(\mathbf{w}) + \frac{2}{R_p}\left[u^2(x,0,z,t) + w^2(x,0,z,t)\right]_{\mathcal{I}xz} \\ &- \frac{2}{R_p}\left[ \frac{\partial u^2}{\partial x} + \frac{\partial v^2}{\partial x} + \frac{\partial w^2}{\partial x} + \frac{\partial u^2}{\partial z} + \frac{\partial v^2}{\partial z} + \frac{\partial w^2}{\partial z} \right]_{\mathcal{I}xyz} \\ &+ \frac{2}{R_p}\left[ \left[ u\frac{\partial u}{\partial y} + w\frac{\partial w}{\partial y} \right]_0^2 \right]_{\mathcal{I}xz} - 2\left[ uv\frac{\partial \widehat{U}}{\partial y} \right]_{\mathcal{I}xyz} . \end{aligned} \tag{2.52}$$

To find upper bounds with respect to the terms containing derivatives in $x$ and $z$, the derivation is based on a Poincaré type inequality. Integrating by parts, using Young's inequality, $cd \leqslant \eta c^2/2 + d^2/(2\eta)$ with $\eta = 2$, and upper-bounding leads to:

$$\begin{aligned}\int_0^{L_x} f^2(x)\,\mathrm{d}x &\leqslant L_x f^2(L_x) + 2\int_0^{L_x} x^2 f_x^2\,\mathrm{d}x + \frac{1}{2}\int_0^{L_x} f^2(x)\,\mathrm{d}x \\ \Leftrightarrow \int_0^{L_x} f^2(x)\,\mathrm{d}x &\leqslant 2L_x f^2(L_x)\,\mathrm{d}x + 4\int_0^{L_x} x^2 \frac{\partial f^2}{\partial x}(x)\,\mathrm{d}x. \end{aligned} \tag{2.53}$$

As $x \in [0, L_x]$, the last integral in (2.53) can be further upper-bounded:

$$\int_0^{L_x} f^2(x)\,\mathrm{d}x \leqslant 2L_x f^2(L_x)\,\mathrm{d}x + 4L_x^2 \int_0^{L_x} \frac{\partial f^2}{\partial x}(x)\,\mathrm{d}x. \tag{2.54}$$

Inequality (2.54) can be applied to a function of three variables by summing over one direction at a time. For the $u$ velocity component, these expressions are:

$$\left[u^2(x,y,z,t)\right]_{\mathcal{I}xyz} \leqslant 2L_x \int_0^{L_z}\int_0^{2} u^2(L_x,y,z,t)\,\mathrm{d}y\,\mathrm{d}z + 4L_x^2\left[\frac{\partial u^2}{\partial x}\right]_{\mathcal{I}xyz}, \tag{2.55}$$

$$\left[u^2(x,y,z,t)\right]_{\mathcal{I}xyz} \leqslant 2L_z \int_0^{L_x}\int_0^{2} u^2(x,y,L_z,t)\,\mathrm{d}y\,\mathrm{d}x + 4L_z^2\left[\frac{\partial u^2}{\partial z}\right]_{\mathcal{I}xyz}. \tag{2.56}$$

Equations (2.55) and (2.56) and the corresponding ones involving $w$ lead to:

$$-\frac{2}{R_p}\left[\frac{\partial u^2}{\partial x} + \frac{\partial v^2}{\partial x} + \frac{\partial w^2}{\partial x}\right]_{\mathcal{I}xyz} \leqslant -\frac{E(\mathbf{w})}{2R_p L_x^2}, \tag{2.57}$$

$$-\frac{2}{R_p}\left[\frac{\partial u^2}{\partial z} + \frac{\partial v^2}{\partial z} + \frac{\partial w^2}{\partial z}\right]_{\mathcal{I}xyz} \leqslant -\frac{E(\mathbf{w})}{2R_p L_z^2}. \tag{2.58}$$

The boundedness of the equilibrium profile gives:

$$\begin{aligned}-2\left[uv\frac{\partial \widehat{U}}{\partial y}\right]_{\mathcal{I}xyz} &\leqslant 2\left[|u|\,|v|\right]_{\mathcal{I}xyz} \leqslant 2\left[u^2+v^2\right]_{\mathcal{I}xyz} \\ &\leqslant 2\left(E(\mathbf{w}) - \left[w^2\right]_{\mathcal{I}xyz}\right) \leqslant 2E(\mathbf{w}). \end{aligned} \tag{2.59}$$

Substitution of (2.59) into (2.52) leads to

$$\begin{aligned}\frac{\mathrm{d}E(\mathbf{w})}{\mathrm{d}t} \leqslant & - \frac{\alpha E(\mathbf{w})}{2} + \frac{2}{R_p}\left[u^2(x,0,z,t) + w^2(x,0,z,t)\right]_{\mathcal{I}xz} \\ & + \frac{2}{R_p}\left[\left[u\frac{\partial u}{\partial y} + w\frac{\partial w}{\partial y}\right]_0^2\right]_{\mathcal{I}xz}, \end{aligned} \tag{2.60}$$

where $\alpha = {R_p}^{-1} - 4 + {R_p}^{-1}L_x^{-2} + {R_p}^{-1}L_z^{-2}$.

## 2.4 Fukagata-Iwamoto-Kasagi identity

For this particular section, the notations hereafter are set to be consistent with Ricco & Hahn (2013) thereby allowing the reader to easily compare with it. The Fukagata-Iwamoto-Kasagi (FIK) identity (Fukagata *et al.*, 2002) is obtained by considering the streamwise momentum equation and developing the flow into a mean and fluctuating part: $f = f_m + f_t$. This formulation is general and in the case of rotating actuators, only the contribution from the latter are added into the decomposition as $f = f_m + f_t + f_d$. When control is applied at the boundaries of the domain, it is convenient to derive a formulation where boundary terms appear explicitly for their computation. This equation is expressed in outer units. From the streamwise momentum equation

$$\frac{\partial u}{\partial t} + \frac{\partial (uu)}{\partial x} + \frac{\partial (uv)}{\partial y} + \frac{\partial (uw)}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{R_p}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial x^2}\right). \tag{2.61}$$

The time average and space average operators are applied.

$$\langle f \rangle = \frac{1}{L_x L_z}\int_0^{L_z}\int_0^{L_x} f \,\mathrm{d}x\,\mathrm{d}z \quad , \quad \bar{f} = \frac{1}{t_f - t_i}\int_{t_i}^{t_f} f\,\mathrm{d}x\,\mathrm{d}z \tag{2.62}$$

With these definitions: $f_m = \langle \bar{f} \rangle$ and $\overline{u_t} = 0$. Applying time and space averaging operators to (2.61) with the aforementioned decomposition leads to:

$$-\frac{\partial p_m}{\partial x} = \frac{\partial}{\partial y}\left[u_m v_m + \langle \overline{u_t v_t} \rangle - \frac{1}{R_p}\frac{\partial u_m}{\partial y}\right] \tag{2.63}$$

The wall-normal direction domain is $\Omega = [0, 2]$. Integrating (2.63) from 0 to $y$:

$$-\frac{\partial p_m}{\partial x} y = [u_m(y) v_m(y)] - [u_m(0) v_m(0)] + \langle \overline{u_t(y) v_t(y)} \rangle - \langle \overline{u_t(0) v_t(0)} \rangle - \frac{1}{R_p}\left[\partial u_m/\partial y - \partial u_m/\partial y(0)\right] \tag{2.64}$$

Taking $y = 1$ and $y = 2$ into (2.64) and using the fact that at the channel centre $\partial u_m/\partial y(1) = 0$:

$$-\frac{\partial p_m}{\partial x} = [u_m(1)v_m(1)] - [u_m(0)v_m(0)] + \langle\overline{u_t(1)v_t(1)}\rangle - \langle\overline{u_t(0)v_t(0)}\rangle + \frac{1}{R_p}\partial u_m/\partial y(0) \quad (2.65)$$

$$-2\frac{\partial p_m}{\partial x} = [u_m(2)v_m(2)] - [u_m(0)v_m(0)] + \langle\overline{u_t(2)v_t(2)}\rangle - \langle\overline{u_t(0)v_t(0)}\rangle - \frac{1}{R_p}\left[\partial u_m/\partial y(2) - \partial u_m/\partial y(0)\right] \quad (2.66)$$

Subtracting (2.65) from (2.66):

$$-\frac{\partial p_m}{\partial x} = [u_m(2)v_m(2)] - [u_m(1)v_m(1)] + \langle\overline{u_t(2)v_t(2)}\rangle - \langle\overline{u_t(1)v_t(1)}\rangle - \frac{1}{R_p}\partial u_m/\partial y(2) \quad (2.67)$$

Equations (2.65) and (2.67) are considered in the sequel. Relating $C_f$ and $\partial u_m/\partial y$ at $y = 0$ and $y = 2$ (respectively labelled $C_{f,\mathrm{lw}}$ and $C_{f,\mathrm{lw}}$):

$$C_{f,\mathrm{lw}} = \frac{9}{2R_p}\partial u_m/\partial y(0) \quad , \quad C_{f,\mathrm{uw}} = -\frac{9}{2R_p}\partial u_m/\partial y(2) \quad (2.68)$$

Replacing the previous expressions, equations (2.65) and (2.67) can be written in a compact form:

$$-\frac{\partial p_m}{\partial x} = A_0 + \frac{2C_{f,\mathrm{lw}}}{9} \quad (2.69)$$

and for the upper-wall

$$-\frac{\partial p_m}{\partial x} = A_2 + \frac{2C_{f,\mathrm{uw}}}{9} \quad (2.70)$$

with $A_0$ and $A_2$ given by:

$$A_0 = [u_m(1)v_m(1)] - [u_m(0)v_m(0)] + \langle\overline{u_t(1)v_t(1)}\rangle - \langle\overline{u_t(0)v_t(0)}\rangle \quad (2.71)$$

$$A_2 = [u_m(2)v_m(2)] - [u_m(1)v_m(1)] + \langle\overline{u_t(2)v_t(2)}\rangle - \langle\overline{u_t(1)v_t(1)}\rangle \quad (2.72)$$

Replacing equation (2.69) into (2.63) and integrating twice from 0 to $y$:

$$\begin{aligned} A_0 y^2 + \frac{2C_{f,\mathrm{lw}}}{9}\left(\frac{y^2}{2} - y\right) &= \int_0^y \left[u_m(y)v_m(y) + \langle\overline{u_t(y)v_t(y)}\rangle\right] \mathrm{d}y \\ &- \left[u_m(0)v_m(0) + \langle\overline{u_t(0)v_t(0)}\rangle\right] y - \frac{1}{R_p}\left[u_m(y) - u_m(0)\right] \end{aligned} \quad (2.73)$$

Integrating (2.73) in the interval [0,2] with $\int\limits_0^2 u_m(y)\,\mathrm{d}y = \frac{4}{3}$, transforming and rearranging terms leads to the final expression:

$$\begin{aligned} C_{f,\mathrm{lw}} &= 9A_0 - \frac{27}{4}\int\limits_0^2 (2-y)\left[u_m(y)v_m(y) + \overline{\langle u_t(y)v_t(y)\rangle}\right]\,\mathrm{d}y \\ &+\frac{27}{2}\left[u_m(0)v_m(0) + \overline{\langle u_t(0)v_t(0)\rangle}\right] + \frac{9}{2R_p}\left[1 - \frac{3}{2}u_m(0)\right] \end{aligned} \tag{2.74}$$

Applying similar steps for the upper-wall:

$$\begin{aligned} C_{f,\mathrm{uw}} &= 9A_2 + \frac{27}{4}\int\limits_0^2 y\left[u_m(y)v_m(y) + \overline{\langle u_t(y)v_t(y)\rangle}\right]\,\mathrm{d}y \\ &-\frac{27}{2}\left[u_m(2)v_m(2) + \overline{\langle u_t(2)v_t(2)\rangle}\right] + \frac{9}{2R_p}\left[1 - \frac{3}{2}u_m(2)\right] \end{aligned} \tag{2.75}$$

The expression that is used in the numerical evaluation of $C_f$ consists in the average of (2.74) and (2.75).

## 2.5 Quadrant analysis

From the previous section, it is clear that the Reynolds shear-stress are those contributing to the modidification of $C_f$. It is therefore interesting to quantify their contribution not only by directly computing their overall contribution, but also compute these contributions in terms of the of the sign of the streamwise and wall-normal velocity components. This brings to consider the contributions of the Reynolds shear stresses in the FIK identity for each quadrant (Yakeno *et al.*, 2014).
The quadrant classification is based on the well-known Q$i$ ($i \in [\![1,4]\!]$) events (Willmarth & Lu, 1972). The events in Q2 and Q4 contribute positively to the downward momentum flux, with the Q2 events corresponding to ejections, while Q4 events correspond to sweeps. The contributions of the Q1 and Q3 events characterize outward and inward interactions (refer to figure 2.2). The expressions derived in the previous section can be written in terms of the normalization and usual conventions taken everywhere else in this manuscript. Therefore by combining the FIK identity and selecting over the turbulent flow sign, it is possible to quantify the contribution of each event into $C_f$. The quantity of interest entering into the expression for the skin-friction drag is therefore given by

$$\mathcal{Q} = -\frac{6}{\mathcal{U}_b^2}\int_0^1 (1-y)uv_{rey}\mathrm{d}y. \tag{2.76}$$

the quantity $\mathcal{Q}$ can then take four values depending on the events selection.

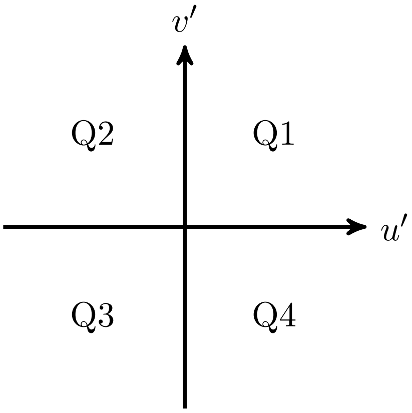


Figure 2.2: Delimitation of the quadrant regions depending on the sign of the fluctuating velocity.

# Chapter 3

# Numerical methods

## 3.1 Introduction

The complex behaviour of turbulence relies on the set of Navier-Stokes equations. Analytic solutions only exist for the simplest turbulent flows. Attempts to overcome this issue have been carried out through a statistical theory of turbulence. However, this approach faces a new problem that relies on the closure of its formulation. The closure problem arises from the coupling of successive moments of the distribution function. This coupling originates from the Reynolds stresses encompassing the nonlinear terms. To remedy this problem, modelling of the Reynolds stresses is necessary. In this way, the set of equations is closed thus avoiding the infinite hierarchy issue. Although these methods allow to model turbulence phenomena, it may not be the most accurate one. A full description of turbulent flows, where the relevant quantities such as the velocity fields and the pressure are function of time and space can be obtained by solving numerically the Navier-Stokes equations using DNS.

For globally understanding turbulence problems without introducing any bias related to numerical modelling, one can focus on the most accurate numerical description that is DNS. One of the issues when using this computational approach resides in the fact that the range of scales in turbulent flows increases dramatically with the Reynolds number. Therefore a wide range of scales on the spatial domain needs to be resolved at every time step. More precisely, the underlying physics dictates that scales on the order of the Kolmogorov length scales have to be consistently resolved.

In order to understand the relationship between the number of grid points, the estimation that is first shown is based on an isotropic turbulence model. For a channel flow, the power law estimate simply dictates that the number of degrees of freedom, i.e. the number of grid points, $n_x \times n_y \times n_z$, for a three-dimensional uniform grid increases as $R_\tau^{9/4}$, where $R_\tau$ is the Reynolds number based on the wall-shear stress velocity (Piomelli, 2008). Indeed the Reynolds number is the parameter governing the range of scales present in a DNS. The grid resolution required for a particular problem is expected to scale with this number. To see this, the required number of collocation points is

$$\mathcal{N} \propto \left(\frac{L^*}{\eta^*}\right)^3, \tag{3.1}$$

where $L^*$ is the integral length scale of the largest eddies, which is actually proportional to the channel height $h^*$. The quantity $\eta^*$ is the Kolmogorov scale which accounts for the typical size of the smallest eddies

$$\eta^* = \left(\frac{\nu^{*3}}{\epsilon^*}\right)^{\frac{1}{4}}, \tag{3.2}$$

where $\nu^*$ is the kinematic viscosity and $\epsilon^*$ the energy dissipation rate. The latter can be approximated by the following expression

$$\epsilon^* = \frac{u_\tau^{*3}}{h^*}, \tag{3.3}$$

where $u_\tau^*$ is the friction velocity at the wall, ratio of the averaged shear stress to the fluid density, $\sqrt{\tau^*/\rho^*}$. Combining (3.1)-(3.3) results in the number of collocation points

$$\mathcal{N} \propto \left(\frac{u_\tau^* h^*}{\nu^*}\right)^{\frac{9}{4}} = R_\tau^{\frac{9}{4}}. \tag{3.4}$$

However, the present DNS code is used with periodic directions only in the streamwise and spanwise directions which are both equally spaced number of grid points, while requiring a mesh refinement for the wall-normal direction in the vicinity of the solid wall. This is required as the latter is the main source of vorticity and where local extrema of turbulent kinetic energy occur. Therefore the number of grid points estimate for a turbulent channel flow is not as straightforward as the one based on the isotropic turbulence case. Therefore for the turbulent channel flow the estimate will not be as stringent as the isotropic turbulent case. To verify that the number of grid points and the computational box size are chosen adequately in the turbulent channel flow case, the verification can be done a posteriori by using correlations (box size adequacy check) and spectra (grid points adequacy check). The latter should verify that there is no energy density pile up in Fourier space at the highest wavenumber (Moser & Moin, 1984; Kim *et al.*, 1987*a*). This problematic was exhibited in DNS of turbulent channel flow performed at higher Reynolds numbers (Moser *et al.*, 1999).

The NSE and continuity equations are a set of partial differential equations (PDE), that first need to be discretized in order to be solved numerically using DNS. The great challenge in DNS resides in its massive computational cost due to the non-linearity and the non-locality of the Navier-Stokes equations. The seminal article presenting this method was based on the simulation of a channel flow (Kim *et al.*, 1987*a*). From there, other numerical studies have focused on improving this method and building up on this fundamental simulation.

### 3.1.1 Space grid discretization

The Navier-Stokes equations discretization in space requires for the continuous space to be divided into a set of discrete points and the relevant physical quantities to be stored at these points. This leads to a set of ordinary differential equations for each point. As NSE are elliptic in space all the domain should be resolved simultaneously. Grid generation consists in dividing the physical flow domain into smaller domains called control volumes or cells. There are two major type of grids: **collocated** grids where scalars and the vector type quantities (pressure and velocity in the present case) are both stored on the same control volumes and **staggered** grids where scalars are stored at the centre of control volumes, while the vectors are stored at the mesh faces. For most DNS simulations, the staggered method is used. The justification for using this type of grid distribution follows from stability considerations. Indeed, discretization of the partial differential flow equations on a mesh with collocated variables, can give rise to a spurious mode (actually a spurious wave) for the pressure, when the cell-face velocities are linearly interpolated between the neighbouring nodes without pressure stabilization and when the pressure term in the momentum equations is approximated by central difference. This mode is not seen by the discretized equations, and results in a solution without physical meaning. A remedy for this problem is the use of a staggered grid for the variables. One of the variants of the pressure-correction scheme of this type is the MAC (Marker-And-Cell) (Harlow &

Welch, 1965). If rectangular grids are used, the choice of a staggered grid arrangement is the most natural choice for a straightforward discretization of the governing equations. However, this approach is not comfortable, especially in a three-dimensional environment making use of body-fitted grids.

By using the staggered grids, this issue is overcome. Figure 3.1 sketches a 2D staggered grid. The 3D case can be straightforwardly extended from there. In this figure, the location for the physical quantities to be stored are shown. This is the current configuration in which the Incompact3d code is used. In order to understand the difference and implications resulting from the use of collocated and staggerd grids, a brief justification of both methods through the momentum and continuity equations is given in the next section.

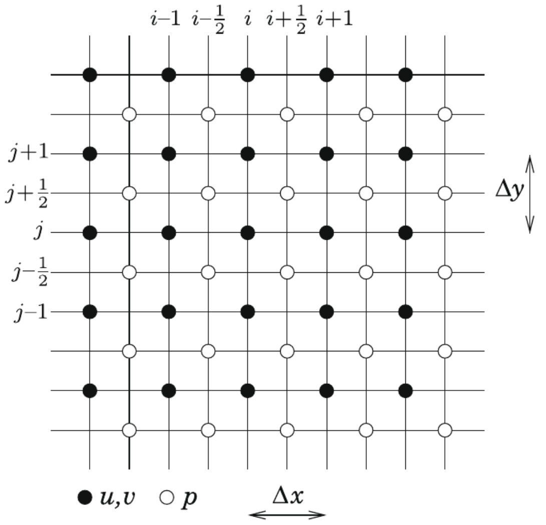


Figure 3.1: Partially staggered grid representation where the pressure is staggered by half a mesh from the velocity field.

### Collocated storage of variables

Initially, suppose that pressure and velocity are collocated (stored at the same positions) and that advective velocities (the cell-face velocities used to calculate mass fluxes) are calculated by linear interpolation. This configuration is shown in figure 3.2. In the scalar momentum equation the net pressure force is

$$p_w - p_e = \frac{1}{2}(p_{i-1} + p_i) - \frac{1}{2}(p_i + p_{i+1}) = \frac{1}{2}(p_{i-1} - p_{i+1}) \tag{3.5}$$

In the scalar momentum equation, pressure forces appear as a source of momentum:

$$\text{net pressure force} = (p_w - p_e)A.$$

where $A$ accounts for an area. The discretized momentum equation can schematically be expressed as:

$$\underbrace{a_p u_p - \sum_F a_F u_F}_{\text{net flux}} = \underbrace{A(p_w - p_e)}_{\text{pressure forces}} + \text{other forces.}$$

Hence $u_p = d_p(p_w - p_e)$ with $d_p = \frac{A}{a_p}$. The discretized momentum equation can then symbolically be written as:

$$u_i = \frac{1}{2} d_i (p_{i-1} - p_{i+1}) + \cdots \tag{3.6}$$

In the continuity equation the net outward mass flux depends on:

$$\begin{aligned} u_e - u_w &= \frac{1}{2}(u_i + u_{i+1}) - \frac{1}{2}(u_{i-1} + u_i) = \frac{1}{2}(u_{i+1} - u_{i-1}) \\ &= \frac{1}{4}\left[d_i(p_i - p_{i+2}) - d_{i-1}(p_{i-2} - p_i)\right] \end{aligned} \tag{3.7}$$

Thus, both mass and momentum equations only produce links between pressures at alternate nodes. As a summary we can state that the combination of (1) collocated $u$, $p$ and (2) linear interpolation for advective velocities leads to decoupling of odd nodal values $p_1$, $p_3$, $p_5$, $\cdots$ from even nodal values $p_2$, $p_4$, $p_6$, $\cdots$. This odd-even decoupling or checker-board effect leads to indeterminate oscillations in the pressure field and usually causes calculations to crash. This brings about the concept of staggered grids in order to overcome this limitation.

### Staggered grid

In the staggered-velocity grid arrangement, velocity components are stored half-way between the pressure nodes. This leads to different sets of control volumes. The usual convention is that each velocity node has the same index ($i$, $j$ or $k$) as the pressure node to which it points. Other scalars are stored at the same position as pressure. On a Cartesian mesh, in the momentum equation, pressure is stored at precisely the points required to compute the pressure force. As before, the discretized momentum equation can be written (refer to figure 3.5):

$$u_i = d_i(p_{i-1} - p_i) + \cdots \tag{3.8}$$

In the continuity equation, velocity is stored at precisely the points required to compute mass fluxes. The net mass flux involves:

$$\begin{aligned} u_{i+1} - u_i + \cdots &= d_{i+1}(p_i - p_{i+1}) - d_i(p_{i-1} - p_i) + \cdots \\ &= d_i p_{i-1} + (d_i + d_{i+1})p_i - d_{i+1}p_{i+1} \end{aligned} \tag{3.9}$$

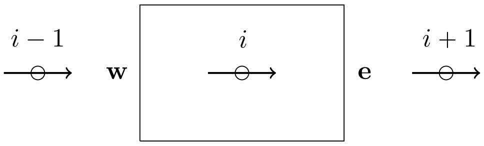


Figure 3.2: Collocated configuration: velocity field (arrows) at the same location as the pressure (open circles).

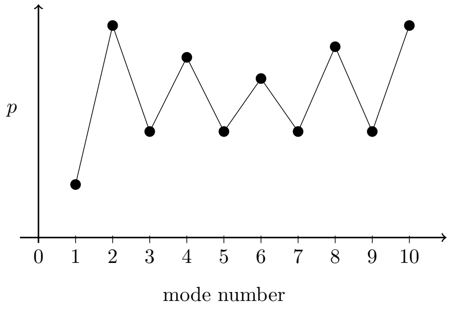


Figure 3.3: Illustration of the odd-even decoupling phenomenon.

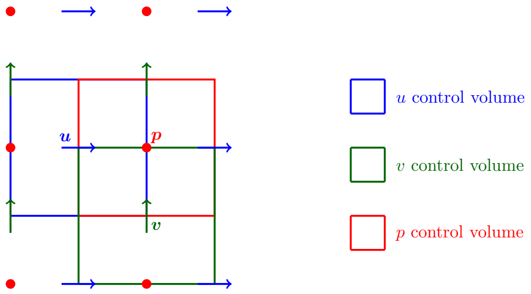
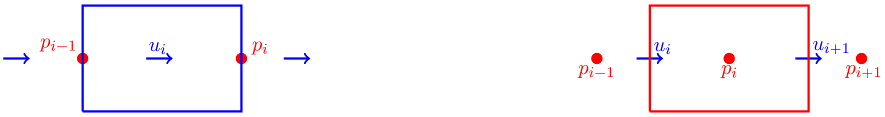


Figure 3.4: Close-up of one cell in the case of a staggered grid configuration in two-dimensional case.



(a) momentum equation cell configuration.

(b) continuity equation cell configuration.

Figure 3.5: Configuration of the staggered grid for the momentum and the continuity equation.

## 3.2 Time integration of the Navier-Stokes equations

The set of Navier-Stokes equations are elliptic in space and parabolic in time. In order to have the solution at a certain time, only the information at the previous time step is required. The spatial discretization of parabolic PDEs yields a system of ODEs for each grid point. The difficulty in solving a time accurate solution for an incompressible flow relies on the fact that the continuity equation does not contain a time derivative explicitly. The constraint of mass conservation is achieved by implicit coupling between the continuity equation and the pressure in the momentum equations. To solve the time advancement of the NSE, Incompact3d code uses a fractional step method.

Before describing the latter methods it is also worth mentioning two other techniques used for the numerical resolution of the NSE. The first is the so-called velocity-vorticy formulation. Within this framework, the vorticity transport equation is central to the dynamics of the system. This equation is well-known and commonly used. On the other hand, the kinematics relies on a vector-based elliptic equation. The formulation was first derived by Fasel (1976). It embodies several advantages from a numerical standpoint. First the two variables are the most relevant for tackling problems where the flow is vortex-dominated. Then, it offers a non-negligible advantage over the primitive variable formulation: all the difficulties arising from the treatment of the pressure boundary conditions are eliminated. Finally it can efficiently handle non-inertial frames. The other numerical method was developed by Kleiser & Schumann (1980). A comprehensive review is given in Canuto *et al.* (2007). In this method, a spectral approach is developed. The proper boundary condition for the pressure is defined implicitly through the condition on the velocity field bound to be divergence free at the walls of the channel. Numerically, the correct boundary value for the pressure is obtained through an influence (or capacitance) matrix technique. In this way the problem is equivalently reduced to cascading Helmholtz problems which proves to be efficient. This methodology was used within the `channelflow` code (Gibson, 2006*a*).

### 3.2.1 The fractional-step method

The fractional-step or projection method (also known as splitting-up method) aim is to carry out the time integration of the momentum equation and enforce the continuity equation, thereby solving the NSE. This method stems from the work by Leray (1933) but from a numerical standpoint, this procedure was independently developed by Chorin

(1968) and Temam (1969). The method uses the so-called Hodge-decomposition theorem (Hodge, 1989). In its simplest version, it states that any square-integrable vector field ($\mathbf{v}(\mathbf{x})$) can be decomposed as the sum of a curl-free ($\nabla\Phi$) and divergence-free ($\mathbf{u}(\mathbf{x})$) parts also known as the Helmholtz-Leray decomposition:

$$\mathbf{v}(\mathbf{x}) = \mathbf{u}(\mathbf{x}) + \nabla\Phi, \tag{3.10}$$

Using a proper inner product, coming from the definition of the Leray projector such that $\langle \mathbf{u}, \nabla\Phi \rangle = 0$, this implies that the two contributions to the vector field are orthogonal. Note that the uniqueness of the decomposition for this vector field needs to be ensured by imposing relevant boundary conditions. This can be done by specifying the normal component of the flow: $\hat{\mathbf{n}} \cdot \mathbf{u} = u_{\mathrm{bc}}$, where $\hat{\mathbf{n}}$ is the unit normal. The divergence-free contribution of an arbitrary vector $\mathbf{v}$ can be obtained through the projection onto the orthogonal subspace of divergence-free vectors by removing the gradient of a well-chosen scalar potential $\Phi$. This projection is expressed through a projector $\mathcal{P}$:

$$\mathbf{u} = \mathcal{P}(\mathbf{v}), \tag{3.11}$$

One can then write an approximation for the momentum equation:

$$\frac{\partial \mathbf{u}^{\circledast}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla q = \nu \nabla^2 \mathbf{u}^{\circledast}, \tag{3.12}$$

which is advanced within a time interval $[t_0, t_1]$. The divergence-free velocity is obtained at any time using the definition of the projector, i.e. $\mathbf{u}(t) = \mathcal{P}(\mathbf{u}^{\circledast}(t))$. In (3.12) $\nabla q$ represents an approximation to the pressure gradient. In practical terms, the projection is carried out through the derivation of an elliptic equation. Thus, applying Hodge decomposition theorem, one can write:

$$\mathbf{u}^{\circledast} = \mathbf{u} + \nabla\Phi, \tag{3.13}$$

By taking the divergence of (3.13) along with $\nabla \cdot \mathbf{u} = 0$, an elliptic equation is derived:

$$\nabla^2\Phi = \nabla \cdot \mathbf{u}^{\circledast}, \tag{3.14}$$

The pressure can then be recovered at any time through:

$$\nabla p = \nabla \left( q + \frac{\partial \Phi}{\partial t} \right) - \nu \nabla^2 \nabla \Phi \tag{3.15}$$

Equation (3.14) is obtained by substituting (3.13) into (3.12). The most general method is to advance (3.12) for a single time step, then compute $\mathbf{u}$ at the end of the time step using the projection and finally replace $\mathbf{u}^{\circledast}$ with the new value of $\mathbf{u}$ at the start of the next time step. Numerous projection methods are discussed in the literature, differing from each other in the following points, considered either separately or as a combination:

- the approximation of the advective terms
- the approximation for $q$
- if (3.12) is advanced in an explicit or implicit way
- the way the pressure gradient as given by (3.15) is treated. With respect to pressure treatment, this point is the most problematic and theoretical issues are also discussed extensively by Temam (1991) and Sani *et al.* (2006).

The general projection method described above is put into context through a brief description of some the most commomly used methods from the literature.

In Chorin (1968), at each time step, an incomplete form of the momentum equation is integrated to get an approximate velocity field. This field is not divergence-free. Therefore a corrector step is necessary to enforce this velocity field to verify the divergence-free constraint. This correction step is applied to the approximate velocity field and is an orthogonal projection, i.e. it projects the initial velocity field onto a divergence-free field without modifying the vorticity. This so-called *projection* step thereby lends its name to the method.

The original method proposed by Chorin was then modified by Kim & Moin (1985) to be applied for finite volumes on a staggered grid and has been used extensively for unsteady flow simulations. Basically the projection method drops the pressure gradient from the momentum equation and a factorization technique is used. Special boundary condition were imposed on the intermediate velocity field ensuring a second order approximation in time. The method introduced by Kim & Moin (1985) is based on a pressure correction step. It is similar to the projection method but the pressure gradient term is retained in the momentum equation. The Poisson equation is solved for a pressure correction subsequently used to correct the intermediate velocity field and enforce continuity. It is worth noting that a detailed analysis of the projection methods was carried by Gresho & Chan (1990). Therein they segregated the basic method and pressure correction method, labelling them respectively P1 and P2. It was analytically proven that P1 with physical boundary conditions for the intermediate velocity was first order in time while P2 was second order in time. The pressure correction method can be used in an iterative way, thereby the momentum and pressure correction system of equations is solved repeatedly at each time step. This minimizes the error associated in a single step method.

## 3.3 The Incompact3d code

Incompact3d is a CFD code for undertaking direct numerical and large-eddy simulations of incompressible flows. Within the frame of DNS, resolving the dynamics of the turbulent eddies at the smallest scales requires high resolution multi-scale flow simulations. This involves using powerful supercomputing systems. Very few DNS codes are able to efficiently carry out computations on large supercomputing facilities using mesh nodes scaling up to billions.

For reduced geometries, the most accurate codes are those using spectral methods either based on Fourier or Chebychev representation. This numerical strategy can be encountered in homogeneous turbulence, transitional or turbulent channel flows with simple geometry. However, when considering more complex geometries these type of codes fall short in terms of simplicity of the numerical strategies. Non-trivial numerical methods are required in order to combine sophisticated meshes and higher order schemes (collocation and spectral elements method). One of the strength of the Incompact3d code is that the spectral accuracy obtained using high-order compact finite difference schemes allows to study problems dealing with complex geometries, which usually becomes a deadlock for fully spectral codes. Incompact3d aims at finding a compromise between the accuracy of spectral codes and the versatility of more realistic industrial codes in terms of complex geometry treatment. The strategy followed comes in a complementary way to the collocation method and spectral elements relying on higher order finite difference schemes computed on a Cartesian grid. This strategy is simple and efficient. Using a Cartesian mesh enables the implementation of a high order compact finite difference scheme. The choice for a $6^{th}$-order scheme has been found to be the best compromise between the required accuracy and the computational cost. It can also capture the physics of problems based on more complex geometries when combined with an *Immersed Boundary Method* (IBM). This method mimics the effects of a solid surface on the fluid with a forcing applied in the body region. With the latter method, any solid wall geometry can be enforced, but the problem can still make use of a Cartesian mesh since this new geometry comes as a forcing term in the Navier-Stokes equations.

When using higher order schemes, one of the main problems arising in terms of numerical development and efficiency is the implementation of incompressibility. This is done using the method described in §3.2.1, in the context of the modified wave number framework. Indeed, equivalence rules between spectral and physical spaces can readily be employed (described in the sequel) to solve the Poisson equation arising from the projection method. The spectral treatment of the Poisson equation offers three major advantages: easy staggering of the pressure mesh (necessary to avoid the occurrence of spurious pressure oscillations), the possibility to introduce mesh stretching in one direction without losing the non-iterative nature of the Poisson solver, cost limited to less than 10% of the overall computational expense for regular meshes used here (and around 30% for stretched meshes). The code description relative to a channel flow configuration will be given in details in the next sections. A brief introduction to the code numerical strategies is given hereafter.

Incompact3d solves the incompressible Navier-Stokes equations using $6^{th}$ order compact finite difference schemes in the spatial domain. For the time integration, various multistep schemes are at hand, namely Adams-Bashforth or Runge-Kutta. It is necessary to mention about stability limits as in the code the time step is set to a constant from the start of the simulation. Indeed, the limit that is associated to the viscous term is $\nu^* \Delta t^* / \Delta x^{*2} < C_\nu$, with $\nu^*$ the kinematic viscosity, and $C_\nu$ a constant depending on the spatial and temporal schemes. The other stability limit is based on the definition of the Courant-Friedrichs-Lewy (CFL) number. This number is related to the convective term which is $c_s^* \Delta t^* / \Delta x^* < C_{\text{CFL}}$, where $c_s$ is a flow characteristic velocity and $C_{\text{CFL}}$ is a constant depending on the spatial and temporal schemes. From these considerations, it becomes clear that when decreasing the Reynolds number, the considered problem can become viscous dominated and the stability limit is based on the viscous case. Therefore, as the number of points is increased, because of the Reynolds number scaling, one needs to decrease the time step accordingly based on the viscous stability limit. Therefore if a problem needs to reach very high Reynolds numbers for wall turbulence problems (where the mesh is refined near the wall), reducing the time step can become a deadlock. Therefore an implicit treatment of the viscous term would be necessary.

The flow domain $L_x \times L_y \times L_z$, is discretized on a Cartesian mesh comprising $n_x \times n_y \times n_z$ mesh nodes. The wall-normal direction can be stretched by using a mapping technique

(Cain *et al.*, 1984). Inflow/Outflow, periodic, free-slip, no-slip conditions are used. A total of 10 sets of boundary conditions can be combined within the code depending on the problem under consideration. The incompressibility condition is enforced by using a fractional step method, requiring to solve a Poisson equation for the pressure. By using the numerical methods in Kim & Moin (1985); Swarztrauber (1977); Wilhelmson & Ericksen (1977) adapted to higher order schemes and stretched grids, the Poisson equation is fully solved in reciprocal space by means of Fast Fourier Transform (FFT) algorithm. One has to be aware that the Fourier representation can be applied in the three spatial directions whatever the set of boundary conditions. To this end, relevant cosine expansions are used. Within the modified wave number framework, this technique allows for the implementation of the divergence-free condition up to machine precision. For the computation of FFTs, the code offers two alternatives: a generic FFT library or the well-known FFTW package (Frigo & Johnson, 2005). The treatment of the pressure is based on a partially staggered grid, where the latter field is shifted by half a mesh in all spatial directions allowing for reduced spurious oscillations.

The code uses the finite difference schemes in order to deal with a wide variety of boundary conditions. The underlying idea for using high order compact finite difference schemes is to express the operators through equivalence rules with the wavenumber framework. This procedure mimics the behaviour of spectral numerical methods and allows to achieve spectral-like accuracy. Incompact3d is a Fortran 90 code. At the time of writing the thesis, it comprised a total of about 34,000 lines with 50 source files. In terms of efficient computation, an optimised version of the code already exists, parallelized with Message Passing Interface (MPI). This version of the code employing a dual domain decomposition (DD) strategy has been used. The 2D-DD allows running up to $\mathcal{O}(10^5)$ cores (Laizet & Li, 2011). Its performance was measured on several supercomputing platforms by carrying out weak and strong scaling tests. Figure 3.6 shows an example of strong scaling of the code on HECToR supercomputer. In the next section, the DD strategy will be described in further details.

### 3.3.1 Domain decomposition strategy

An early parallel version of Incompact3d used a 1D-DD (or slab decomposition). Simulations on 1024 cores with up to $5 \times 10^9$ mesh nodes could be achieved (Laizet *et al.*,

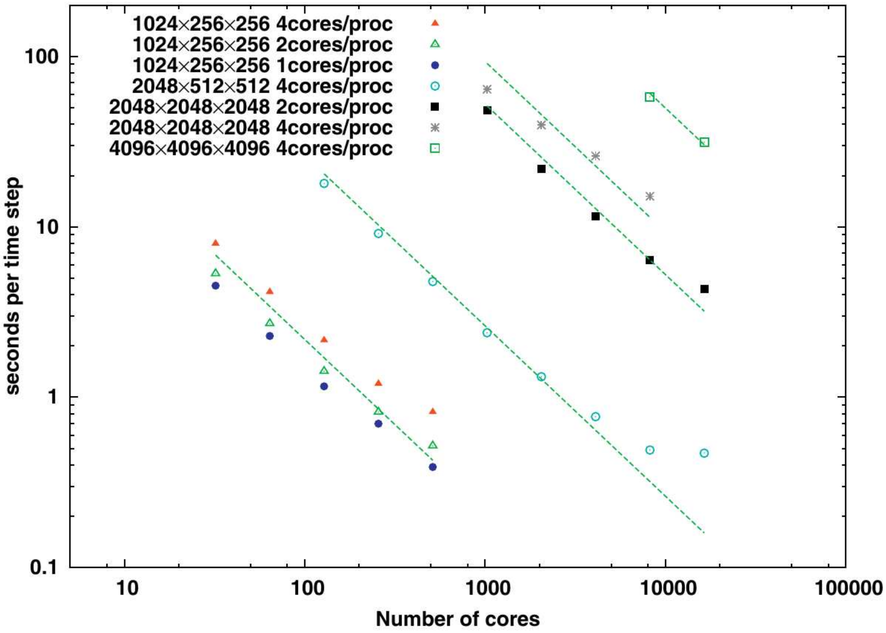


Figure 3.6: Strong scaling on HECToR (from Laizet & Li (2011)).

2010). While no incompatibility with parallel computing was encountered in the explicit time advancement, the spatial operators were more problematic due to their implicit nature. Indeed the evaluation at a given point requires the use of all neighbouring points. The problems are modelled on a three-dimensional Cartesian topology, using spatially implicit numerical schemes (as is the case with Incompact3d with compact $6^{th}$ order finite-difference). Due to the nature of these schemes, one is left with solving banded matrices (generally tridiagonal or pentadiagonal) linear system in the evaluation of spatial derivatives or computing spatial interpolations. The solution to this system then requires the inversion of a large tridiagonal matrix which is known to be time consuming because of repetitive communications. Two methods are at hand for performing such computations on distributed-memory systems. The first option is either to rely on distributed algorithms based on parallel tridiagonal solver or a parallel FFT algorithm processing distributed data. The second approach is based on the dynamical redistribution (transposition operations) of data among processors in order to apply serial algorithms in local memory. The latter approach is usually preferred due to its simplicity. Indeed, existing serial algorithms remain unchanged, porting serial code can be straightforward as long as the original code structure still holds, while the only major addition is the data transposition step.

The slab decomposition implies dividing the computational domain into equally sized subdomains in $z$ direction (allowing equal load balance) as presented in figure 3.7: In figure 3.7, a 3D domain is chosen to be decomposed in Y and X directions separately. On the left part, all the calculations in the X-Z planes can be performed in local memories while data along a Y mesh-line is distributed (state (a)). When it becomes necessary to compute along Y mesh-lines (for instance to perform Y-derivatives, or 1D FFTs along Y), one can redistribute the data among the processors to reach state (b) shown on the right of figure 3.7, in which any computation in Y becomes local. Using the standard MPI libraries the swapping operation (or transpose operation) between state (a) and state (b) is achieved using `MPI_ALLTOALL(V)`.

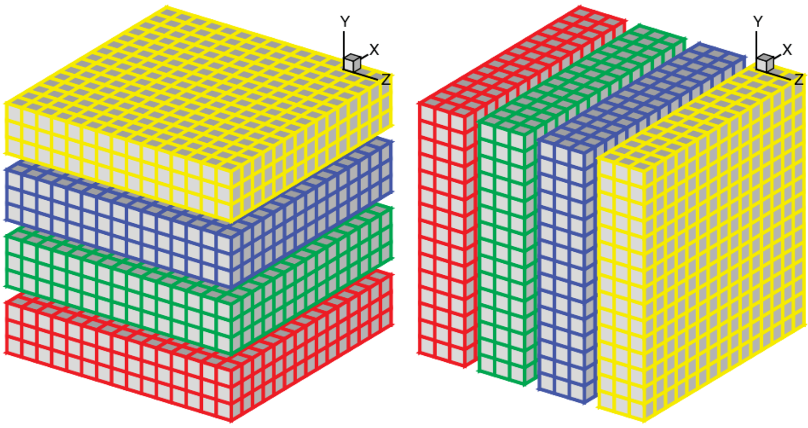


Figure 3.7: 1D domain decomposition using 4 processors.

The aforementioned description applies to the particular case of the slab version of the Incompact3d code. Each MPI-process is assigned to one of the subdomains to carry out simultaneously the time advancement. The values of all variables are then available to all subdomains. At this point, it is not possible to compute any derivative or interpolation in $z$ direction. A global transpose operation is necessary with a new DD in $y$ direction in order to get the relevant data onto the required MPI process. This operation allows for the computation in $z$ direction for each new slice. These global transpose operations need the parcelling of small blocks of data on each MPI process, labelling these with the address of the MPI process requiring the information, transferring the data, reading it by receiving MPI processes and finally reconstructing the flow field. These steps are all carried out using the specific MPI instruction `MPI_ALL_TO_ALL(V)`. From a general viewpoint, an All-to-All communication instruction is a computer communication method in which each sender passes messages to all receivers within a group. This is in contrast with the point-to-point communication in which each sender communicates with one receiver only. Therefore in MPI, this collective call ensures that all MPI processes send the same amount of data to each other and also receive the same amount of data from each other. This technique is also used in the parallel FFTW library implementation. The main shortcoming in the 1D-DD was the limitation imposed on the number of computational cores ($n_c$), i.e. $n_c < \min(n_y, n_z)$, because of the workload on each computational core and/or long wall clock time for production runs. This fact can be straightforwardly explained using a cubic mesh with size $N^3$. Indeed, the obvious constraint is that the maximum number of processors $N_{\text{procs}}$ that can be used in a 1D-DD should be $N$, since each slab has to contain at least one plane of data. For a cubic mesh with $10^9$ points, the latter constraint translates into $N_{\text{procs}} \leqslant 10^3$. This stands as a severe limitation since most of the high end supercomputing platforms are nowadays endowed with $\mathcal{O}(10^5)$ cores (list, 2016). In addition, since the systems treated in DNS are dealing with large arrays, memory becomes an issue if the workload cannot be distributed among a large number of processors. These issues led to the implementation of a dual domain decomposition strategy in Incompact3d.

In order to overcome these shortcomings, a 2D-DD has been implemented by Laizet & Li (2011). This decomposition (also called pencil, drawer or block decomposition) is an extension of the slab decomposition. The label (X, Y, or Z) of the pencil is related to the direction of the block. The library integrated in the code for the 2D-DD is called 2DECOMP&FFT (Li, 2012). It allows to apply the domain decomposition strategy for FFTs as well. Figure 3.8 shows the same three-dimensional domain considered for the slab decomposition, but this time partitioned into two dimensions simultaneously. This strategy is based on the 1D-DD but is much more complex although not slower that the latter, in terms of communications.

In the 1D-DD, the algorithm needed to swap between two states. Now in the 2D-DD, the new algorithm needs to carry out global transpose operations between three states, enabling up to six global transpose operations to be performed. Nevertheless, in the present version of the code, only four of these operations are used: (X-pencil↔Y-

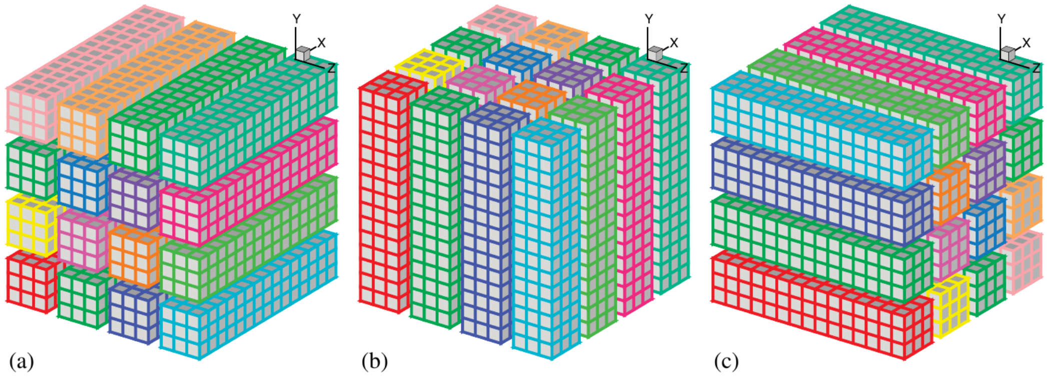


Figure 3.8: 2D domain decomposition example using a 4 × 3 processor grid:
(a) X-pencil, (b) Y-pencil, (c) Z-pencil.

pencil or (a)↔(b)), (Y-pencil↔Z-pencil or (b)↔(c)),(Z-pencil↔Y-pencil or (c)↔(b)), (Y-pencil↔X-pencil or (b)↔(a)). There are two separate communicator groups. For a $p_{\text{row}} \times p_{\text{col}}$ 2D-MPI grid, $p_{\text{row}}$ groups of $p_{\text{col}}$ MPI processors need to exchange data among themselves (X-pencil↔Y-pencil). Then $p_{\text{col}}$ groups of $p_{\text{row}}$ MPI processors need to exchange data among themselves (Y-pencil↔Z-pencil). All the derivatives and interpolations operations are carried out in one spatial direction at a time. This means that derivatives and interpolations in $x$ direction ($y$ direction, $z$ direction respectively) are performed in X-pencil (Y-pencil, Z-pencil respectively). By splitting the communications into sub-groups, it is possible to reduce significantly the time spent in global communications.

The 3D FFTs required by the Poisson solver are split into a series of 1D FFTs computed in one direction at a time. With the aim of reducing the number of global transpose operations, it is necessary to gather some derivatives/interpolations operations for each spatial direction and compute these simultaneously. The structure of the code is sketched in figure 3.9. This figure shows how the pencil swap operations are distributed along the code algorithm. For the Poisson solver in the spectral space, a single division needs to be performed and the modified wave numbers combined with the transfer functions are all independent from each other. When computing 3D forward FFTs, Z-pencil in the physical space is first considered, then the X-pencil in the spectral space and finally the Z-pencil again after the 3D FFT backward. This order is followed in order to reduce the global transpose operations. For the code using periodic boundary conditions in the three spatial directions, 55 global transpose operations need to be computed at each time step. This number may appear to be large but despite a large number of global transpose operations, this strategy is suitable for simulations using thousands of computational cores due to its favourable scalability features. The new strategy based on dual-DD offers two major advantages in terms of parallelization: first of all, parallelization is possible without reducing the order of the schemes, and scalability is excellent because, even though the schemes are implicit in space, there is no data communication (i.e. overlapping) at the boundaries of each subdomain. This parallelization has also benefited from the periodic

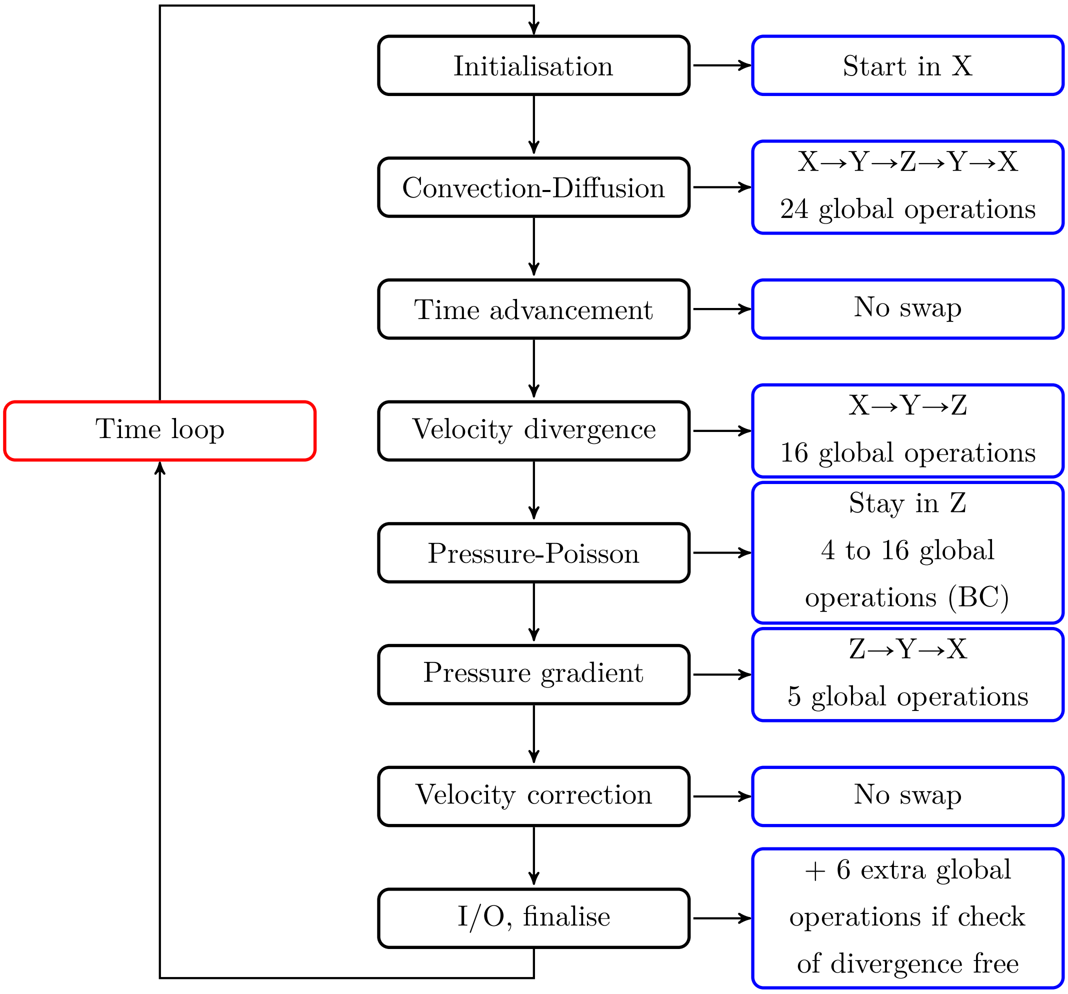


Figure 3.9: Structure of the Incompact3d code.

boundary conditions which are imposed in the lateral directions. In terms of performance, Incompact3d was tested on multiple supercomputing platforms and has proved to be highly scalable Laizet *et al.* (2010).

## 3.4 Governing equations

The DNS of the channel flow that are going to be carried out in the present study with Incompact3d are scaled in outer units. Therefore the governing equations are given here in vector form based on equations (2.5), (2.6) and (2.7) with the only difference that the convective terms are written using skew-symmetric form. The implementation of this term in the latter form enables to reduce aliasing errors, and remain energy conserving for the spatial discretization (Kravchenko & Moin, 1997).

$$\frac{\partial \mathbf{U}}{\partial t} = -\boldsymbol{\nabla} P - \frac{1}{2}[\boldsymbol{\nabla}(\mathbf{U} \otimes \mathbf{U}) + \mathbf{U} \cdot \nabla \mathbf{U}] + \frac{1}{R_p}\boldsymbol{\nabla}^2 \mathbf{U} \tag{3.16}$$

$$\boldsymbol{\nabla} \cdot \mathbf{U} = 0, \tag{3.17}$$

where $P(\mathbf{x}, t)$ represents the pressure field and $\mathbf{U}(\mathbf{x}, t)$ the velocity field. Note that equations (3.16)-((3.17) are the forced incompressible Navier-Stokes equations.

### 3.4.1 Time advancement

The time advancement of equation (3.16) can be expressed as

$$\frac{\mathbf{U}^{\circledast} - \mathbf{U}^k}{\Delta t} = a_k \mathbf{F}^k + b_k \mathbf{F}^{k-1} - c_k \boldsymbol{\nabla} \tilde{P}^k \tag{3.18}$$

$$\frac{\mathbf{U}^{k+1} - \mathbf{U}^{\circledast}}{\Delta t} = -c_k \boldsymbol{\nabla} \tilde{P}^{k+1} \tag{3.19}$$

$$\mathbf{F}^k = -\frac{1}{2}[\boldsymbol{\nabla}(\mathbf{U}^k \otimes \mathbf{U}^k) + \mathbf{U}^k \cdot \boldsymbol{\nabla} \mathbf{U}^k] + \nu \boldsymbol{\nabla}^2 \mathbf{U}^k \tag{3.20}$$

with

$$\tilde{P}^{k+1} = \frac{1}{c_k \Delta t} \int_{t_k}^{t_{k+1}} P \mathrm{d}t \tag{3.21}$$

The pressure is expressed through its time-averaged value on a given sub-step $c_k \Delta t$, indicated by the tilde symbol. $\tilde{P}^{k+1}$ is obtained by resolving the Poisson equation as follows:

$$\nabla^2 \tilde{P}^{k+1} = \frac{\nabla \cdot \mathbf{U}^{\circledast}}{c_k \Delta t} \tag{3.22}$$

This allows to verify the incompressibility condition after the velocity correction step in (3.19). The temporal discretization is based on explicit schemes. The coefficient couples $a_k$,$b_k$ ($c_k = a_k + b_k$), on $n_k$ sub-time steps, $k = 1, \cdots, n_k$, with $t_1 = t_n$ and $t_{n_k} = t_{n+1}$ ($\Delta t = t_{n+1} - t_n$ being the full time step). Depending on the flow configuration, two temporal schemes can be chosen, namely a third-order Runge-Kutta scheme, with $n_k = 3$ or a second-order Adams-Bashforth with $n_k = 1$ and ($a_1$,$b_1$)=(3/2,-1/2). The governing equations ((3.16)-(3.17)) are directly solved in a computational domain $L_x \times L_y \times L_z$ discretized on a Cartesian mesh of $n_x \times n_y \times n_z$ nodes. At each end of the domain in $x$, $y$ and $z$, periodic, free-slip, no-slip or open conditions can be imposed, depending on the chosen flow configuration. Periodic or free-slip boundary conditions can be set directly via the spatial discretization without specific care in the time advancement. However, the use of Dirichlet conditions on the velocity (for no-slip or open conditions) needs to be defined according to the time advancement procedure. The Dirichlet boundary conditions on the velocity can be imposed just after the first step in the time advancement ((3.18)) directly on $\mathbf{U}^{\circledast}$. This step allows a $\mathcal{O}(\Delta t^2)$ accuracy on the final velocity $\mathbf{U}^{k+1}$. To solve the pressure, conventional homogeneous Neumann conditions are used ($[U_{1,j,k} - U_{0,j,k}]/\Delta x = 0$, (Wilhelmson & Ericksen, 1977)). This procedure will be further detailed in the description of the pressure treatment using the Poisson equation. The Poisson equation is solved using spectral methods to obtain the pressure. Indeed, this comes from the fact that the incompressibility condition (3.17) is enforced at the end of each sub-time step in a fractional-step method:

$$\boldsymbol{\nabla} \cdot \mathbf{U}^{k+1} = 0, \tag{3.23}$$

### 3.4.2 Spatial discretization

In this section, the spatial discretization of the convective and viscous terms are described. The code uses $6^{th}$ order compact finite difference schemes for the spatial discretization. The mesh is Cartesian and in the channel flow configuration, it is stretched in the wall-normal direction. The grid is collocated for convective and diffusive terms while being staggered for the Poisson equation and for the correction step (within the fractional-step method). Considering a simplified one dimensional problem, the first derivative on a collocated mesh is given by:

$$\alpha f'_{i-1} + f'_i + \alpha f'_{i+1} = a\frac{f_{i+1} - f_{i-1}}{2\Delta x} + b\frac{f_{i+2} - f_{i-2}}{4\Delta x}. \tag{3.24}$$

By choosing $\alpha = 1/3$, $a = 14/9$ and $b = 1/9$, the first derivative of $f$ is $6^{th}$ order accurate achieving a so-called quasi-spectral accuracy. Note that the $6^{th}$ order scheme, combines a 3-5 stencil, which is a compromise between the order and the extent of the scheme. It presents a Hermitian structure when $\alpha \neq 0$. A similar reasoning can be applied for the computation of the second derivative:

$$\alpha f''_{i-1} + f''_i + \alpha f''_{i+1} = a\frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta x^2} + b\frac{f_{i+2} - 2f_i + f_{i-2}}{4\Delta x^2} + c\frac{f_{i+3} - 2f_i + f_{i-3}}{9\Delta x^2}, \tag{3.25}$$

for this scheme to be $6^{th}$ order accurate, $\alpha = 2/11$, $a = 12/11$, $b = 3/11$ and $c = 0$, achieving spectral-like accuracy, as in the first derivative computation. The advantage of this scheme is that it can control the aliasing errors when using it in the computation of the viscous term. It can be defined in such a way that it is over-dissipative at the highest wave numbers. In a more practical point of view, the previous schemes and the following schemes modified in order to account for specific boundary conditions are part of the `derive.f90` file. The term denoted in (3.20) as $\mathbf{F}^k$, which embodies the convective and viscous terms is computed in the code by using the discretized first and second order derivatives given by (3.24)-(3.25).

Concerning the boundary conditions, the code offers the choice between four different configurations. These configurations can be applied indifferently to the three spatial directions. The periodic and free-slip (symmetric/antisymmetric condition) boundary conditions enable the use of the schemes given by (3.24)-(3.25) for all the nodes. This means that these two schemes need only to be modified at the edge of the considered domain, i.e. at $i = 1$ and $i = n_x$ (note that the index is not specific to the $x$ direction as it stands for a general index). At these points, based on the representation of first and second derivatives, values of $f_0$, $f_{-1}$, $f_{n_x+1}$, and $f_{n_x+2}$ are needed, hence requiring the computation of so-called ghost-cells. However, the solution is provided by replacing each of these terms by their respective counterparts: $f_1$, $f_2$, $f_{n_x-1}$, and $f_{n_x-2}$. For instance for a periodic boundary condition, the following substitution are used:

$$f_0 \rightarrow f_{n_x} \quad , \quad f_{-1} \rightarrow f_{n_x-1} \quad , \quad f'_0 \rightarrow f'_{n_x} \quad , \quad f''_0 \rightarrow f''_{n_x} \tag{3.26}$$

the free-slip requires the following substitutions (symmetric/antisymmetric conditions respectively)

$$f_0 \rightarrow f_2 \quad , \quad f_{-1} \rightarrow f_3 \quad , \quad f_0^{'} \rightarrow -f_2^{'} \quad , \quad f_0^{''} \rightarrow f_2^{''} \tag{3.27}$$

and

$$f_0 \rightarrow -f_2 \quad , \quad f_{-1} \rightarrow -f_3 \quad , \quad f_0^{'} \rightarrow f_2^{'} \quad , \quad f_0^{''} \rightarrow -f_2^{''} \tag{3.28}$$

The above substitutions follow similar schemes for the other edge of the domain. In the specific the channel flow configuration, no-slip boundary conditions apply in $y$ direction (Dirichlet for the velocity), while periodic boundary conditions are used in $x$ and $z$ directions. In the no-slip boundary condition case, no intrinsic assumption is drawn. Therefore no ghost-cells are stepping in. Single-sided formulations are devised to approximate the first and second derivatives:

$$f_1^{'} + 2f_2^{'} = \frac{1}{2\Delta x}(-5f_1 + 4f_2 + f_3), \tag{3.29}$$

$$f_1^{"} + 11f_2^{''} = \frac{1}{\Delta x^2}(13f_1 - 27f_2 + 15f_3 - f_4). \tag{3.30}$$

The schemes given in equations (3.29))-(3.30) are $3^{rd}$ order. For the adjacent nodes, a three-point formulation based on Padé schemes is used:

$$\frac{1}{4}f_1^{'} + f_2^{'} + \frac{1}{4}f_3^{'} = \frac{3}{2}\frac{f_3 - f_1}{2\Delta x} \tag{3.31}$$

$$\frac{1}{10}f_1^{''} + f_2^{''} + \frac{1}{10}f_3^{''} = \frac{6}{5}\frac{f_3 - 2f_2 + f_1}{\Delta x^2}, \tag{3.32}$$

these schemes being $4^{th}$ order accurate.

The use of these Hermitian schemes (since in the formulation $\alpha \neq 0$), involves inverting tridiagonal matrices. The most computationally expensive derivative calculation relates to the choice of periodic boundary conditions, as a cyclic matrix (special kind of Toeplitz matrix, where each row is a cyclic shift of the row above it) has to be inverted. However, this step remains cheaper compared to pseudo-spectral codes, due to the presence of nonlinear convective terms.

### Compact finite difference schemes and grid stretching

Before describing the resolution of the Poisson equation, some light is shed over the grid staggering, the modification of operators in this configuration and the physical/spectral equivalence of these operators Lele (1992), as these are necessary in the derivation of the Laplacian. In the code, the velocity components always keep the same position. The staggering is applied on the pressure through a shift of its location by half a mesh in all spatial directions. The operators staggered by half a mesh (here $\Delta x/2$ since a simplified 1D version is considered) are given as:

$$\alpha f'_{i-1/2} + f'_{i+1/2} + \alpha f'_{i+3/2} = a\frac{f_{i+1} - f_i}{\Delta x} + b\frac{f_{i+2} - f_{i-1}}{3\Delta x}, \tag{3.33}$$

where the coefficients are defined as $\alpha = 9/62$, $a = 63/62$ and $b = 17/62$. The spectral-like behaviour of this scheme needs an interpolation to be defined:

$$\alpha f^I_{i-1/2} + f^I_{i+1/2} + \alpha f^I_{i+3/2} = a\frac{f_{i+1} - f_i}{2} + b\frac{f_{i+2} - f_{i-1}}{2} \tag{3.34}$$

this combination with $\alpha = 3/10$, $a = 3/4$ and $b = 1/20$ is $6^{th}$ order accurate. One needs to focus on the Fourier operators to define the equivalence of operators between physical and spectral space. In the following, a simplified 1D case that can be generalized to higher dimensions is considered. If a function $f$ is periodic in a $[0, L_x)$ physical domain:

$$f(x + L_x) = f(x) \tag{3.35}$$

then the discrete Fourier transform can be defined as

$$\hat{f}_l = \frac{1}{n_x}\sum_{i=1}^{n_x} f_i e^{-\imath k_x x_i} \quad , \quad \forall l \in [\![-n_x/2, n_x/2 - 1]\!] \tag{3.36}$$

its inverse being defined as (where the wave number is $k_x = 2\pi/L_x$)

$$f_i = \sum_{l=-n_x/2}^{n_x/2-1} \hat{f}_l e^{\imath k_x x_i}. \tag{3.37}$$

In order to obtain the shifted grid transforms, one needs only to make the following substitution $i \to i + 1/2$. When $f$ is symmetric in $x = 0$ and $x = L_x$

$$f(x) = f(-x) \quad , \quad f(x + L_x) = f(L_x - x) \tag{3.38}$$

the discrete cosine Fourier transform is defined with $k_x = \pi l/L_x$ as

$$\hat{f}_l = \frac{2}{n_x - 1}\left[\frac{1}{2}f_1 + \sum_{i=2}^{n_x-1} f_i \cos(k_x x_i) + \frac{1}{2}(-1)^l f_{n_x}\right] \quad , \quad \forall l \in [\![0, n_x - 1]\!] \tag{3.39}$$

the inverse is given by

$$f_i = \frac{\hat{f}_0}{2} + \sum_{l=1}^{n_x-2} \hat{f}_l \cos{(k_x x_i)} + \frac{1}{2}(-1)^{i+1}\hat{f}_{n_x-1}, \tag{3.40}$$

the shifted equivalent transforms into

$$\hat{f}_l = \frac{2}{n_x-1}\sum_{i=1}^{n_x-1} f_{i+1/2}\cos{(k_x x_{i+1/2})} + \frac{1}{2}(-1)^l f_{n_x} \quad , \quad \forall l \in [\![0, n_x-2]\!] \tag{3.41}$$

and the inverse expression

$$f_{i+1/2} = \frac{\hat{f}_0}{2} + \sum_{l=1}^{n_x-2} \hat{f}_l \cos{(k_x x_{i+1/2})}. \tag{3.42}$$

From the previous definitions, the Fourier coefficients relative to the compact scheme equation (3.24) can be related to the Fourier coefficients defined in equation (3.37)

$$\hat{f}_l' = \imath k_x' \hat{f}_l \tag{3.43}$$

All the spectral like accuracy concept becomes clear through the modified wave number:

$$k_x'\Delta x = \frac{a\sin{(k_x\Delta x)} + b/2\sin{(2k_x\Delta x)}}{1+2\alpha\cos{(k_x\Delta x)}}, \tag{3.44}$$

which relates to the true wave number $k_x$. A similar formulation can be derived for the symmetric/antisymmetric case using the cosine transform:

$$k_x'\Delta x = \frac{2a\sin{(k_x\Delta x/2)} + 2b/3\sin{(3k_x\Delta x/2)}}{1+2\alpha\cos{(k_x\Delta x)}}, \tag{3.45}$$

This process can be undertaken for deriving the mid-point interpolating operators ((3.34))

$$\hat{f}_l^I = T_x(k_x\Delta x)\hat{f}_l, \tag{3.46}$$

the relation is then based, not on a modified wave number, but a transfer function $T_x(k_x\Delta x)$:

$$T_x(k_x\Delta x) = \frac{2a\cos{(k_x\Delta x/2)} + 2b/3\cos{(3k_x\Delta x/2)}}{1+2\alpha\cos{(k_x\Delta x)}}. \tag{3.47}$$

In summary, the usual equivalence principle between multiplication in Fourier space and derivation/interpolation in physical space is given through the aforementioned definitions when approximating with central finite difference schemes given by (3.24), (3.34)-(3.35). The equivalence mentioned is exact. This means that computing the derivative with (3.34) in physical space with the required boundary conditions leads to the same result given by using (3.44)).

### Pressure discretization

One of the main difficulties, in the treatment of Navier-Stokes equations is the enforcement of the incompressibility condition. From the fractional-step method, it follows that equation (3.22) needs to be solved. This process can be highly time consuming when using compact finite difference schemes, which would discourage the use of such schemes in favour of second-order schemes. However, when this computation is carried out in Fourier space the transformed operations are more straightforward in terms of implementation and less computationally expensive, since Fast Fourier Transforms (FFT) algorithms are used. This framework allows the derivation of solutions from a Poisson equation with homogeneous Neumann conditions (Swarztrauber (1977),Wilhelmson & Ericksen (1977)). The principle of this strategy is based on the previous references.

The pressure can now be further described. In Fourier space, a simplified three-dimensional form can be written as

$$\hat{P}_{lmn} = \frac{1}{n_x n_y n_z} \sum_i \sum_j \sum_k P_{ijk} W_x(k_x x_i) W_y(k_y y_j) W_z(k_z z_k), \tag{3.48}$$

along with its inverse

$$P_{ijk} = \sum_l \sum_m \sum_n \hat{P}_{lmn} W_x(-k_x x_i) W_y(-k_y y_j) W_z(-k_z z_k). \tag{3.49}$$

In (3.48))-(3.49), $W_x$, $W_y$, $W_z$ functions are part of an orthogonal basis and $(k_x, k_y, k_z)$ are the wave numbers. In the case of periodic boundary conditions, these functions correspond to the usual FFT definition, while they are defined as cosine FFT when applying free-slip or $\mathbf{U}/P$-Neumann boundary conditions (either in their collocated or staggered version). Since FFT routines can be found in various scientific libraries, the Fourier transforms that need to be performed, are relatively easy to implement. The first step in solving the Poisson equation (3.22), is to calculate the r.h.s of this equation. The derivatives involved in the r.h.s are computed with collocated or staggered schemes. This ensures compatibility between physical and spectral space operators. Once the required Fourier transform operations have been applied to $\mathcal{D} = \boldsymbol{\nabla} \cdot \mathbf{U}^{\circledast\circledast}$, solving the Poisson equation implies only a division for each Fourier mode $\hat{\mathcal{D}}_{lmn}$ by $F_{lmn}$

$$\hat{\hat{P}}^{k+1}_{lmn} = \frac{\hat{\mathcal{D}}_{lmn}}{F_{lmn}}, \tag{3.50}$$

Depending on the chosen grid configuration, the expression is consistently changed. For a collocated configuration

$$F_{lmn} = -(k_x^{'2} + k_y^{'2} + k_z^{'2})c_k\Delta t, \tag{3.51}$$

($k_x^{'}$,$k_y^{'}$,$k_z^{'}$) stand for the modified wave numbers in the spatial directions. In the partially staggered case, $F_{lmn}$ includes the mid-point interpolation operator by using the relevant transfer functions

$$F_{lmn} = -[(k_x^{'}T_yT_z)^2 + (k_y^{'}T_xT_z)^2 + (k_z^{'}T_xT_y)^2]c_k\Delta t. \tag{3.52}$$

Note that for the collocated case ((3.48)), there are various configurations for the set ($k_x$,$k_y$,$k_z$) for which $F_{lmn}$ is 0 in (3.47). This specific case is the singular modes of the pressure where $k_x^{'2} + k_y^{'2} + k_z^{'2} = 0$. Considering $x$ direction, the modified wave number $k_x^{'}$ given in (3.44) (collocated case) can be 0 for two cases: $k_x = 0$ and $k_x\Delta x = \pi$. The former is the cutoff wave number. In addition to the trivial singular mode ($k_x$,$k_y$,$k_z$)=(0,0,0), seven other singular modes can be deduced in a three-dimensional configuration. This is due to the simultaneous cancellation of the three modified wave numbers ($k_x^{'}$,$k_y^{'}$,$k_z^{'}$)=0. When solving the Poisson equation in physical space, the aforementioned singular mode significantly affects the performance of the solver and trigger non-physical oscillations for the pressure.

The fact that the Poisson equation is solved in spectral space, allows for singular modes to be discarded by skipping all the modes where $k_x^{'2} + k_y^{'2} + k_z^{'2} = 0$. This description focused on the collocated case. For the partially staggered configuration, the situation is not as complex as the collocated case. Indeed, the modified wave number given in this case by (3.44) vanishes only for $k_x = 0$ and peaks at $k_x\Delta x = \pi$. Four additional singular modes, can though be identified. This comes from the cancellation of the transfer functions $T_x$, $T_y$, $T_z$ at the following respective values: $k_x\Delta x = \pi$, $k_y\Delta y = \pi$ and $k_z\Delta z = \pi$. The singular modes trigger, as in the collocated case, spurious oscillations in the oblique directions w.r.t the Cartesian mesh. However, in the domain where these singular modes appear, i.e. $(k_x^{'}T_yT_z)^2 + (k_y^{'}T_xT_z)^2 + (k_z^{'}T_xT_y)^2 = 0$, the amplification is less pronounced than in the fully collocated case.

The correction step in (3.19) is performed by computing the modified pressure in physical space using inverse Fourier transform (3.49). After this step, the modified pressure gradient $\boldsymbol{\nabla}\tilde{P}^{k+1}$ is computed using either a collocated or a staggered scheme. Once the correction step completed, the divergence-free conditions can be enforced up to machine precision.

As is commonly adopted in DNS simulations, the wall-normal direction is stretched such that the points near the wall are clustered near the top and bottom walls, while remaining coarser in between. A difficulty arises for the treatment of the pressure when stretching is carried out. Indeed, the procedure for the discretization of the pressure is only valid for a regular mesh in the three spatial directions. However, as a stretched grid needs to be used in the wall-normal direction, the aforementioned discretization is no longer valid. To overcome this issue, a one-to-one operation mapping between the physical and the computational grids needs to be used.

A mapping is a topological operation, used to alter the geometry such that the function to which it applies is smoother and therefore more regular in the transformed coordinate system. This allows to space the mesh points uniformly in the new coordinate system. For instance, if $\xi$ is the computational coordinate, $z$ the physical coordinate, then a mapping can be defined as $z = h(\xi)$.

However, as Fourier transforms are involved in the computation of the pressure, a general mapping can introduce a serious difficulty in terms of algorithm and computational cost. Indeed, the main problem is linked to the use of a product in physical space that corresponds to a convolution product of the in spectral space. For this reason, the spectral equivalence based on the modified wave number needs to be reconsidered. Following the stretching technique introduced by Cain *et al.* (1984), the Poisson solver is based on a specific function mapping and expressed using only few Fourier modes. In this way, the spectral and non-iterative nature of the pressure treatment can be preserved without significant loss of accuracy.

In order to understand the main ideas of this mapping, the analysis is restricted to a one-dimensional case. First it is assumed that $y$ is the physical coordinate while $s$ is the computational coordinate such that $y = h(s)$, $0 \leqslant s \leqslant 1$, $0 \leqslant y \leqslant L_y$. The function $h(s)$ maps the equally spaced coordinate $s$ with the stretched physical space coordinate $y$. From the definition of the mapping, the first and second derivatives can be obtained by simply using chain rule:

$$\frac{\partial f}{\partial y} = \frac{1}{h'}\frac{\partial f}{\partial s}, \tag{3.53}$$

$$\frac{\partial^2 f}{\partial y^2} = \frac{1}{h'^2}\frac{\partial^2 f}{\partial s^2} - \frac{h''}{h'^3}\frac{\partial f}{\partial s}. \tag{3.54}$$

Expressed in physical space, these rules are straightforward to implement for compact finite difference schemes where the finite difference operations are performed on the regular coordinate $s$ (instead of $y$). The issue comes from the treatment of the Poisson equation which requires to define similar operations in spectral space. Indeed, for ensuring the divergence free condition, the Laplacian in the Poisson equation has to be discretized using an iterated first derivative operator, based on (3.53). In practice, the multiplication $\frac{1}{h'}\frac{\partial f}{\partial s}$ has to be expressed in spectral space. Assuming that $f$ can be represented by one of the discrete Fourier transforms previously given, the regular coordinate being $s$, the objective is to define appropriately the Fourier representation of the metric $1/h^{'}$. For completeness, the example for $f$ and $1/h^{'}$ is treated:

$$f_i = \sum_{m=-n_y/2}^{n_y/2-1} \hat{f}_m e^{-\imath k_m s_i}, \tag{3.55}$$

and the metric can be written in terms of a Fourier series as well:

$$\frac{1}{h^{'}_i} = \sum_{m=-n_y/2}^{n_y/2-1} \hat{a}_m e^{-\imath k_m s_i}, \tag{3.56}$$

along with the $y$ derivative of $f$

$$\left.\frac{\partial f}{\partial y}\right|_i = \sum_{m=-n_y/2}^{n_y/2-1} \left.\widehat{\frac{\partial f}{\partial y}}\right|_m e^{-\imath k_m s_i}, \tag{3.57}$$

with $k_m = 2\pi m$ the wave number associated with the uniform distribution of $n_y$ points $s_i = (i-1)/n_y$ on the regular coordinate $s$. Then the expression in spectral space is given by

$$\left.\frac{\partial f}{\partial y}\right|_i = \frac{1}{h^{'}_i}\left.\frac{\partial f}{\partial s}\right|_i, \tag{3.58}$$

This leads to the following convolution product expression (with $-n_y/2 \leqslant m \leqslant n_y/2-1$)

$$\left.\frac{\partial f}{\partial y}\right|_m = \sum_{p=-n_y/2}^{n_y/2-1} \hat{a}_p \left.\widehat{\frac{\partial f}{\partial s}}\right|_{m-p}, \tag{3.59}$$

The drawback of the convolution operation resides in the fact that it requires $n_y^2$. In terms of computational cost, the use of a such mapping appears to be penalising. In addition to this issue, the convolution product in equation (3.59) does not mimic exactly the multiplication in physical space as given by (3.58) due to the truncation of the modes outside from the range $-n_y/2 < m < n_y/2 - 1$. This truncation eliminates the aliasing errors. However, because they are present when the operation is performed in physical space as in (3.59)), the physical/spectral equivalence is lost. This is the reason why the method introduced by Cain *et al.* (1984) is used. In this technique, a metric, expressed with only three Fourier modes in spectral space, is prescribed:

$$\frac{1}{h'} = \frac{1}{L_y}\left[\frac{\alpha}{\pi} + \frac{1}{\pi\beta}\sin^2\pi(\gamma s+\delta)\right] = \frac{1}{L_y}\left[\frac{\alpha}{\pi} + \frac{1}{2\pi\beta}\left(1 - \frac{e^{2\imath\pi(\gamma s+\delta)} + e^{-2\imath\pi(\gamma s+\delta)}}{2}\right)\right]. \tag{3.60}$$

From there by integration, the expression of the mapping function is derived. Since this expression is complex, an example of a mapping function used in the code is given in figure 3.10 with the "at-scale" corresponding clustered channel flow configuration in figure 3.11. The clustering is so fine that in the present representation the very near-wall region lines cannot be differentiated.

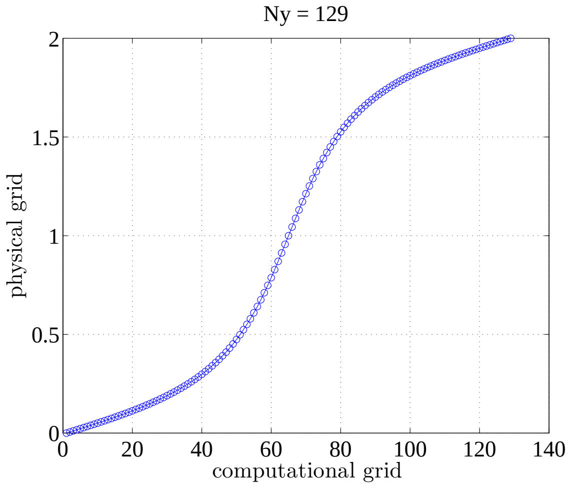


Figure 3.10: Example of a mapping function used for stretching the wall-normal direction.

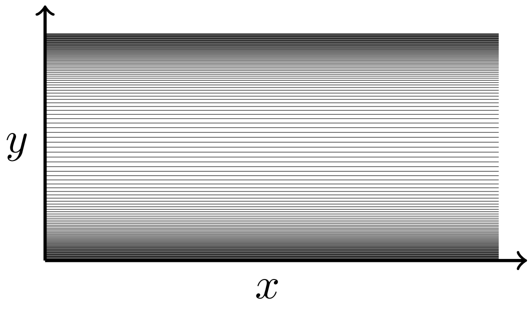


Figure 3.11: Channel flow configuration with clustering in the wall-normal direction.

## 3.5 Dicussion and recommendations

This section is intended to address further some key points in the study of turbulent channel flow performed through DNS. For each of the points that are presently addressed, a brief description is first given, and implications/recommendations relative to the Incompact3d code are discussed anywhere necessary.

### 3.5.1 Finite difference

When dealing with very fundamental fluid flow modelling problems, it is a necessity to consider very accurate numerical schemes for either DNS or LES. This is the reason why spectral methods relying on Fourier or Chebyshev computation are used to represent the flow with high fidelity in simple geometries. For instance, the first principle problems considering this methodology are homogeneous turbulence, transitional or turbulent channel flow and turbulence in a cavity. Each of these use Fourier and Chebyshev either separately or as a combination. However, when the complexity of the geometry becomes higher, the spectral approach (whether using Fourier or Chebyshev representation) falls short in terms of accuracy. This is the reason why one may direct the research in terms of high-order schemes whether spectral or finite differences. This task is not straightforward as discussed in Deville *et al.* (2002), in terms of accuracy and efficiency of these schemes for incompressible flows.

On the other hand for more realistic flow configurations which can thereby include complex geometries, DNS and LES still need to be improved. flows in non-academic geometries remain to be performed. For this purpose the Incompact3d code is the best alternative to other codes. It proposes a trade-off between spectral methods' accuracy and the more "pragmatic" approach of industrial codes. It uses high-order method applied to finite difference on a Cartesian mesh. The computation uses high-oder compact finite difference schemes and allows to capture the flow features whether in simple or complex geometries (through immersed boundary methods). The Cartesian geometry makes it very straightforward for the distribution of tasks in a parallel way.

In spatial discretization, the inolved error can be decomposed into two main components: differentiation error and error related to the nonlinearity of the equations. By using Fourier analysis, one can introduce the "modified wavenumber" concept. This allows to quantify the first type of error, i.e. the differentiation error. Writing a Fourier mode as $f = e^{ikx}$, it is possible to express the derivatives in terms of their exact (based on the spectral definition) and the one relying on finite difference methods expressed in terms of the modified wavenumber, $k'$. The difference between these two wavenumbers allows to precisely quantify the differentiation error. The second important point involves to resolve the smallest wavenumber with a given accuracy. The smallest wavenumber can be expressed as a multiplicative factor of the Kolmogorov length scale $\eta^*$. Within the modified wavenumber framework it is then possible to compute an estimate of the required mesh spacing for a given finite difference scheme to fufill a determined accuracy. Proceeding in this way, for second, fourth, sixth and spectral schemes, one can show that for a given accuracy to be reached, the mesh spacing is determined by the following ordering (expressed in terms of $\eta^*$): $\alpha_2\eta^* < \alpha_4\eta^* < \alpha_6\eta^* < \alpha_{\text{spectral}}\eta^*$, with $\alpha$ factors in $\mathbb{R}^+$. This ordering means that the second order scheme accuracy is lower and therefore requires larger resolution. The previous reasoning thereby justifies the choice for high-order compact finite difference schemes, in addition to the ease of implementation of finite difference schemes for parallel computing purposes.

### 3.5.2 Time advancement

As mentioned earlier, the time scales involved in turbulent flow span over a wide range, thereby categorizing such flows as stiff systems for time advancement. Stiff systems are usually seen in CFD computations (for instance steady aerodynamic flows). These are then carried out with implicit time advancement algorithms allowing the use of larger time steps. One would therefore question the use of explicit time advancement schemes as is the case in Incompact3d code and more particularly in the DNS of flow controlled systems.

Indeed imposing additional requirements on the wall, puts more constraints on the choice of the time step, subsequently narrowing the range for the choice of this parameter. The underlying constraint is more profound. Indeed, the necessity of accuracy in time over a wide scales' range prevents the use of substantially large time steps in DNS. This follows from the fact that using large timesteps implies that small scales can potentially incur large errors, subsequently leading to a corrupted solution. This can easily be seen by applying von Neumann stability analysis on the one-dimensional advection equation on a periodic domain. The solution can be represented as a sum of Fourier modes. Using the modified wavenumber ($k'$) framework, a complex amplification factor (i.e. the ratio of the solution at two consecutive time steps). Integration leads to an amplification factor that depends on the modified wavenumber and the CFL number. For this equation it can numerically be shown that there is a strong influence of the time step on small scale errors. However, this should not be considered as an obstacle to rule out implicit schemes and exclusively focus on explicit schemes for the time advancement in DNS. Indeed, it actually depends on the physics of the considered problem. The spatial grid have also to be considered to correctly represent the accurate range of frequencies. Therefore implicit time advancement schemes can be taken advantage of as the discrete equations encompass frequencies which are higher than those required by the physics. In DNS of incompressible wall-bounded flows, implicit time advancement is generally used for the viscous terms and explicit time advancement for the convection terms. The possibility for using a fully implicit algorithm in a turbulent channel flow was assessed by Choi & Moin (1994). They showed that choosing very large timesteps would cause the turbulent state to transition to a laminar state. However, one has to keep in mind that for the particular case of flow control through riblets (Choi *et al.*, 1993), fully implicit schemes needed to be used due to the stringent clustering over the edges of the riblets. Such a scheme is necessary to be competitive CPU-wise. In our particular case, the code was exclusively used in an explicit time advancement version. The code scaling being very efficient, the question of implementing an implicit time advancement scheme was not the main issue even in the context of flow control. Simulation up to friction Reynolds number of 590 were carried out only by reducing the time step and still being able to obtain relevant results within a reasonable time frame. A newer semi-implicit version of the code has been implemented by Cédric Flageul. For further details, one can directly contact him or look at updates on the Incompact3d code webpage.

### 3.5.3 Choice for the Incompact3d code

This section briefly mentions a few technical points and tries to give in a more qualitative way the choices made from the start of this project with respect to the code. The Incompact3d code is a reliable code as from its very first version it has been used within different areas and has undergone several revisions and additions requiring benchmark tests to be carried out. For further details and applications one can refer to Laizet & Lamballais (2009); Laizet & Li (2011); Laizet *et al.* (2010); Laizet & Vassilicos (2011). One of the major step dates back to the work undertaken by Dr Sylvain Laizet during his PhD thesis. The modifications and implementation from there have given rise to the most advanced version the code which is still being used and from which the present project departed from. For the purpose of this PhD project, two other codes were tested in an early stage: the code by Orlandi and the one by Gibson (2006*b*). Both of these codes proved to be inefficient in terms of computation as they were implemented in serial. From there it became evident that a code involving a parallel strategy was a requirement since a large number of flow control strategies needed to be implemented involving a costly validation process in terms of CPU-time usage. Therefore the choice of the Incompact3d code became obvious once its capabilities more particularly in terms of parallel implementations were studied and assessed first on the HECToR supercomputer. Not only was it very efficient, thanks to the pencil decomposition, but the understanding of the implementation was made really clear. Indeed the pencil decomposition in the 3 directions and the exchange of information among these, enables to carry out implementations locally and pass information efficiently through the transpose operations among the pencils.

During this project the code was tested on 4 different clusters, Tier-1 to Tier-3. For all cases, the porting process was straightforward and didn't require a large number of steps in terms of tuning the code with respect to the platform. The most straightforward way relied on the Fortran GNU compiler with optimization option -O2. Indeed, there are other choices depending on the machine choice and on how aggressive the optimization needs to be. In this case additional preprocessing options must be tuned with respect to the architecture. In particular, for testing a new implementation the GNU compiler proved to be more stable than the Intel compiler. When running production simulations, the Intel compiler can be used with more aggressive pre-processing optimization options with -O3 along with -xAVX -axSSE4.1. Finally all the FFT operation (for instance for the Poisson equation resolution), were performed using the generic FFT relying on the 2D domain decomposition library. However, it is also possible to use the third party library called FFTW (Frigo & Johnson, 2005).

# Chapter 4

# Validation and implementation of wall-forcing methods

As this project focuses on flow control techniques, the Incompact3d code needs first to be benchmarked against several of these methods. Indeed, this code has never been used in a channel flow configuration along with control techniques. Therefore this is the first time that these techniques have been implemented in the Incompact3d code. Before describing the implementation of the control techniques, it is important to benchmark the uncontrolled turbulent channel flow configuration and check the consistency of the statistics with different grid resolution versus well-established turbulent channel flow (Kim *et al.*, 1987*a*; Moser *et al.*, 1999). This step is necessary as the flow control techniques rely on the uncontrolled flow skin-friction to quantify the drag-reduction enforced by the control technique.

The code sensitivity to the grid resolution is tested. For this purpose, six different cases are simulated and the statistics of these compared to the reference case by Moser *et al.* (1999); Moser (2007) The conventions and tools defined in 2 are being used in the present and subsequent chapters.

## 4.1 Simulations parameters

All the simulations are performed at constant mass flow rate within a box of size $(L_x, L_y, L_z) = (4\pi, 2, 4\pi/3)$. The Reynolds number based on the Poiseuille flow $R_p = 4200$ correspond-

| case | $n_x$ | $n_y$ | $n_z$ | $\beta$ |
|---|---|---|---|---|
| 1 | 128 | 129 | 128 | 0.259065151 |
| 2 | 256 | 129 | 128 | 0.259065151 |
| 3 | 128 | 129 | 256 | 0.259065151 |
| 4 | 256 | 129 | 256 | 0.259065151 |
| 5 | 128 | 129 | 128 | 0.114740949 |
| 6 | 128 | 257 | 128 | 0.259065151 |

Table 4.1: Overview of the parameters used for the simulations.

ing to a nominal friction Reynolds number of $R_{\tau,r} = 180$. In the context of DNS, it is necessary to adequately resolve all the scales. The periodicity in $x$ and $z$ imposes to have a large enough domain in these two directions in order to capture the largest scales. An additional constraint is to capture the scales based on the Kolmogorov scale. However, as mentioned in Moin & Mahesh (1998), this constraint is very strict as DNS that have been performed without resolving these scales have shown good consistency. Therefore the size of the channel flow and resolution are stemming from the DNS from by Moser *et al.* (1999). In table 4.1, Case 1 is taken as a benchmark to which all the five following cases will be compared, in order to assess the grid resolution sensitivity. Case 2,3 and 4 are performed to check the sensitivity in the streamwise and spanwise directions separately and using their combination. Case 5 and 6 are performed to check the sensitivity of the grid resolution in the wall-normal direction. This is the reason why the $\beta$ parameter necessary to build the stretching in the wall-normal direction are different from each other. In summary, Case 5 and Case 6 have the same $\Delta y_{\min}$ but different number of points in $y$ direction. These two cases will be compared to Case 1 which has the same resolution in $x$ and $z$ as Cases 5 and 6 but has a resolution that is twice as those used in Cases 5 and 6. This allows to check not only the resolution in $y$ but also the impact of the number of points for the same $\Delta y_{\min}$, thereby testing the form of the mapping. Note that the code being used for the computation of the $\beta$ stretching parameter assumes that the first adjacent node to the wall is located at $y^+ = 1$. This number contrasts with those used in Kim *et al.* (1987*a*); Moser *et al.* (1999); Moser (2007) where $y^+ \approx= 0.05$ for the first adjacent node. However, with this value the basic statistics are in excellent agreement with the aforementioned results and statistics involving derivatives of the fields are also benchmarked very accurately. Therefore this assumption for the first adjacent node from the wall is not an issue for obtaining the required results.

The fields are saved every `imodulo` time steps. For all the cases, this value is set such that $\Delta t \times$ `imodulo` $= 1.25$. Therefore the fields are saved every $t = 1.25$ which is equivalent to $10\nu^*/u_{\tau,r}^{*2}$. A checkpoint file is also generated in the case of any unexpected behaviour from the code. This file is saved at time intervals `isave` which generally larger than `imodulo`. The time advancement is based on a two-step Adams-Bashforth scheme. In its default version the code outputs the primitive variables fields, i.e. the streamwise, wall-normal, spanwise and pressure, along with the vorticity norm. These are labelled as `ux`), `uy`, `uz`, `pp` and `vort` respectively. In the newly reorganized version of the code undertaken in this work, various alternatives for saving fields have been added and are described in Appendix A. As one wants to spend as less possible time as possible in the generation of fields during a simulation, in the following only the primitive variables are computed. An external data analysis tool has been written from scratch. It uses the MPI standard and 2D-DD to post-process the data from the code. It allows to compute a large array of statistics based only on the primitive variables, making it unnecessary to generate other intermediate fields (e.g. derivatives of the velocity fields, vorticity, $\cdots$) as these are computed "online" in this data analysis tool. Note that one should stress the fact that these intermediate fields are not generated as output files but only generated in variables to be used for the statistics. For instance, the output of the DNS does not give the streamwise component of the vorticity. However, when the data is processed, this component is computed but not written as an output. In this way, as the derivatives in the post-processing tool are the same as the one used in the code for generating the primitive variables, the derivatives of the latter will be consistent in terms of their order. Indeed, one could simply compute the derivatives by first order finite difference schemes but it is best practice to keep consistency from the generation of data to its post-processing. Figure 4.1 shows the simplified steps carried out in the parallel diagnostics tool. The process of fields saving uses the features from MPI-I/O since the computation is undertaken on a so-called local grid (indexing relative to the domain decomposed Cartesian grid) and then reduced with MPI calls to get the fields on the global grid (indexing relative to the physical domain). As a large number of fields is being used, one can release the constraint over the time step which generally should be on

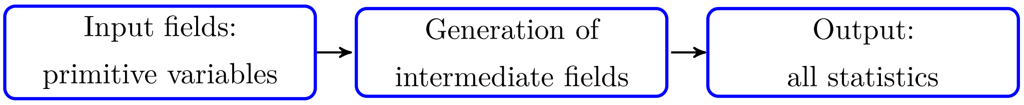


Figure 4.1: Steps undertaken in the parallel diagnostics tool.

the order of one eddy turnover time. As in Ricco & Hahn (2013), the turbulence statistics are computed after discarding the initial temporal transient during which the flow adapts to the new drag-reducing regime. The duration of the transient is estimated by direct observation of the time history of the space-averaged wall-shear stress and is typically of the order of 100 $h^*/U_p^*$ $(1150\nu^*/u_{\tau,r}^{*2})$. The statistics are calculated by averaging instantaneous saved flow fields for a total time window of $650h^*/U_p^*$ $(5015\nu^*/u_{\tau,r}^{*2})$. All the procedure for starting a simulation and obtain a turbulent state are described in AppendixA.

Before showing and comparing the statistics for the previous simulations, a few points need to be clarified in the way the statistics are computed. In turbulence analysis, averages in time and space operations are usually required. If $x$ is an event and $x_n$ the $n^{th}$ realisation of this event, then the ensemble average of $x$ is given by:

$$X = \breve{x} = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} x_i. \tag{4.1}$$

An ensemble average is a convenient theoretical concept since it is directly related to the probability density function, which can generally be obtained by the theoretical analysis of a given physical system. Preparing an infinite number of identical systems in a real situation is irrelevant but it is what would strictly be needed using the ensemble average definition. A time average is more directly related to experiments. Theoretical predictions based on ensemble average and time average are equivalent if and only if the system is a so-called ergodic ensemble. It is often mentioned that ensemble averaging and time averaging are identical for statistically stationary systems. In the diagnostics tool that has been implemented, the structure of the modules has been devised in a way that follows the previous equivalence statement for a statistically stationary system. In the cases considered in this study (whether in an uncontrolled channel flow or a controlled one), the statistics are computed in a time window in which a stationary state has been reached.

| Case | $\mathcal{P}_{sp} + \frac{100\mathcal{D}}{\mathcal{P}_{x,r}}(\%)$ | Case | $\mathcal{P}_{sp} + \frac{100\mathcal{D}}{\mathcal{P}_{x,r}}(\%)$ |
|---|---|---|---|
| 1 | 0.36 | 4 | 0.61 |
| 2 | 0.49 | 5 | 0.34 |
| 3 | 0.32 | 6 | 0.39 |

Table 4.2: Verification of the power balance equality.

## 4.2 Statistics

In this section the statistics of Cases 1 to 6 as summarised in Table 4.1 are shown and compared with the reference data from Moser *et al.* (1999). The code used in their simulations is a variation of the version from Kim *et al.* (1987*a*), as it the time discretization for the nonlinear terms are performed with a low-storage third-order Runge–Kutta scheme.

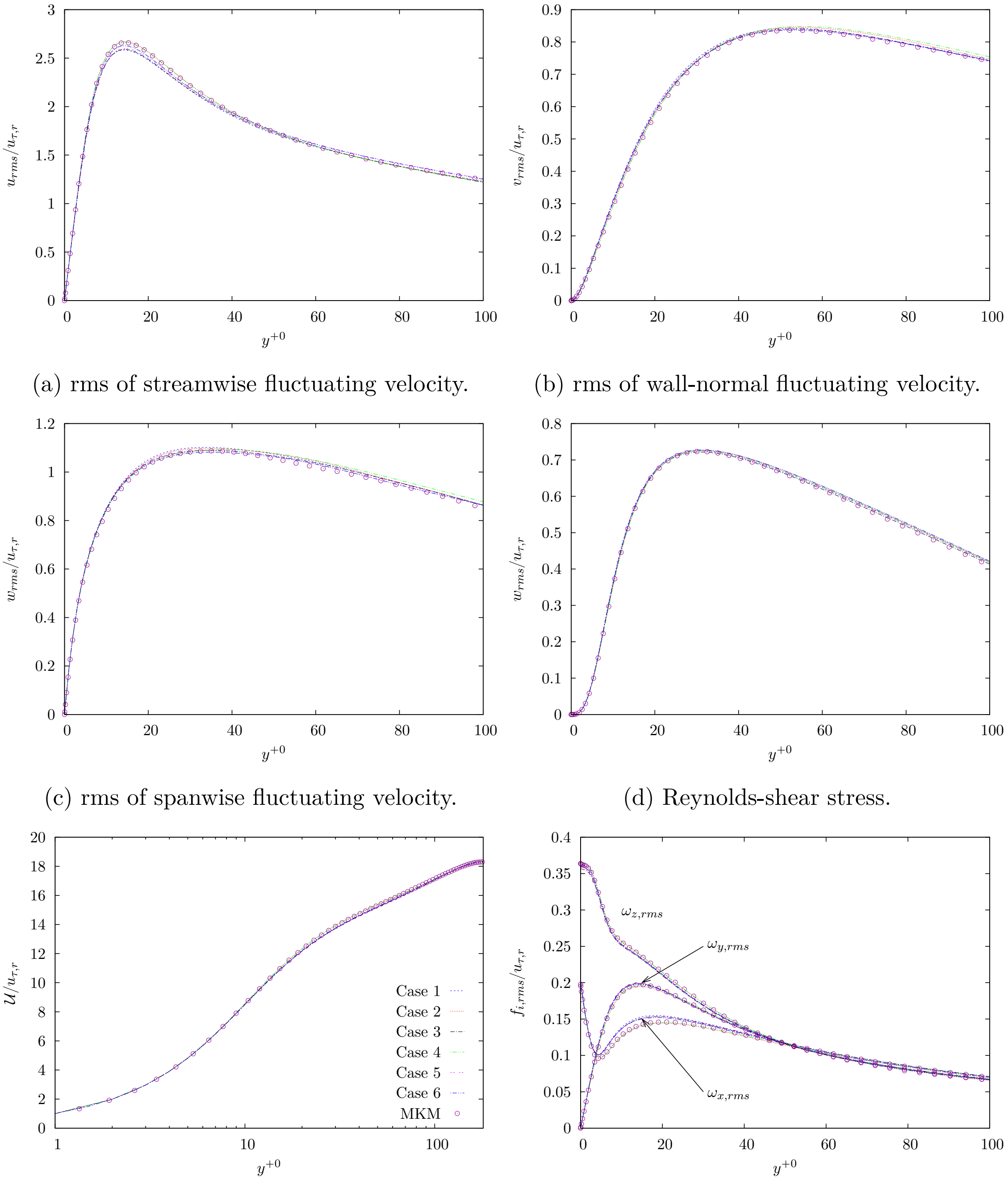
The power balance terms in equation (2.19) are also computed. For the uncontrolled flow one should verify

$$\mathcal{P}_{sp} + \frac{100\mathcal{D}}{\mathcal{P}_{x,r}} = 0. \tag{4.2}$$

The results are gathered in Table 4.2 In addition to the mean/rms of velocity and vorticity, the energy balance terms and two-point correlation in the spanwise direction are computed. The latter quantity is computed with

$$R_{uu,z}(\Delta z, y) = \frac{(L_x L_z)^{-1} \left[\overline{u'(x,y,z,t)u'(x,y,z+\Delta z,t)}\right]_{\mathcal{I}xz}}{u_{rms}^2}. \tag{4.3}$$

The correlation $R_{uu,z}$ is directly related to half-distance spacing of the streaks (Kline *et al.*, 1967; Panton, 2001). The parameter that is relevant in the flow control problems is the skin-friction coefficient. This quantity is also computed and as a further consistency check, it is computed by using the data from the Reynolds-shear stress. Indeed, the formulation proposed by Fukagata *et al.* (2002) allows from the knowledge of $uv_{rey}$ to obtain the skin-friction coefficient. This simple formulation decomposes the contribution to the skin-friction into a laminar and a turbulent part directly related to the Reynolds-shear stress production. Table 4.3 also shows that the skin-friction coefficient computed by space- and time-averaging the wall-shear stress matches the value computed through



(a) rms of streamwise fluctuating velocity.

(b) rms of wall-normal fluctuating velocity.

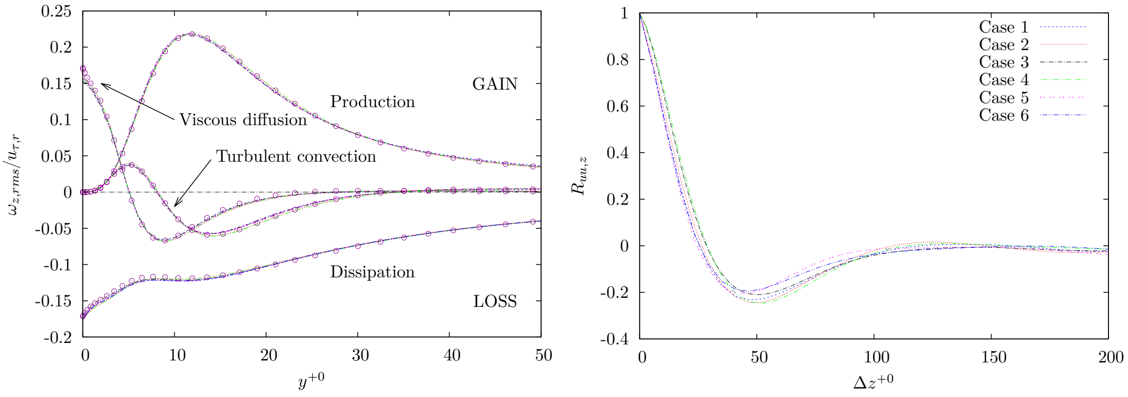
(c) rms of spanwise fluctuating velocity.

(d) Reynolds-shear stress.

(e) Mean streamwise velocity.

(f) rms of the vorticity components.

Figure 4.2: Rms and mean statistics.



(a) Energy balance terms.

(b) Spanwise correlation of the streamwise velocity component.

Figure 4.3: The contributions from the turbulent kinetic energy are shown along with the spanwise two-point correlation length.

| Case | Incompact3d | FIK | Case | Incompact3d | FIK |
|---|---|---|---|---|---|
| 1 | 8.18 | 8.176 | 4 | 8.10 | 8.101 |
| 2 | 8.14 | 8.145 | 5 | 8.17 | 8.169 |
| 3 | 8.18 | 8.189 | 6 | 8.18 | 8.184 |

Table 4.3: Skin-friction coefficient $C_f(\times 10^3)$ computed from the code using the definition (2.13) and through the FIK identity (B.13).

the Fukagata-Iwamoto-Kasagi (FIK) identity:

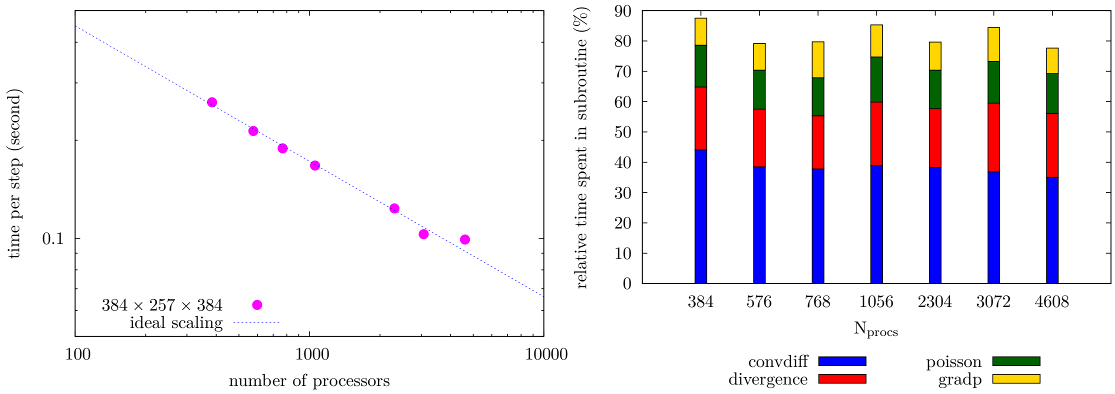
$$C_f = \frac{6}{\mathcal{U}_b R_p} - \frac{6}{\mathcal{U}_b^2} \int_0^1 (1-y) uv_{rey} \mathrm{d}y \tag{4.4}$$

## 4.3 Scaling of the code

In this section, a brief discussion and demonstration of the scaling of the code is given. Indeed, at the time of starting this project, the code was not available online and was released a few months after. Therefore, the code needed to be tested extensively before being used in this project and more particularly in a channel flow geometry. Therefore, in addition to all the tests performed in terms of grid resolution and computation of statistics, a few strong scaling tests were performed first on the HECToR supercomputer and then on the ARCHER supercomputer. The tests that are presented hereafter have been performed on the ARCHER supercomputer. Note that on ARCHER, there are 24 processors per node. In general the code has shown to perform 6 to 8 times faster on ARCHER than on HECToR. Obviously, these were not intended to be as large as those already given in the literature of the code (Laizet & Li, 2011), but were rather a useful step in the learning process of the code and the aspects relative to supercomputing. Ideally, one would need to use a very large number of mesh points to fully take advantage of the capability of the code. By this, it is meant that if the code has the capability to perform ideal linear scaling, this behaviour is best achieved by tackling a very large problem. However as the computing time was limited, choosing DNS problems with a large number of collocation points would have required a large amount of CPU hours. Therefore as a simple configuration, a turbulent channel flow at $R_{\tau,r} = 180$ was first generated to reach a stationary state and used as a starting point for the computation of the strong scaling tests. Table 4.4 gives the details of the parameters used in the strong scaling tests. For these tests there are approximately $38 \times 10^6$ mesh points. From the generated turbulent state, the simulations are restarted and performed for $5 \times 10^4$ time steps. The domain decomposition consisting in the set of parameters $(p_{\text{row}}, p_{\text{col}})$ for each case was determined through the auto-tuning mode. A probing module was incorporated into the code and used to obtain further information relative the time spent by each subroutine. An example of a file with this information is given in Listing 1. The number of processors for this strong scaling are multiple of the maximum number of processors per code available per node on the ARCHER supercomputer, i.e 24. Therefore the number of processors ($\mathrm{N}_{\text{procs}}$) used are

| $n_x$ | $n_y$ | $n_z$ | $\beta$ | $\Delta t^{+0}$ | compiler |
|---|---|---|---|---|---|
| 384 | 257 | 384 | 0.884740949 | 1.25 | GNU |

Table 4.4: Parameters used for the strong scaling test.



(a) Strong scaling in a channel flow configuration performed on the ARCHER supercomputer.

(b) Relative time spent in the most time consuming subroutines.

384, 576, 768, 1056, 2304, 3072, 4608. Listing 1 is based on one of the case from the strong scaling, with $N_{\text{procs}} = 768$. The total number of time steps are clearly accounted for in the probing tool. Note the number of times the call to divergence being twice as much as the total number of time steps, exhibiting the two-step fractional step method. In addition,

```
-----------------------------------------------------
Timer name       #calls    Time(s)    Incl.(%)
-----------------------------------------------------
Incompact3d       1        8457.216   100
convdiff          50000    3196.093   37.7913
pre_correc        50000    3.944      0.0466347
divergence        100000   1482.485   17.5292
poisson           50000    1060.357   12.5379
gradp             50000    1002.480   11.8535
corgp             50000    816.915    9.65938
channel           50000    318.307    3.76373
statistic         50000    413.188    4.88563
visu_insta        100      196.265    2.32068
```

Listing 1: Example of output for performance probing.

the main subroutine calls are shown here. Figure 4.4b shows the relative importance of the most expensive subroutines in terms of calls relative to the total simulation time for all the number of processors used in the strong scaling. It is clear that the subroutine performing the computation of the convective and diffusive terms is the most expensive as the computation spends about 35-40% of its overall time in this subroutine.

After the necessary step of benchmarking the code in the uncontrolled configuration, it is ready for the implementation of all the flow control techniques that are required in the present study. This implementation is incorporated to the reorganised version of the code as described in Appendix A making it a reliable automated suite for flow control problems under a channel flow configuration.

Before introducing the flow control work, it is necessary to benchmark the code against existing flow control techniques. Indeed, the base code, does not include any implementation of flow control methods. Therefore the implementation of four wall-forcing methods has been undertaken in Incompact3d, namely the spanwise wall oscillation, steady rotating discs, oscillating discs and opposition control technique. All of the following simulations have been performed after their implementation in the newly reorganised version of the code as described in Appendix A. In addition to this reorganised version and the implementation of all the wall-forcing methods, a parallel diagnostics tool was also implemented from scratch to compute all the relevant statistics and flow visualizations. This version is also described in Appendix A. This was therefore the occasion to test the reliability of the overall implementation.

## 4.4 Spanwise wall oscillation

The DNS of the incompressible NSE are used to study the turbulent wall-shear stress in a channel flow configuration, where the forcing is applied through spanwise sinusoidal oscillations of the walls. Several studies have focused on this method to control the flow (Laadhari *et al.*, 1994; Baron & Quadrio, 1996; Dhanak & Si, 1999; Quadrio & Ricco, 2004; Duggleby *et al.*, 2007). One of the early simulations of high frequency spanwise wall oscillations was performed by Akhavan *et al.* (1993). At $R_\tau = 200$, they obtained the maximum skin-friction drag reduction of 40%.

This study is based on the work by Quadrio & Ricco (2004), where a maximum drag reduction of 44.7% was obtained with a specific set of parameters, along with a large reduction of the mean streamwise wall friction. The wall forcing method is characterised through the amplitude of the spanwise oscillating velocity $W_m$ and its period $T$. The expression of the spanwise forcing is given by

$$w = W_m \sin\left(\frac{2\pi}{T}\right) \tag{4.5}$$

The set of parameters achieving the highest net power and the highest drag-reduction rate (cases 20 and 25 respectively in Quadrio & Ricco (2004)) are chosen to carry out these simulations as benchmarks. The turbulent state is first generated at a nomina $R_{\tau,r} = 200$ with $(L_x, L_y, L_z) = (21.0, 2.0, 4\pi/3)$ and $(nx, n_y, n_z) = (384, 129, 128)$. For Case 20, $(W_m^{+0}, T^{+0}) = (4.5, 125)$ while for Case 25 $(W_m^{+0}, T^{+0}) = (27, 125)$. In figure 4.5, the all the phases of the simulations are exhibited, along with the selected time windows for obtaining the skin-friction values, i.e. $C_{f,r}$ and $C_f$. Their computation results in $\mathcal{R} = 17.4\%$ for Case 20 and $\mathcal{R} = 44.8\%$ for Case 25. These values are in very good agreement with those obtained in Quadrio & Ricco (2004). For Case 20, the power spent was also computed using the parallel diagnostics tool resulting in $\mathcal{P}_{sp} = 9.6\%$ to be compared with Quadrio & Ricco (2004) value of $\mathcal{P}_{sp} = 9.9\%$. In figure 4.6, the amplitude of the vorticity at the wall are shown, just before enforcing the spanwise wall oscillations and at the end of the simulation where a new stationary state has been reached within the controlled phase. One can clearly notice a decrease in the intensity of these structures and the influence of the spanwise wall motion on the overall vorticity. Indeed

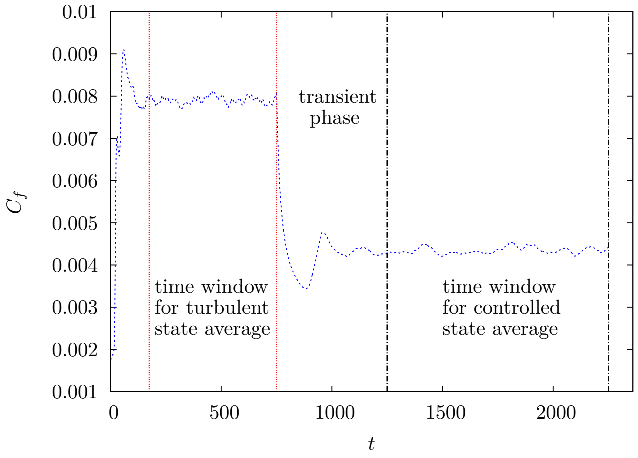


Figure 4.5: Skin-friction time trace including all the phases.

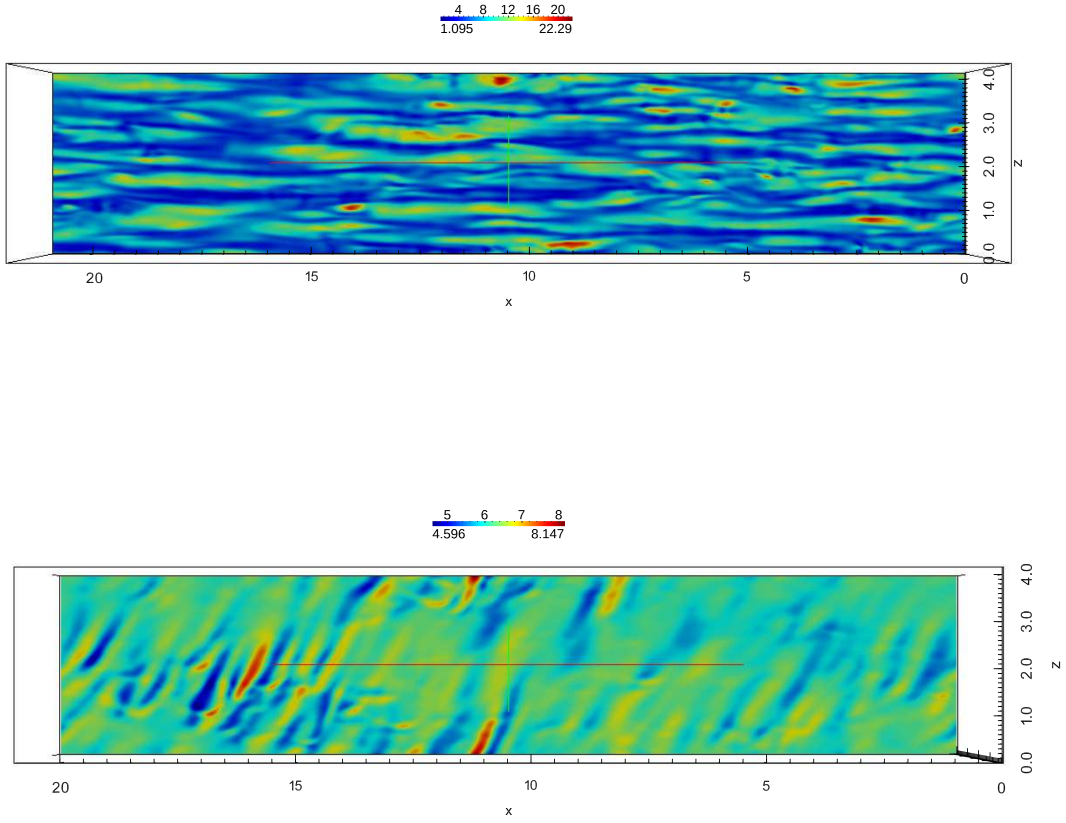


Figure 4.6: Amplitude of the vorticity at the wall before imposing spanwise wall forcing (top) and at the end of the controlled phase (bottom).

these structures move back and forth in the spanwise direction. This behaviour was also exhibited in Akhavan *et al.* (1993) and was identified as the responsible for the reduction in turbulence production. The crucial point is that these structures tend to realign to resume the production cycle of turbulence, whereby showing that the spanwise forcing is responsible for the breakup of this cycle, subsequently reducing the skin-friction drag.

## 4.5 Steady and oscillating flush-mounted discs

The second forcing method, consists in placing flush-mounted steady rotating discs at the wall. The first reference to such a forcing method was proposed as part of a patent by Keefe Keefe (1998). However, no experiment or numerical simulations were performed to assess the drag-reducing capability of this apparatus. Recently, Ricco & Hahn (2013) performed DNS of turbulent channel flow integrating this forcing at the wall. A maximum drag-reduction rate of 23%. The net power savings, based on a parametric study of the tip velocity and the diameter of the discs, gives about 10%. This figure takes into account the power needed for putting the discs into motion. The configuration is based on figure 1.2.

The velocity components of the discs ($u_{\text{disc}}$ and $w_{\text{disc}}$) that need to be passed to Incompact3d are generated with an external code as a data file. The procedure for generating these and passing them to the code are given in §A.3 of Appendix A. Figure A.1 gives a summary of these steps. The flow structure is then analysed using ParaView. The scripts and the automated procedure to obtain the flow visualizations are also detailed in Appendix A. Figure 4.7 shows the norm of the discs velocity components, i.e. $\sqrt{U_{\text{disc}}^2 + W_{\text{disc}}^2}$ flush-mounted at the wall of Case 24 in Ricco & Hahn (2013). For this case, $(L_x, L_y, L_z) = (6.79\pi, 2.0, 2.26\pi)$, $Re_{\tau,r} = 180$. The discs are characterised by a diameter $D^{+0} = 640$ and a tip velocity $W^{+0} = 9$. The sensitivity of the results have also been checked with different resolutions and show good consistency with each other and the case to which these are compared to, i.e Case 24 in Ricco & Hahn (2013). Setting the discs on the top and lower wall, these simulations are performed and then averaged over a time window of $t = 1450h^*/U_p^*$. The results are gathered in Table 4.5. The skin friction in Ricco & Hahn (2013)'s case is $C_f = 6.65 \times 10^{-3}$. The three-dimensional representation of the flow is computed by splitting the total velocity vector into three components as $\boldsymbol{U} = \boldsymbol{U}_{\text{m}} + \boldsymbol{U}_{\text{d}} + \boldsymbol{U}_{\text{t}}$. Here, $\boldsymbol{U}_{\text{d}} = (U_{\text{d}}, V_{\text{d}}, W_{\text{d}}) = \overline{\boldsymbol{U}} - \boldsymbol{U}_{\text{m}}$ with $\boldsymbol{U}_{\text{m}} = \left[\overline{\boldsymbol{U}}\right]_{\mathcal{I}xz}$ Proceeding in this way, it is then possible to obtain the discs flow representation as $\sqrt{U_d^2 + W_d^2}$ is computed. Their representation at different wall-normal level are given in figures 4.7 and 4.8. The same features that occurred in the simulations by Ricco & Hahn (2013) are obtained, namely the characteristic tube-shaped patterns at the centre and the edges of the flow. As the level in the wall-normal direction is increased, the disc flow on top of each disc progressively vanishes. The only parts that persist are the tubular patterns already seen in 4.8. These features are therefore not simply a numerical artefact but are related

| $n_x$ | $n_y$ | $n_z$ | $C_f \times 10^3$ |
|---|---|---|---|
| 384 | 129 | 256 | 6.62 |
| 480 | 129 | 224 | 6.62 |
| 384 | 129 | 320 | 6.61 |
| 512 | 257 | 320 | 6.63 |

Table 4.5: Impact on the skin-friction coefficient with different grid resolutions.

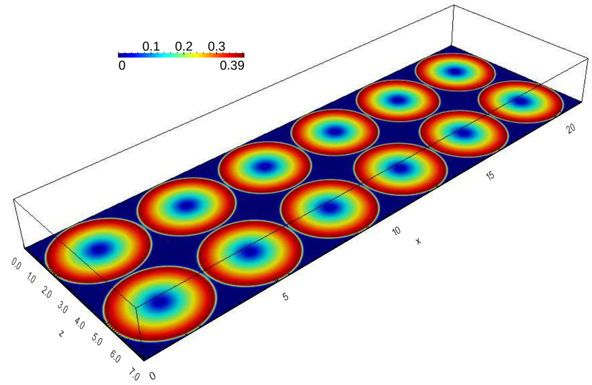


Figure 4.7: Velocity norm of the flush-mounted discs using Case 24 parameters from Ricco & Hahn (2013).

to the drag-reduction mechanism.

In addition to the drag reduction rate, another feature that is important within the flow control technique framework is the power spent. For this particular case $\mathcal{P}_{sp}$ can be expressed through the discs flow velocity as shown in (4.6)

$$\mathcal{P}_{sp}(\%) = -\frac{100R_p}{R^2_{\tau,r}\mathcal{U}_b}\left[U_\mathrm{d}(x,0,y)\left.\frac{\partial U_\mathrm{d}}{\partial y}\right|_{y=0} + W_\mathrm{d}(x,0,y)\left.\frac{\partial W_\mathrm{d}}{\partial y}\right|_{y=0}\right]_{\mathcal{I}xz} \tag{4.6}$$

This quantity was computed for this case and $\mathcal{P}_{sp} = 23.4\%$ to compared with $\mathcal{P}_{sp} = 23.3\%$ in Ricco & Hahn (2013).

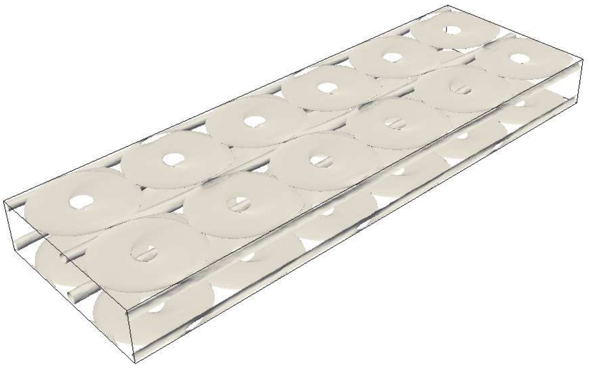


Figure 4.8: Isocontour of the discs flow with $\sqrt{U_{\mathrm{d}}^2 + W_{\mathrm{d}}^2} = 0.09$.

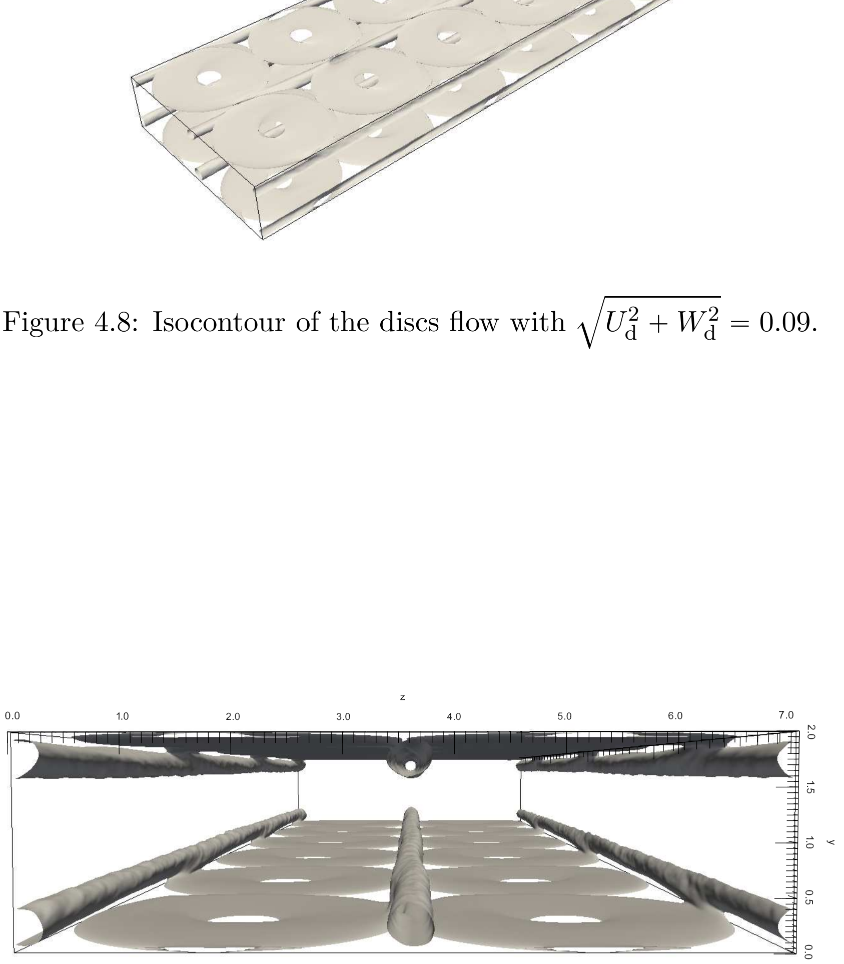
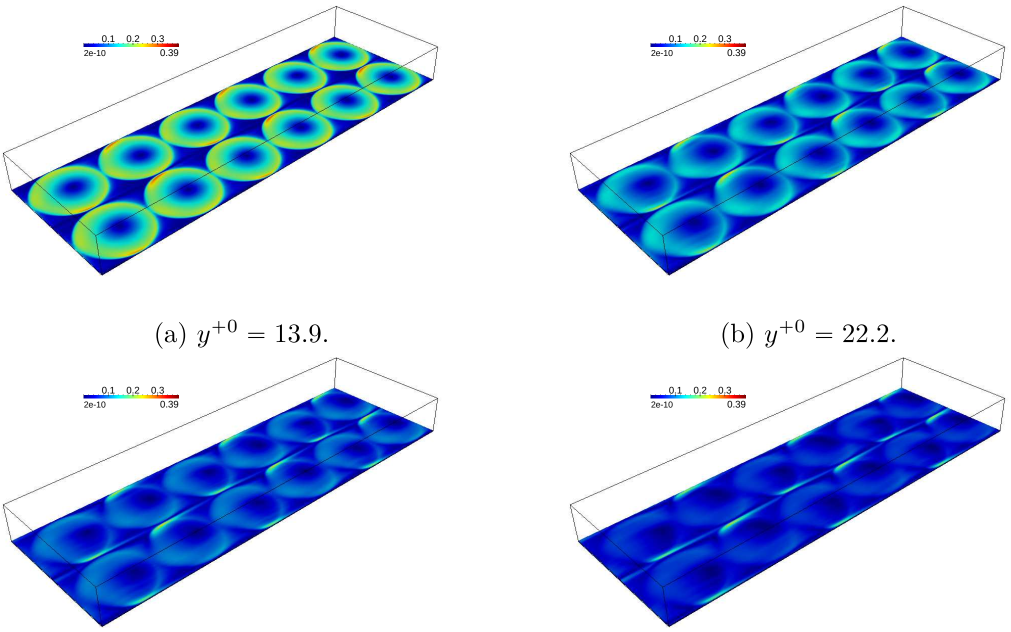


Figure 4.9: Cross flow isocontour representation of discs flow keeping the same level as in figure 4.8.



(a) $y^{+0} = 13.9$.

(b) $y^{+0} = 22.2$.

(c) $y^{+0} = 27.7$.

(d) $y^{+0} = 33.3$.

Figure 4.10: Discs flow representation at different $y^{+0}$ levels.

The wall forcing method described in the previous section has been modified in Wise & Ricco (2014). In this configuration, the discs tip velocity is subject to an oscillatory motion, i.e. $W_{\text{tip}} = W \cos(2\pi/T)$ with $T$ the oscillating period. The present case, focuses on the set of parameters reaching optimal drag-reduction rate of 20%. In this case, the simulation is also performed at $R_{\tau,r} = 180$ with the following features: $(L_x, L_y, L_z) = (6.79\pi, 2.0, 3.39\pi)$, $(n_x, n_y, n_z) = (384, 129, 384)$ and $(D^{+0}, W^{+0}, T^{+0}) = (960, 12, 1000)$. These values results in $\mathcal{R} = 19.8\%$ which compares well with the aforementioned reference. The three-dimensional representation of the flow is also given using the same flow decomposition. The time trace of the skin-friction coefficient is shown in figure 4.11. One can notice that the time trace during the controlled phase exhibits an oscillatory motion coupled with the usual turbulent behaviour. The time separating two peaks is approximately $t = 65 \Leftrightarrow t^{+0} = 500 = T^{+0}/2$.

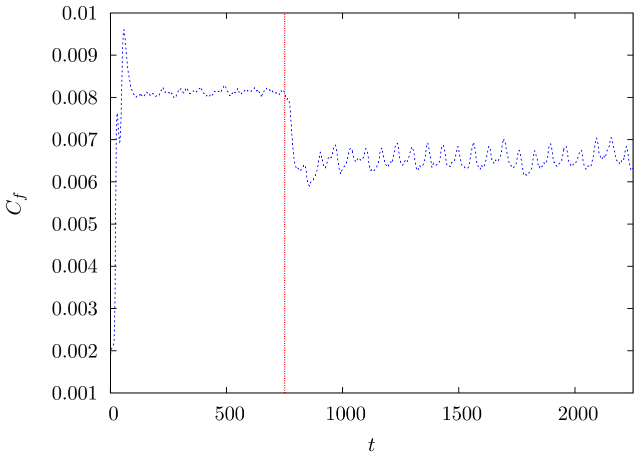


Figure 4.11: Skin-friction coefficient for the oscillating discs case.

## 4.6 Opposition control

The opposition control technique studied in this section was first simulated by means of DNS by Choi *et al.* (1994). The objective in this control technique is to reduce drag by attenuating the detrimental effects of the presence of streamwise vortices in the turbulent boundary layer. Indeed, these structures increase momentum transport in the wall vicinity by impinging high speed fluid near the wall and driving low speed fluid away from the wall. This method will be described in further details in 6 as the implementation is focused on hereafter. To achieve this, the method proposed by Choi *et al.* (1994) was to select a given plane and to measure the wall-normal velocity at this level on the whole plane. After this step the opposite velocity is fed back at the wall. The value was found to be $y^+ \simeq 10$ and yielded 25% drag reduction. A higher value for $y^+$ ($\simeq 26$) proved to induce drag increase. Note that DNS based on measurements of the spanwise velocity were also performed in Choi *et al.* (1994) and found to be able to reduce drag. However, in terms of practical implications, i.e. use of blowing suction, the wall-normal velocity is focused on in the present study. However, in subsequent studies (Koumoutsakos *et al.*, 1997; Hammond *et al.*, 1998) it was found that mentioned that a slightly better drag-reduction rate could be achieved by choosing a detection plane $y^{+0} \simeq 15$ resulting in 25% drag reduction, while with $y^+ \simeq 25$, drag is increased. It should be mentioned that the grid retained in these articles are $(n_x, n_y, n_z)$=(256,130,256) with $L_x = 4\pi$, $L_y = 2.0$, $L_z = 2\pi$. The algorithm is as follows: spatial derivatives in the streamwise and spanwise directions are computed using Fourier transforms, while the wall normal direction uses a conservative second-order finite difference scheme. The time advancement is based as in Choi *et al.* (1994) on a Runge-Kutta scheme. For all the simulations carried out in the aforementioned references constant mass flow rate is enforced. For the present simulation, the parameters are as follows: $(n_x, n_y, n_z)$=(256,129,256) and $l_x = 4\pi$, $l_y = 2.0$, $l_z = 4\pi/3$ and $R_p = 4200$. Three detection planes are simulated with nominal values $y^{+0} \in \{10, 15, 25\}$ and a further scanning over the wall-normal planes revealed that at $y^{+0} = 15.05$, the largest value for the drag reduction could be obtained, i.e. 24.6%. This value is consistent with those found in Koumoutsakos *et al.* (1997); Hammond *et al.* (1998); Chung & Talha (2011) where $\mathcal{R} \approx 25\%$.

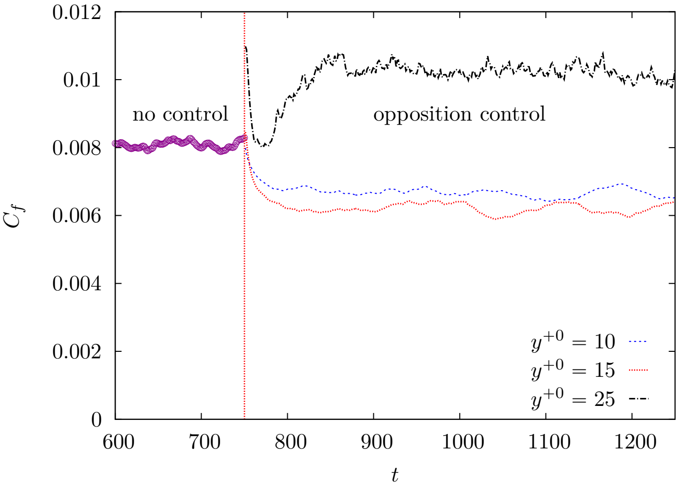


Figure 4.12: Skin-friction time trace for the opposition control technique.

# Chapter 5

# Turbulent flows over hydrophobic surfaces with shear-dependent slip length

## 5.1 Introduction

A key area of turbulence research has been flow control, i.e. the development of methods that modify the flow to achieve a beneficial effect, such as the attenuation of turbulent kinetic energy to obtain drag reduction (Gad-el Hak, 2000).

Our research interest is on hydrophobic surfaces, whose main characteristic is a finite effective slip velocity at the wall (Philip, 1972). These surfaces may achieve drag reduction for both laminar and turbulent flows (Watanabe *et al.*, 1999; Min & Kim, 2004), delay the transition to turbulence (Min & Kim, 2005), and operate over a wide range of Reynolds numbers relevant for technological applications, such as flows over marine vessels (Fukagata *et al.*, 2006). In particular, we are motivated by recent experimental and numerical research works that suggest that the characteristic slip length of the wall velocity may be a function of the wall-shear stress (Lauga & Stone, 2003; Min & Kim, 2004; Choi & Kim, 2006; Busse & Sandham, 2012). The crucial observation is that this dependence is likely to be true especially for liquids in the turbulent regime flowing past hydrophobic surfaces because these flows exert shear stresses that are much larger than in the laminar regime. Most of hydrophobic surfaces feature alternating patches of solid wall and trapped air pockets. The interaction between the viscous flow and the air pockets gives rise to the drag reduction effect. The inspiration for their design comes from the water-repellent lotus leaves (Feng *et al.*, 2002). More recently, liquid-infused rigid porous surfaces, the so-called Slippery Liquid-Infused Porous Surfaces (SLIPS) (Wong *et al.*, 2011; Wilson *et al.*, 2013) mimicking the features of the nepenthes pitcher plant, have shown very interesting hydrophobic, anti-biofouling and self-cleaning properties. Drag reduction over SLIPS has been reported in laminar Wexler *et al.* (2015); Jacobi *et al.* (2015) and turbulent regimes Rosenberg *et al.* (2016).

### 5.1.1 Laminar and transitional flows over hydrophobic surfaces

The remarkable hydrophobic properties of these surfaces have spurred scientists to investigate their effect on laminar flows with the aim of reducing the friction drag (Lauga & Stone, 2003; Ou & Rothstein, 2005; Busse *et al.*, 2013). One of the first experimental works of a laminar flow over superhydrophobic surfaces showed that 14% drag reduction could be attained (Watanabe *et al.*, 1999), while Ou *et al.* (2004) reported a 40% drag reduction.

The effect of hydrophobic surfaces has mainly been modelled in two ways. In the first model, which traces back to Navier (1823), the fluid obtains a finite slip velocity at the boundary and a linear relation between the local wall velocity and the shear-rate has been assumed to exist, i.e. $u_{\mathrm{wall}} = b\, \partial u/\partial y|_{\mathrm{wall}}$, where the constant $b$ is called the slip length. The second model distinguishes between the interaction of the liquid with the solid portions of the wall, modelled by the standard no-slip condition, and the dynamics between the liquid and the trapped air pockets, often modelled simply through a shear-free boundary. Philip (1972) used the second framework and extracted analytical solutions for the laminar Poiseuille pipe flow. Lauga & Stone (2003) extended Philip (1972)'s work to the pipe-flow case with different orientation of the micro-patterns and correlated these analytical results with the effective slip length for the first time.

The research works on stability and transition to turbulence are more limited. The most notable effort is by Min & Kim (2005), who demonstrated numerically that the critical channel-flow Reynolds number for linear stability increases when the walls are hydrophobic and that the laminar-turbulent transition can be significantly delayed.

### 5.1.2 Turbulent flows over hydrophobic surfaces

Inspired by the success of hydrophobic surfaces to reduce laminar drag, research efforts were soon directed toward turbulent drag reduction. Daniello *et al.* (2009) proved experimentally that turbulent drag reduction as high as 50% can be obtained with hydrophobic surfaces. Drag reduction experiments in free-stream transitional and turbulent boundary layer flows over flat surfaces sprayed with hydrophobic nanoparticles were carried out by Aljallis *et al.* (2013). A crucial observation was the eventual depletion of the surface at high-shear rates and the subsequent drag increase. The experimental work by Bidkar *et al.* (2014) showed that sustained turbulent drag reduction of up to 30% can be achieved over random-textured hydrophobic surfaces. Turbulent drag reduction of 14% over the SLIPS has been measured experimentally by Rosenberg *et al.* (2016).

In the direct numerical simulations (DNS) by Min & Kim (2004), the hydrophobic surface was implemented through Navier (1823)'s model, thereby enforcing an effective slip length. Maximum drag reduction occurred for slip in the streamwise direction only, while slip along the spanwise direction was detrimental for drag reduction. Min & Kim (2004)'s parametric study on the influence of slip lengths was extended in the DNS work of Busse & Sandham (2012). In a later work, Hasegawa *et al.* (2011) numerically studied a turbulent channel flow with streamwise-varying micro-grooves. The boundary conditions were expressed through a mobility tensor, relating the slip velocity and the wall-shear stress, in line with other works on flows over anisotropic hydrophobic patterns (Stroock *et al.*, 2002; Bazant & Vinogradova, 2008).

The DNS by Martell *et al.* (2009) modelled a superhydrophobic surface through periodically patterned micro-cavities filled with trapped air, confirming most of the experimental findings by Daniello *et al.* (2009). Martell *et al.* (2010) numerically simulated flows at three Reynolds numbers, demonstrating that, even though the Reynolds number changed, the same drag reduction is obtained as long as the scales of the wall texture are the same in wall units. Martell *et al.* (2010) and Lee *et al.* (2015) both proved that the drag reduction performance improves as the bulk Reynolds number increases if the texture scales are kept constant when scaled in outer units.

Fukagata *et al.* (2006) proposed a theoretical formula that analytically predicts the dependence of drag reduction on the slip length and the Reynolds number. They showed that increasing the Reynolds number leads to a weak decrease of the drag-reducing effect when slip is along the streamwise direction only. This negligible effect was also reported by

Busse & Sandham (2012). Further discussion on the physics of turbulent drag reduction by hydrophobic surfaces can be found in Rothstein (2010) and in the more recent DNS works by Jelly *et al.* (2014) and Lee *et al.* (2015), who reported the changes of turbulent kinetic energy balance, in particular the strengthening of the energy production near the slip patches and a detailed study of secondary and tertiary flows induced by the wall texture.

### 5.1.3 Motivation and objectives of shear-dependent slip length hydrophobic surface study

In this paper, for the first time theoretical and numerical results of laminar and turbulent flows bounded by hydrophobic walls exhibiting shear-dependent slip lengths are presented. We have been motivated by several discussions in experimental articles (Bonaccurso *et al.*, 2003; Choi *et al.*, 2003; Churaev *et al.*, 1984; Craig *et al.*, 2001) and numerical articles (Lauga & Stone, 2003; Min & Kim, 2004; Busse & Sandham, 2012; Schönecker *et al.*, 2014), from which it emerges that a shear-dependent slip length is likely to occur especially in the turbulent regime as the wall-shear stress can reach high values. Churaev *et al.* (1984) first experimentally reported slip lengths increasing with the shear rate. Lauga & Stone (2003) point out that the high wall shear may stretch the air pockets, thereby increasing the portion of the wall surface covered by air and causing the effective slip length to depend on the shear stress. Choi & Kim (2006) show that, in both water and mixed water-glycerin flows, the slip length depends on the wall shear, although they state that this effect may be influenced by viscous heating at high shear rates. Shear-dependent slip lengths were also shown by Choi *et al.* (2003) at smaller scales. Although the linear Navier's model was used by Min & Kim (2004), they remark that experimental works show that the slip length in general depends on the shear rate. Busse & Sandham (2012) further advocate that future research ought to consider this dependence to improve the modelling of hydrophobic surfaces under high-shear turbulent flows. Schönecker *et al.* (2014) point out that the hydrophobic slip depends on the dynamics of the enclosed gas and that the gas viscosity impacts on the slip length, implying that the latter depends on the shear rate. In the laminar case, steps in this direction have been taken by Schönecker & Hardt (2013), who computed a streamwise-dependent slip length for flows over rectangular air-filled cavities. More recently, the direct numerical simulation study by Jung *et al.* (2016) of a turbulent flow over thin air layers showed that in high-drag-reduction cases the computed slip length may depend on the shear at the water-air interface.

Furthermore, the SLIPS hydrophobic surfaces (Wong *et al.*, 2011; Wilson *et al.*, 2013), studied for the first time below a turbulent flow by Rosenberg *et al.* (2016), may also exhibit shear-dependent slip lengths. The liquid trapped in the porous substrate is usually a Newtonian oil, but non-Newtonian liquids could also be a sensible choice because they would stick well to the porous rigid substrate, an essential requirement for these textures to function properly. It is therefore likely that the interaction between the flowing water and the trapped oil would be characterized by shear-dependent slip lengths. Schönecker & Hardt (2015) further remark that the viscosity of the trapped oil in the SLIPS, and consequently the shear at the liquid-oil interface, must be considered to model these surfaces. Furthermore, when representing the SLIPS by the slip-length model, the issue of capturing accurately the near-wall spatially inhomogeneous interaction with the air-pockets pattern is avoided because the liquid infusing the substrate is uniformly distributed below the flowing liquid.

As a first study on laminar and turbulent flows over hydrophobic surfaces which show wall-slip properties that depend on the wall-shear stress, we have chosen to extend the slip-length model employed by Min & Kim (2004) and Busse & Sandham (2012). This approach clearly implies that, when representing surfaces with trapped air pockets, the precise texture features are not modelled and that the characteristic lengths of the hydrophobic surface are smaller than the near-wall viscous scales of the turbulence. The other option to model these surfaces would have been to resolve the complex interaction between the turbulent flows and the textured patterns of alternating patches of solid surfaces and air pockets. The modelling of the slip/no-slip pattern would have been more realistic, but, in order to synthesize the dependence of the wall slip on the wall shear, the widely-adopted boundary conditions of zero velocity over the solid wall and of zero shear over the air pockets would not have been adequate because the corresponding effective slip length would not have been shear dependent. This approach would have required the precise characterization of the interaction between the liquid flow and the gas, i.e. the resolution of the flow dynamics of the air motion in the pockets, as amply discussed by Schönecker *et al.* (2014).

A linear dependence between the slip length and the wall shear has been chosen, motivated by the experimental findings by Churaev *et al.* (1984) and Choi & Kim (2006). Although slip is considered along both the streamwise and spanwise directions, the shear-dependence of the slip length is only modelled along the streamwise direction because this direction experiences the highest shear. The turbulent flow is studied numerically by DNS, carried out by the Incompact3d code (Laizet & Lamballais, 2009; Laizet & Li, 2011).

The first objective is to solve the Navier-Stokes equations analytically for the laminar flows in the confined channel-flow and pipe-flow geometries. The laminar channel flow is then studied through nonlinear Lyapunov stability analysis. The rigorous two-dimensional approach by Balogh *et al.* (2001) is extended to the three-dimensional case and the shear-dependent laminar solution is chosen as the base flow. We stress that, although not useful to explain the physics of drag reduction in the turbulent regime because of the very small critical Reynolds number, the stability analysis is useful to arrive at rigorous nonlinear stability conditions. The feedback-control wall boundary conditions found from the stability analysis coincide with the hydrophobic slip-length model. For the first time, the conceptual link between the extracted feedback-law boundary conditions and the hydrophobic-surface model is advanced.

Other objectives are to extend the theory of Fukagata *et al.* (2006) to the shear-dependent slip-length case, to evince how the parameters describing the hydrophobic surface affect the drag reduction rate, and to carry out a comparison between Fukagata *et al.* (2006)'s theoretical results and the DNS results. The final aim is to study the drag-reducing turbulent flow through statistical analysis. The power exerted by the liquid turbulent flow on the hydrophobic surface is investigated and the principal strain rates of the near-wall turbulent flow are studied for the first time in a drag-reducing flow.

In §5.2, the laminar-flow analysis is presented. The laminar flow solutions for the channel-flow and the pipe-flow geometries are found in §5.2.1 and the Lyapunov stability analysis is discussed in §5.2.2. In §5.3, the turbulent-flow analysis is presented. The Fukagata *et al.* (2006)'s theory for drag-reduction prediction is contained in §5.3.1, the results on the drag reduction properties and turbulence statistics are found in §5.3.2, and the power spent on the hydrophobic surface is discussed in §5.3.3. In §5.3.4, the numerical results on the turbulent vorticity are presented and the study of the principal strain rates is found in §5.3.6. In §5.4 a summary of the results is given.

## 5.2 Laminar flow

This section presents the analytical results for laminar flows over hydrophobic surfaces in §5.2.1 and the nonlinear Lyapunov stability analysis of the laminar channel flow in §5.2.2.

### 5.2.1 Analytical laminar solutions

The laminar channel-flow solution with shear-dependent slip-length hydrophobic walls is first derived analytically. Lengths are scaled by the channel half-height $h^*$, velocities by the maximum Poiseuille velocity $U_p^*$ with uncontrolled walls, and the time $t^*$ by $h^*/U_p^*$. Quantities non-dimensionalized through these units are not indicated by any symbol and dimensional quantities are marked by the superscript $*$. The Reynolds number is defined as $R_p = U_p^* h^*/\nu^*$, where $\nu^*$ is the kinematic viscosity of the fluid. The streamwise, wall-normal, and spanwise directions are $x^*$, $y^*$, and $z^*$, respectively, and $y \in [0,2]$. The velocity vector field is defined as $\mathbf{W} = (U(x,y,z,t), V(x,y,z,t), W(x,y,z,t))$ and the pressure is $P(x,y,z,t)$. The velocity and the pressure satisfy the incompressible continuity and Navier-Stokes equations. The hydrophobic surface is modelled through the following boundary condition at the bottom wall:

$$U(0) = l_s \left.\frac{\partial U}{\partial y}\right|_{y=0} = a \left( \left.\frac{\partial U}{\partial y}\right|_{y=0} \right)^2 + b \left.\frac{\partial U}{\partial y}\right|_{y=0}, \tag{5.1}$$

and analogously for the upper wall at $y = 2$. The constant $b$ is positive and, as suggested by experiments (Churaev *et al.*, 1984; Choi *et al.*, 2003; Choi & Kim, 2006), $a$ is also positive. The boundary condition (5.1) is also consistent with the shear-dependent slip length computed from the molecular dynamics simulations carried out by Thompson & Troian (1997), i.e. $l_s = l_{s0}(1 - \dot{\gamma}/\dot{\gamma}_c)^{-1/2}$, where $\dot{\gamma}$ and $\dot{\gamma}_c$ are the scaled shear rate and a critical shear rate, respectively. Indeed, the Taylor expansion for small $\dot{\gamma}$ leads to $l_s = l_{s0} + l_{s0}\dot{\gamma}/(2\dot{\gamma}_c) + \mathcal{O}(\dot{\gamma}^2)$. As the flow is symmetric along the channel centreline, the other boundary condition may be chosen as:

$$\left.\frac{\partial U}{\partial y}\right|_{y=1} = 0. \tag{5.2}$$

In the case of fully-developed two-dimensional laminar channel flow, $\mathbf{W} = (U(y), 0, 0))$. The streamwise velocity $U$ satisfies a simplified form of the $x$-momentum equation,

$$\frac{1}{R_p}\frac{\mathrm{d}^2 U}{\mathrm{d}y^2} - \frac{\mathrm{d}P}{\mathrm{d}x} = 0. \tag{5.3}$$

The solution is

$$U(y) = R_p \frac{\mathrm{d}P}{\mathrm{d}x}\left(\frac{y^2}{2} - y + aR_p\frac{\mathrm{d}P}{\mathrm{d}x} - b\right). \tag{5.4}$$

It is useful to introduce the bulk velocity,

$$\mathcal{U}_b = \frac{1}{2}\int_0^2 U(y)\mathrm{d}y = R_p\frac{\mathrm{d}P}{\mathrm{d}x}\left(aR_p\frac{\mathrm{d}P}{\mathrm{d}x} - b - \frac{1}{3}\right). \tag{5.5}$$

The special case of constant slip length ($a$=0) is first studied. In the constant-bulk-velocity case, $\mathcal{U}_b = 2/3$. The streamwise pressure gradient is

$$\frac{\mathrm{d}P}{\mathrm{d}x} = \frac{-2}{R_p(3b+1)}. \tag{5.6}$$

To enforce the same mass flow rate, the hydrophobic surface leads to a smaller pressure gradient than in the uncontrolled case. The pressure gradient tends to zero as $b$ increases. By substituting (5.6) into (5.4), for $a = 0$ one finds

$$U(y) = \frac{-2}{3b+1}\left(\frac{y^2}{2} - y - b\right), \tag{5.7}$$

which was also derived by Min & Kim (2005). In the limit of large slip length, $b \to \infty$, the plug flow case is found, $U = \mathcal{U}_b$. For the case of constant pressure gradient, $\mathrm{d}P/\mathrm{d}x = -2/R_p$. For $a = 0$, $\mathcal{U}_b$ increases linearly with the slip length, $\mathcal{U}_b = 2b + 2/3$.

In the shear-dependent slip-length case, $a \neq 0$, and when $\mathcal{U}_b$ is constant, the pressure gradient is found as follows. Expression (5.5) is first solved for the pressure gradient,

$$\left.\frac{\mathrm{d}P}{\mathrm{d}x}\right|_{1,2} = \frac{3b+1}{6aR_p}\left[1 \pm \sqrt{1 + \frac{36a\mathcal{U}_b}{(3b+1)^2}}\right]. \tag{5.8}$$

The minus-sign solution is selected by Taylor expansion of the square-root term in (5.8) for small $a$ and $b = \mathcal{O}(1)$, i.e. $[1 + 36a\mathcal{U}_b/(3b+1)^2]^{1/2} = 1 + 18a\mathcal{U}_b/(3b+1)^2 + \mathcal{O}(a^2)$, to match (5.8) with the pressure-gradient solution (5.6) for the constant-slip-length case. We further set $\mathcal{U}_b = 2/3$ and the result is

$$\frac{\mathrm{d}P}{\mathrm{d}x} = \frac{3b+1}{6aR_p}\left[1 - \sqrt{1 + \frac{24a}{(3b+1)^2}}\right]. \tag{5.9}$$

For $b = \mathcal{O}(1)$ and $a \to \infty$,

$$\frac{\mathrm{d}P}{\mathrm{d}x} \sim \frac{1}{R_p}\sqrt{\frac{2}{3a}}, \tag{5.10}$$

i.e. the pressure gradient is independent of $b$ and decreases as $a$ increases more slowly than when $b$ increases and $a = 0$, as shown by (5.6). When the pressure gradient is constant, $\mathcal{U}_b = 4a + 2b + 2/3$, that is the bulk velocity increases linearly with both $a$ and $b$, and the growth rate is larger for $a$. The equivalent slip length can be computed in the laminar case by substituting (5.4) into (5.1), i.e.

$$l_s = b - aR_p \frac{\mathrm{d}P}{\mathrm{d}x}. \tag{5.11}$$

The solution for the laminar flow in a pipe with a hydrophobic wall featuring a shear-dependent slip length is briefly described in the following in order to relate the bulk velocity to the pressure gradient.

The channel flow analysis in §5.2.1 is herein extended to the case of pipe flow. In this appendix, lengths are scaled by the pipe radius $R^*$ and velocities by the maximum Poiseuille velocity $U_p^*$. The Reynolds number is defined as $R_p = U_p^* R^* / \nu^*$ and the radial direction is denoted by $r \in [0, 1]$, where $r = 0$ indicates the pipe axis. In the case of fully-developed axial-symmetrical laminar pipe flow, $\mathbf{W} = (U(r), 0, 0)$, where $U$ satisfies the simplified $x$-momentum,

$$\frac{1}{rR_p}\frac{\mathrm{d}}{\mathrm{d}r}\left(r\frac{\mathrm{d}U}{\mathrm{d}r}\right) - \frac{\mathrm{d}P}{\mathrm{d}x} = 0. \tag{5.12}$$

The boundary conditions are

$$\begin{aligned} U(1) &= a\left(\left.\frac{\mathrm{d}U}{\mathrm{d}r}\right|_{r=1}\right)^2 - b\left.\frac{\mathrm{d}U}{\mathrm{d}r}\right|_{r=1}, \\ \left.\frac{\mathrm{d}U}{\mathrm{d}r}\right|_{r=0} &= 0. \end{aligned} \tag{5.13}$$

The solution to (5.12) is

$$U(r) = \frac{R_p}{4}\frac{\mathrm{d}P}{\mathrm{d}x}\left(r^2 + aR_p\frac{\mathrm{d}P}{\mathrm{d}x} - 2b - 1\right). \tag{5.14}$$

The solution for $a = 0$ is also given in Watanabe *et al.* (1999). The bulk velocity is

$$\mathcal{U}_b = 2\int_0^1 U(r) r \mathrm{d}r = \frac{R_p}{8}\frac{\mathrm{d}P}{\mathrm{d}x}\left(2aR_p\frac{\mathrm{d}P}{\mathrm{d}x} - 4b - 1\right). \tag{5.15}$$

$$\mathcal{U}_b = 2\int_0^1 U(r) r \mathrm{d}r = \frac{R_p}{8}\frac{\mathrm{d}P}{\mathrm{d}x}\left(2aR_p\frac{\mathrm{d}P}{\mathrm{d}x} - 4b - 1\right), \tag{5.16}$$

The relationship (5.16) is useful to compute the slip-length parameters $a$ and $b$ from the experimental data of mass flow rate of mercury in thin quartz capillaries as a function of the pressure gradient reported by Churaev *et al.* (1984) in their figure 4 on page 579 and reproduced in figure 5.1 (left). It is clear that a constant-slip-length behaviour only occurs at small pressure gradients (dashed line), while a quadratic behaviour as that predicted by (5.16) ensues for larger pressure gradients (solid line). By rescaling (5.16) and fitting the experimental data, $a^*$=7·10$^{-6}$µm s and $b^*$=0.16µm are found. We are also interested in laminar flows over surfaces characterized by larger slip lengthsWatanabe *et al.* (1999), i.e. of the order of tens of µm. To the best of our knowledge, the cone-and-plate rheometer data for a NanoTurf superhydrophobic surface reported by Choi & Kim (2006) are the only ones that show shear-dependent slip lengths of this magnitude in the laminar regime (refer to their figure 4 (bottom)). As shown in figure 5.1 (right), the dependence of the slip length on the shear rate is linear with $a^*$=0.12µm s and $b^*$=36µm. Note that, although featuring slip lengths of different orders of magnitude, both Churaev *et al.* (1984) and Choi & Kim (2006) show a linear dependence of the slip length on the wall-shear stress, i.e. consistent with our model (5.1).

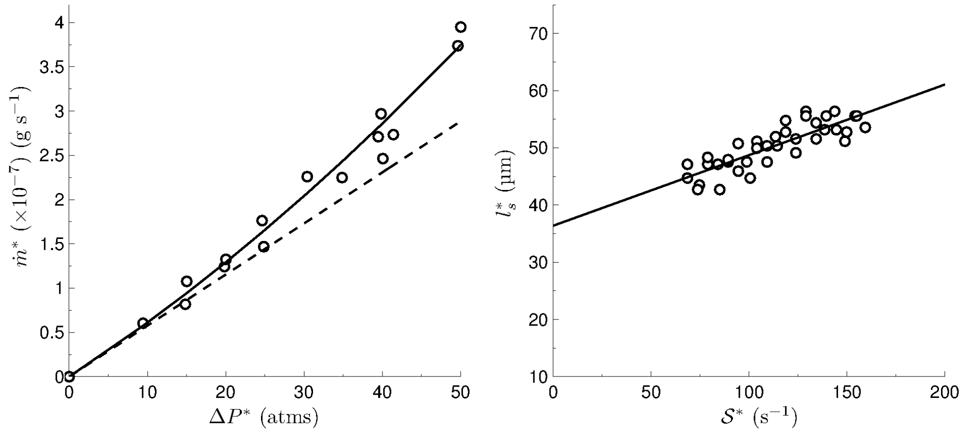


Figure 5.1: Left: Mass flow rate as a function of pressure difference for mercury flow in thin quartz capillary tubes, measured by Churaev *et al.* (1984) (refer to their figure 4 on page 579). Right: Slip length as a function of shear rate for water flow in a cone-and-plate rheometer, measured by Choi & Kim (2006) (refer to their figure 4).

### 5.2.2 Nonlinear Lyapunov stability analysis

The Lyapunov nonlinear stability analysis of the laminar flow studied in §5.2.1 is performed in this section by using the formalism described in §2.3. The shear-dependent slip-length condition is derived from (2.39). The boundary conditions are

$$u(x,y_w,z,t) = a\left(\frac{\partial u}{\partial y}\right)^2(x,y_w,z,t) + (1-y_w)\,b\frac{\partial u}{\partial y}(x,y_w,z,t) \tag{5.17}$$

and corresponding ones for the spanwise velocity component $w$. Note that the different signs only apply to $b$ and not to $a$ because $a$ multiplies $(\partial u/\partial y)^2$ and therefore the symmetrical condition over the two channel walls is respected. Following the same reasoning as in the constant $k$ case, expressions for $\partial u/\partial y$ and $\partial w/\partial y$ are found from (5.17) and from the corresponding ones for $w$. In the shear-dependent slip-length case, the inequality for the perturbation energy is:

$$\begin{aligned}\frac{\mathrm{d}E(\mathbf{w})}{\mathrm{d}t} \leqslant & -\frac{\alpha E(\mathbf{w})}{2} - \frac{2}{R_p}\left(\frac{2}{b+\sqrt{b^2+4a}}-1\right)\left[u^2(x,0,z,t)+w^2(x,0,z,t)\right]_{\mathcal{I}xz} \\ & - \frac{4}{R_p\left(b+\sqrt{b^2+4a}\right)}\left[u^2(x,2,z,t)+w^2(x,2,z,t)\right]_{\mathcal{I}xz}.\end{aligned} \tag{5.18}$$

Hereafter the detailed derivation is given. The derivation carried out in §2.3 is extended to the shear-dependent slip-length case. In the sequel, except for the final formula (5.32), the dependence on $x$, $z$, $t$ is dropped for compactness. Only the procedure for the streamwise velocity is described as the one for the spanwise velocity is analogous.

The discriminant of (5.17) is

$$\Delta = b^2 + 4au(y_w), \tag{5.19}$$

(where $y_w = 0, 2$) which must be positive because $\partial u/\partial y \in \mathbb{R}$ and must be different from zero because otherwise $\partial u/\partial y$ would not be related to $u$. If $\Delta = 0$, the double root is $-b/2a$, which diverges for $a \to 0$ and $b = \mathcal{O}(1)$. The roots for the bottom wall are, $\forall a \neq 0$:

$$\frac{\partial u^{\ominus}}{\partial y}(0) = \frac{-b+\sqrt{b^2+4au(0)}}{2a}\,,\quad \frac{\partial u^{\circledast}}{\partial y}(0) = \frac{-b-\sqrt{b^2+4au(0)}}{2a}, \tag{5.20}$$

and the roots for the top wall are

$$\frac{\partial u^{\odot}}{\partial y}(2) = \frac{b+\sqrt{b^2+4au(2)}}{2a}\,,\quad \frac{\partial u^{\oplus}}{\partial y}(2) = \frac{b-\sqrt{b^2+4au(2)}}{2a}. \tag{5.21}$$

The arguments of the square-root terms must be positive. To ensure this, the amplitude of the streamwise velocity perturbation at the wall is first imposed to be bounded, $|u(0)| \leqslant$

1 and $|u(2)| \leqslant 1$. This is fully consistent with the objective of the analysis, i.e. the stabilization of the laminar flow, because $\max_y \widehat{U} = 1$. It follows that $b^2-4a \leqslant b^2+4au(0) \leqslant b^2+4a$ and $b^2-4a \leqslant b^2+4au(2) \leqslant b^2+4a$, and a sufficient condition for $b^2+4au(0)$ and $b^2+4au(2)$ to be positive is

$$a \leqslant b^2/4. \tag{5.22}$$

The choice of the relevant roots in (5.20) and (5.21) is dictated by the limit $a \to 0$ with $b = \mathcal{O}(1)$, i.e. the constant-slip formulas (2.41) must be recovered from the shear-dependent slip-length formulas. For this purpose, (5.20) and (5.21) are Taylor-expanded to first order with $a \to 0$ and $b = \mathcal{O}(1)$. The Taylor expansion for $\partial u^{\ominus}/\partial y(0)$ leads to:

$$\frac{\partial u^{\ominus}}{\partial y}(0) = \frac{u(0)}{b}, \tag{5.23}$$

and similarly for (5.20) and (5.21). The constant-slip formulas (2.41) are recovered as $a \to 0$, and $\partial u^{\ominus}/\partial y$ and $\partial u^{\oplus}/\partial y$ are chosen for the lower and upper wall, respectively. The velocity gradients $\partial u^{\ominus}/\partial y$ and $\partial u^{\oplus}/\partial y$ in (5.20) and (5.21) and the corresponding spanwise velocity are inserted in $\mathcal{I}_{uw}$ in (2.39) to find:

$$\mathcal{I}_{uw} = \frac{2}{R_p}\left[u(2)\frac{\partial u^{\oplus}}{\partial y}(2) - u(0)\frac{\partial u^{\ominus}}{\partial y}(0) + w(2)\frac{\partial w^{\oplus}}{\partial y}(2) - w(0)\frac{\partial w^{\ominus}}{\partial y}(0)\right]_{\mathcal{I}xz}. \tag{5.24}$$

The term containing $u^{\ominus}$ in (5.24) expands as:

$$\begin{aligned} -\frac{2}{R_p}\left[u(0)\frac{\partial u^{\ominus}}{\partial y}(0)\right]_{\mathcal{I}xz} &= \frac{b}{aR_p}\left[u(0)\left(1-\sqrt{1+\frac{4au(0)}{b^2}}\right)\right]_{\mathcal{I}xz} \\ &= -\frac{4}{bR_p}\left[\frac{u^2(0)}{1+\sqrt{1+\frac{4au(0)}{b^2}}}\right]_{\mathcal{I}xz}. \end{aligned} \tag{5.25}$$

The expression for the term containing $u^{\oplus}$ is analogous. Using the boundedness argument $|u(0)| \leqslant 1$ and $|u(2)| \leqslant 1$ employed in §2.3, one finds

$$-\frac{4}{bR_p\left(1+\sqrt{1-\frac{4a}{b^2}}\right)} \leqslant -\frac{2}{R_p}\left[u(0)\frac{\partial u^{\ominus}}{\partial y}(0)\right]_{\mathcal{I}xz} \leqslant -\frac{4}{bR_p\left(1+\sqrt{1+\frac{4a}{b^2}}\right)}. \tag{5.26}$$

Using (5.26) in (5.25), one finds

$$-\frac{4}{bR_p}\left[\frac{u^2(0)}{1+\sqrt{1+\frac{4au(0)}{b^2}}}\right]_{\mathcal{I}xz} \leqslant -\frac{4}{bR_p}\left[\frac{u^2(0)}{1+\sqrt{1+\frac{4a}{b^2}}}\right]_{\mathcal{I}xz}. \tag{5.27}$$

Equations (5.25) and (5.27) can be used in the second integral of (5.32):

$$\left[u^2(0) - u(0)\frac{\partial u^\ominus}{\partial y}(0)\right]_{\mathcal{I}xz} \leqslant -\left[\frac{2}{b\left(1+\sqrt{1+\frac{4a}{b^2}}\right)} - 1\right]\left[u^2(0)\right]_{\mathcal{I}xz}. \tag{5.28}$$

For the system to decay exponentially, hence achieving global stability:

$$\frac{2}{b\left(1+\sqrt{1+\frac{4a}{b^2}}\right)} \geqslant 1, \tag{5.29}$$

i.e. $a \leqslant 1-b$. The derivation involving $u^\oplus$ is analogous,

$$\frac{2}{R_p}\left[u(2)\frac{\partial u^\oplus}{\partial y}(2)\right]_{\mathcal{I}xz} = \frac{b}{aR_p}\left[u(2)\left(1-\sqrt{1+\frac{4au(2)}{b^2}}\right)\right]_{\mathcal{I}xz} = -\frac{4}{bR_p}\left[\frac{u^2(2)}{1+\sqrt{1+\frac{4au(2)}{b^2}}}\right]_{\mathcal{I}xz}. \tag{5.30}$$

By bounding (5.30), one finds:

$$-\frac{4u^2(2)b}{R_p}\frac{1}{1+\sqrt{1-\frac{4a}{b^2}}} \leqslant \frac{2}{R_p}u(2)\frac{\partial u^\oplus}{\partial y}(2) \leqslant -\frac{4u^2(2)b}{R_p}\frac{1}{1+\sqrt{1+\frac{4a}{b^2}}}. \tag{5.31}$$

The boundary term on the left-hand-side of (5.31) is thus always bounded by a negative term. Substitution of (5.28) and (5.31) into (2.39) leads to:

$$\begin{aligned}\frac{\mathrm{d}E(\mathbf{w})}{\mathrm{d}t} \leqslant & -\frac{\alpha E(\mathbf{w})}{2} - \frac{2}{R_p}\left(\frac{2}{b+\sqrt{b^2+4a}} - 1\right)\left[u^2(0)+w^2(0)\right]_{\mathcal{I}xz} \\ & - \frac{4}{R_p\left(b+\sqrt{b^2+4a}\right)}\left[u^2(2)+w^2(2)\right]_{\mathcal{I}xz}.\end{aligned} \tag{5.32}$$

Expression (2.42) is recovered from (5.32) in the limit $a \to 0$ with $b = \mathcal{O}(1)$.

In the limit $a \to 0$, (2.42) is recovered from (5.18). As the limits $L_x, L_z \to \infty$ have been taken, $\left[u^2(x,0,z,t)+w^2(x,0,z,t)\right]_{\mathcal{I}xz} = \left[u^2(x,2,z,t)+w^2(x,2,z,t)\right]_{\mathcal{I}xz}$. It follows that (5.18) simplifies to:

$$\frac{\mathrm{d}E(\mathbf{w})}{\mathrm{d}t} \leqslant -\frac{\alpha E(\mathbf{w})}{2} - \frac{2}{R_p}\left(\frac{4-b-\sqrt{b^2+4a}}{b+\sqrt{b^2+4a}}\right)\left[u^2(x,0,z,t)+w^2(x,0,z,t)\right]_{\mathcal{I}xz} \tag{5.33}$$

In summary, the stability conditions are

$$R_p < \frac{1}{4}, \quad a \leqslant 4-2b, \quad a \leqslant b^2/4. \tag{5.34}$$

The first stability condition relating the positive $a$ and $b$ in (5.34) is found by imposing the coefficient multiplying the second term on the right-hand-side in (5.33) to be negative.

The inequality changes to the more restrictive $a \leqslant 1 - b$ if (5.18) is used. As in the two-dimensional case studied by Balogh *et al.* (2001), the condition on the Reynolds number is very restrictive and proper of laminar microfluidic flows. Therefore, the nonlinear stability analysis does not provide information on the physical mechanism that leads to the attenuation of the turbulent kinetic energy.

We can verify whether the flow parameters in Choi & Kim (2006), pertaining to a laminar flow in a thin gap between a stationary plate and a spinning cone (i.e. a very good model for the idealized Couette flow), satisfy our stability conditions (5.34) because these are also valid for Couette flow (which is verified by substituting the Couette constant shear in inequality (2.59)). A Reynolds number of 1/4, based on their rheometer's gap and tip speed, is found for an angular velocity of 0.15 rad/s, which is in the range of values that the rheometer can achieve. By scaling their slip parameters, $a^*$=0.12µm s and $b^*$=36µm, by the rheometer's tip speed and gap thickness, the first stability condition, $a \leqslant 4 - 2b$, is always satisfied. The second condition, $a \leqslant b^2/4$, is satisfied when the rheometer's tip speed is smaller than 0.029 m/s (angular velocity smaller than 0.6 rad/s), which again is in the realizable range of Choi & Kim (2006)'s experimental rig.

## 5.3 Turbulent flow

The turbulent flow decomposition and the numerical procedures are contained in §2.1 and the Fukagata-Kasagi-Koumoutsakos theory for drag reduction production is described in §5.3.1. The numerical results are found in the remaining §5.3.2-§5.3.6.

The pressure-driven turbulent flow between infinite parallel flat plates with hydrophobic properties has been studied by DNS at low Reynolds number. The open-source Navier-Stokes solver Incompact3d as described in the previous chapters, has been modified to model the hydrophobic surfaces characterized by constant and shear-dependent slip lengths. The present simulations have been performed on the Polaris cluster at the University of Leeds and the ARCHER UK National Supercomputing Service.

The simulations have been carried out at $R_p = 4200$ at constant mass flow rate, i.e. $\mathcal{U}_b = 2/3$, and the uncontrolled friction Reynolds number is $R_{\tau,r} = 179.5$. The dimensions of the computational domain are $L_x = 4\pi$, $L_y$=2, and $L_z = 4\pi/3$. The time step is $\Delta t = 0.0025$ ($\Delta t^{+0} = 0.019$). The grid sizes are $\Delta x^{+0} = 8.5$ and $\Delta z^{+0} = 3$, and the minimum $\Delta y^{+0} = 0.4$ near the wall. The simulations with hydrophobic walls have been started from a fully-developed turbulent flow with the no-slip condition. As in Ricco &

Hahn (2013), the turbulence statistics are computed after discarding the initial temporal transient during which the flow adapts to the new drag-reducing regime. The duration of the transient is estimated by direct observation of the time history of the space-averaged wall-shear stress and is typically of the order of $100h^*/U_p^*$ $(1150\nu^*/u_{\tau,r}^{*2})$. The statistics are calculated by averaging instantaneous flow fields saved at intervals of $10\nu^*/u_{\tau,r}^{*2}$ for a total time window of $850h^*/U_p^*$ $(6520\nu^*/u_{\tau,r}^{*2})$.

In the code, 6th-order compact finite difference schemes are used for the spatial derivatives in the convective and diffusive terms. For modelling of the hydrophobic surfaces, the wall boundary conditions (5.17) are implemented through single-sided two- and three-point formulas. Both schemes have been tested thoroughly without notable differences. The constant-slip-length results have been compared successfully with Min & Kim (2004)'s and Busse & Sandham (2012)'s.

### 5.3.1 Fukagata-Kasagi-Koumoutsakos theory for shear-dependent slip-length surfaces

The theoretical analysis by Fukagata *et al.* (2006) (FKK hereinafter) is extended to the case of shear-dependent slip length. As in the constant-slip-length case used in FKK, the starting point is to express the mean streamwise slip velocity $\mathcal{U}(0)$ as a function of the wall-normal gradient of the mean velocity:

$$\mathcal{U}(0) = a \left( \left. \frac{\mathrm{d}\mathcal{U}}{\mathrm{d}y} \right|_{y=0} \right)^2 + b \left. \frac{\mathrm{d}\mathcal{U}}{\mathrm{d}y} \right|_{y=0} . \tag{5.35}$$

Note that in the constant-slip-length case ($a = 0$), (5.35) is found from (5.1) because the order of the integral operators used in (2.9) and the wall-normal derivative operator can be switched as the relationship is linear. In the shear-dependent case, this is obviously not possible because of the square of the wall-normal gradient. To make progress and continue along the lines of FKK's theoretical formulation, (5.35) is nevertheless assumed to hold as is shown in the following. In this appendix the error in assuming that (5.35) is valid is quantified. Expression (5.35) is found by first space- and time-averaging (5.1). As in the constant-slip-length case studied by FKK, the second term on the right-hand-side of (5.35) is obtained directly because the order of the integral operators used in (2.9) and the wall-normal derivative operator can be switched. By applying the space- and time-averaging operators (2.40) and (2.11) to the first term on the right-hand-side (5.1), one finds:

$$\mathcal{A} = \frac{1}{L_x L_z} \left[ \overline{\left( \frac{\partial U}{\partial y}\bigg|_{y=0} \right)^2} \right]_{\mathcal{I}xz} . \tag{5.36}$$

In order to express (5.36) as a function of the mean velocity $\mathcal{U}$, the square of the mean-flow wall-normal gradient is instead considered:

$$\mathcal{B} = \left[ \frac{1}{L_x L_z} \left[ \overline{\frac{\partial U}{\partial y}\bigg|_{y=0}} \right]_{\mathcal{I}xz} \right]^2 = \left( \frac{\mathrm{d}\mathcal{U}}{\mathrm{d}y}\bigg|_{y=0} \right)^2 = \frac{C_f^2 R_p^2 \mathcal{U}_b^4}{4}. \tag{5.37}$$

The last two in (5.37) follow from (2.9) and (2.13). The percent relative error $\mathcal{E}$ between $\mathcal{A}$ and its approximation $\mathcal{B}$ is:

$$\mathcal{E}(\%) = 100 \times \left| \frac{\mathcal{A} - \mathcal{B}}{\mathcal{A}} \right|. \tag{5.38}$$

The error $\mathcal{E}$ is less than 1%.

Along the same lines, two definitions of the average slip length are proposed. These can be defined as

$$\mathcal{L}_1 = (L_x L_z)^{-1} \left[ \overline{l_s} \right]_{\mathcal{I}xz}, \tag{5.39}$$

where $l_s(x, z, t)$ is defined in (5.1), or as

$$\mathcal{U}(0) = \mathcal{L}_2 \, \frac{\mathrm{d}\mathcal{U}}{\mathrm{d}y}\bigg|_{y=0} . \tag{5.40}$$

As the two lengths show very good agreement, the average slip length is indicated by $\mathcal{L}$.

Equation (5.35) is first transformed into:

$$\mathcal{U}(0)^+ = a u_{\tau,r} \left( R_{\tau,r} \, \frac{\mathrm{d}\mathcal{U}^+}{\mathrm{d}y^+}\bigg|_{y=0} \right)^2 \left(u_\tau^{+0}\right)^3 + b \, \frac{\mathrm{d}\mathcal{U}^+}{\mathrm{d}y^+}\bigg|_{y=0} u_\tau^{+0} R_{\tau,r}. \tag{5.41}$$

As $\mathrm{d}\mathcal{U}^+/\mathrm{d}y^+|_{y=0} = 1$, then using $u_\tau = u_\tau^{+0} u_{\tau,r}$ and $\mathcal{U}(0) = \mathcal{U}(0)^+ u_\tau^{+0} u_{\tau,r}$, (5.41) becomes

$$\mathcal{U}(0) = a \left(u_\tau^{+0}\right)^4 \left(u_{\tau,r} R_{\tau,r}\right)^2 + b \left(u_\tau^{+0}\right)^2 u_{\tau,r} R_{\tau,r}. \tag{5.42}$$

The bulk velocity, $\mathcal{U}_b$, is expressed as the sum of the mean slip velocity and an effective bulk velocity $\mathcal{U}_{be}$,

$$\mathcal{U}_b = \mathcal{U}(0) + \mathcal{U}_{be}. \tag{5.43}$$

The bulk velocity is assumed to satisfy Dean (1978)'s formula,

$$\mathcal{U}_b = \left(\kappa^{-1} \ln R_{\tau,r} + F\right) u_{\tau,r}, \tag{5.44}$$

where both the constant $F$ and the von Kármán constant $\kappa$ are assumed to be independent of the Reynolds number. Formula (5.44) follows directly from the assumption that the mean-velocity profile is logarithmic in the channel core. As amply verified by experimental and numerical data (Hoyas & Jiménez, 2006; Marusic *et al.*, 2010; Bernardini *et al.*, 2014), this is not the case at the low Reynolds number of the present study and it has been argued that a truly logarithmic behaviour is only obtained at an infinite Reynolds number (Bernardini *et al.*, 2014). Nevertheless, the use of (5.44) has proved to be successful in the constant-slip-length cases as excellent theoretical predictions for $\mathcal{R}$ were obtained by FKK. Therefore, the logarithmic behaviour is also assumed to hold in the present shear-dependent slip-length cases and the predictive power of the framework is checked a posteriori when the theoretical results are compared with the DNS data in §5.3.2.

As suggested by Busse & Sandham (2012), $\kappa$ and $F$ are computed from our DNS data. The von Kármán constant $\kappa$ is estimated through the diagnostic function (Hoyas & Jiménez, 2006; Bernardini *et al.*, 2014):

$$\kappa^{-1} = y^+ \frac{\mathrm{d}\mathcal{U}^+}{\mathrm{d}y^+}. \tag{5.45}$$

Once $\kappa$ is known, $F$ is computed via (5.44). We find $\kappa = 0.4$ and $F = 2.67$. As in FKK, the effective bulk velocity is also assumed to follow the logarithmic law,

$$\mathcal{U}_{be} = \left[\kappa^{-1} \ln\left(u_\tau^{+0} R_{\tau,r}\right) + F\right] u_\tau^{+0} u_{\tau,r}. \tag{5.46}$$

Combining equations (5.42) and (5.46), one finds

$$\left(\kappa^{-1} \ln R_{\tau,r} + F\right) \frac{1 - u_\tau^{+0}}{\left(u_\tau^{+0}\right)^2} = a\left(u_\tau^{+0}\right)^2 u_{\tau,r} {R_{\tau,r}}^2 + b R_{\tau,r} + \frac{\ln u_\tau^{+0}}{\kappa u_\tau^{+0}}. \tag{5.47}$$

Using $u_\tau^{+0} = \sqrt{1 - \mathcal{R}^\star}$, $\mathcal{R}^\star = \mathcal{R}/100$ and $u_{\tau,r} = R_{\tau,r}/R_p$, (5.47) becomes

$$\frac{a\left(1 - \mathcal{R}^\star\right) R_{\tau,r}^2}{R_p} + b = \left(\kappa^{-1} \ln R_{\tau,r} + F\right) \frac{1 - \sqrt{1 - \mathcal{R}^\star}}{R_{\tau,r}(1 - \mathcal{R}^\star)} - \frac{\ln\left(1 - \mathcal{R}^\star\right)}{2\kappa R_{\tau,r}\sqrt{1 - \mathcal{R}^\star}}. \tag{5.48}$$

The value of $\mathcal{R}^\star$ is found through a Monte Carlo simulation Dunn & Shultis (2011). As expected, the constant-slip-length formula (13) in FKK is recovered from (5.48) when $a = 0$. There is an interesting interpretation of the left-hand-side of (5.48). It can be written as follows:

$$\frac{a\left(1 - \mathcal{R}^\star\right) R_{\tau,r}^2}{R_p} + b = a \left.\frac{\mathrm{d}\mathcal{U}}{\mathrm{d}y}\right|_{y=0} + b = b - a\frac{\mathrm{d}P}{\mathrm{d}x} R_p = \mathcal{L}_1. \tag{5.49}$$

It represents the averaged slip length $\mathcal{L}_1$, as defined in (5.39). Therefore the extended FKK equation (5.48) has the same form of the original FKK equation where $\mathcal{L}_1$ replaces the constant slip length $b$. Also, once written in terms of the pressure gradient $\mathrm{d}P/\mathrm{d}x$, the average slip length has the same form of the equivalent slip length of the laminar case given in (5.11). It follows that turbulent flows with the same averaged slip length are characterized by the same reduction of wall friction. In §5.3.2, this property is successfully checked via DNS and the $\mathcal{R}$ values computed from (5.48) for different $a$ and $b$ values are compared with the DNS data.

### 5.3.2 Turbulent drag reduction and velocity statistics

Numerical simulations in the shear-dependent slip-length cases are carried out by first varying $a$ and $b$, the constants for the hydrophobic model along the streamwise direction. Figure 5.2 shows the very good comparison between the $\mathcal{R}$ values computed via DNS (black circles) and the theoretical predictions obtained through the FKK theory (solid lines), studied in §5.3.1.

The drag reduction increases monotonically with $a$ for fixed $b$ and with $b$ for fixed $a$. For fixed $a$, the growth of $\mathcal{R}$ as $b$ increases is more intense for small $a$ values and the drag reduction has a very weak dependence on $a$ for $a \leqslant 10^{-4}$. For $b = 0.02$ and $a$ increasing from $a = 0.001$ to $a = 0.01$, the drag reduction increases from $\mathcal{R} = 33.4\%$ to $\mathcal{R} = 51\%$, which is the maximum $\mathcal{R}$ computed in our study.

As discussed in §5.3.1, an averaged slip length $\mathcal{L}$ is defined. For $b = 0.02$ and $a$ increasing from $a = 0.001$ to $a = 0.01$, the average slip length $\mathcal{L}$ increases from 0.025 ($\mathcal{L}^{+0} = 4.52$) to 0.06 ($\mathcal{L}^{+0} = 10.5$). Flows with the same $\mathcal{L}$ have the same drag reduction, which is verified even when two extreme cases at the maximum $\mathcal{R}$=51% with the same $\mathcal{L} = 0.06$, one with $a = 0.0159$ and $b = 0$ and the other with $a = 0$ and $b = 0.06$, are compared. For this to occur, we notice that $\mathcal{L}^*$ is scaled in outer units and not in viscous units of the hydrophobic case. Our results with $a \neq 0$ agree with the constant-slip-length ones by Min & Kim (2004) and Busse & Sandham (2012) for the same $\mathcal{L}$.

These numerical results confirm the theoretical prediction of monotonic growth of $\mathcal{R}$ with $\mathcal{L}$, given by the FKK equation (5.48) once (5.49) is used. From the definition of $\mathcal{L}$ and from the agreement of $\mathcal{R}$ values for the same $\mathcal{L}$, it also follows that flows with the same $\mathcal{L}$ have the same averaged wall-slip velocity $\mathcal{U}(0)$. From the Fukagata-Iwamoto-Kasagi (FIK) identity (Fukagata *et al.*, 2002), herein extended to include the effect of wall hydrophobicity

Hasegawa *et al.* (2011); Lee *et al.* (2015),

$$C_f = \frac{6}{\mathcal{U}_b R_p} - \frac{6}{\mathcal{U}_b^2} \int_0^1 (1-y) uv_{rey} \mathrm{d}y - \frac{6\,\mathcal{U}(0)}{R_p \mathcal{U}_b^2}, \tag{5.50}$$

it is found that flows with the same $\mathcal{R}$ and $\mathcal{U}(0)$ must have an equally weighted $y$-integrated contribution of the Reynolds stresses $uv_{rey}$. Our numerical calculations confirm this and further show that that the $uv_{rey}$ profiles agree throughout the channel. However, despite the same $uv_{rey}$, the rms profiles of the velocity components do not overlap. For the cases with maximum $\mathcal{R}$=51% ($a = 0.0159$, $b = 0$ and $a = 0$, $b = 0.06$), the $u_{rms}$ profiles differ up to $y = h/3$, their peaks show a 14% difference, and $u_{rms}(0)$ differ by 30%. This demonstrates that locally the behaviour of wall turbulence over these surfaces is markedly different and that the property of same $\mathcal{R}$ for same $\mathcal{L}$ is only to be considered in spatial and temporal averaged terms.

In the constant-slip case ($a$=0), the space- and time-averaged wall velocity $\mathcal{U}(0)$ has also been verified to agree with the following

$$\mathcal{U}(0) = \frac{3b}{3b+1} \left[ \mathcal{U}_b - R_p \int_0^1 (1-y) uv_{rey} \mathrm{d}y \right], \tag{5.51}$$

which is found by averaging the wall boundary conditions (5.1) with $a = 0$, and by substitution of (2.13) into (B.13). As expected, $\lim_{b \to \infty} U(0) = \mathcal{U}_b$, i.e. the laminar plug-flow found in §5.2.1 is recovered because the Reynolds stresses vanish slowly when the turbulent production decreases as the mean-flow wall-normal gradient drops, as shown by Busse & Sandham (2012).

The effect of slip along the spanwise direction is also considered. Along $z$, a constant slip length is considered ($a = 0$) because the wall-shear stress is smaller than along the streamwise direction. In all the tested cases, degradation of drag reduction is found, which confirms the original result by Min & Kim (2004) for constant slip length along both directions. This effect is more intense for small $\mathcal{L}$. $\mathcal{R}$ decreases from 29% to 21.5% when, along $x$, $a = 0.0036$ and $b = 0$, and the $b$ value along $z$ changes from null to 0.02. $\mathcal{R}$ changes only from 51% to 48% when, along $x$, $a = 0.01$ and $b = 0.02$, and $b$ along $z$ again increases from null to 0.02.

The rms of the three velocity components and the Reynolds stresses are shown in figure 5.3 for increasing values of $a$ and $b = 0.02$. The value of $u_{rms}$ at the wall increases with $a$ and the effect of the hydrophobic surface is to attenuate the turbulence activity through the domain, confirming the main results by Min & Kim (2004) for the constant-slip-length

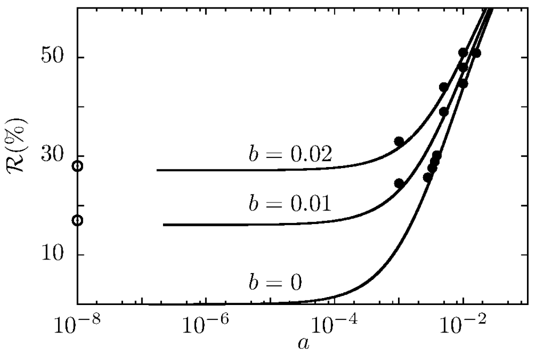


Figure 5.2: Comparison between the $\mathcal{R}$ values computed via DNS (white circles for $a \to 0$ and black circles for finite $a$) and the theoretical prediction obtained through the modified FKK formula (5.48) (lines).

case. The modification is strengthened as $a$ increases, which is consistent with $\mathcal{R}$ becoming larger as the average slip length increases. The streamwise velocity is the less affected, while the wall-normal and the spanwise velocities are attenuated by the same amount. The Reynolds stresses $uv_{rey}$ are the most affected, with the peak decreasing by more than 50%. Figure 5.4 shows the $u_{rms}$ and $uv_{rey}$ scaled with the viscous units of the hydrophobic flow. Near the wall, where the streamwise-velocity boundary conditions are altered, the $u_{rms}$ display a marked differences, i.e. $u^+_{rms}(0)$ and the peak of $u^+_{rms}$ grow with $\mathcal{L}$ as expected. The changes at higher wall-normal locations are less significant and are thus mostly due to the modification of the Reynolds number. The collapse of the Reynolds stresses is confined very near the wall.

It is paramount to verify that the cases studied above can be realized experimentally. The maximum $\mathcal{R}$ case is considered, for which $\mathcal{L}^{+0} = 10.5$. It is assumed that this scaled value corresponds to $\mathcal{L}^* = 100$µm, which is a sensible choice according to several experimental and theoretical works (Choi & Kim, 2006; Daniello *et al.*, 2009; Lauga & Stone, 2003). From these values of $\mathcal{L}^{+0}$ and $\mathcal{L}^*$ the ratio $u^*_\tau/\nu^*$ can first be found. Assuming the liquid to be water ($\nu^*$=$10^{-6}$ m$^2$s$^{-1}$), the channel height $2h^* = 3.4$mm and the bulk velocity $\mathcal{U}^*_b$=1.6ms$^{-1}$ can be educed from the Reynolds numbers $R_{\tau,r} = 180$ and $R_p = 4200$. These values can be realized in a laboratory. Table 5.1 shows more estimated values for channel-

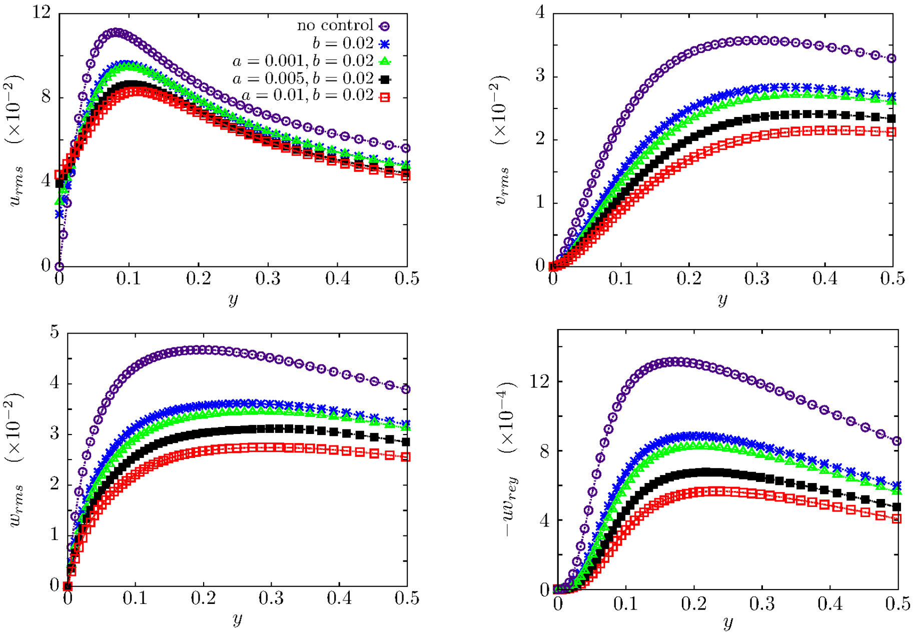


Figure 5.3: Profiles of the rms of the streamwise (top left), wall-normal (top right) and spanwise (bottom left) velocity components and of the Reynolds stresses (bottom right), scaled in outer of the uncontrolled flow.

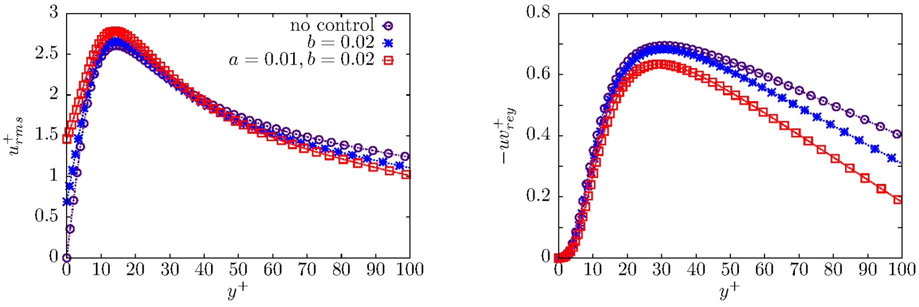


Figure 5.4: Profiles of the rms of the streamwise velocity component (left) and of the Reynolds stresses (right), scaled with the viscous units of the hydrophobic flow.

| fluid | $\nu^*$ (m$^2$s$^{-1}$) | $R_{\tau,r}$ | $R_p$ | $2h^*$ (mm) | $\mathcal{U}_b$ (ms$^{-1}$) |
|---|---|---|---|---|---|
| water | $10^{-6}$ | 180 | 4200 | 3.4 | 1.6 |
| 30% water+ glycerin | $2.5 \times 10^{-6}$ | 180 | 4200 | 3.4 | 4.1 |
| water | $10^{-6}$ | 400 | 10400 | 7.6 | 1.8 |
| 30% water+ glycerin | $2.5 \times 10^{-6}$ | 400 | 10400 | 7.6 | 4.6 |
| water | $10^{-6}$ | 1100 | 33060 | 20 | 2.1 |
| 30% water+ glycerin | $2.5 \times 10^{-6}$ | 1100 | 33060 | 20 | 5.3 |

Table 5.1: Estimates for the channel heights and the bulk velocities for different fluids and Reynolds numbers for $\mathcal{L}^* = 100$μm and $\mathcal{L}^{+0} = 10.5$.

flow experiments, where the same slip lengths in viscous units and in physical dimensions are assumed (the empirical relationship $R_p = 11.05 R_{\tau,r}^{1.143}$ was used for the estimates at higher Reynolds numbersPope (2000)). Our estimated flow quantities are comparable with those of Rosenberg *et al.* (2016), who, for the first time, measured turbulent drag reduction (maximum $\mathcal{R} = 14\%$) in a Couette flow over SLIPS. The friction Reynolds number was $R_\tau = 140$, the maximum velocity was 4.4 m/s, and the gap thickness was 2 mm. Although no information was reported on whether their slip length depended on the shear rate, $\mathcal{L}^*$=138±55 μm ($\mathcal{L}^+ \approx 10$) is comparable to ours and to Choi & Kim (2006)'s.

A further comment is due on the results by Choi & Kim (2006), shown in figure 5.1 (right). By extrapolating the data, such surface would produce a slip length of 100μm when $\mathcal{S}^*$=450s$^{-1}$. We compare these quantities with our predictions in table 5.1. For the first case in table 5.1, $b^*$=36μm is assumed, $\mathcal{S}^*$ is about 10,000s$^{-1}$ and $a^*$=0.01μm s. The shear rate is about 20 times larger Choi & Kim (2006)'s and the constant of proportionality $a^*$ is one order of magnitude smaller than Choi & Kim (2006)'s. It follows that in a turbulent flow a much lower $a$ than that found by Choi & Kim (2006) would lead to significant shear-dependent effects because the wall-shear stress is much larger. This analysis proves that in wall-bounded turbulent flows, where the shear rate are orders of magnitude larger than in the laminar flows, hydrophobic surfaces are likely to feature slip lengths with shear dependence.

Further evidence of shear-dependent slip lengths emerges from the recent DNS investigation by Jung *et al.* (2016), where turbulent channel flows at $R_{\tau,r} = 180$ over thin air layers have been simulated for the first time. Their figure 5f demonstrates that the slip length depends on the wall-shear stress for high-drag-reduction cases with zero mass flow rate in the air layer (refer to their figure 1b for a schematic of the flow domain). We have interpolated the data in their figure 5f with a power law, i.e. $u_s^{+0} = a_j(0.01\ \mu_r \partial u^{+0}/\partial y\big|_{y=0})^{\beta}$, where $\mu_r$ is the ratio between the viscosities of water and air. The least squares fitting method leads to $a_j = 0.006$ and $\beta = 2.02$. This means that for this type of idealized hydrophobic surfaces our boundary condition (5.17) with $b = 0$ and $a = 0.04$ (computed by rescaling $a_j$) is a very good model relating the instantaneous streamwise slip velocity and the streamwise velocity gradient at the water-air interface. According to our figure 5.2, this value of $a$ would lead to $\mathcal{R}$ above 60%, which is consistent with the wall-shear stress reduction computed by Jung *et al.* (2016). It is certainly necessary to carry out further experimental and modelling work for flows at high wall-shear stress, especially in the turbulent flow regime, in line with the numerical study of Jung *et al.* (2016) and the experimental study of Rosenberg *et al.* (2016). The main objectives would be to identify hydrophobic surfaces featuring shear-dependent slip lengths and to obtain further constitutive relations between the slip length and the shear rate.

### 5.3.3 Power spent by the turbulent flow on the hydrophobic surface

In wall-bounded flow control problems, the performance of a flow system must be evaluated by the drag reduction and by the power exchanged through the surface. To the best of our knowledge, this is the first time that $\mathcal{P}_{sp} = 100\mathcal{W}/\mathcal{P}_{x,r}$, i.e. the percent power that the fluid exerts on the hydrophobic surface with respect to the power required to pump the fluid along $x$ in the uncontrolled case, is taken into account (refer to (2.20)-(2.21) and §2.1. This power is obviously null in the uncontrolled case. For the shear-dependent slip case with $a = 0.01$ and $b = 0.02$, $\mathcal{R} = 50\%$ and $\mathcal{P}_{sp} = 16\%$, and for the constant-slip case with $b = 0.02$ ($a = 0.0$), $\mathcal{R} = 29\%$ and $\mathcal{P}_{sp} = 12\%$.

In the case of a hydrophobic surface modelled by an alternating pattern of in-plane no-slip/free-shear strips without penetration, the power spent on the surface, given by equation (2.27), is null because $U(0) = 0$ over solid portions of the wall and $\partial U/\partial y(0) = 0$ over air pockets (Martell *et al.*, 2009). In reality, the turbulent flow expends energy to shear the enclosed air pockets by viscous action. This power transfer is responsible for the detachment and disappearance of the air bubbles trapped in the surface, which leads to the degradation of its drag-reduction properties. As argued by Aljallis *et al.* (2013) and Govardhan *et al.* (2009), the loss of drag reduction is not due to surface damage, but to the high wall shear and pressure that cause the depletion of air from the wall, to a higher water-wetted area, and thereby drag increase. Further work is certainly needed to compare the power spent at the wall computed via the effective slip model and the power exerted by the flow on the air pockets. In the case of SLIPS (Wong *et al.*, 2011; Rosenberg *et al.*, 2016), power is instead expended by the flowing liquid onto the liquid substrate that infuses the rigid porous matrix, mainly by the shear stress at the interface between the two liquids.

An exchange of power at the surface in controlled wall-bounded turbulent flows obviously also occurs in several flow control techniques such as spanwise wall oscillation (Jung *et al.*, 1992), wall travelling waves (Quadrio & Ricco, 2011; Roggenkamp *et al.*, 2015), and spinning discs (Ricco & Hahn, 2013). These are active methods because power is introduced into the fluid system from the exterior of the domain. This follows mathematically from the tangential velocity induced by the wall actuation decaying on average along $y$ in a thin viscous layer within the turbulent flow. In the hydrophobic-surface case, a passive technique, power is instead exerted by the fluid on to the surface because, on average, both the slip-wall velocity and the wall-normal gradient of the streamwise velocity at the wall are positive. Therefore, $\mathcal{P}_{sp}$ for hydrophobic surfaces is of opposite sign when compared with that of active techniques. To compute the net power saved for active techniques, the power supplied at the wall $\mathcal{P}_{sp}$ is subtracted from the saved $\mathcal{P}_x$ (which coincides with $\mathcal{R}$ when the mass flow rate is constant), as discussed in Ricco & Hahn (2013). For the hydrophobic-surface case, the net power saved instead coincides with the saved $\mathcal{P}_x$ as $\mathcal{P}_{sp}$ is not supplied externally.

Passive techniques have often been classified as methods that do not involve exchange of energy through the boundaries. Riblets are one of these methods. Hydrophobic surfaces (and also compliant surfaces) can still be categorized as passive, although they absorb energy from the fluid in motion. Hydrophobic surfaces can thus be named passive-absorbing methods while geometry-modifying techniques, such as riblets, can be called passive-neutral.

Another point on the power spent ought to be discussed. As remarked following (2.42), the feedback boundary conditions extracted from the Lyapunov stability analysis coincide with those used to represent hydrophobic surfaces. Therefore two different physical systems are modelled through the same boundary conditions (2.41) and (5.17). In Balogh *et al.* (2001), the boundary conditions are proposed to model an active technique for which the wall-shear stress is measured locally by distributed flush-mounted sensors to activate actuators which, in response to the wall-shear stress measurements, induce a wall streamwise velocity. As the boundary conditions in Balogh *et al.* (2001)'s case and in the hydrophobic case coincide, Balogh *et al.* (2001)'s surface absorbs power from the flow just like in the hydrophobic case. This sounds in contrast with Balogh *et al.* (2001)'s idea of modelling an active drag reduction technique, which by definition requires an injection of power from the exterior of the system. This apparent contradiction is resolved if one accounts for the electrical and mechanical power spent by the sensors and actuators below the walls, which is not modelled by the boundary conditions (2.41) and (5.17).

### 5.3.4 Vorticity, vortices, and streaks

The rms of the vorticity vector components are shown in figure 5.5 for the uncontrolled, constant-slip-length, and shear-dependent slip-length cases. The graphs on the left show the profiles scaled in outer units, while the graphs on the right are nondimensionalized using viscous units based on the drag reduction friction velocity.

In outer units, the fluctuations of all the vorticity components are strongly attenuated when compared to the uncontrolled case, indicating a strong reduction of turbulent activity. Like the uncontrolled case, the hydrophobic $\omega_{x,rms}$ profiles display a local minimum at the edge of the viscous sublayer and a higher local maximum, located in the buffer region, a sign of the presence of streamwise vortices (Kim *et al.*, 1987*a*). The wall-normal position of the local minimum is only slightly moved upward, while the second maximum is more significantly shifted away from the wall in the hydrophobic case, a behaviour also observed in the opposition control flows (Choi *et al.*, 1994; Jiménez, 1994) and in flows over porous walls (Jimenez *et al.*, 2001). The attenuation and upward shift of $\omega_{x,rms}$ is consistent with the wall-shear stress reduction as high skin-friction regions are closely related to streamwise vortices Kravchenko *et al.* (1993). When scaling in drag-reducing viscous units, a marked difference in the $\omega^+_{x,rms}$ profiles still occurs, particularly in the buffer region and beyond. This proves that these changes are not an effect of the friction Reynolds number, which decreases when the wall-shear stress is reduced, but the indication of a true flow modification throughout the whole channel.

The $\omega_{y,rms}$ and $\omega_{z,rms}$ profiles show a significant reduction throughout the channel for the

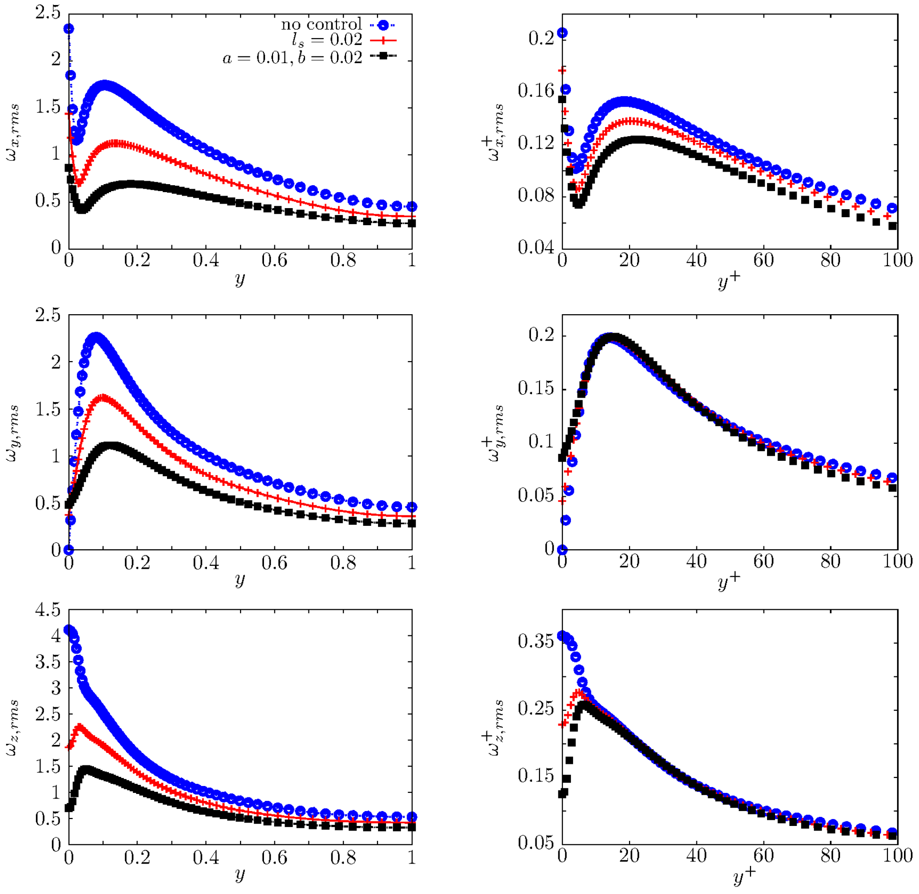


Figure 5.5: Rms profiles of the streamwise (top), wall-normal (middle), and spanwise vorticity (bottom). Quantities in the left graphs are scaled by the uncontrolled $u_{\tau,r}$ and quantities in the right graphs are scaled by the drag-reducing $u_\tau$.

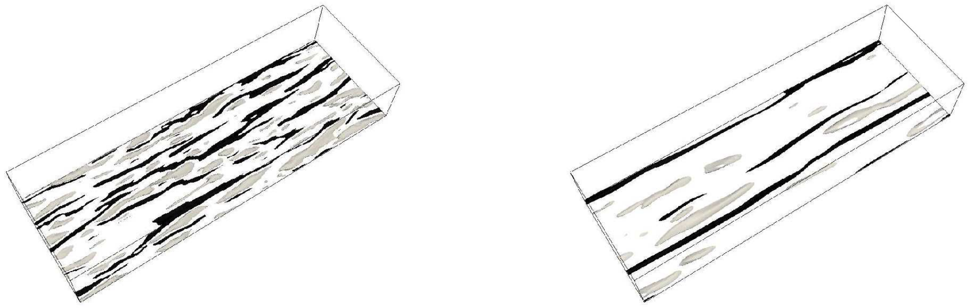


Figure 5.6: Low- (black) and high-speed (grey) streaks for the (a) no-slip and (b) shear-dependent slip-length cases at $y^{+0} = 12$ ($y = 0.07$), defined according to (5.52).

hydrophobic cases when scaled in outer units. When scaled in viscous units, these profiles are only altered up to about $y^+ = 10$, showing very good collapse at higher locations. This demonstrates that, differently from the streamwise velocity, the changes at $y^+ > 10$ are solely due to the change of Reynolds number caused by the drag reduction. The collapse of $\omega^+_{y,rms}$, which quantifies the alternation of low- and high-speed streamwise elongated regions, clearly shows that the low-speed streaks maintain their kinematic properties when scaled in viscous units. The strongest near-wall reduction is displayed by $\omega_{z,rms}$ as a direct consequence of the non-zero wall slip because $\omega'_z$ is dominated by $\partial u'/\partial y$ at the wall. These smaller fluctuations of $\omega_{z,rms}$ lead to a decrease of mean wall-shear stress via nonlinear interactions. A further comment on the velocity and vorticity statistics very near the wall ($y^+ < 10$) is due. Although the slip-length model is representative of either lotus-leaf-type surfaces with trapped air pockets or pitcher-plant-type SLIPS, very near the wall these statistics are likely not be the exact representation of the first kind of surfaces because of the spatial inhomogeneity of the texture (alternating solid patches and air pockets). However, they more precisely model the behaviour over SLIPS because the liquid infused in the porous substrate is homogeneously distributed as a thin layer below the overflowing liquid.

The low-speed streaks, streamwise-elongated regions of slow fluid compared to the mean flow (Kline *et al.*, 1967; Panton, 2001), are further analyzed to evince how these structures

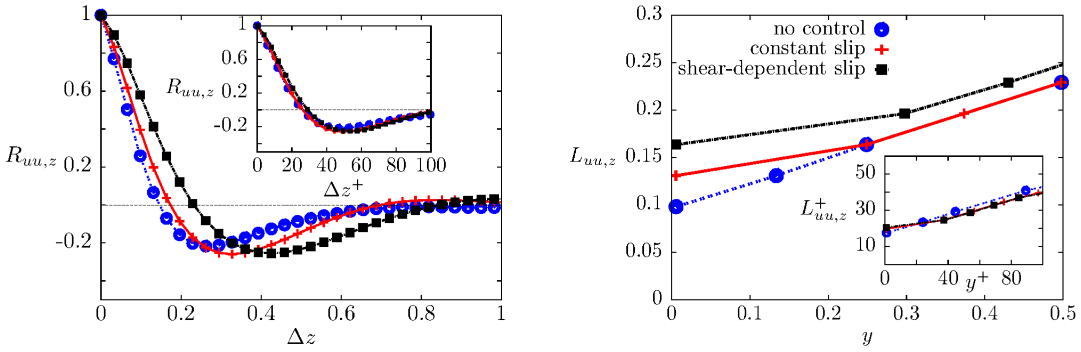


Figure 5.7: Two-point spanwise correlation for the fluctuating streamwise velocity (left), defined in (5.53), at $y^{+0} = 12$ ($y = 0.07$) and spanwise-correlation length (right), defined in (5.54), as a function of $y$.

are affected by the hydrophobicity. Low- and high-speed streaks were defined as follows:

$$\text{Streak detection} \rightarrow \begin{cases} \text{Low speed if :} \quad u'(x, y, z, t) \leqslant -\chi \max\limits_{y} u_{rms}(y) \\ \text{High speed if :} \quad u'(x, y, z, t) \geqslant \chi \max\limits_{y} u_{rms}(y), \end{cases} \tag{5.52}$$

where $\chi = 0.9$ is the threshold parameter. Figure 5.6 shows the streaks in the $x - z$ plane at $y^{+0} = 12$ ($y = 0.07$), defined according to (5.52). The low-speed streaks over the hydrophobic surface appear more sporadically and more stretched along the streamwise direction than in the uncontrolled case. The high-speed streaks are also less numerous, more elongated, and wider than in the uncontrolled case.

To quantify the spreading of the low-speed streaks, we study the streamwise-velocity correlation functions along the spanwise direction $R_{uu,z}$, defined as

$$R_{uu,z}(\Delta z, y) = \frac{(L_x L_z)^{-1} \left[\overline{u'(x, y, z, t) u'(x, y, z + \Delta z, t)}\right]_{\mathcal{I}xz}}{u_{rms}^2}. \tag{5.53}$$

The correlation $R_{uu,z}$ is shown in figure 5.7 (left) for $y^{+0} = 12$ ($y = 0.07$). For the no-slip case, the first minimum is at $\Delta z^{+0} = 50$, resulting in the widely-reported streak spacing of 100 wall units (Kline *et al.*, 1967; Panton, 2001). The minimum shifts to higher separation $\Delta z$, which indicates a larger spanwise streak spacing. The correlation $R_{uu,z}$ is also expressed versus $\Delta z^+$, scaled in drag-reducing viscous units, and shown in the inset of figure 5.7 (left). The uncontrolled, constant and shear-dependent models collapse on top of each other and present a minimum at $\Delta z^+ = 50$. This confirms the results of drag-reduction viscous scaling shown in figure 5.5 by $\omega^+_{y,rms}$, which is a measure of the alternating high and low streamwise velocity fluctuations near the wall.

The spanwise correlation length $L_{uu,z}$ is computed from $R_{uu,z}$ as

$$L_{uu,z}(y) = \Big\{\min(\Delta z) \mid R_{uu}(\Delta z, y) < e^{-1}\Big\} \tag{5.54}$$

to quantify the streak width further (Busse & Sandham, 2012). Figure 5.7 (right) shows that $L_{uu,z}$ increases with $y$ and, for a given $y$, $L_{uu,z}$ attains the largest values in the shear-dependent slip-length case, especially in the near-wall region. The inset of figure 5.7 (right) further demonstrates that the characteristic spanwise spacing of the low-speed streaks displays a good scaling in drag-reduction viscous units.

### 5.3.5 Probability density functions

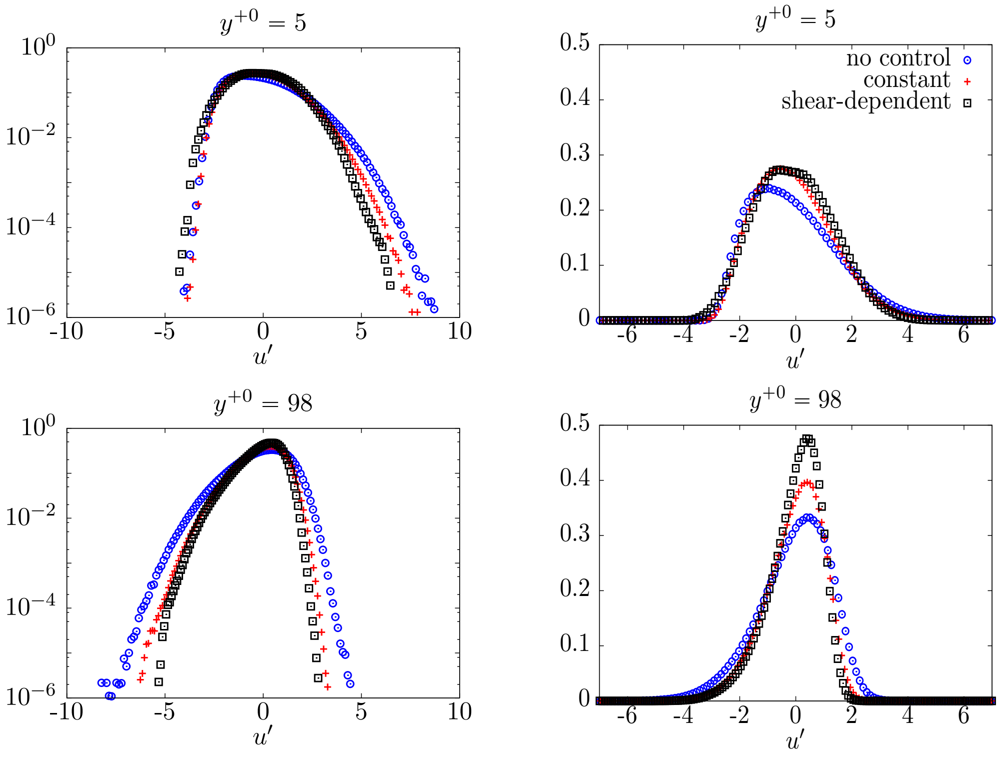


Figure 5.8: From top to bottom the PDF the fluctuating streamwise velocity are shown in log (left) and linear scale (right).

The flow is further analyzed through the probability density function (PDF) of the fluctuating streamwise and spanwise velocity components (Pope, 2000). The PDF is useful to detect a strongly intermittent velocity signal, which is characterized by a long tail in the PDF trend at high velocity. The results are shown in figure 5.8 for the streamwise velocity scaled in outer units. Logarithmic scale is used to highlight the differences for small PDF values and linear scaling is used to observe the changes of the maximum PDF values.

The near-wall streamwise velocity is more intermittent for high velocities than for the low velocities, and it becomes less so when the wall is hydrophobic. The attenuation due to hydrophobicity is more pronounced for $u' > 0$ than for $u' < 0$, and this effect is more significant at $y^{+0} = 5$ ($y = 0.03$) than at $y^{+0} = 10$ ($y = 0.06$) (not shown here). The near-wall fluctuations of $u'$ contribute significantly to $\omega_z$, which is dominated by the wall-normal gradient of $u'$ (Choi *et al.*, 1994). It further follows that, if such fluctuations are reduced, the drag will be reduced because the mean wall-shear stress will be less modified by these fluctuations through nonlinearity. The reduction of $\omega_{z,rms}$, shown in figure 5.5, is therefore consistent with the attenuated $u'$ fluctuations. At $y^{+0} = 98$ ($y = 0.55$), i.e. in the logarithmic region, the negative high-velocity fluctuations are instead more altered by the hydrophobic surface.

Coherent vortical structures in the wall region are closely related to the production of Reynolds stresses (Wallace *et al.*, 1972; Jiménez & Moin, 1991; Robinson, 1991*b*,*a*). Therefore, the flow is further examined by comparing vortex population in the uncontrolled and controlled cases. In the literature, various definitions have been proposed (Hunt *et al.*, 1988; Jeong & Hussain, 1995; Chong *et al.*, 1998; Cucitore *et al.*, 1999), all of which rely on quantities relative to the velocity gradient tensor $\boldsymbol{\nabla}\mathbf{u} = \boldsymbol{S} + \boldsymbol{\Omega}$, where $\boldsymbol{S}$ and $\boldsymbol{\Omega}$ respectively stand for the symmetric and antisymmetric part of $\boldsymbol{\nabla}\mathbf{u}$ also known as the strain rate and rotation tensors. In the following, the second eigenvalue of $\boldsymbol{S}^2 + \boldsymbol{\Omega}^2$ tensor ($\lambda_2$) is used (Jeong & Hussain, 1995) to compare the vortical structures between the no-slip configuration and the finite slip models. In this framework, a vortex core is detected by requiring for $\lambda_2$ to be negative. Figure 5.9 shows that the vortical structures are strongly reduced when the streamwise slip control is applied. This reduction being larger in the shear-dependent slip model. Quasi-streamwise vortices are important in the self-sustaining cycle (Bernard *et al.*, 1993*b*; Panton, 2001) subsequently being a source for new structures in the flow.

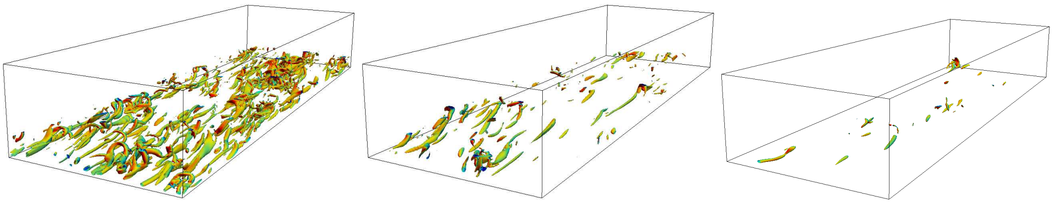


Figure 5.9: View of the channel showing instantaneous vortical structures with $\lambda_2 = -1$ respectively from left to right: no-slip flow, constant slip length model with $l_s = 0.02$ and the shear-dependent slip-length model with $(a, b) = (0.01, 0.02)$.

### 5.3.6 Principal strain rates

To gain further insight in the physical mechanisms, we analyze the orientation of the vorticity vector $\boldsymbol{\omega}$ and the eigenvalues of the strain rate tensor $\boldsymbol{S}$, called principal strain rates and denoted by $s_i$, $i \in [\![1, 3]\!]$. The associated eigenvectors $\boldsymbol{e}_i$ are the principal axes of the strain rate tensor. The vorticity $\boldsymbol{\omega}$ and the eigenvectors $\boldsymbol{e}_i$ define three angles $\theta_i$ that satisfy $\cos\theta_i = \boldsymbol{\omega} \cdot \boldsymbol{e}_i/(|\boldsymbol{\omega}|\,|\boldsymbol{e}_i|)$. The compressional eigendirection is $\boldsymbol{e}_3$ and the extensional one is $\boldsymbol{e}_1$ (Hamlington *et al.*, 2008). The intermediate eigenvector $\boldsymbol{e}_2$ tends to align with $\boldsymbol{\omega}$. The associated eigenvalues are ordered as $s_3 \leqslant s_2 \leqslant s_1$, with $s_1 > 0$ and $s_3 < 0$. This is the first time this approach is employed to study a drag-reduction flow.

The PDF of $\cos\theta_i$ associated with the extensional and compressional eigendirections are first computed and shown in figure 5.10 (left) at $y^{+0} = 10$ $(y = 0.06)$. The alignment of the second eigendirection (not shown here) is not affected in the hydrophobic case. The extensional and compressional eigendirections instead show more pronounced peaks at $\cos\theta = 0$. Hydrophobic surfaces thus enhance the likelihood of the extensional and compressional eigendirection to be perpendicular to the vorticity vector. Furthermore, the extensional eigendirection from $y^{+0} = 10$ to $y^{+0} = 40$ $(y = 0.22)$ present the same ratios in the PDF maximum between the uncontrolled-wall and hydrophobic cases (not shown).

The alignment of the eigendirections and $\boldsymbol{\omega}$ can be related to the turbulence dynamics. The $\boldsymbol{\omega}$ alignment with the eigendirections of the strain rate tensor $S_{ij}$ can be interpreted by the vorticity equation:

$$\frac{D\omega_i}{Dt} = S_{ij}\omega_j + \frac{1}{R_p}\nabla^2\omega_i, \tag{5.55}$$

where $D/Dt$ is the substantial derivative, $S_{ij}$ are the components of the strain rate tensor and $\omega_i = -\epsilon_{ijk}\Omega_{jk}$ (where $\epsilon_{ijk}$ is the Levi-Civita symbol), with $\Omega_{jk}$ being the components of the rotation tensor. The first term in the right-hand-side of (5.55) is also found in the vortex stretching term:

$$\omega_j \frac{\partial u_i}{\partial x_j} = \omega_j S_{ij} + \omega_j \Omega_{ij}. \tag{5.56}$$

The second term on the right-hand-side of (5.56) vanishes, while the amplitude of the first term can be expressed as (Betchov, 1956; Hamlington *et al.*, 2008):

$$|S_{ij}\omega_j| = \omega\sqrt{s_i^2(\boldsymbol{e}_i \cdot \boldsymbol{e}_\omega)^2}, \tag{5.57}$$

where $\omega^2 = \omega_i\omega_i$ and $\boldsymbol{e}_\omega$ is the vorticity unit vector. It is clear from (5.57) that an attenuation of either the alignment term $\boldsymbol{e}_i{\cdot}\boldsymbol{e}_\omega$, the vorticity amplitude $\omega$ or the eigenvalues $s_i$ contributes to a reduction of vortex stretching.

After taking the product of (5.55) and $\omega_i$, the enstrophy production $\omega_i S_{ij}\omega_j$ can be linked to the quantities in (5.57) to explain the changes in enstrophy dynamics. The enstrophy production can be written as:

$$\omega_i S_{ij}\omega_j = \underbrace{\omega^2 s_1 \cos^2\theta_1}_{\text{I}} + \underbrace{\omega^2 s_2 \cos^2\theta_2}_{\text{II}} + \underbrace{\omega^2 s_3 \cos^2\theta_3}_{\text{III}}. \tag{5.58}$$

In (5.58) term I is always positive, term III is always negative, and II is positive in average. As shown in figure 5.10 (right), as terms I and III almost compensate, the main contribution to the enstrophy production is due to term II. In the hydrophobic case, $\cos\theta_1$ and $\cos\theta_3$ are strongly attenuated near the wall because the extensional and compressional eigenvectors tend to be perpendicular to the vorticity. The observation for $\boldsymbol{e}_1$ is also consistent with Buxton *et al.* (2011), who mention that the perpendicular orientation of $\boldsymbol{e}_1$ with respect to $\boldsymbol{\omega}$ underlines an enstrophy attenuating mechanism. Figure 5.11 shows that the total enstrophy production is significantly reduced compared to the uncontrolled-wall case, reflecting the attenuation of the intensity of vortical structures.

### 5.3.7 Quadrant analysis

The quantity given by (2.76) and described in details in 2.5, is now computed for the rigid-wall case and for the hydrophobic cases. This representation exhibits the evolution of each quadrant. In particular for control problems, the sweep and ejection quadrants are those which should be focused on.

Figure 5.12 shows that Q2 and Q4 events are reduced in the hydrophobic cases, especially in the shear-dependent slip-length case. For a fixed $b$, the larger $a$, the larger is the

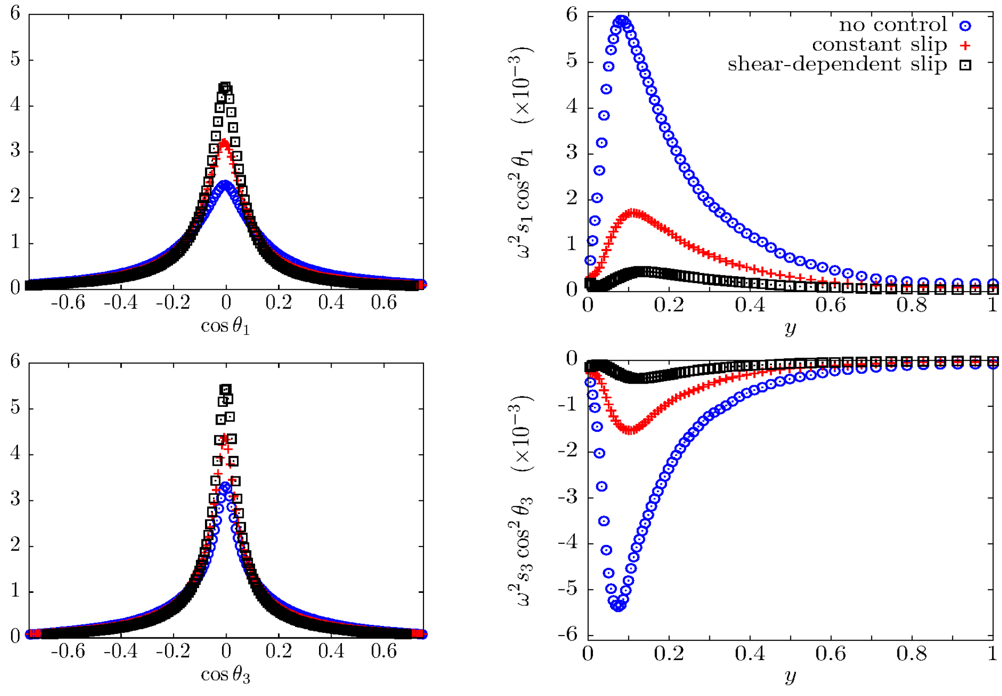


Figure 5.10: Left: PDF of the extensional (top) and extensional (bottom) eigendirections. Right: contributions to the enstrophy production term given by (5.58) for the extensional (top) and compressional (bottom) eigendirections at $y^{+0} = 10$ ($y = 0.06$).

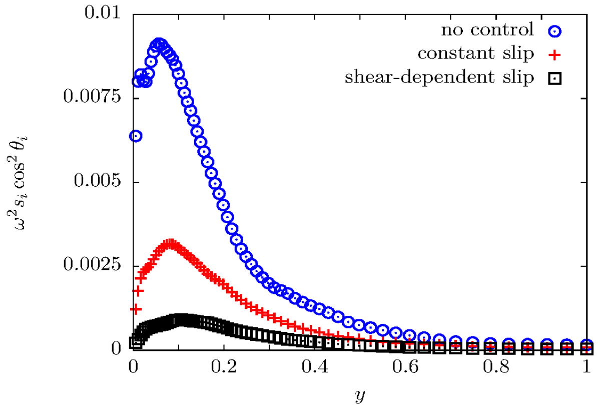
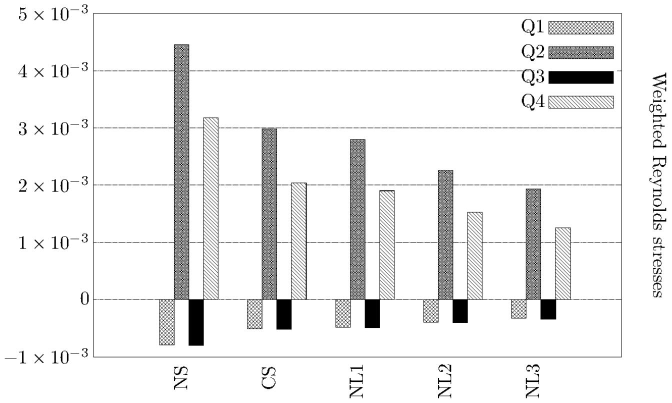


Figure 5.11: Total enstrophy production rate from (5.58).

reduction in ejections and sweep events. This is consistent with the fact that the largest $\mathcal{R}$ occurs for the shear-dependent slip-length model with parameters $(a, b) = (0.01, 0.02)$.



1

Figure 5.12: Quadrant contribution of the weighted Reynolds shear-stress $\mathcal{Q}$, given by equation (2.76). The no-slip case (NS) the constant slip model (CS) with $l_s = 0.02$ and three shear-dependent slip-length cases (NL1) using $a = 0.001, b = 0.02$, (NL2) associated with $a = 0.005, b = 0.02$ and (NL3) with $a = 0.01, b = 0.02$ are computed.

## 5.4 Summary and outlook

In this paper laminar and turbulent channel flows with hydrophobic surfaces featuring shear-dependent slip lengths have been investigated theoretically and numerically. The slip length has been assumed to depend linearly on the wall-shear stress and therefore two constants, $a$ and $b$, model the hydrophobic surface. In the turbulent flow case, the slip length is time-dependent and spatially inhomogeneous as it depends on the local instantaneous velocity gradient at the wall.
The main results are summarized in the following.

- *Laminar channel-flow solution*
  The laminar channel-flow solution with shear-dependent slip length has been derived analytically. If the shear-dependent slip length is substituted into the formula for the velocity profile, the final expression has the same form of the constant slip-length formula derived by Min & Kim (2005). The increase of mass flow rate under

constant pressure gradient conditions and the decrease of wall-shear stress under constant mass flow rate conditions have been quantified. The constants $a$ and $b$ have been extracted from experimental data of laminar flows by Churaev *et al.* (1984) and Choi & Kim (2006).

- *Nonlinear Lyapunov stability analysis*
  We have carried out a three-dimensional nonlinear Lyapunov stability analysis of the channel flow between hydrophobic walls featuring a shear-dependent slip length. The stability conditions have been expressed in terms of inequalities involving the Reynolds number $R_p$ and the constants $a$ and $b$. As for a standard channel, the critical Reynolds number is very small, $R_p$=1/4, which is proper of microfluidic flows. Therefore, this analysis has not been useful to shed light on the mechanism of turbulent drag reduction. Nevertheless, it has been instructive to extract the stability bounds and because we have recognized that the feedback-control laws found through the analysis coincide with the slip hydrophobic-wall conditions.

- *Fukagata-Kasagi-Koumoutsakos theory*
  The theoretical formula for drag reduction prediction by Fukagata *et al.* (2006) has been extended to the shear-dependent slip-length case. The computed drag reduction values show very good agreement with the direct numerical simulation results.

- *Turbulent drag reduction*
  It increases monotonically with both $a$ and $b$, and also with $\mathcal{L}$, the average slip length, scaled in outer units. It is found that flows featuring the same $\mathcal{L}$ have the same drag reduction and the same Reynolds stresses profiles, irrespectively of the values of $a$ and $b$. The rms profiles of the streamwise velocity nevertheless do not overlap, demonstrating that the local behaviour of wall turbulence over these surfaces is markedly different and that the property of same $\mathcal{R}$ for same $\mathcal{L}$ is only to be considered in averaged terms. If hydrophobicity along the spanwise direction is taken into account, the drag reduction effect deteriorates. Furthermore, by rescaling our numerical slip parameters and flow conditions, we have found that even a quite weak dependence of the slip length on the wall shear can produce substantial differences in the drag-reducing properties because of the large turbulent wall shear. These $a$ values are much smaller than the experimental ones reported by Choi & Kim (2006) for a laminar flow.

- *Viscous-units scaling of near-wall statistics*
  Scaling the vorticity rms profiles with the drag-reduction friction velocity reveals that the streamwise vortices are strongly attenuated, while the low-speed streaks maintain their characteristics spacing. This is confirmed by rescaling the velocity correlations along the spanwise direction.

- *Power spent by the turbulent flow on the hydrophobic surface*
  Because of the local slip, the wall-shear stress exerts power on the hydrophobic surface, which is a non-negligible portion of the power required to propel the fluid along the streamwise direction. This shearing action is responsible for the detachment of the air bubbles from their pockets, which leads to surface degradation and the progressive loss of the drag-reducing properties. While the slip-length hydrophobic model accounts for this power expenditure, if hydrophobic surfaces are modelled as alternating patterns of no wall slip (solid boundary) and shear-free slip (air pockets), this power is null. Future research should therefore focus on the viscous effects between the turbulent liquid flow and the air pockets. For lotus-leaf-type surfaces, further analysis should focus on the precise specification of the texture geometry and of the flow motion inside the air pockets. This simulation would required coupled Navier-Stokes solvers for the liquid and gas flows with changing interface geometry to resolve fully the interaction between the turbulent liquid flow and the flow in the air pockets. Such study would clarify the influence of the liquid and gas viscosities and also reveal the role of fluctuating pressure and kinetic energy exchange at the wall. These two latter quantities also contribute to the power exchange at the wall (Hinze, 1975) and are not modelled if the wall-normal velocity is assumed to vanish at the interface between the turbulent flow and the gas bubbles. In the turbulent regime, steps in this direction have been taken by Garcıa-Mayoral *et al.* (2014), who relaxed the no-penetration condition at the wall, and by Jung *et al.* (2016), who simulated a turbulent channel flow over thin air layers. Studies in the laminar regimes include Schönecker & Hardt (2013) and Schönecker & Hardt (2015). Further theoretical work on the geometrical changes the liquid-gas interface due to pressure and its impact on the drag reduction properties in microfluidic flows has been carried out by Davis & Lauga (2009). In order to quantify the power spent by the liquid flow on the lotus-leaf surfaces, one idea would be to carry out an energy balance at the wall and to measure the kinetic energy of the bubbles as they detach from the

surface as a consequence of the shearing and pressure action of the liquid flow. This study should include a detail analysis of the stability of the sheared air pockets.

- *Principal strain rates*
  In the hydrophobic case, the compressional and extensional eigenvectors of the strain rate tensor show a marked tendency to orient perpendicularly to the vorticity vector. This in turn causes a reduction of the vortex stretching term in the vorticity equation and an attenuation of the enstrophy production.

# Chapter 6

# Spinning rings and wall-distributed forcing combination

## 6.1 Introduction

As mentioned in the previous chapters, flow control is a key component in fluid mechanics. It is only recently that closed-loop have been used in flow control problems. Although presenting more potential for modifying the dynamics of a system, closed-loop control strategies are not as widely used as open-loop or feed-forward techniques. As they aim at continuously reacting to intermittent events occurring in the flow, feedback control systems are generally more expensive to implement either in experiments or numerical simulations. For this reason open-loop flow control techniques are predominantly used. The latter enforce a predetermined fixed response for all times and states developing in the system. Tackling these structures is a daunting task as the controller interacts with a set of turbulent fluctuations developing in the flow (e.g. coherent structures and bursts). The main drawback lies in the way these react to small-scale fluctuations: open-loop systems are acting indifferently on strong or weak turbulent events, while closed-loop systems consistently adapt to the response from the flow.

In the flow control literature and more particularly in drag-reduction problems, control methods have only relied on a single strategy to act upon the flow. The combination of control techniques, to the best of our knowledge, has not been dealt with to assess their performance. In the present study, three control techniques are chosen with the view of a potential practical implementation. This choice entails a combination of active-passive (rotating actuators-hydrophobic surfaces) and active-active (rotating actuators-blowing/suction) control strategies. The objective of the present study therefore focuses on understanding how the combination of two control methods can be used and to assess the impact of this combination on the drag reduction rate.

## 6.2 Strategy for the combination

Despite the recent advances in Micro-Electoromechanical Systems (MEMS) (Gad-el-Hak, 2001), the main obstacle to a foreseeable future large-scale implementation of closed-loop control techniques remains the cost. Nevertheless, numerical studies have tried to exploit the properties of these techniques. However, the opposition control technique studied numerically by Choi *et al.* (1994); Hammond *et al.* (1998); Chung & Talha (2011) (4.6) is particularly interesting on the ground of its simplicity for fundamental numerical studies. This technique relies on the intimate relationship between the so-called quasi-streamwise vortices (QSV) and the wall-shear stress upstream of these structures Kravchenko *et al.* (1993); Bernard *et al.* (1993*b*). Kravchenko *et al.* (1993) showed that higher skin-friction values are correlated to streamwise vortices close to the wall. Therefore, by mitigating regions where QSV are present. Therefore the reduction of high skin-friction regions should induce a reduction of the overall drag. Opposition control uses sensors at a fixed plane in the flow located in the range where $y^{+0} \in [10, 25]$ to measure the wall-normal velocity. This technique can either be used through in-phase or out-of-phase configuration: wall-normal velocity is measured in the flow and fed back at the wall either with a positive or negative sign (hence the qualification of opposition). In this way, the boundary conditions that are enforced at the wall necessarily imply a zero wall-normal velocity at the plane where sensors are located. By suppressing sweep/ejection events related to energy containing coherent structures present in the near-wall region (Robinson, 1991*b*), this techniques achieved up to 25% drag- reduction rate based on out-of-phase wall-normal control. In addition, this technique exhibits a high ratio between the power saved and the ideal power input (neglecting losses from an implementational viewpoint such as valve losses). Therefore the power input is negligible. The skin-friction reduction is also proportional to the surface under control Choi *et al.* (1994). A final point relative to the opposition control technique concerns the Reynolds number dependence. Indeed, LES have shown that this technique's scaling with the Reynolds number is not very good. Pamiès *et al.* (2007) conducted LES at $R_{\tau,r} = 960$ and reached an optimal drag reduction value of 17.9%.

The active method introduced in 4.5 shows that the wall tangential velocity varies linearly with the radial coordinate $r$ from the centre (zero velocity) to the edge (maximum $W$ tip velocity). It follows that the most energetic part contributing to the flow is contained in the outermost region of the discs. Following this reasoning, an annular rotating device can replace the full discs Wise & Ricco (2014). This configuration is shown in figure 1.2 with $r$ and $R$ the minor and major radius respectively. Their ratio is given by $a = r/R$ as shown in figure 6.1 The rotating rings configuration is particularly interesting as it can be combined with other control methods. Indeed, the inner region of the rings (delimited within the radius $r$) is not moving, leaving the possibility to act upon it with another control technique. This study focuses on such combinations. Therefore one will distinguish between primary control technique, i.e. the steady rotating rings, and the secondary control method (either active or passive) complementing the primary one. The two secondary control techniques employed in the present study are based on opposition control and hydrophobic surfaces. The latter have been studied in details in the previous chapter and are going to be used in their original setting, i.e. with constant linear slip length. Small-scale systems design makes such surfaces promising candidates for various industrial applications. This argument remains valid for the other secondary control method based on opposition control. With this perspective and the parameter estimates given in Ricco & Hahn (2013) for a practical implementation of the rotating actuators, it is not unreasonable to consider a combination of the primary and secondary control methods for targeting skin-friction drag reduction. The combination of the two control techniques should pertain as long as the secondary control methods require low power input into the system. It is presently the case for the opposition control technique as mentioned earlier and also verified for hydrophobic surfaces as a passive technique.

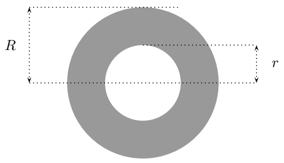


Figure 6.1: Schematic representation of a ring actuator

The methodology for this study is based on the DNS of a turbulent channel flow in which the primary control, consisting of steady rotating rings embedded at both walls, is combined with the secondary control based either on opposition control or hydrophobic surfaces. The secondary control is imposed in all the regions where the rotating rings are not present.

## 6.3 Primary and secondary control methods

The Incompact3d code is used for the present study with constant mass flow rate at $R_p = 4200$ with $\mathcal{U}_b = 2/3$. The dimensions of the computational domain used are given in Table 6.1. These values are used in the ring-shaped rotating actuators configuration as shown in figure 6.2.

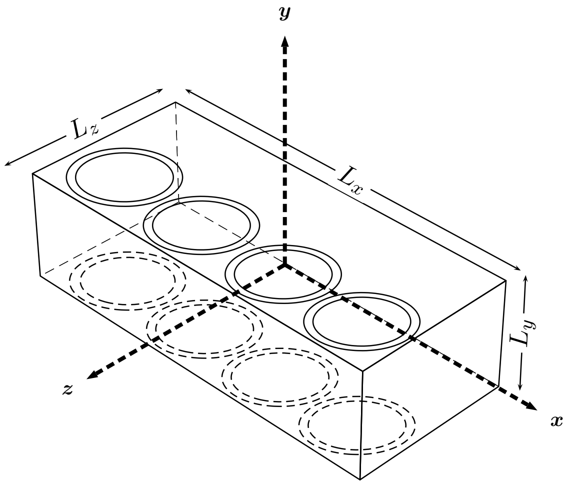


Figure 6.2: Channel flow configuration with rotating ring devices embedded at the walls. For clarity only half of the rings are represented on each wall in the spanwise direction.

The features of the three cases presently needed in this study are summarised in Table 6.1. These configurations are first used as rotating discs and show consistent results with Ricco & Hahn (2013) as shown in Table 6.2. The subsequent simulations with rotating actuators will be based on these three configurations. The statistics are computed by averaging instantaneous flow fields saved at intervals of $10\nu^*/u_{\tau,r}^{*2}$ for a total time window of $850h^*/U_p^*$ $(6520\nu^*/u_{\tau,r}^{*2})$.

As one wants to combine two control methods, it is necessary to examine the surface on which these are going to be applied. Indeed, the aim here is to use the rotating actuators as a primary control technique and apply on top a secondary control method either active of passive. Therefore, it is necessary to focus on the surface on which the secondary

| **Configuration** | $L_x$ | $L_y$ | $L_z$ | $D^{+0}$ | $W^{+0}$ | $n_x$ | $n_y$ | $n_z$ |
|---|---|---|---|---|---|---|---|---|
| Case RA1 | $4.53\pi$ | 2.0 | $2.26\pi$ | 640 | 9 | 256 | 129 | 256 |
| Case RA2 | $6.79\pi$ | 2.0 | $3.39\pi$ | 960 | 6 | 384 | 129 | 384 |
| Case RA3 | $6.79\pi$ | 2.0 | $3.39\pi$ | 960 | 9 | 384 | 129 | 384 |

Table 6.1: Parameters used in the DNS. The diameter $D$ and tip velocity of the discs $W$ are expressed in inner fixed-wall units.

| **Configuration** | $\mathcal{R}(\%)$ (Incompact3d) | $\mathcal{R}(\%)$ (Ricco & Hahn, 2013) |
|---|---|---|
| Case RA1 | 19.2 | 19.5 |
| Case RA2 | 19.3 | 19.2 |
| Case RA3 | 22.8 | 22.9 |

Table 6.2: Comparison of the drag-reduction rate $\mathcal{R}(\%)$ values for the steady rotating discs.

control methods can be enforced. In this way, the contribution from the surface available for applying the secondary control techniques are quantified.

For this purpose a compromise between the performance obtained with the primary control and the one brought either by the opposition control or hydrophobic surfaces is necessary to be achieved. Thus, using rotating rings (Wise & Ricco, 2014) appears to be a natural choice. The configuration presented in figure 6.2, while reducing the effective rotating area allows to gain surface for being used by the secondary control methods. Figure 6.4 shows the velocity norm for both configurations. The outermost regions of the discs are clearly spinning faster than those in the vicinity of the centre. As such, these are contributing the most in the drag-reducing capability of the rotating discs. To justify this statement, the configurations based on Cases RA1 to RA3 in Table 6.1 are used with rotating actuators based on the annular geometry and compared with rotating actuators consisting of full discs. By scanning over the ratio of the minor to the major radius, $a$, the optimal drag reduction rate that can be achieved occurs for $a = 0.6$. This is checked with the Incompact3d code for all box sizes, as shown in figure 6.3. These results are consistent with Wise & Ricco (2014). In Wise & Ricco (2014) only the configuration based on Case 1 with rotating rings was tested for the ratio $a = 0.6$. Configurations based on Cases 2

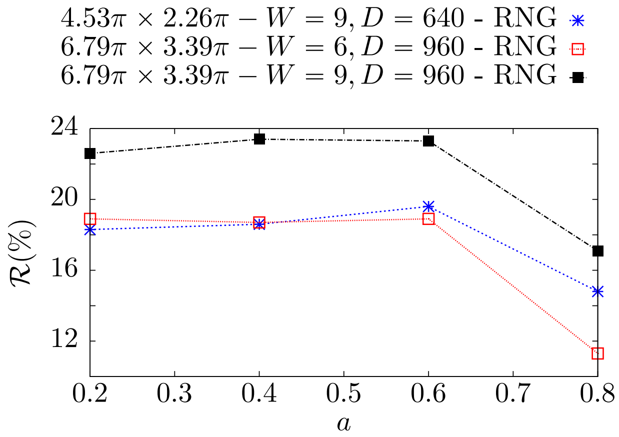


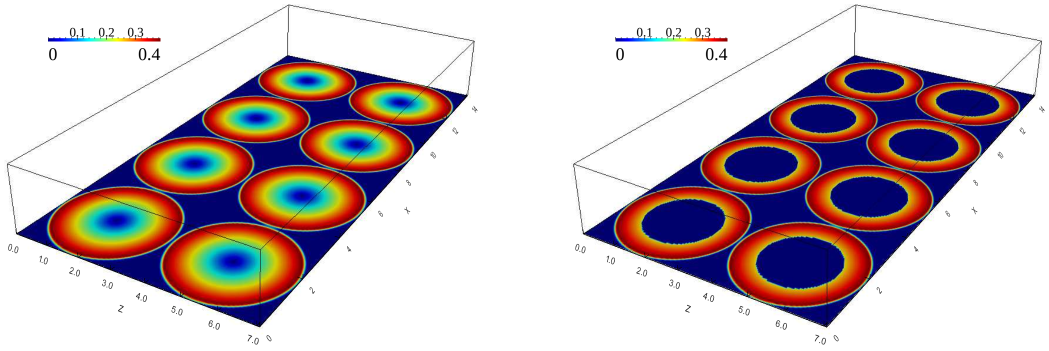
Figure 6.3: Scan over the aspect ratio for the primary control technique for all the box sizes under consideration in this study.

| **Configuration** | $\mathcal{R}(\%)$ (discs) | $\mathcal{R}(\%)$ (rings, $a = 0.6$) |
|---|---|---|
| Case RA1 | 19.2 | 19.6 |
| Case RA2 | 19.3 | 18.4 |
| Case RA3 | 22.8 | 23.3 |

Table 6.3: Comparison of the drag-reduction rate for the rotating actuators based on discs and rings respectively.

and 3 are presently computed and also reached an optimum $\mathcal{R}$ value for $a = 0.6$. The results are gathered in Table 6.3 and show that the rings and discs configuration differ by less than 1% for $\mathcal{R}(\%)$. This confirms that the drag-reducing effect is mostly induced by the outermost regions of the rotating actuators. Therefore, in the rest of this study, we will focus on the primary control based on ring-shaped actuators.

The first secondary control technique combined with the rotating actuators is based on opposition control. This method is used through wall-normal out-of-phase control. Hence $\forall t$, $v(x, y_d, z) = -v(x, 0, z)$ and $v(x, L_y - y_d, z) = -v(x, L_y, z)$, where $y_d$ is the detection (or virtual sensing) plane. The configuration used for devising the plane achieving the largest drag reduction value is given by ($L_x$,$L_y$,$L_z$)=($4\pi$,2.0,$4\pi/3$) and ($n_x$,$n_y$,$n_z$)=(256,129,256). As opposition control and the rotating rings actuators are combined through Cases RA1 to RA3 in Table 6.1, the first step is to consider these configurations where opposition



(a)

(b)

Figure 6.4: Representation of the discs (6.4a) and rings (6.4b) velocity norm $(\sqrt{U_{\text{disc}}^2 + W_{\text{disc}}^2})$ at the lower wall. For the rings configuration $a = 0.6$

| **Resolution** | $L_x$ | $L_y$ | $L_z$ | $n_x$ | $n_y$ | $n_z$ | $\mathcal{R}(\%)$ |
|---|---|---|---|---|---|---|---|
| Res0 | $4\pi$ | 2.0 | $4\pi/3$ | 256 | 129 | 256 | 24.6 |
| Res1 | $4.53\pi$ | 2.0 | $2.26\pi$ | 256 | 129 | 256 | 24.8 |
| Res2 | $6.79\pi$ | 2.0 | $3.39\pi$ | 384 | 129 | 384 | 24.7 |

Table 6.4: DNS computations with opposition control only for three different channel configuration and grid points. In terms of configuration and grid pints Res1 and Res2 are exactly equivalent to Case RA1 and Case RA2 in Table 6.1. $\mathcal{R}(\%)$ is computed by averaging over $t = 700h^*/U_p^*$.

control is solely applied. In terms of simulations only two resolutions are needed: the one from Case RA1 and Case RA2 as Case RA3 has the same number of grid points and channel dimensions as Case RA2. The results are presented in Table 6.4 and shown in figure 6.5. The wall-normal resolution and stretching are kept the same for all the simulations. Figures 6.6-6.7 show a snapshot of the wall-normal velocity at the wall and at the plane where the virtual sensors (presently at $y^{+0} = 15.05$) are located. One can notice that the structures are exactly the same apart from the opposite sign in their amplitude. This confirms the proper implementation of the opposition control technique.

In Choi *et al.* (1994), it is reported that the drag reduction rate is proportional to the area on which opposition control is applied. This assertion needs first to be confirmed in the Incompact3d code. Therefore, the configuration corresponding to Res0 in Table 6.4 is

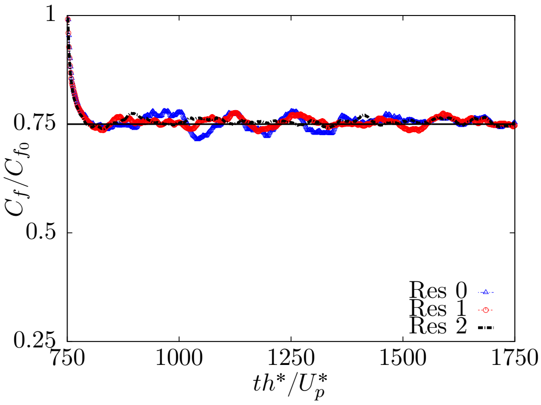


Figure 6.5: Normalised friction for the three cases in Table 6.1 using only opposition control.

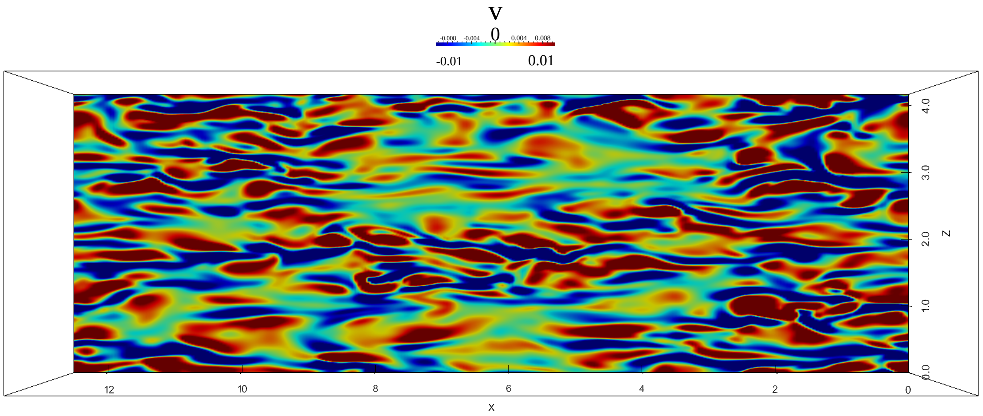


Figure 6.6: Wall normal velocity at $y^{+0} = 0$

chosen and only half of the channel flow is actuated by opposition control. This procedure results in $\mathcal{R} = 12.6\%$, consistent with halving the area under control.

The previous check is important since the secondary control techniques are applied on areas where the primary control technique is not enforced. As such, one needs to quantify how much surface is under control for the secondary control techniques when these are combined with the primary control. This surface is given by the white area in figure 6.8.

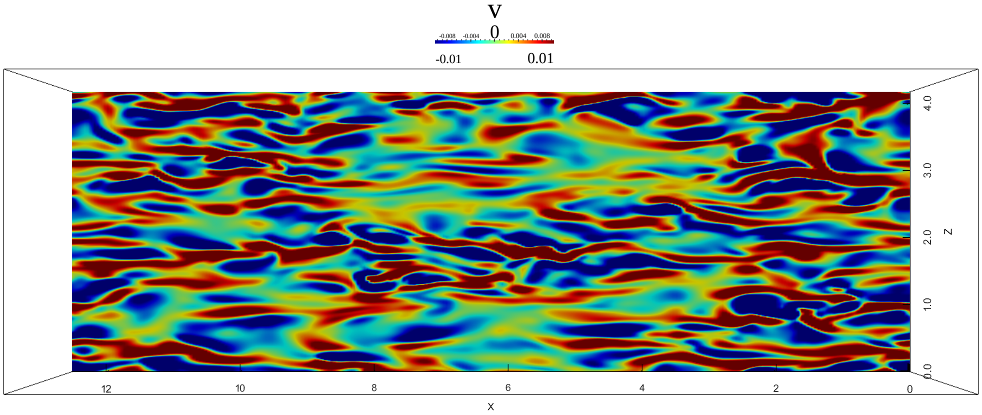


Figure 6.7: Wall normal velocity at $y^{+0} = 15.05$

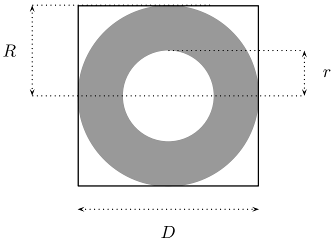


Figure 6.8: A single ring shown in a square frame. This pattern constitutes a single unit which is replicated on the full channel flow.

The white surface in figure 6.8 on which the secondary control is applied is computed as

$$\mathscr{A}_{\text{control}} = D^2\left[1 + \frac{\pi}{4}(a^2 - 1)\right]. \tag{6.1}$$

When the rotating actuators are discs ($a = 0$), the controlled area lies between the outer part of the discs and the edges of the square in which the ring is inscribed.

## 6.4 Flow analysis

In order to assess the performance for the combination of the primary and secondary control techniques, one should compare each type of control separately. The secondary control techniques are first computed with full and half of the wall surface covered. In the following sections the three cases given in Table 6.1 are focused on and analysed.

The dimensions and number of grid points for Case RA1 are considered (refer to Table 6.1). Therefore two series of simulations need to be performed.

The first one considers the rings+opposition control (RNG+OC henceforth) while the second is based on rings+hydrophobic surfaces (RNG+HDR henceforth). In both cases, the control techniques are compared independently and as a combination. For the combination of rings+opposition control, the value for the aspect ratio is taken to be $a = 0.6$ since with this value optimal $\mathcal{R}(\%)$ is achieved as previously shown. When rotating actuators are either combined with opposition control or hydrophobic surfaces, the surface on which these secondary control techniques apply is also about 50%. This value is computed by setting $a = 0.6$ into equation (6.1). This is the reason why the secondary control techniques have also been simulated separately only with 50% of the surface covering the channel walls. The results comparing the values for $\mathcal{R}(\%)$ are gathered in Table 6.5. The tables for cases RA2 and RA3 are also shown. However, all the following analysis will focus on case RA1 as the two other cases are similar to the former.

| Control | RNG | OC (100%) | OC (50%) | RNG+OC(50%) |
|---|---|---|---|---|
| $\mathcal{R}(\%)$ | 19.6 | 24.8 | 12.5 | 27.3 |
| Control | RNG | HDR (100%) | HDR (50%) | RNG+HDR(50%) |
| $\mathcal{R}(\%)$ | 19.6 | 28.7 | 15.1 | 27.1 |

Table 6.5: Drag reduction values for the control techniques taken either separately or as a combination (Case RA1). The following flags are chosen for the different techniques: RNG (rotating rings), OC (opposition control), HDR (hydrophobic). The value in parenthesis indicates the percentage area covered by the secondary control method. Case RA1

| Control | RNG | OC (100%) | OC (50%) | RNG+OC(50%) |
|---|---|---|---|---|
| $\mathcal{R}(\%)$ | 17.6 | 24.7 | 12.7 | 25.6 |
| Control | RNG | HDR (100%) | HDR (50%) | RNG+HDR(50%) |
| $\mathcal{R}(\%)$ | 17.6 | 28.7 | 15.8 | 28.5 |

Table 6.6: Case RA2

| Control | RNG | OC (100%) | OC (50%) | RNG+OC(50%) |
|---|---|---|---|---|
| $\mathcal{R}$(%) | 23.5 | 24.7 | 12.7 | 30.5 |
| Control | RNG | HDR (100%) | HDR (50%) | RNG+HDR(50%) |
| $\mathcal{R}$(%) | 23.5 | 28.7 | 15.8 | 31.1 |

Table 6.7: Case RA3

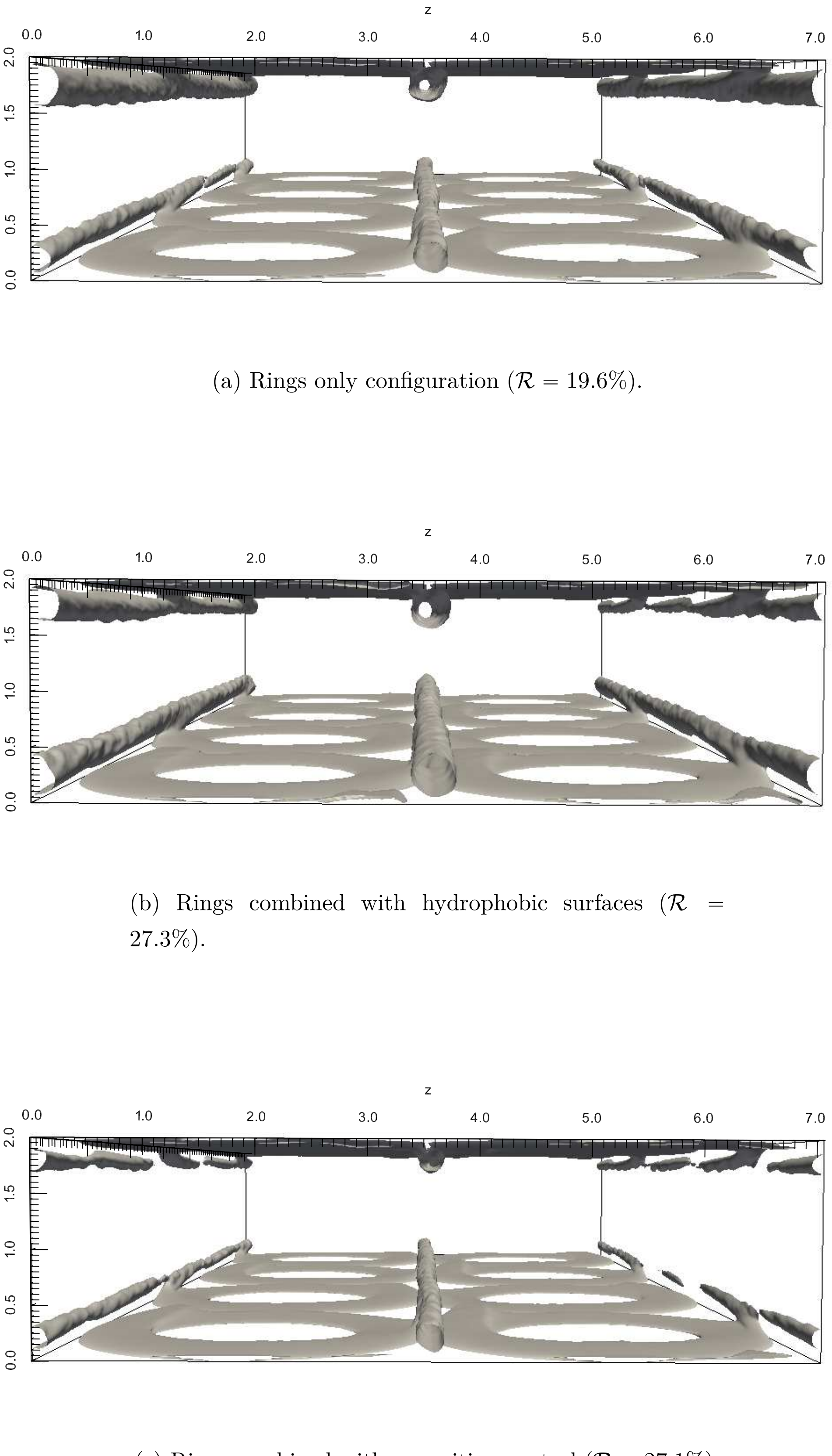
The first naive conclusion would consider that the drag-reduction rate resulting from the combination RNG+OC(50%) or RNG+HDR(50%) would be the sum of the contribution from each control technique separately. In this case one would have obtained respectively 32.1% for RNG+OC(50%) and 34.7% for RNG+HDR(50%). Here it is clearly less than these expected values. Table 6.5 shows that, although the combination of the primary and secondary control techniques yields higher drag-reduction rates, these values are not the sum of the contributions taken independently. The non-additivity in the drag-reduction rate of the control techniques equally applies to both combinations, i.e. active-passive and active-active. The same conclusions can be drawn for cases RA2 and RA3 as shown in Tables 6.6-6.7. The flow is further analysed using visualizations based on various quantities.

### 6.4.1 Disc flow analysis

The objective is explain why drag reduction rates are not additive. The 2D/3D flows are analysed by focusing on the disc flow. First the 3D representation of the norm of the disc flow is computed. For this purpose the flow is split into its three contributing parts, as described in 4.5 and briefly reminded hereafter:

$$\boldsymbol{U}(x,y,z,t) = \boldsymbol{U}_m + \boldsymbol{U}_d + \boldsymbol{U}_t. \tag{6.2}$$

with the mean flow defined as $\boldsymbol{U}_m(y) = U_m, 0, 0 = \left[\overline{\boldsymbol{U}}\right]_{\mathcal{I}xz}$, the disc flow $\boldsymbol{U}_d(x,y,z) = (U_d, V_d, W_d) = \overline{\boldsymbol{U}} - \boldsymbol{U}_m$ and $\boldsymbol{U}_t$ the turbulent fluctuations. The averaging operators definitions are those given through (2.40) and (2.11). Figure 6.9 gives the isosurface representation for $\sqrt{U_d^2 + W_d^2}$ in the three configurations. The latter quantity is computed in order to be consistent with the conventions taken in Ricco & Hahn (2013). First, the same type of interdiscs streamwise-elongated patterns as those represented in Ricco & Hahn (2013) are obtained at the centre of the channel as well at the edges forming



(a) Rings only configuration ($\mathcal{R} = 19.6\%$).

(b) Rings combined with hydrophobic surfaces ($\mathcal{R} = 27.3\%$).

(c) Rings combined with opposition control ($\mathcal{R} = 27.1\%$).

Figure 6.9: Three-dimensional representation of the disc flow amplitude with an isosurface level set at 0.08 (expressed in outer units). This level is kept the same for the three configurations.

another tube by periodicity along the spanwise direction. This observation confirms that it is actually the outermost spinning regions of the discs that contribute the most to the flow, thereby sustaining these peculiar structures even when rings, instead of discs, are used as actuators. Therefore, in the following, the term disc flow will actually define the flow contribution from the rings.

The quantity $\sqrt{U_d^2 + V_d^2 + W_d^2}$ was computed (not shown here) as well but no notable difference was observed in the isosurface representation for the three configurations. This confirms that the disc flow in the wall-normal direction is very weak and only the components in both the streamwise and spanwise directions are the main contributors to the disc flow. A first qualitative observation is that the interdiscs streamwise-elongated patterns for the combined cases, i.e. rings and secondary control methods, are modified compared to the case where only the primary control technique is solely applied. When the primary control is combined with hydrophobic surfaces, these particular patterns become slightly wider. On the other hand, when the primary control is combined with opposition control, these patterns are weakened.

To further investigate these structures, slices in $y-z$ planes (cross-flow) and $x-z$ for various components of the disc flow are computed for the three configurations and compared. Note that in the following analysis, the amplitude level on the colorbar is set to the same level for all the figures for a chosen component.

In figure 6.10, the cross-flow is represented for the streamwise component of the disc flow, $U_d$, at different $x$ positions. For the three control methods (primary alone and combined with secondary methods), the contour level is set at 50. For different $x$ positions, one can clearly notice that the periodicity of the structures is enforced for all the control techniques. One can also observe that the three control methods exhibit a common pattern for the structure of $U_d$. This particular feature for all these configurations is located at the centre of the channel forming a cell-like structure spanning from the top to the bottom of the channel.

The second point to notice concerns the variation of these structures relative to their location. By observing figure 6.9, one would a priori think that these tubular patterns are relatively unmodified along the streamwise direction. However, looking at the planes located at $x = 0$ and $x = 3.516$ (corresponding to a fourth of the channel in the streamwise direction), the cell-like structures exhibit a local steepening in the gradient, particularly

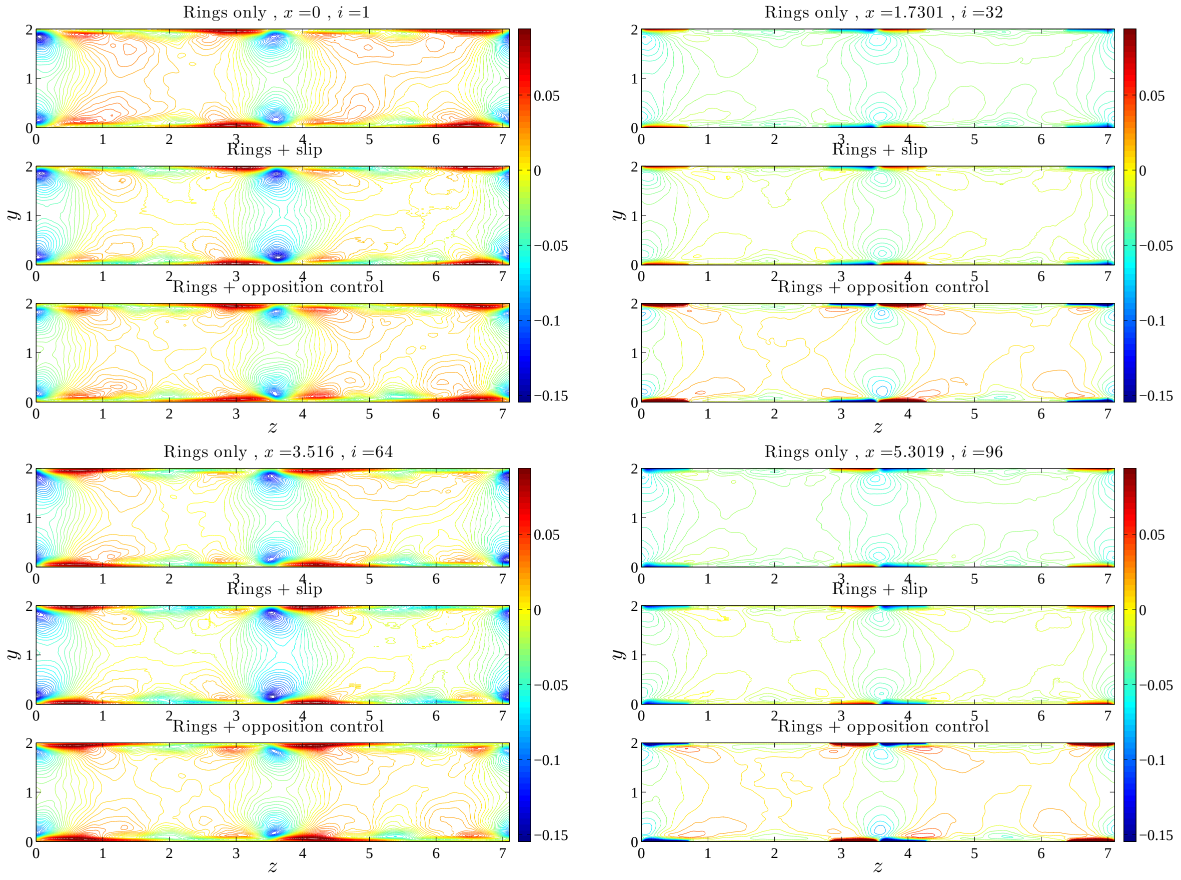


Figure 6.10: Cross-flow representation of the streamwise component of the disc flow, $U_d$, for the three control methods at different $x$ positions.

noticeable for the combination of the primary control with hydrophobic surfaces. This behaviour contrasts with the same type of structures for planes located at the centre of each disc, e.g. either at $x = 1.7301$ or $x = 5.3019$.

In order to compare with figure 6.9, the contours of the amplitude $\sqrt{U_d^2 + W_d^2}$ for the disc flow are computed in the same way as for the streamwise case. The results are shown in figure 6.11. From these figures, one can observe the clear strengthening of the streamwise-elongated central structures for the combination of rings and hydrophobic surfaces. This is true for $x = 0$, $x = 3.516$ and the following starting indices of a new rotating actuator. In addition the centre of the rings show an enhanced flow both for the rings only case and rings combined with opposition control. This has to be opposed to the rings and hydrophobic surfaces combination for which at the same locations the disc flow is weakened. Therefore, the finite slip length surface at these locations are effectively affecting the disc flow.

Another point concerns the correlation between the amplitude on top of discs' locations and the amplitude of the streamwise-elongated structures. Indeed, when the amplitude on the top of the discs' locations is weakening, the amplitude of the tubular patterns becomes stronger and vice e versa. This can be observed for all $x$ plane locations and for all three configurations.

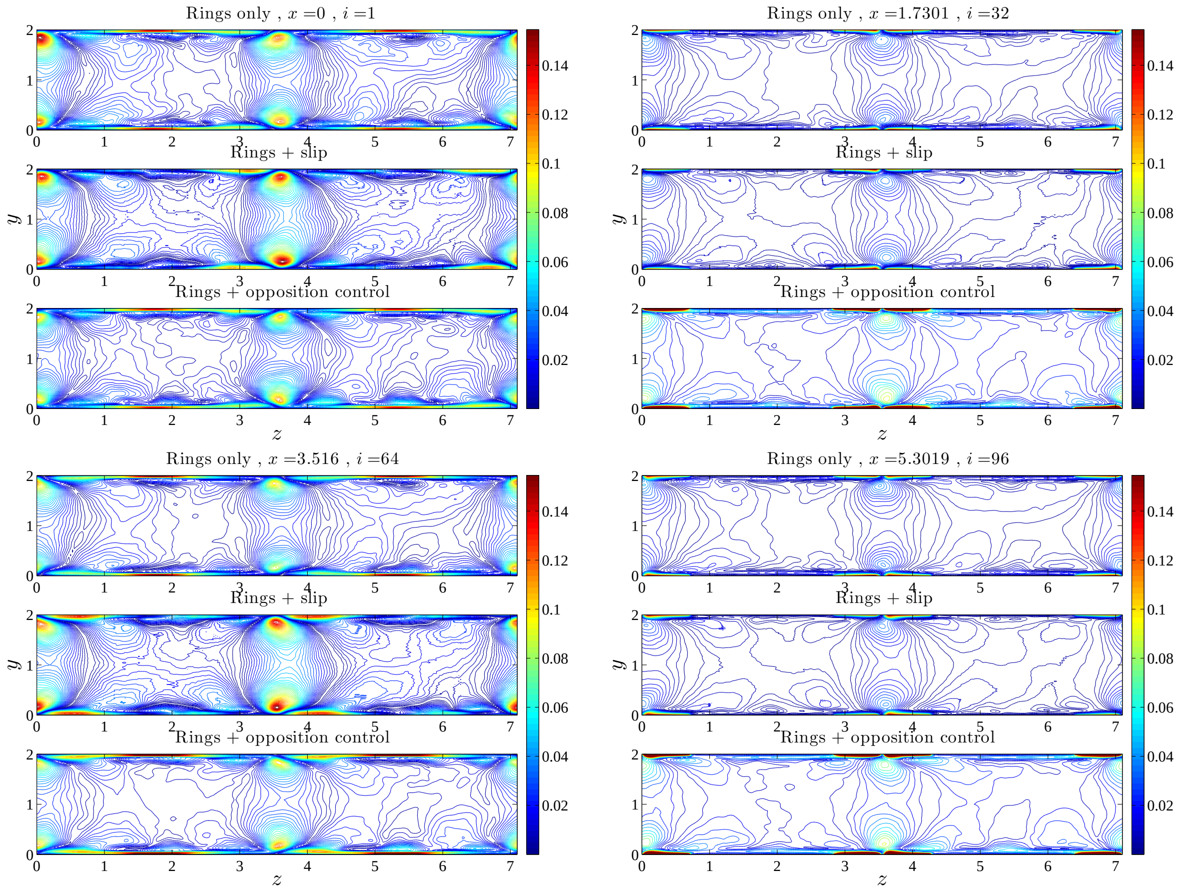


Figure 6.11: Cross-flow representation of the streamwise component of the disc flow, $\sqrt{U_d^2 + W_d^2}$, for the three control methods at different $x$ positions.

The three-dimensional representation of $U_dV_d$ is shown in figure 6.12 for the three configurations. For these, the same isosurface level of the rings only configuration is chosen as a reference. The first remark concerns the fact that for all cases, this representation still exhibits the peculiar patterns already observed in $U_d$ and for the norm. This entails that the contribution from the wall-normal disc flow is very weak as is also seen from the amplitude of $U_dV_d$ compared to $U_d$.

The second point concerns the modification of sheared structures developing very near the wall and located beneath the streamwise elongated structures. For rings only and rings combined with opposition control two alternative sheared bipolar cells develop on each side of the streamwise elongated central structure. The aforementioned sheared structures overall keep the same patterns for these two cases. However, the weakening of the streamwise-elongated structures is also exhibited in the combined case with opposition control. On the other hand for the rings combined with hydrophobic surfaces, the bipolar cells closer to the elongated structure and those towards the centre of the rings are parting while increasing in size. This change of structure impact the size of the streamwise elongated structures as these become wider.

From these three cases, one can draw some conclusions. First the tubular patterns are not specific to the norm of the discs' flow, as the Reynolds-shear stress of the disc flow also exhibit such patterns. Secondly the local changes in the patterns of $U_dV_d$ in the vicinity of the rotating actuators plays a role in the formation of the tubular structures. Finally, depending on the chosen control, the structures are modified in a different way, implying that it is not necessarily the change in $U_dV_d$ that can explain the change in the skin-friction coefficient. The cross flow for $U_dV_d$ is computed as well following the same methodology as for $U_d$. The first point to notice is the opposite sign of the two tubular cells at the top and bottom walls, following the discs' orientation. In all three configuration in the plane located at the centre of the discs, a secondary smaller structure appears marking the location of the junction of the larger tubular structure with the rotating actuators.

Additional visualizations are presented hereafter by computing the streamwise and spanwise discs flow components for the vorticity. These quantities are compared for the rings only, rings combined with hydrophobic surfaces, and rings combined with opposition control. Concerning the discs' flow streamwise vorticity component, it can be noticed in figures 6.14 that the amplitude of the vortical structures forming near the edges and at the centre of the flow are weakening both for the combination of hydrophobic surfaces with rings and opposition control with rings. For the latter, the change in amplitude in the aforementioned regions is even more acute. This means that although the disc flow meant to reduce the drag is weakening, there exist a rebalance coming from the combination with the secondary control methods. With respect to the discs' flow spanwise vorticity (refer to figure 6.15) component, one can notice the very peculiar structures in X shapes which are enhanced in the case of the combined control methods. It appears that the combined methods are playing the role of a shield with respect to the vortical structures in the regions located on top of the discs.

### 6.4.2 Quadrant analysis

The flow is then analysed quantitatively by computing the contributions of each quadrant for the Reynolds-shear stresses. Indeed, in Ricco & Hahn (2013); Wise & Ricco (2014), it is mentioned that the contribution from the streamwise-elongated tubular patterns can be considered as jets playing a beneficial role on the drag-reduction mechanism. These are intimately related to the Reynolds-shear stresses from the rotating actuators' flow denoted henceforth by $[U_dV_d]_{\mathcal{I}xz}$. The latter are oriented backwards and tend to oppose the flow which in turn induces the drag-reducing effect made explicit through the FIK identity. Indeed their contribution directly enters in the FIK formulation (Fukagata *et al.*, 2002) for the computation of the skin-friction coefficient. As there is no time modulation/oscillation of the discs' tip velocity, $[U_dV_d]_{\mathcal{I}xz}$ is a time-independent quantity as is shown in the following:

$$U_dV_d = (\overline{U} - [\overline{U}]_{\mathcal{I}xz})(\overline{V} - [\overline{V}]_{\mathcal{I}xz}) = (\overline{U - [U]_{\mathcal{I}xz}})(\overline{V - [V]_{\mathcal{I}xz}}), \tag{6.3}$$

Other necessary expressions for the computation of the diagnostics are detailed as follows:

$$[\overline{UV}]_{\mathcal{I}xz} = U_mV_m + [U_dV_d]_{\mathcal{I}xz} + [\overline{U_tV_t}]_{\mathcal{I}xz}\,, \tag{6.4}$$

From (6.3), it follows that:

$$[U_dV_d]_{\mathcal{I}xz} = [\overline{U}\ \overline{V}]_{\mathcal{I}xz} - U_mV_m, \tag{6.5}$$

Replacing (6.5) into (6.4), one gets:

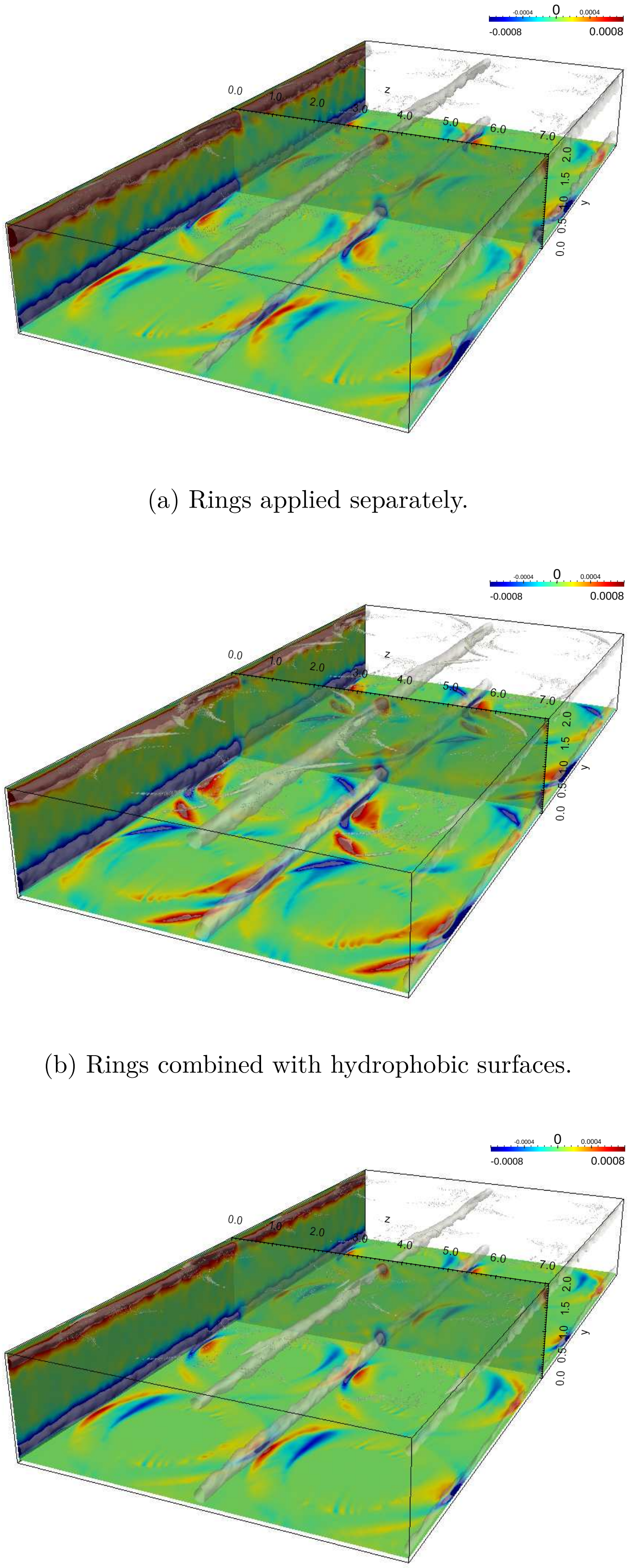
$$[\overline{U_tV_t}]_{\mathcal{I}xz} = [\overline{UV}]_{\mathcal{I}xz} - [\overline{U}\ \overline{V}]_{\mathcal{I}xz}\,. \tag{6.6}$$

The computation for the different quadrant contributions for the discs and the turbulent flow allows to quantify their relative importance in the FIK identity. The delimitation of each region was shown in 2.2. By taking into account these regions in the computation of the disc and turbulent flow as defined in (6.2), the Reynolds-shear stress specific to these flows are obtained. Further, these are integrated following (2.76) in §5.3.7 to compute the aforementioned contributions and shown in figures 6.16-6.17.

In the particular case of the combination of rings and hydrophobic surfaces, the term contributing from the mean flow is added to be relatively compared with the other quadrant contributions. From figure 6.16, one can notice that quadrants Q1 and Q3 contributions are increased for the combination of rings and hydrophobic surfaces compared to the case where rings are applied separately. The fact that Q3 contributes more is a sign of the strengthening of the tubular structures, thereby opposing the flow and inducing the drag-reducing effect. Although this gives a beneficial effect, in the meantime, the sweep/ejection events as quantified by quadrants Q2 and Q4 are also being increased (in absolute value) compared to the rings only case.

Comparing the quadrants between the rings only case and the combination of rings with opposition control shows that the contributions the sweep/ejection events relative to Q2 and Q4 are being decreased. On the other hand Q3 contribution is clearly reduced. This is consistent with the weakening of the streamwise-elongated structures as observed earlier. The drag-reduction mechanism based on the disc flow can be seen as a balance between the modification of the sweep/ejection and Q3 events. The mean flow term in the FIK identity the case with hydrophobic surfaces comes as an additional contribution, which is offsetting the increase in Q3 events (compared to the Q3 events of the opposition control case), thereby explaining the close skin-friction values for these two combined cases.

In figure 6.17, the sweep/ejection events accounted through Q2 and Q4 are being decreased for both combined cases compared to the rings only configuration. The important point to note is that while these contributions are decreasing for both combined cases, the levels are almost the same, consistent with their close drag-reduction rates.



(a) Rings applied separately.

(b) Rings combined with hydrophobic surfaces.

(c) Rings combined with opposition control.

Figure 6.12: Three dimensional representation of the Reynolds-shear stress based on the disc flow. The level of the isosurface is kept the same for all the configurations at 0.0075.

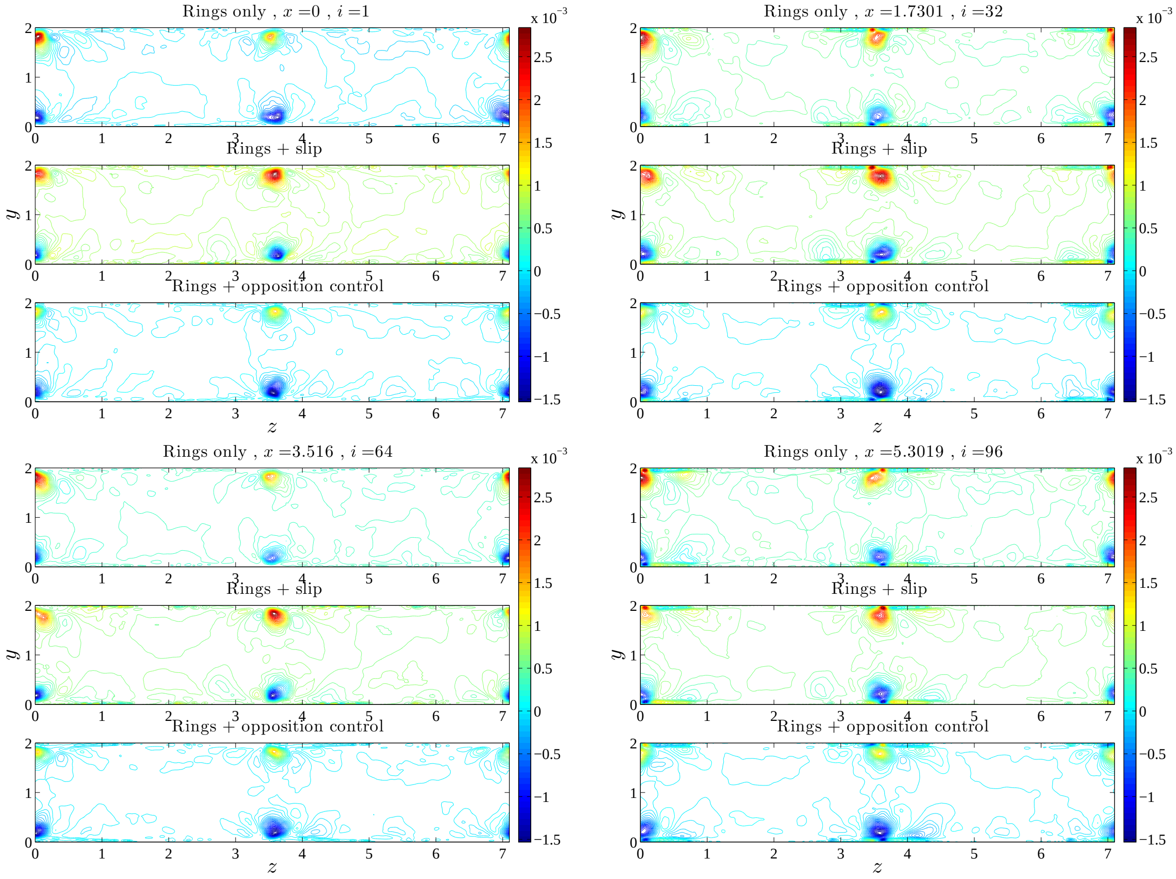
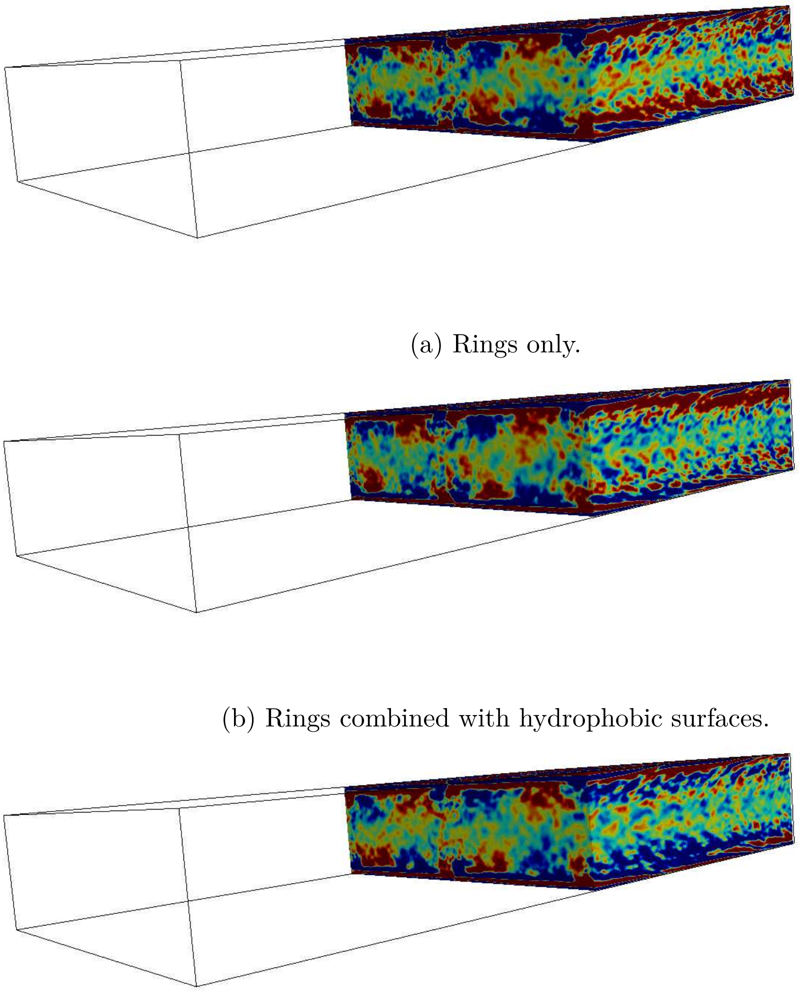


Figure 6.13: Cross-flow representation of the Reynolds-shear stress of the disc flow, $U_dV_d$, for the three control methods at different $x$ positions.

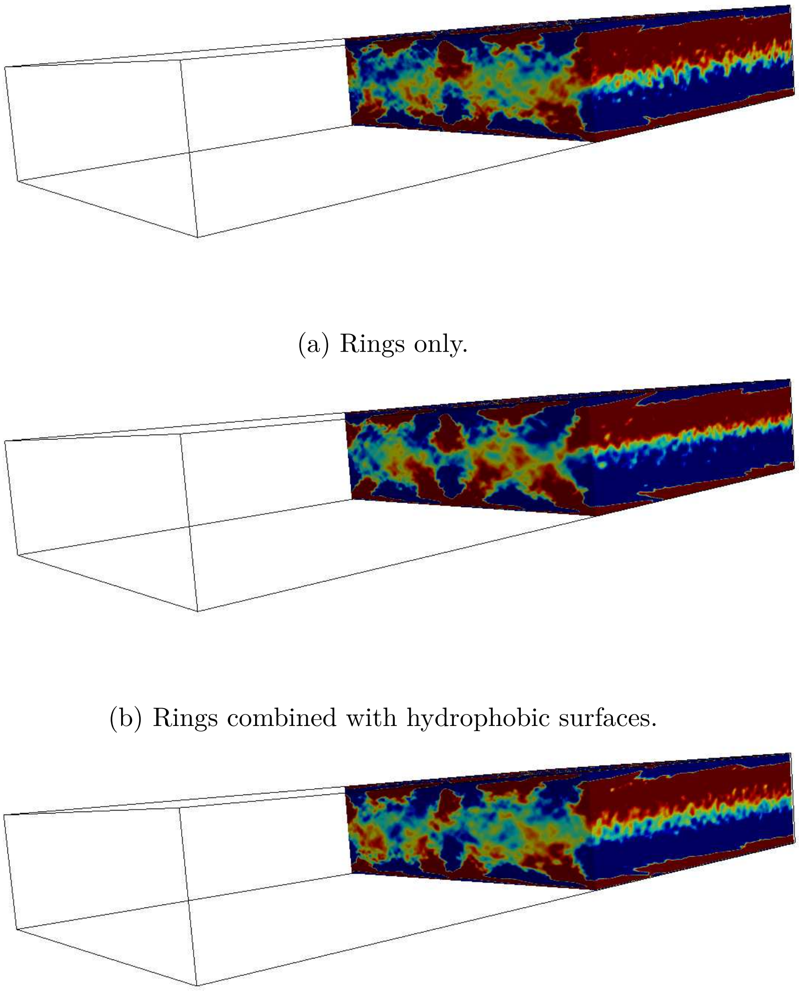


(a) Rings only.

(b) Rings combined with hydrophobic surfaces.

(c) Rings combined with opposition control.

Figure 6.14: Representation of the discs' streamwise component of the vorticity for the three control methods. The color scale lies within the interval [-0.05,0.05] (in Poiseuille units) for all three cases.



(a) Rings only.

(b) Rings combined with hydrophobic surfaces.

(c) Rings combined with opposition control.

Figure 6.15: Representation of the discs' spanwise component of the vorticity for the three control methods. The color scale lies within the interval [-0.05,0.05] (in Poiseuille units) for all three cases.

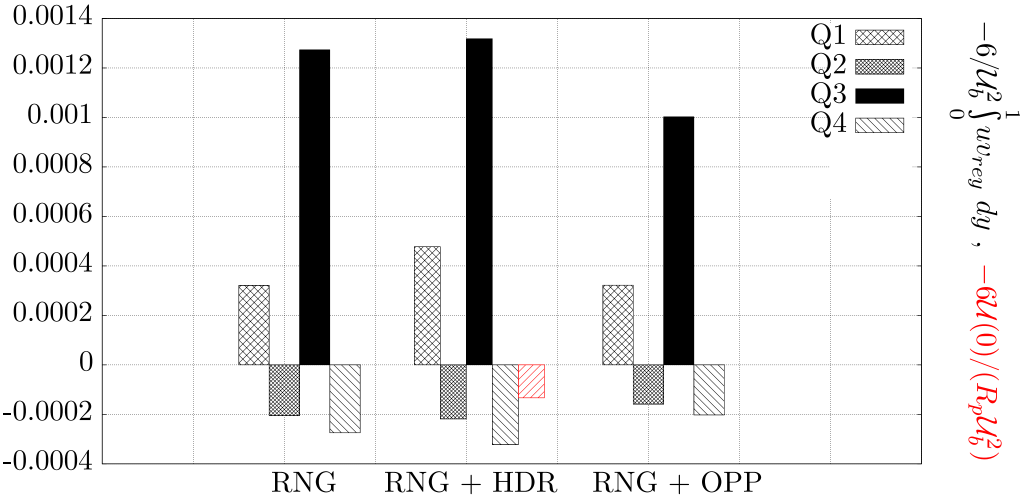


Figure 6.16: Contribution from the Reynolds shear-stress based for the disc flow quantified for each quadrant.

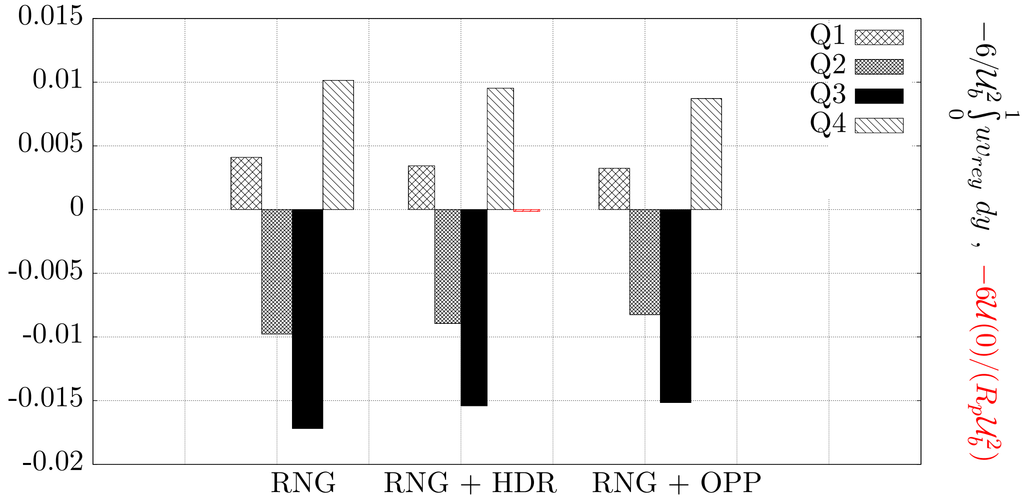


Figure 6.17: Contribution from the Reynolds shear-stress based for the turbulent flow quantified for each quadrant.

## 6.5 Statistics

This analysis focuses on a given number of statistical quantities. In addition to standard rms quantities, the Reynolds shear-stresses are computed in three specific regions. Indeed, at the wall, there are different regions each on which a different type of control applies. For this reason three different regions are devised as shown in figure 6.18.

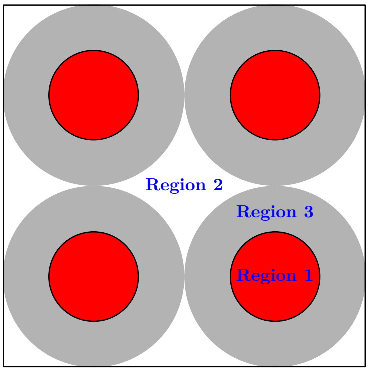


Figure 6.18: Definition of the regions selected for the computation of the statistics. Only a section of the full channel wall is shown. Region 1,2 and 3 respectively comprise the red, white and light grey areas in their entirety.

In figure 6.19a, one can notice the decrease and shift in the minimum for the rms of the streamwise component around $y^{+0} \simeq 10$ using the combined control methods. The second point concerns the boundary layer thickness for the disc flow. Indeed the amplitude of the slope for the tangent to the origin increases. Its intersection with the abscissa stands for the aforementioned boundary-layer. The exact definition of the boundary-layer is based on slope of both $u_{rms}$ and $w_{rms}$. However, the main contribution comes from the streamwise component as the one from the spanwise component shown in figure 6.19c are almost the same.

The rms of the streamwise velocity for the disc flow and turbulent flow show two specific behaviour for the combined controlled cases. The combination with hydrophobic surfaces show that a shift at the origin occurs for the disc flow. This implies that the hydrophobic surface is effectively impacting the disc flow. This is not unexpected as the boundary condition specified is only based on the streamwise direction as a finite slip length is used in this direction. Therefore the combined effect of the mixed boundary condition is clearly seen and one does not prevail or annihilate the other. A similar behaviour occurs for the combination with opposition control where a decrease and shift is incurred by blowing/suction. This becomes clearer when analysing the Reynolds shear-stresses. An alternating maximum/minimum pattern is exhibited near the wall which accounts for the displaced virtual origin encountered in blowing/suction control as presented in Choi *et al.* (1994).

In figure 6.19f the contribution for the disc flow in the combined case with hydrophobic surfaces dominates over the two other configurations from $y^{+0} \simeq 10$. This is consistent with the previous computation of the quadrant as the Q3 event incurred a large increase. On the other hand, the turbulent contributions follows an ordering that is clearly separate between the rings only case and those that are combined.

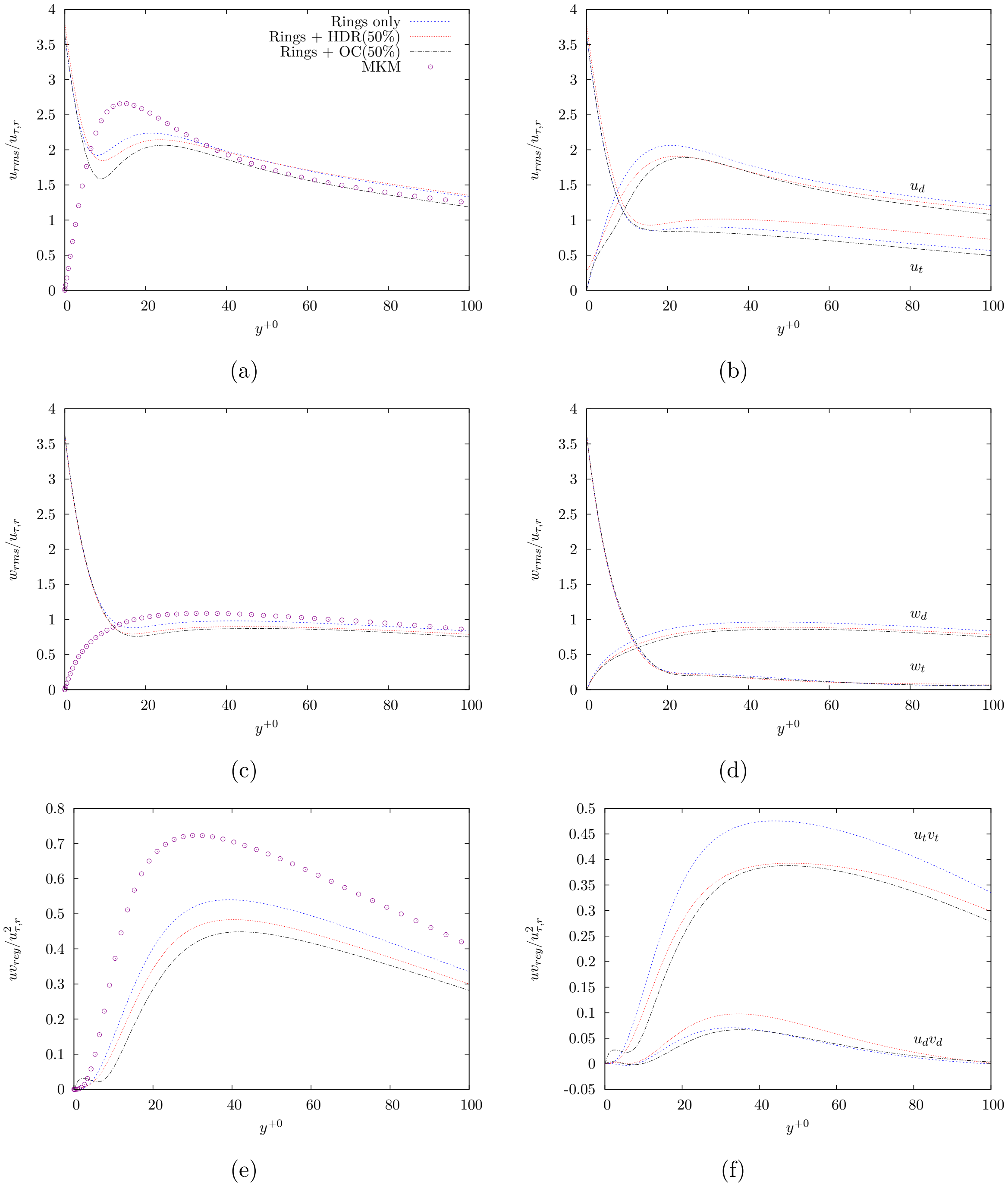
As previously mentioned, the statistics are also computed in three separate regions. In figures 6.20a-6.20b and 6.20c, for each configuration, the three regions are represented, while figures 6.20d-6.20e and 6.20f show for each region the three configurations.

The essential points to note are described hereafter. For region (1), the Reynolds shear-stress for the combined rings and hydrophobic surfaces configuration dominates over the rings only configuration, the latter being larger than the combined case with opposition control (see 6.20d). On the other hand in region (3), the two other configurations dominate over the combined rings and hydrophobic surfaces configuration. This computation allows to precise the role of each region that contribute to the Reynolds shear-stress and thereby to the drag-reduction mechanism. The second point concerns the behaviour of the combined case with opposition control. In region (1), it has the smallest contribution compared to the other two configurations. The behaviour particular to blowing/suction with an alternative maximum/minimum near the wall, only appears in region (2) although the start of this pattern could be guessed in region (1). This behaviour is not seen at all for region (3). Therefore, the particular behaviour described before for opposition control is only specific to region (2). It therefore appears that although blowing/suction is also applied in region (1), the rotating actuators seem to shield the effect expected from this secondary control method.

When comparing figures 6.20a-6.20b and 6.20c, it is interesting to note that it is the combination of rings and hydrophobic surfaces that makes the Region 1 contribution to dominate over the two others up to $y^{+0} \approx 20$. In the two other configurations, i.e. rings only and rings combined with opposition control, Region 1 contribution always keeps being lower and upper bounded by the contributions from Regions 2 and 3 with the interval $y^{+0} \in [0; 20]$. This therefore suggests that the dynamics involved on the rings (i.e. Region 3) and on Region 2 seem to have a shielding effect which is though overcome in the case of the combination of rings and hydrophobic surfaces. By comparing figures 6.20b and 6.20c, one can notice that the contributions from all the regions are very similar and in terms of amplitude modification, there is a trade-off among the three regions' contributions, thereby explaining the close values of the primary control method respectively with hydrophobic surfaces and opposition control.

The other noticeable point concerns figures 6.20d, 6.20e and 6.20f. Indeed, the idea behind the separation of the diagnostics being computed per region, was among other to decipher the impact of each type of control on the different statistics and here more particularly the Reynolds-shear stresses. The latter being related to the skin-friction drag-reduction coefficient through the FIK identity. One can notice in the aforementioned figures discriminated by regions: in figure 6.20d, in Region 1, it is the hydrophobic surfaces contribution that dominates the other two control techniques. Then in 6.20e, there is a competition on the dominating control method behaviour. Note that all the control methods are not applied at the same time, therefore when mentioning competition between these, it should be understood as a competition in a separate way. In fact, very near the wall, the strong behaviour well-known from the opposition control technique is first dominating. However, in terms of amplitude, the effect of the hydrophobic surface which applies on Region 2 becomes dominant up to $y^{+0} \approx 20$, where it peaks. After this, the contributions from the rings taken separately and the rings combined with opposition control dominate the hydrophobic one, remaining close to each other. When opposition control combined with rings is used, in Region 2, there is exclusively enforcement of opposition control only. Therefore, it would have been natural to think that the effect provided by the opposition control technique would have dominated over the rings. But this is not the case and can explain why the drag reduction of the two techniques taken separately are not adding up when these are combined. In addition the amplitudes for the opposition control technique remains very close to the rings only case. Finally in figure 6.20f, where the rings are rotating, one can clearly see that the two other combined methods are being dominated by the rings only control. However, the impact on the hydrophobic surfaces is more pronounced than the combined opposition control and rings.

When the Reynolds-shear stresses are computed on regions separately, one can clearly see the domain of influence of each control technique. In fact, although the flow comprises the three components, each region behaves as a mask over it, allowing to exhibit the dynamics of each control method. It appears that the two regions that are exhibiting the effects of the control methods are Region 1 and Region 3. Indeed, in Region 1, when applying the different control methods, the strong response from the combination of hydrophobic surfaces combined with rings. It clearly shows that the rings on its own cannot mitigate the effect of the hydrophobic surfaces. However, the rings-only flow represents a non negligible part in the Reynolds-shear stresses' contributions. This is another sign of the impossibility of additivity of the contribution of the two methods taken separately.



(a)

(b)

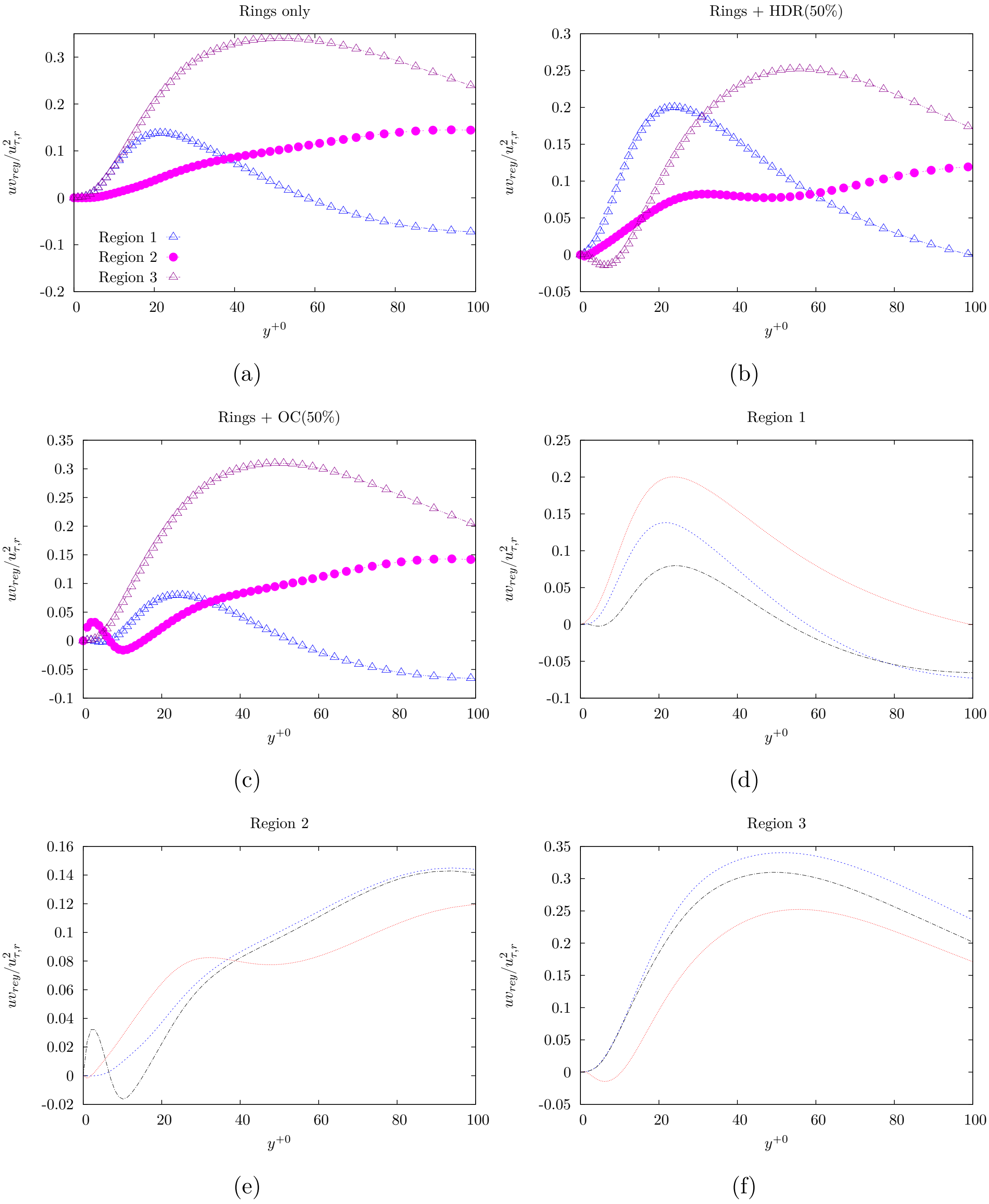
(c)

(d)

(e)

(f)

Figure 6.19: Streamwise (labels (a) and (c)), spanwise (labels (b) and (d)) rms and Reynolds stresses values. Both streamwise and spanwise rms are also computed by using the decomposition of the flow through $\boldsymbol{U} = \boldsymbol{U}_\mathrm{m} + \boldsymbol{U}_\mathrm{d} + \boldsymbol{U}_\mathrm{t}$ as given by (6.2). MKM refers to the uncontrolled flow (Moser *et al.*, 1999).



(a)

(b)

(c)

(d)

(e)

(f)

Figure 6.20: Reynolds stresses are shown for a given control technique on different regions (a)-(b)-(c) and for a given regions the three control methods are compared to each other (d)-(e)-(f). For (d)-(e)-(f) the legend follows the previous page conventions.

## 6.6 Conclusion

In this study, the combination of active and passive control methods are focused on. The underlying idea is based on practical considerations. Indeed, well-known control methods have not been so far studied in a complementary way. This was the occasion for testing the drag-reduction method proposed by Ricco & Hahn (2013), use it in a modified setup (rotating rings), allowing for the latter to be complemented with secondary control methods. Although higher drag-reduction rates were achieved, the full capacity of the control methods taken independently cannot be taken advantage of. This was made explicit by the analysis of the flow either in terms of the flow structures and the terms entering the FIK identity.

The idea of the statistics being computed in separate regions allows to devise the dynamics and the dominant contribution in each of these regions. This allows to obtain a qualitative view on competing dynamics in each region and explain empirically why the combination of controlled method is not the result of each control technique taken separately. This study still present potential for further development. However, it sets the ground for future studies that are able to combine control methods as presented here in the form of a primary and secondary control methods which have not been combined in this manner.

# Chapter 7

# Summary and future work

The main objective of this project was the study flow control techniques using Direct Numerical Simulations. For this purpose, all the major well-known flow control techniques were implemented into the high order compact finite-difference scheme Incompact3d code. The implementations were extensively benchmarked and showed consistent results proving the reliability of the implementation and the efficiency of the code.

After this step, we have focused on specific surfaces called hydrophobic surfaces. Inspired by recent experimental work (Choi & Kim, 2006), a linear shear-dependent slip length has been chosen. First, the Navier-Stokes equations were solved analytically for the laminar flows in the confined channel-flow geometry. The impact of the slip is exhibited through the decrease of pressure gradient under constant mass flow rate conditions and by the increase of mass flow rate under constant pressure gradient conditions. The laminar channel flow is also studied through nonlinear Lyapunov stability analysis. The two-dimensional approach is due Balogh *et al.* (2001) was extended to the three-dimensional case and the shear-dependent laminar solution is chosen as the base flow. The stability analysis is useful to devise rigorous nonlinear stability conditions. It was found that the feedback-control wall boundary conditions coincide with the hydrophobic slip-length model. This is the first time this conceptual link between the extracted feedback-law boundary conditions and the hydrophobic-surface model is exhibited. The theory by Fukagata *et al.* (2006) was further extended to the shear-dependent slip-length case, to obtain through DNS the relevant parameters affecting the drag reduction rate using these shear-dependent hydrophobic surfaces. The results from the DNS and the theoretical formulation proved to be consistent with the developed theory for the shear-dependent model. Finally the power exerted by the liquid turbulent flow on the hydrophobic surface is investigated and the principal strain rates of the near-wall turbulent flow are studied for the first time in a drag-reducing flow.

In a second part, the study of combination control was advanced. To the best of our knowledge, in the context of flow control, such a methodology has not yet been used. Indeed, when a type of flow control technique is chosen to be applied, it is generally distributed on the whole boundary leaving no possibility for other techniques to be simultaneously implemented. In the context of DNS within a channel flow configuration, such a combination focused on rotating actuators (primary control) and two other techniques (secondary control), namely constant-slip hydrophobic surfaces and opposition control.

Finally an extensive work has been carried out on the code itself to make it a standard tool for the generation of reliable flow control results. From an implementational viewpoint, through the course of this project, the Incompact3d code was transformed in two ways. It has now the capability of tackling various flow control problems: spanwise wall oscillations, streamwise/spanwise constant hydrophobic surfaces, streamwise/spanwise shear-dependent hydrophobic surfaces, steady and oscillating rotating discs, steady and oscillating rotating rings, relevant combination of the previous techniques. In a second step modularity, repeatability and ease in the use of the code have been the target. For these to become effective, a new general structure was applied to the code, handling all the necessary steps from the installation to the launch of the simulation in an automatic and as comprehensive as possible. Finally a self-contained guide for this new version including the flow control techniques has been written with the view of becoming a base for future developments contributing to the online guide by Dr Sylvain Laizet.

# Appendix A

# Incompact3d **reference: reorganised version for flow control problems**

This reference guide intends to give an exhaustive description of the CFD code Incompact3d. The code is based on high-order compact finite difference schemes, uses various explicit temporal schemes and is parallelized based on the Message Passing Interface (MPI) paradigm. The parallelization of the code relies on a 2D domain decomposition (or pencil decomposition). A general purpose dedicated library called 2DECOMP&FFT (`http://www.2decomp.org/`) enables an efficient treatment of the parallel operations in the code. The code was developed by Dr. Sylvain Laizet, as described in his PhD thesis (`https://workspace.imperial.ac.uk/tmfc/Public/2005_LAIZET_PHD.pdf`) and its capabilities have been extended over the years as summarised in Laizet & Lamballais (2009); Laizet & Li (2011). This guide particularly focuses on a reorganised version of the code into which active and passive control features in the channel flow configuration were implemented during the course of the present thesis.

## A.1 Introduction

To take full advantage of the code, it is advised to use it on a supercomputing cluster. Indeed, contrary to a desktop PC (even configured with multiple cores), a cluster hardware is optimised for running on multiple processors. In addition, as the code output files are written using MPI-I/O, an efficient parallel file system is required for the code to swiftly output data. Finally on supercomputing clusters, it is possible to quickly switch between compilers and use pre-installed libraries, some of which are optimised w.r.t. the supercomputer hardware (Cray machines for instance on the ARCHER supercomputer). However, for completeness, the required steps to run on a desktop PC are briefly described hereafter. On a desktop computer, one should ensure that an MPI implementation is available along with compiler. Any standard Linux system should include the gcc/gfortran compilers. It is advised to use the version from the Argonne National Laboratory called MPICH2. In order to have access to the MPI routines for running the code, the steps described in Listing 2 (assuming an Ubuntu Linux distribution) should be followed. Once installed,

Listing 2: Installing MPI on local computer.

`mpirun`, `mpifxx`, `mpicxx` (where `xx` stands for the versions used in Fortran or C languages) commands are enabled. On the other hand, on supercomputing clusters, MPI libraries should be installed by default.

The code can be retrieved from the new website of the Incompact3d code at (`http://www.incompact3d.com/`).At the time this guide was written the code was hosted on Google Code. However, in April 2015, it was announced that this system would be shut down in August 2015 and would no longer host any project. As a consequence the Incompact3d code have been archived on Google Code website and be in a read-only state. Therefore searching on the web for Incompact3d would also give links to the aforementioned Google Code website. However, the new version and anything related to the code is now hosted on `http://www.incompact3d.com/` which is a dedicated platform. Links on this website also direct to the old Google code website which is now in read-only mode. In the Download section of the website, the version of the code relative to the channel flow configuration results described in Laizet & Lamballais (2009) is given as a tarball file. With this file it is possible to run a standard channel flow simulation enabling to compare the statistics (such as $f_{rms}$, where $f$ stands for the primitive flow variables) to those in Kim *et al.* (1987*b*); Moser *et al.* (1999). A short user guide is also present. The present guide relative to this reorganised version of the code including all the flow control problems implemented in the Incompact3d  code will become available online as well.

An other version of the code is available upon request from the author of this guide.

| | | | | |
|---|---|---|---|---|
| doc | external | INSTALL | makefile | makefiles |
| newprob | src | testsuite | tools | |

Table A.1: Structure of the base directory.

This version aims at reorganising the whole code making the installation and compilation more automated. Besides these technical changes, active and passive control features in a channel flow configuration are implemented in this new version. This revamped version stems from the base trunk revision. The new structure consists in the files and directories described in A.1 and called hereinafter the base directory. The description that is given in the present guide is relative to this last version. Once entering in the base directory as shown in A.1, it is possible to read the INSTALL file to quickly launch a basic simulation. However, it is recommended to read this guide, located in doc, before launching a simulation. Each of the directory or files mentioned above are next described in further details.

## A.2 Directory doc

This directory contains the guide for the code and the present restructured version with flow control capabilities. This guide gives details on how to how set up a new problem and launch a simulation, several other features and how to use the channel flow geometry for flow control problems are also described. Only the pdf file is present in this directory. If any modification is added to the present version, it will be updated and put online along with the modified sources.

## A.3 Directory external

This directory contains three subdirectories: checkpoint disks_bin disks_gen

The directory checkpoint contains checkpoint (or restart) files which are the files written in binary format created at the end of a simulation (this will be described in further details in §A.7). This allows to store all the primitive variable fields at the end of a simulation in order to restart from the state at which the simulation was either stopped in the case of lack of walltime or ended because it reached the final time step as specified in the parameters file. The checkpoint files that are stored in `checkpoint` directory are needed for development purposes as discussed in §A.8. The general idea is that when one modifies any source files of the code in `src`, apart from this modification, the code should be able to recover well established results for a given configuration. In this way, it is possible to check whether the modifications implemented in the source code do not alter the base behaviour of the code for the configuration under consideration. This is the reason why several directories relative to each type of physical problem considered have been created. For instance, if one wants to apply some modifications to the channel flow base problem and check the consistency of the code with a standard channel flow Kim *et al.* (1987*a*); Moser *et al.* (1999), the checkpoint that is going to be used is located in external/checkpoint/standard/nx_128_ny_129_nz_128 Therefore the checkpoint files are stored in `checkpoint` such that these can be used by the script files in the `testsuite` directory and will be taken from `checkpoint/standard`. Note that some basic checkpoint files have been created and added but the user can also add its own checkpoint files and store them in the relevant directory. In this case, it will also be necessary to modify the paths in the scripts used in the testsuite directory, described in details in §A.8.

The next directory in the `external` directory is called `disks_gen` and contains the program and scripts necessary for generating the flush-mounted rotating actuators velocities (streamwise and spanwise) that are passed as boundary conditions to the code when control through rotating actuators (Ricco & Hahn, 2013) are used. Two cases can be modelled: rotating disks and rotating rings. As such, two different directories have been created with these names. In each of these, all the process of creation of the files containing the rotating disks/rings velocity ($u_{\mathrm{disk}}$ and $w_{\mathrm{disks}}$ with $u$ and $w$ the streamwise and spanwise velocity) data is automated. `disks_gen` contains the two directories called `disks` and `donuts`. In the first one rotating disc velocities are generated while the second generates velocities for rotating ring-shaped actuators as shown in figure A.3. **Note that in all programs and scripts the generic name "disks" is used either for rotating discs or rings**. The script called `all.sh` is in charge of all the tasks which are summarised in figure A.1. Once `all.sh` is executed all the processes described in figure A.1 are carried out automatically. The end result are the files `disks_opac_x.dat` and `disks_opac_z.dat` which can then be used in the code. Therefore the script called `scr.sh` is automatically executed and does not need to be manually executed by the user. This script actually automates the process for creating the files `disks_opac_x.dat` and `disks_opac_z.dat`. One can argue that `disks_phys_x.dat` and `disks_phys_z.dat` are already binary files and could be read in the code. However, in order for the `decomp_2d` interface to be able to read and pass the data to the boundaries in the subroutine `pre_correc` in `navier.f90` it is necessary to do such a step. Indeed, decomp2d_io calls are adapted for reading 3d arrays in general, the data from the binary files disks_phys_x.dat and disks_phys_z.dat are stored in a 3d array, but filled with the same information for all $y$ planes. Therefore in all the $y$ planes ($j$ index in the code), the same streamwise and spanwise disk velocity is stored enabling the use of the decomp_2d I/O calls (in `src/decomp_2d.f90`).

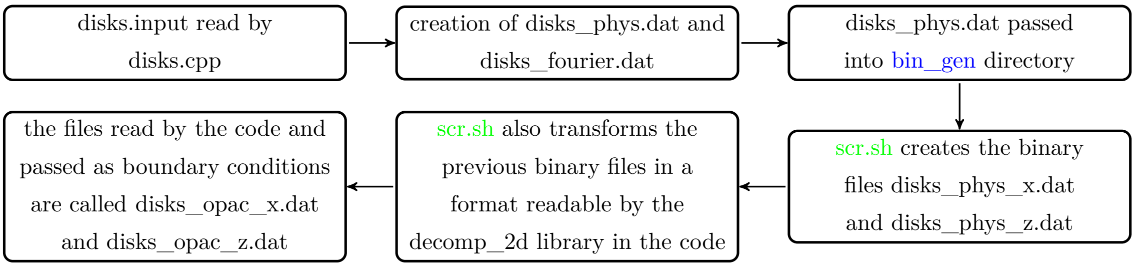

disks.input read by disks.cpp
creation of disks_phys.dat and disks_fourier.dat
disks_phys.dat passed into bin_gen directory
scr.sh creates the binary files disks_phys_x.dat and disks_phys_z.dat
scr.sh also transforms the previous binary files in a format readable by the decomp_2d library in the code
the files read by the code and passed as boundary conditions are called disks_opac_x.dat and disks_opac_z.dat

Figure A.1: Summary of the automated steps following the execution of the script `all.sh`

Note that the file containing the velocity components of the rotating actuators are given both in Fourier and physical space, thereby enabling them to be used for codes that would be spectral as well. In the present case, as the boundary conditions are only considered for the Incompact3d code, one is only concerned with the file `disks_phys.dat`. Figure A.2 shows a visualisation from the files `disks_opac_x.dat` and `disks_opac_z.dat` using ParaView in a channel flow geometry. The visualisation of the data files previously generated can be obtained in the directory `external/disks_gen/disks/bin_gen/pvwiew_diag` for discs and `external/disks_gen/donut/bin_gen/pvwiew_diag` for rings-shaped actuators. In the latter, one only needs to execute the script `scr.sh`. This script retrieves the grid parameters from the file `disks.out` that has been automatically generated in `bin_gen` and carries on with the compilation and execution. This results in `.vtr` files that can then be visualized in ParaView. All the steps mentioned above are also summarised in

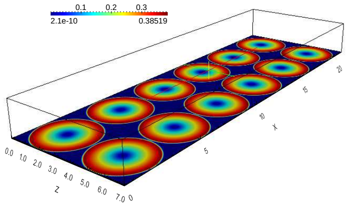


Figure A.2: $\sqrt{u_{\text{disc}}^2 + w_{\text{disc}}^2}$ at the lower wall.

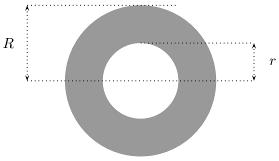


Figure A.3: Configuration of rotating rings (or donuts) device. The minor to major radius ratio is denoted $a$.

external/disks_gen/README. After the execution of the scripts in external/disks_gen, the resulting output is directed in external/disks_bin. These files can then be used as input boundary conditions for the flush-mounted rotating discs/rings actuators.

## A.4 makefile

When one enters the base directory the file makefile is present. Its primary purpose is to compile and link the source code. As such it is a master makefile that uses a specific machine makefile stored in A.5. It is sufficient to type make and the compilation will start. The important point is that this makefile will automatically detect the machine on which the code is being compiled. Therefore based on a hostname selection as shown in Listing 3 a specific machine makefile is selected and the compilation starts with the options specific to the aforementioned machine (paths to include directories, $\cdots$). If one does not find the machine in the list at the beginning of the makefile, is is possible to add the line for specifying it in the selection process of the machines. This is the only way to proceed with the compilation having correctly chosen. In addition you need to create the directory specific to your machine in makefiles directory. These machine specific makefiles are further described in A.5. For the moment the N8-Polaris cluster and ARCHER supercomputer can be selected as part of the machine selection process. Apart from the addition of new machines in the list as shown in Listing 3, makefile should not be modified further.

Listing 3: Excerpt of the master makefile.

## A.5 Directory makefiles

In this directory, all the specific machine makefiles are gathered. In the present version, directories relative to ARCHER supercomputer and the N8-Polaris cluster are specified. Each of these directories contains default files: makefile, parameter file, submit command file specific to the batch system of the selected machine and a script file to launch some channel flow problems. Therefore in archer and polaris there are four default files. A directory called new_machine, serves as a model to build a directory for a new machine that is not presently in this directory. For instance suppose that you are on ARCHER supercomputer, in the base directory of Incompact3d. Once you type make, the master makefile determines that you are on the ARCHER supercomputer and selects the archer directory in the makefiles and creates a directory called bin in the base directory in which it copies the machine makefile, i.e. here archer.mk. After compilation the directory bin contains, in addition, a directory called obj_archer (called obj_polaris if compiling on the N8-Polaris cluster), that contains the object (".o" extension) files from the compilation. Besides bin contains the executable that is symbolically linked in the prob# directory to which the default parameter, script and submit command files have been copied to, after launching the master makefile in the base directory. The creation of prob directories is further described in §A.6.

## A.6 Script newprob

The script newprob written in Perl is in charge of the creation of a new problem directory. This script has to be executed from the base directory. Once executed, it will create the first time a directory called prob01. If a prob directory prob<N> already exists, executing newprob will create incrementally a prob directory, i.e. prob<N+1>. Note that the prob directory will contain:

### A.6.1 the executable inc3d_<machine name>

It is symlinked to the directory bin (created at the compilation) in the base directory (see §A.5). Therefore every time the code is compiled from the base directory the executable that is in bin will be modified and this modification will propagate the executable that is subsequently mirrored in the prob directory.

### A.6.2 the parameter file

The file inc3d.prm contains all the parameters that the code can handle. The user has to change it accordingly to the problem under investigation. Hereafter a detailed description of each parameter that is used in this file is given. The value in brackets indicates the type of the parameter.

- xlx,yly,zlz[real]: Size of the box in $x$, $y$ and $z$ directions respectively. These quantities represent the extent of the box. These are normalized with a characteristic length. For example for a channel flow, it is the channel half-height $\delta^*$. In the following, unless otherwise stated, "*" superscript indicates a dimensional quantity.

- re[real]: Reynolds number. For a channel flow, $re = R_p = U_p^* \delta^* / \nu^*$. $U_p^*$ is the centreline velocity for a Poiseuille laminar flow and $\nu^*$ is the kinematic viscosity of the fluid.

- retau[real]: Friction Reynolds number corresponding to the value of re. It is used together with re for conversion from + units to outer units. This is mainly used in the subroutines related to wall control in forcing.f90.

- sc[real]: Schmidt number, used when dealing with mass transfer problems. In the case of channel flow, this parameter is automatically deactivated when the itype parameter is set for this configuration (described below).

- u1,u2[real]: maximum and minimum values for the inflow condition. For a channel flow, where a parabolic equilibrium profile is modelled, these values are set to 1.

- noise,noise1[real]: two turbulence intensity levels. noise is used as a parameter in the initialization (subroutine init in navier.f90). noise1 is used as an inflow condition (subroutine inflow in navier.f90)

| | | | | | |
|---|---|---|---|---|---|
| 1 | constant flow field | 2 | channel flow | 3 | wake flow |
| 4 | mixing layer | 5 | channel flow | 6 | taylor-green vortices |
| 7 | cavity flow | 8 | flat plate boundary layer | 9 | water tank |

Table A.2: Flow type options.

- `dt`[real]: time increment $\Delta t$. The time step normalization is expressed consistently with $\delta$* and $R_p$. The code runs at constant $dt$. Therefore a trial simulation with a few time steps should be tested w.r.t. the spatial resolution for the time step to lead to stable computations. Following this step, a production run can be carried out. Choosing the value for $\Delta t$ is a trade-off between stability and computational time.

- `nclx,ncly,nclz`[integer]: flags specifying the boundary conditions (BC) in the $x$, $y$ $z$ directions respectively. `ncl*` = 0 specifies periodic BC, `ncl*` = 1 stands for free-slip and `ncl*` = 2 no-slip/Dirichlet BC.

- `itype`[integer]: this parameter sets the flow type. Various choices are at hand. These are described in Table A.2.

- `iin`[integer]: inflow conditions. Depending on this value different type of initialization are used for the turbulence to kick off (based on a standard random seed and clock system random number generator). This parameter is used in the subroutine `init`: if `iin`=1 then a random noise is generated, while when `iin`=2, a previously correlated random noise is read from file.

- `ifirst`[integer]: first time step for the simulation.

- `ilast`[integer]: last time step for the simulation.

- `nscheme`[integer]: temporal scheme used in the time advancement of the Navier-Stokes equation. Four different schemes can be used as detailed in Table A.3.

- `istret`[integer]: flag for mesh refinement. It is described in Table A.4. For a channel flow `istret`=2. Along with this parameter, the value for the parameter `beta` should be set as well.

- `beta`[real]: refinement/stretching parameter. It is computed by using the program `calc_beta.f90` located in the directory `tools`.

| `nscheme` | scheme | order |
|---|---|---|
| 1 | Adams-Bashforth | second |
| 2 | Runge-Kutta | third |
| 3 | Runge-Kutta | fourth |
| 4 | Adams-Bashforth | third |

Table A.3: Time advancement schemes.

| `istret` | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| refinement | none | center | both walls | bottom wall |

Table A.4: Refinement parameters choice.

- `iskew`[integer]: this parameter specifies the way the convective terms are computed. If `iskew`=0 the rotational form is chosen while with `iskew`=1, the skew-symmetric form is computed. The skew-symmetric formulation should be used in order to reduce the effect of aliasing errors as described in Kravchenko & Moin (1997). This point is explained in further details in §A.7.2. The conservation properties of finite difference schemes w.r.t. the formulation of the terms in the momentum and continuity equations is also discussed in Morinishi *et al.* (1998).

- `iscalar`[integer]: parameter allowing to include a passive scalar.

- `rot_channel`[logical]: this parameter enables initialisation of a rotating channel flow to trigger a turbulent state. To obtain a turbulent state, the first stage consists in starting with a rotating channel flow: `rot_channel=.true.` and setting the value for the Rossby number (parameter `rossby`). The rotating channel should be run for at least $10^4$ time steps depending on the resolution or the Reynolds number. Once this stage is completed, the rotating channel flow should be deactivated and the channel flow to be run from the saved checkpoint file `sauve.dat`. With this procedure the flow should reach the turbulent level at the required Reynolds number.

- `rossby`[real]: parameter that is taken into account only if the parameter `rot_channel` is set to `.true.` (which sets the rotating channel flow). By default this value is set

| `forcing_type` | control configuration |
|---|---|
| 1 | spanwise wall oscillations |
| 2 | steady rotating discs |
| 3 | hydrophobic surfaces |
| 4 | oscillating discs |
| 5 | opposition control |

Table A.5: Flow control possible configuration choices.

to 1/9. For further details on rotating channel flows see Grundestam *et al.* (2008) and the references therein.

- `forcing`[logical]: sets a wall forcing problem. This logical flag is used in the file `forcing.f90` which gathers all the forcing methods for wall control. This file is described in details in §A.7.11.

- `forcing_type`[integer]: if `forcing` is set to `.true.`, then this parameter has to be chosen as described in Table A.5.

- `slip_type`[integer]: when the `forcing_type`=3, this parameter sets the slip length to be either constant or shear-dependent. For constant slip, `slip_type`=0 and for shear-dependent slip `slip_type`=1

- `slip_direc`[integer]: one `slip_type` is set, then it is needed to know on which direction to apply a slip velocity. Table A.6 gives a summary of the options that can be used for using hydrophobic surfaces as a control, i.e. when `forcing_type`=3. The constant slip model can be used to model the results by Min & Kim (2004). the values for $a_x$, $b_x$, $a_z$, $b_z$ are set through the parameters `a_x`, `b_x`, `a_z`, `b_z`. Note that these parameters are mutually exclusive in the sense that when `slip_direc` is chosen, e.g. `slip_direc`=1, setting values for `a_z` or `b_z` will not be considered. This is something general for the other options as well.

- `tw_ps`[real]: when `forcing_type`=1, control through spanwise wall oscillations is used to model the cases in Quadrio & Ricco (2004). This value is the period of oscillations and needs to be given in "+0", i.e. "+" units based on the uncontrolled

| `slip_type` | `slip_direc` | model |
|---|---|---|
| 0 | 1 | $u_s = l_s \frac{\partial u}{\partial y}\vert_{\text{wall}}$ |
| 0 | 2 | $w_s = l_s \frac{\partial w}{\partial y}\vert_{\text{wall}}$ |
| 1 | 1 | $u_s = a_x \left(\frac{\partial u}{\partial y}\vert_{\text{wall}}\right)^2 + b_x \frac{\partial u}{\partial y}\vert\text{wall}$ |
| 1 | 2 | $w_s = a_z \left(\frac{\partial w}{\partial y}\vert_{\text{wall}}\right)^2 + b_z \frac{\partial w}{\partial y}\vert\text{wall}$ |

Table A.6: Options for modelling hydrophobic surfaces.

flow. The conversion will be automatically carried out w.r.t. the Reynolds number, and thereby converted to Poiseuille units to comply with the code units.

- `wm_ps`[real]: this parameter is required along with `tw_ps` when `forcing_type`=1 for spanwise wall oscillations. This value is the maximum amplitude of the oscillations also to be given in "+0".

- `a_x`,`b_x`,`a_z`,`b_z`[real]: slip length model parameter (refer to Table A.6 for the different possible choices).

- `oscd_ps`[real]: when `forcing_type`=4, the oscillating rotating actuators can be modelled. This parameter gives the value for the oscillation period of the discs in "+0" units.

- `j_detect`[integer]: when `forcing_type`=5, the problem considered in (Choi *et al.*, 1994). In this case one has to set the level the plane at which the virtual detection plane needs to be set. As this value is an integer, it is based on the index of the wall-normal grid. Therefore one has first to choose the location in "+0" units convert into outer units depending on the Reynolds number and find corresponding index.

- `ilit`[integer]: if set to 0: starts a new simulation and if set to 1, the simulation starts from a checkpoint file that is saved as `sauve.dat` at the end of the simulation and contains the flow field information.

- `isave`[integer]: sets the frequency for writing the checkpoint file

- `imodulo`[integer]: sets the frequency for the data files (basically the primitive variables).

- with_der_vel[logical]: if set to .true., the derivative of the velocity fields are computed and written as output in addition to the primitive variables (in visu.f90). This is useful for obtaining for instance the invariants ($Q$, $R$, $\Delta$, $\lambda_2$ criterion, $\cdots$) through the nine elements of the velocity gradient tensor. In the case this value was set to .false., there is still the possibility to compute these derivatives by using the programs in the directory tools/grad_vort_gen (described in §A.7.9). The idea is to reduce the output of the code if one is limited in terms of disc space, while being able to compute useful variables at a later time when disc space is no longer an issue.

- with_vort[logical]: if set to .true. the three components of the vorticity are computed. The description for this logical selection is similar to

- with_der_vel (can be computed within tools/grad_vort_gen as well if set to .false.).

- diagdir[string]: path to the directory for the flow fields output.

- file[string]: extension used for file output

- ivirt[integer]: sets the body force type

- cex[real]: $x$ position of the solid body

- cey[real]: $y$ position of the solid body

- cez[real]: $z$ position of the solid body

- ra[real]: radius of the solid body

### A.6.3 Script file scr.sh

This bash script serves as a model for the user to develop on it. It contains some rules. First, it is assumed that this script file will be run once the parameters in the file inc3d.prm are set correctly and that the number of grid points in module_param.f90 is also correctly set. The reason is that the script automatically computes the value of the stretching parameter $\beta$ (which is given a default value once the prob directory is created) by detecting the value for $n_y$ (the number of grid points in the wall-normal direction) and the Reynolds number $R_p$ based on the maximum velocity of the Poiseuille parabolic profile. Therefore ny in src/module_param.f90 along with re in the parameter file both need to be known also in order to be set correctly before running the script. The script will subsequently compile the program `calc_beta.f90` located in `tools`. For security the user is asked if for confirmation to carry out this computation. Another assumption concerns the location and name of the checkpoint file. If one wants to restart from a previously stopped simulation then as the checkpoint file `sauve.dat` has been created at the end of the previous simulation, it is possible to directly restart from this file without changing its name. On the other hand, if one has a checkpoint file from a simulation and with a random name, then this checkpoint file needs to be copied under the name `sauve.dat` to enable the code to read it. The path of the directory where the executable and submit command file are copied also needs to be specified in the parameters file `inc3d.prm` as it is used in `scr.sh`. This path is assumed to be the one that also contains the checkpoint file, either already having been given the name `sauve.dat` or having a random name. Here again the user is asked whether the checkpoint file with the right name (`sauve.dat`) already exists in the work directory or if this is not the case, is asked to rename the to `sauve.dat` in the work directory.

### A.6.4 Batch file `submit.cmd`

Depending on the machine detected by the script `newprob`, the relevant `submit.cmd` file specific to a machine or supercomputer is chosen. The `submit.cmd` file already contains the required options for the specific machine (here N8-Polaris cluster and ARCHER supercomputer) and default walltime, number of processors and nodes. The user can change these values if needed.

## A.7 Directory `src`

All the f90 source files are included in this directory. There are currently 43 source files. In addition a file called `files.mk` sets all the source files dependency. As such, if one adds a new file or includes a new module use in an existing file, the dependencies should be added in this file.

**Be careful of circular dependency issues**. A directory called `nusd` included in `src`. It contains older versions of some of the files that were already part of the original version of the code. These are just kept for the sake of completeness but do not have to be actually used. In any case, the `makefile` and machine `makefiles` will only compile those source files in the src directory and which dependencies have been set in `files.mk`. Now each of the main features of each f90 file is going to be described.

### A.7.1 `alloc.f90`

This file gathers all the routines to allocate arrays in a general way. It allows to have a more compact way of allocating arrays. Hence instead of having to allocate with `allocatable` command on a given pencil and with all the sizes required, it is possible to allocate only stating the name of the variable using calls similar to `alloc_<pencil>(var)`. The real and complex subroutines for these calls are defined in this file and are part of an interface in `decomp2d.f90`.

### A.7.2 `convdiff.f90`

In this file two subroutines are present: `convdiff` and `scalar`. `convdiff` evaluates the convective and diffusive contributions in the Navier-Stokes equations. These terms involve first and second derivatives. Therefore the computations involved need to use the subroutines that are part of the file `derive.f90` (refer to §A.7.10).

As mentioned in the description of the parameters file §A.6.2, one can compute the convective terms (`CV`) using two formulations: a rotational and skew-symmetric form. The skew-symmetric form allows to reduce aliasing errors, and remain energy conserving for the spatial discretisation (Kravchenko & Moin, 1997). This skew-symmetric form is set by the parameter `iskew`=1 (§A.6.2). In most of the simulations, one should keep this form. In its vectorial form the skew-symmetric form is given by):

$$\texttt{SKW} = \frac{1}{2}[\boldsymbol{\nabla}(\boldsymbol{u} \otimes \boldsymbol{u}) + (\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{u}] \tag{A.1}$$

Each component of the expression given by (A.1) is:

$$\mathrm{SKW}_x = \frac{1}{2}\left(\frac{\partial uu}{\partial x} + u\frac{\partial u}{\partial x}\right) + \frac{1}{2}\left(\frac{\partial uu_y}{\partial y} + u_y\frac{\partial u}{\partial y}\right) + \frac{1}{2}\left(\frac{\partial uu_z}{\partial z} + u_z\frac{\partial u}{\partial z}\right) \tag{A.2}$$

$$\mathrm{SKW}_y = \frac{1}{2}\left(\frac{\partial uu_y}{\partial x} + u\frac{\partial u_y}{\partial x}\right) + \frac{1}{2}\left(\frac{\partial u_y u_y}{\partial y} + u_y\frac{\partial u_y}{\partial y}\right) + \frac{1}{2}\left(\frac{\partial u_y u_z}{\partial z} + u_z\frac{\partial u_y}{\partial z}\right) \tag{A.3}$$

$$\mathrm{SKW}_z = \frac{1}{2}\left(\frac{\partial uu_z}{\partial x} + u\frac{\partial u_z}{\partial x}\right) + \frac{1}{2}\left(\frac{\partial u_y u_z}{\partial y} + u_y\frac{\partial u_z}{\partial y}\right) + \frac{1}{2}\left(\frac{\partial u_z u_z}{\partial z} + u_z\frac{\partial u_z}{\partial z}\right) \tag{A.4}$$

In convdiff the previous expressions are computed by using the derivative definitions in derive.f90 (§A.7.10). As the operation carried out in terms of derivatives are carried out locally, the information is passed to the relevant pencils. Note that one starts with arrays in the X-pencils. Therefore if one needs to compute a derivative w.r.t. the wall-normal direction, then a transpose operation from X to Y is the first step, then the derivative in $y$ can be computed locally and the information passed to other pencils by relevant transpose operations. Following this procedure, all the expressions given by equations (A.2)-(A.4) are computed. For the rotational form (RT):

$$\mathtt{ROT} = (\boldsymbol{\nabla} \times \boldsymbol{u}) \times \boldsymbol{u} \tag{A.5}$$

the procedure follows likewise knowing that each component for (RT) is given by:

$$\mathtt{ROT}_x = v\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right) + w\left(\frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}\right) \tag{A.6}$$

$$\mathtt{ROT}_y = u\left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right) + w\left(\frac{\partial v}{\partial z} - \frac{\partial w}{\partial y}\right) \tag{A.7}$$

$$\mathtt{ROT}_z = u\left(\frac{\partial w}{\partial x} - \frac{\partial u}{\partial z}\right) + v\left(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}\right) \tag{A.8}$$

The diffusive terms are computed in the same lines, with the difference that calls to subroutines derxx, deryy, derzz are used from the file derive.f90.

### A.7.3 decomp2d.f90

The foundation of the code relies on a 2D domain decomposition (or pencil decomposition) using the 2DECOMP&FFT library based on the MPI standard. It is a general purpose library for this type of decomposition and within this framework also computes FFTs. The reference can be found on the website at the following link http://www.2decomp.org/. The domain decomposition file decomp_2d.f90 uses the number of grid points set in module_param.f90 (§A.7.4) through nx,ny and nz within a 2D mapping given by p_row and p_col also set in module_param.f90. Note that p_row×p_col=Nprocs, with Nprocs the total number of MPI tasks.

### A.7.4 module_param.f90

In this file, most of the variables used in the different modules are defined and set. Note that the 3D arrays will be defined in another file called variables.f90. Hence either 1D or 2D arrays are defined in this file. The most important point in this file concerns the parameters `p_row` and `p_col`. These values set the 2D mapping used in the code and their product has to be consistent with the total number of MPI processors used in the submit batch file.

### A.7.5 `incompact3d.f90`

The main operations take place in the incompact3d.f90 file: setting of the parallel environment along with the domain decomposition, initialization of the flow fields, time advancement, statistics and outputting of the primitive variables to data files. The call to subroutine `parameter()` in `parameter.f90` (§A.7.6) is in charge of reading the Incompact3d parameter file (here called `inc3d.prm` §A.6.2).

One of the most important feature of the code is that by using compact finite difference schemes, it allows to tackle more geometrically complex problems. Subsequently a large set of combinations in the parameters setting the boundary conditions (`nclx`,`ncly`,`nclz`) can be used. More precisely, 10 different combinations are enabled to model different type of flow configuration: `(0 - 0 - 0)`, `(1 - 0 - 0)`, `(2 - 0 - 0)`, `(0 - 2 - 0)`, `(1 - 1 - 0)`, `(2 - 1 - 0)`, `(1 - 1 - 1)`, `(2 - 2 - 1)`, `(1 - 2 - 1) and (2 - 2 - 2)`. The channel flow configuration which is the one that will mostly be focused on is given by the configuration `(0 - 2 - 0)`. **Be careful of testing other configurations than those mentioned before**. Note that the attribute called `mytype`, is automatically set through the preprocessing directive `PRECISION` defined in the machine makefile. Once set to `double`, `mytype`=8 for double precision otherwise its value is 4 for single precision.

### A.7.6 `parameters.f90`

In addition to reading the parameters from the Incompact3d parameters file, the grid is set and in the case of a stretched grid, this configuration is built using the `stretching()` subroutine. Finally the coefficients that are needed for the time advancement are set consistently with the parameter `nscheme`. Note that as mentioned in §A.11, the type should always be real/complex (and not double precision/double complex) and their MPI counterpart.

Some of the parameters read from `inc3d.prm` file are written to the screen when the simulation starts. However, an output file has been added to keep track of all the parameters used in a simulation. This operation is carried out in `tools.f90` (§A.7.9) and written in the `diagdir` directory specified in `inc3d.prm`.

When periodic boundary conditions are chosen by setting `ncl*`=0, the first and last point of the domain are equivalent and the following relation holds:

$$\Delta_* = \frac{L_*}{n_*} \tag{A.9}$$

On the other hand, when `ncl*`=1 or 2, then:

$$\Delta_* = \frac{L_*}{n_* - 1} \tag{A.10}$$

Therefore for channel flow configuration with periodic boundary conditions for the streamwise and spanwise directions and Dirichlet boundary conditions for the wall-normal direction, i.e. `(0 - 2 - 0)`, the number of grid points has to be even in $x$ and $z$ and odd in $y$ when set in `module_param.f90` (§A.7.4).

The time advancement coefficients are also specified in `parameters.f90`. The choice of the scheme is made through the parameter `nscheme` in the parameter file (§A.6.2). In `incompact3d.f90` file, there are two loops for this procedure. One that loops over the interval given by `ifirst` and `ilast` and another one looping over the stages of the chosen scheme. For instance for a third order Runge-Kutta time advancement scheme, the number of stages are 3. Therefore for each time step, three evaluations are necessary to pass to the next time step (parameter $n_k$ as described in Laizet & Lamballais (2009)). The formulation for the Adams-Bashforth scheme is given by:

$$y^{k+1} = y^k + a_k \Delta t f(t_k, y^k) + b_k \Delta t f(t_{k-1}, y^{k-1})) \tag{A.11}$$

In the code, these are declared as arrays and the values therein are adapted to the scheme. For the previous Adams-Bashforth case (`nscheme`=1), $a_k = 0.5$ and $b_k = -1.5$. The other multi-step methods are given by the following equations with RK3 (`nscheme`=2), RK4 or Carpenter-Kennedy (`nscheme`=3) and AB3 (`nscheme`=4):

1. RK3

$$
\begin{aligned}
\mathtt{st}_1 &= 0 + \Delta t f(t_k, y^k) \\
y^{k+\frac{1}{3}} &= y^k + \frac{8}{15}\Delta t f(t_k, y^k) \\
\mathtt{st}_2 &= -\frac{17}{60}\mathtt{st}_1 + \Delta t f(t_{k+\frac{1}{3}}, y^{k+\frac{1}{3}}) \\
y^{k+\frac{2}{3}} &= y^{k+\frac{1}{3}} + \frac{5}{12}\Delta t f(t_{k+\frac{1}{3}}, y^{k+\frac{1}{3}}) \\
\mathtt{st}_3 &= -\frac{5}{12}\mathtt{st}_2 + \Delta t f(t_{k+\frac{2}{3}}, y^{k+\frac{2}{3}}) \\
y^{k+1} &= y^{k+\frac{2}{3}} + \frac{3}{4}\Delta t f(t_{k+\frac{2}{3}}, y^{k+\frac{2}{3}})
\end{aligned}
\tag{A.12}
$$

2. RK4: There are 5 coefficients that cannot be written as a ratio of two integers. For further details on the values one can directly look at the `parameter.f90` source file.

3. AB3

$$
y^{k+1} = y^k + \frac{23}{12}\Delta t f(t_k, y^k) - \frac{16}{12}\Delta t f(t_{k-1}, y^{k-1}) + \frac{5}{12}\Delta t f(t_{k-2}, y^{k-2}) \tag{A.13}
$$

### A.7.7 `variables.f90`

All the major 2D and 3D arrays should be declared in this file. As mentioned in the online guide, the 3D arrays should appear as the argument of a subroutine. In the declaration of the arrays, it is important to declare these in the right pencil. One can notice that the arrays in this file are declared in addition to the standard allocatable command, with a `save` keyword. This allows to keep the values at the end of a subroutine. The allocation of the arrays are carried out through the calls to the subroutines defined in `alloc.f90`. These allow to allocate in a shorter way the arrays in the X, Y and Z pencil.

### A.7.8 `schemes.f90`

In this subroutine, all the coefficients for computing the derivatives which are used in `derive.f90` are specified (also based on the boundary condition parameters). These coefficients are used for the first and second derivatives as well as for the interpolation operations. In addition to the subroutine `schemes` which is part of `schemes.f90`, a subroutine called `prepare` is also present. It is a "tool" that is used various times in the subroutine `schemes`. Indeed, as the derivatives are implicit, at least tridiagonal matrices are solved. For this purpose `prepare` is used as part as an LU decomposition to compute the coefficients of the L and U matrices. In this procedure the lower, diagonal and upper elements of the matrix are taken as input: `b`, `c` and `f` arguments. Once treated, the output consists in the coefficients of the lower and upper triangular matrices which are stored in `s` and `w`.

### A.7.9 `tools.f90`

In this file, various utility subroutine are defined. Among others, the stretching subroutine for the mesh refinement is defined and the computation of the flow rate. One of the most important feature, however, is the definition of the subroutine `restart()` that enables to restart from a checkpoint. Depending on the value taken by `nscheme`, the restart procedure is adapted to reconstruct the flow fields and pressure terms.

For the `stretching()` subroutine, one can refer to the paper Laizet & Lamballais (2009) in which the mapping function $h$ is defined. This mapping function is related to a parameter called $\beta$. This parameter is given as an input in the `inc3d.prm` file. It should be computed relatively to the Reynolds number and the number of grid points in the wall-normal direction. This can be carried out by using the Fortran program called `calc_beta.f90` located in the `tools` directory. In `stretching()`, the physical grid coordinates are stored in the array `yp`. The function that is used is built from the equation given in Laizet & Lamballais (2009). The underlying idea for using such a mapping function resides on the solution of the Poisson equation. Indeed, the computation of this equation is carried out in the spectral space and in the case of free-slip/Dirichlet boundary conditions the point is that one direction does not fufill homogeneity in order to apply the Fourier transform. However, the mapping of the physical grid to the computational grid avoids the aforementioned problem. The only operation to carry out for the FFTs, is to multiply by the metric of the mapping. However, when applying this procedure for the computation of the derivatives, quantities like $1/h'\frac{\partial f}{\partial s}$ (where $s$ is the computational grid) appear and their Fourier transform is problematic. It is well-known that the multiplication in the direct space is equivalent to a convolution in spectral space, operation that is computationally very expensive. Therefore using mapping may appear to be problematic. However as mentioned in Laizet & Lamballais (2009) and studied in details in Cain *et al.* (1984), a truncation in the metric allows to alleviate this issue and in the case of a channel flow combined with the use of cosine transforms, solve the Poisson equation.

The subroutine restart, reads and writes the checkpoint file saved as sauve.dat. The file sauve.dat contains all the information about the flow. It is saved every isave time steps (specified in the inc3d.prm) file. This file is therefore present at the end of each simulation whether it has completed within the required time window [ifirst,ilast] or not. By setting ilit=1, it is possible to restart from where the simulation stopped. In restart(), the input/output is managed using MPI-I/O. For more details on the arguments and routines used in MPI-I/O one can refer directly to the MPI standard documentation.

The subroutine test_speed_min_max computes the maxima and minima velocity components (ux,uy,uz). This is used in order to check if the simulation is becoming unstable. At each time step where this subroutine is called, the aforementioned values are computed and compared to a given threshold settled value for the minimum and maximum velocities. If the computed velocity becomes greater than or smaller than this threshold, it means that the simulation has reached an unstable state.

### A.7.10 derive.f90

This subroutine computes the $x$ derivatives depending on the value for ncl*x. The general idea is that, as the spatial derivatives are computed on stencils that couple neighbouring points, a linear system needs to be solved in order to get the relevant values on a given node. In addition, thanks to the way the transpose operations are defined, all the procedures defined for computing the derivatives are implemented locally. In this way one can straightforwardly match the coefficients in schemes with those schemes in (Laizet & Lamballais, 2009). All the coefficients that are used in der* are set in schemes.f90 (§A.7.8). Depending on the value for nclx, the coefficients thereby the schemes used are changed consistently. The first argument of der* stores the value of the derivative, while the second one takes the input variable which derivative needs to be computed. Once the matrix filled with the coefficients of the relevant scheme, it is inverted using standard matrix inversion algorithm (backward-forward substitution). Note that in the case of free slip boundary conditions, symmetric (npaire=1)/antisymmetric (npaire=0) is imposed for the ghost points, translating the condition given at page 5992 of (Laizet & Lamballais, 2009). The Dirichlet type of boundary conditions follows the same type of reasoning for building the terms but this time based on a Padé-three-point formulation. These procedures equally apply for derx, dery and derz. The only difference for the wall-normal direction concerns the stretching, In this case, an additional argument array in $y$ direction is used and called `ppy` which includes the stretching weights that are then multiplying the schemes coefficients.

The subroutines computing the second derivatives used for the diffusive terms in are called `derxx`, `deryy`, `derzz`. The principle for relies on the same paradigm as in the first derivatives computation. Of course new coefficients are set in `schemes.f90` to be used in the relevant subroutines.

### A.7.11 `forcing.f90`

In this file various type of control strategies are implemented. The selection of these are done by setting the logical parameter `forcing` to true `inc3d.prm` and then setting the parameter `forcing_type` to the required control method as described in Tables A.5 and A.6.

### A.7.12 `FR_perf.f90`

This module allows to obtain a timing summary to estimate the amount of computation devoted to each subroutine. This timing tool is initiated in `incompact3d.f90`, with `call perfinit`, then the main program timing starts with `call perfon('`Incompact3d`')`. At the end of the main program time loop, the probing process terminates with the following statement: `call perfout('`Incompact3d`')`. The names in the quotes can be chosen at will. These will appear in the summary text file at the end of the simulation. In order to probe the other module files, one can follow the example relative to the timing of the subroutine `convdiff` in `convdiff.f90` as shown in 4. Each subroutine requiring time probing, should use a similar pattern in the calls to `perfon('user_defined_name')` and perfoff. At the end of the simulation, the output file called `perf_sum.txt` should appear in the output directory. This file gives a summary of the performance. For each probed subroutine, the name, the number of times called, the individual time spent for the whole simulation, and the relative time spent are displayed. This allows to track the most time consuming parts of the code and in the case where other modules and subroutines are added in the code, to assess their impact in terms of computations. One sample example is shown in the following.

```
***************************************************************
***********           performance results          ***********
***************************************************************
--------------------------------------------------------
Timer name             #calls     Time(s)   Incl.(%)
--------------------------------------------------------
Incompact3d              1       17.923        100.0
test_speed_min_m        11        0.034          0.2
convdiff                10        4.802         26.8
intt                    10        0.022          0.1
pre_correc              10        0.001          0.0
divergence              20        3.863         21.6
poisson                 10        1.794         10.0
inversion5_v1           20        0.051          0.3
gradp                   10        1.082          6.0
corgp                   10        1.421          7.9
channel                 10        1.099          6.1
statistic               10        0.210          1.2
```

## A.8 Directory testsuite

This directory is intended to check the behaviour of the code during a development phase. When one modifies the source code for development purpose, the code behaviour should not be altered with respect to standard problem settings. This is the reason why these type of tests allows to check that the modification brought in the source code is not impacting the physics of other problems. For example, if the source code is modified to bring some optimisation within a channel flow configuration, then performing a simulation should still lead to the same results and statistics as in Kim *et al.* (1987*b*); Moser *et al.* (1999). In that sense the testsuite simulations check that this is actually the case without the need of performing a complete simulation. Indeed, each of these tests is performed with a limited number of timesteps. The comparison is based on the standard output of the code and not the large binary data files. The directory testsuite is organised as shown in Table A.7. For these testsuites one can choose between the following cases relative to a channel flow

```
!********************************************************************
!
subroutine convdiff(ux1,uy1,uz1,ta1,tb1,tc1,td1,te1,tf1,tg1,th1,ti1,di1,&
     ux2,uy2,uz2,ta2,tb2,tc2,td2,te2,tf2,tg2,th2,ti2,tj2,di2,&
     ux3,uy3,uz3,ta3,tb3,tc3,td3,te3,tf3,tg3,th3,ti3,di3)
!
!********************************************************************
USE param
USE variables
USE decomp_2d
USE FR_perf_mod

implicit none

real(mytype),dimension(xsize(1),xsize(2),xsize(3)) :: ux1,uy1,uz1
real(mytype),dimension(xsize(1),xsize(2),xsize(3)) :: ta1,tb1,tc1,td1,te1,tf1,tg1,th1,ti1,di1
real(mytype),dimension(ysize(1),ysize(2),ysize(3)) :: ux2,uy2,uz2
real(mytype),dimension(ysize(1),ysize(2),ysize(3)) :: ta2,tb2,tc2,td2,te2,tf2,tg2,
                                                      th2,ti2,tj2,di2
real(mytype),dimension(zsize(1),zsize(2),zsize(3)) :: ux3,uy3,uz3
real(mytype),dimension(zsize(1),zsize(2),zsize(3)) :: ta3,tb3,tc3,td3,te3,tf3,tg3,th3,ti3,di3

integer :: ijk,nvect1,nvect2,nvect3,i,j,k

nvect1=xsize(1)*xsize(2)*xsize(3)
nvect2=ysize(1)*ysize(2)*ysize(3)
nvect3=zsize(1)*zsize(2)*zsize(3)

call perfon('convdiff')

...

call perfoff

end subroutine convdiff
```

Listing 4: Example for setting the performance calls.

geometry: (a) in directory `standard` (no control) (b) `control`. Therefore the following steps are general and apply indifferently to the standard or controlled cases.

1. First the script `testsuite_ref` needs to be launched **BEFORE BRINGING ANY MODIFICATION TO THE CODE**. The user will be asked to choose the type of problem to be benchmarked. Depending on this choice, the standard output of the code will be directed to the `standard` or `control` directories. Whithin each directory, a directory called `out_ref` and another one called `out_test` are present. The directory `out_ref` will gather the output from the execution of the

| | | | |
|---|---|---|---|
| `control` | `inc3d_<machine>` | `inc3d.prm` | `README` |
| `standard` | `testsuite_cmp` | `testsuite_ref` | |

Table A.7: Structure of the `testsuite` directory.

reference testsuite script `testsuite_ref`. Parameter files are also present and can be added if the user wants to add its own reference cases. In the present state of the code, the standard (uncontrolled) directory contains two parameters file while the control directory contains 4 parameters file. Once the choice made, a series of tests (`test_#`, where `#` is the test number) consistent with the machine architecture and preprocessing commands are performed. The subsequent results are stored in the corresponding directories in the directory `out_ref`. These will be used as a reference to be compared with when running the textcolorgreen`testsuite_cmp` script. Note that these tests are performed on 16 processors and run online, meaning that they do not require to be submitted through batch script. This choice has been made as the number of steps is very limited (100 time steps). After running this first script, the

2. In a second step, one needs to run `testsuite_cmp` script **ONCE THE SOURCE CODE HAS BEEN MODIFIED**. Obviously it is possible not to alter the code from its base state: the comparison won't result in any difference. This test allows to compare its output to the one from the reference testsuite. The results from this execution are gathered in the directory `out_test`.

3. Once the two previous steps are carried out, the comparison takes place between the reference and modified version of the code. This process parses the standard output of the code for each previously mentioned tests (respectively in `out_ref` and `out_test`) following variables (as shown in the standard output): `U max`, `V max`, `W max`, `Div U* max`, `Div U* Moyy`). The computation of the error is based on the $\|\cdot\|_2$ norm between the reference and the computed data from `testsuite_cmp` script.

Once all the tests are completed a summary file called `summary_test.dat` is written in the considered problem directory. For example for the standard (no control) case, the tests based on `parameters_1` and `parameters_2` are as follows:

- parameters_1: rotating channel flow to test the triggering of the turbulent state. ($n_x$,$n_y$,$n_z$)=(128,129,128), (p_row,p_col)=(4,4), (xlx,yly,zlz)=($4\pi$,2,$4\pi/3$), dt=0.0025, ifirst=1, ilast=100 rossby=2/18, no wall control. Note that this test is just set here to have an idea for the order of magnitude to obtain the initialisation of the rotating channel flow. Indeed, as this process uses a random number generator based on the wall clock of the system, comparing this reference data with the data from the testsuite_cmp script will result in differences even if the code has not been modified at all. Therefore it is actually the second test that should either give 0 or be within the machine precision limit in the norm computation of the difference of the

- parameters_2: channel flow starting from a previous checkpoint file (called sauve_stat in the directory external/checkpoint/nx_128_ny_129_nz_128) that has reached a turbulent state. ($n_x$,$n_y$,$n_z$)=(128,129,128), (p_row,p_col)=(4,4), (xlx,yly,zlz)=($4\pi$,2,$4\pi/3$), dt=0.0025, ifirst=300001, ilast=300100, no wall control.

## A.9 Directory tools

This directory contains the program calc_beta.f90 for computing the stretching parameter $\beta$. The only parameters that are required in this program are the Reynolds number based on the maximum Poiseuille velocity and the number of grid points in the wall normal direction if one wants to use this file separately. However, as mentioned in §A.6.3, the file scr.sh automatically computes this parameter and settles it in the inc3d.prm file.

Next a directory called in grad_vort_gen enables to compute the terms that enter in the velocity gradient tensor and the vorticity. This is considered from a practical viewpoint. Indeed, when running a simulation by default the primitive variables are given as output files. In the default setting of the parameter file, the derivatives and vorticity are not computed. This allows to save time (for production runs) and disk space. In the current version of the code, by setting the parameter with_dervel and with_vort to .true. it is possible to compute these quantities during the simulation. Note that the overhead for computing these files can be checked at the end of the simulation in the summary of the code performance as described in §A.7.12. The aforementioned directory allows to compute these quantities from the primitive variables of a previously completed simulation. The structure is similar to the directories in the base directory for running a physical problem. The parameters and directories from which the velocity fields are read have therefore to be consistently set. The computation of the component of the velocity gradient tensor elements then allows the computation of the invariants for flow visualisation purpose.

Another directory directory can be found in `tools`. It is called `parallel_diag` and enables to compute the statistics from the code output. Theses statistics are computed in parallel using the same domain decomposition strategy as used in Incompact3d. With this tool, even if velocity, vorticity, Reynolds stresses and energy balance terms are all computed. The advantage resides in the fact that even if the code is not set to output the vorticity and velocity gradient tensor, by setting the relevant flags in the parameter file (`with_dervel` and `with_vort` set to `.true.`), it is still possible to obtain their statistics as these are computed online in this diagnostics tool. The implementation is carried out in such a way that it is also possible to obtain statistics when discs are implemented. In this case, the discs flow statistics are also computed along with the turbulent part relative to the latter flow. For further details on the flow decomposition in the case with discs one can refer to Ricco & Hahn (2013).

In summary the vorticity and velocity gradient tensor elements can be accessed through:

- Direct computation during a simulation

- If not computed during the simulation, these can be obtained from the primitive variables using the implementation in `tools/grad_vort_gen`

- Whether they are computed or not during the simulation, their statistics can be computed using the implementation in `tools/parallel_diag`

The remaining files are four bash scripts. These are intended for renaming and deleting within a range output `dat` files. The script `rundiv` allows to have the plot of the mean and maximum divergence from the standard output file (which is written at screen in interactive mode). These plots can be checked during the simulation.

## A.10 Frequently Asked Questions

- What value should be set for the Rossby number?
  One should keep the default value in the inc3d.prm file that is 2/18. It should be sufficient to kick off the turbulent channel.

- Where do I get the latest version of the code?
  At `http://www.incompact3d.com/`, in the section Download. Alternatively, one can directly type the following commands in the terminal:

  1. `wget http://www.incompact3d.com/uploads/5/8/7/2/58724623/channel.tar`
  2. `tar -xvf channel.tar`

  In your terminal you should be able to checkout the latest version of the Incompact3d code. The subversion control version client can be installed in a Linux system (Ubuntu here mentioned) by typing the following command in the terminal: `sudo apt-get install subversion` Note that at this time the code is still available on the aforementioned Google Code page, but the website is meant to go down in the course of the year 2015 and shut down later on. Further information will be available on the new website dedicated to the Incompact3d  code.

## A.11 Style guide

1. Style for writing modules

   (a) Modules should be used wherever things belong together.

   (b) A module encapsulates a number of methods and data structures. The usage of modular variables is allowed if only one instance is used throughout the whole code. Otherwise, a new type containing those structures and type should be declared.

   (c) In principle the data in a module is private to the module (the same applies even stronger to the data of a type). All modification of the data should be done by methods of the module.

   (d) The usage of `implicit none` is mandatory.

(e) The structure of the module should be:

```
module matrix_module
implicit none

type matrix
...
end type

interface initialize_matrix_module
  module procedure mp_initialize_module
end interface

interface finalize_matrix_module
  module procedure mp_finalize_module
end interface

interface initialize
  module procedure mp_initialize_matrix
end interface
...

private :: mp_initialize_matrix, mp_initialize_module

integer, private :: modular variables (if any)

contains
  subroutine mp_initialize_module(...)
  end subroutine

  subroutine mp_finalize_module(...)
  end subroutine
end module matrix_module
```

2. Style for writing global functions and subroutines (ideal case)

    (a) Functions should use only arguments and local variables, the usage of global variables should be reduced to an absolute minimum.

    (b) Functions should be short, and each function has ideally exactly one defined purpose and no side effects. It only changes the output arguments of the function. The only side effect allowed in some cases is the influence on private module variables.

    (c) The first block in a function contains the passed argument type definitions. Then separated the local variables are defined.

    (d) A function should be testable separately to make sure it fulfils its purpose correctly.

    (e) Find describing names, starting with a verb in lower case and the objects it is acting on separated by underscores, `calculate_rhs`, `count_number_of_rows` should be considered as an ideal, but names may become too long.

3. Before committing any changes to the `subversion` repository (for developers only), the testsuite has to be run and all tests need to be successful. This has to be done on as many machines as possible.

4. Do not use `double precision` or `double complex` types in type declaration. Always use `real` and `complex`. These types are extended to double precision in the machine dependent makefile. In the same way use only `MPI_COMPLEX_TYPE` and `MPI_REAL_TYPE` for the MPI routines, as these names (in upper case) are replaced by the actual types in the preprocessing phase of compilation.

# Appendix B

# Implementation and validation of the hydrophobic boundary conditions

This appendix first presents how the wall boundary conditions are modified to account for the hydrophobicity. Secondly, the numerical calculations performed to validate the implementation of the hydrophobic-surface boundary conditions are described.

In the code, 6th-order compact finite difference schemes are used for the spatial derivatives in the convective and diffusive terms. These have to be modified at the boundaries, according to the specified problem. For periodic or free-slip boundary conditions, the schemes, given by equations (13) and (14) in Laizet & Lamballais (2009), can be used at all nodes by using ghost nodes. Dirichlet boundary conditions are instead implemented through single-sided formulations, which are 3rd- or 4th-order. This also occurs in the hydrophobic case, for which the boundary conditions for the streamwise velocity are of mixed type, i.e. they involve the wall velocity and its wall-normal gradient (Robin type in the constant slip length case, Behr (2004)), as given in equation (5.1).

The following formulas pertain to the bottom wall and the derivation for the top wall is analogous. The simplest way to implement the constant-slip-length case of (5.1) numerically ($a = 0$) is to use a first-order scheme:

$$U_0 = b\frac{U_1 - U_0}{y_1 - y_0}, \tag{B.1}$$

where the subscript 0 and 1 denote quantities at the wall and at the node adjacent to the wall, respectively. Expression (B.1) is solved for $U_0$:

$$U_0 = \frac{bU_1}{b + y_1 - y_0}. \tag{B.2}$$

A three-point finite-difference formula is also used to implement the boundary conditions. On a uniform grid, this scheme is second-order accurate, but when a non-uniform grid is considered, lower accuracy ensues (Bowen & Smith, 2005). If $(y_0,f(y_0))$, $(y_1,f(y_1))$ and $(y_2,f(y_2))$ are the interpolating points, the three-point formula for the derivative of a function $f$ at the node $y_j$ is given by:

$$\begin{aligned} f'(y_j) = \quad f \quad & (y_0)\frac{2y_j - y_1 - y_2}{(y_0 - y_1)(y_0 - y_2)} + f(y_1)\frac{2y_j - y_0 - y_2}{(y_1 - y_0)(y_1 - y_2)} \\ + \quad & f(y_2)\frac{2y_j - y_0 - y_1}{(y_2 - y_0)(y_2 - y_1)} + g(\Delta y_j), \end{aligned} \tag{B.3}$$

where $g$ is a function representing the higher-order terms (Bowen & Smith, 2005). Applying (B.3) at the boundary, in the constant-slip-length case, the formula reads:

$$U_0 = \frac{b(\alpha_1 U_1 + \alpha_2 U_2)}{1 - b\alpha_0}, \tag{B.4}$$

where $U_2$ denotes the velocity at the second grid point from the wall, and

$$\alpha_0 = \frac{2y_0 - y_1 - y_2}{(y_0 - y_1)(y_0 - y_2)}, \alpha_1 = \frac{y_0 - y_2}{(y_1 - y_0)(y_1 - y_2)}, \alpha_2 = \frac{y_0 - y_1}{(y_2 - y_0)(y_2 - y_1)}. \tag{B.5}$$

In the shear-dependent slip-length case, the first-order formulation leads to a second-order polynomial:

$$a(U_1 - U_0)^2 + \Delta y_{\min}(b + \Delta y_{\min})(U_1 - U_0) + (\Delta y_{\min})^2 U_1 = 0, \tag{B.6}$$

which is is solved by setting $\zeta = U_1 - U_0$. The discriminant is:

$$D = (\Delta y_{\min})^2 \left[(b + \Delta y_{\min})^2 + 4aU_1\right]. \tag{B.7}$$

If $D > 0$, two distinct real solutions exist:

$$\zeta_{1,2} = \frac{\Delta y_{\min}\,(b + \Delta y_{\min})}{2a}\left[-1 \pm \sqrt{1 + \frac{4aU_1}{(b + \Delta y_{\min})^2}}\right]. \tag{B.8}$$

In order to choose the correct solution, the constant-slip-length solution (B.2) must be recovered in the limit $a \to 0$. The square-root term in (B.8) is Taylor-expanded for small $a$,

$$\sqrt{1 + \frac{4aU_1}{(b + \Delta y_{\min})^2}} = 1 + \frac{2aU_1}{(b + \Delta y_{\min})^2} + \mathcal{O}(a^2), \tag{B.9}$$

which proves that the solution with the minus sign in front of the square-root term in (B.8) is ruled out because $\zeta$ diverges as $a \to 0$. The limit of the other solution coincides with (B.2).

| **Case** | $L_x$ | $L_z$ | $\Delta x^{+0}$ | $\Delta y^{+0}_{\text{min}}$ | $\Delta z^{+0}$ | $\mathcal{R}(\%)$ **Two-point** | $\mathcal{R}(\%)$ **Three-point** |
|---|---|---|---|---|---|---|---|
| MK4 | 7 | 3.5 | 10 | 0.3 | 5 | 29 | - |
| Resolution 1 | $4\pi$ | $4\pi/3$ | 17 | 1.0 | 5.9 | 28.7 | 28.8 |
| Resolution 2 | $4\pi$ | $4\pi/3$ | 8.5 | 1.0 | 2.95 | 28.7 | 28.5 |
| Resolution 3 | $4\pi$ | $4\pi/3$ | 8.5 | 0.43 | 2.95 | 28.5 | 28.6 |

Table B.1: Drag reduction $\mathcal{R}(\%)$ from Min & Kim (2004) (MK4) and computed through Incompact3d with different grid resolutions and with the two numerical schemes for the constant-slip-length case, $b = 0.02$, with slip along $x$ only.

The shear-dependent three-point formula is found as follows. Use of (B.3), (B.5), and equation (5.1) leads to:

$$U_0 = a(\alpha_0 U_0 + \alpha_1 U_1 + \alpha_2 U_2)^2 + b(\alpha_0 U_0 + \alpha_1 U_1 + \alpha_2 U_2), \qquad \text{(B.10)}$$

which must be solved for $U_0$. Setting $X = \alpha_0 U_0 + \alpha_1 U_1 + \alpha_2 U_2$, (B.10) becomes

$$aX^2 + \left(b - \alpha_0^{-1}\right) X + K = 0, \qquad \text{(B.11)}$$

where $K = (\alpha_1 U_1 + \alpha_2 U_2)/\alpha_0$. As for the first-order case, the correct solution is obtained by Taylor expansion,

$$X_1 = \frac{b\alpha_0 - 1}{2a\alpha_0}\left(1 + \sqrt{1 - \frac{4aK\alpha_0^2}{(b\alpha_0 - 1)^2}}\right) = -\frac{K\alpha_0}{(b\alpha_0 - 1)} + \mathcal{O}(a^2). \qquad \text{(B.12)}$$

By replacing $K$ and $X$ given by (B.12) into (B.12), (B.10) becomes (B.4).

The validation results for the constant-slip-length case with slip along $x$ only are compared with Min & Kim (2004)'s in table B.1. Different grid resolutions and the two numerical schemes for the boundary conditions have been used. The calculations agree with Min & Kim (2004)'s and the grid independence is proved.

The code is further tested for slip along $x$ only, slip along $z$ only, and slip along both directions, and with different slip length values, $b \in \{0.005, 0.002, 0.01, 0.02\}$, using the two-point formulation of the boundary conditions. The three-point formulation was also tested and the results did not show significant alteration in the $\mathcal{R}$ values, the maximum deviation being 0.5%. As in Min & Kim (2004), the case with slip only along $x$ yields the maximum $\mathcal{R}$ value. The Min & Kim (2004)'s data and our data computed through Incompact3d again show good agreement.

|  | $x$ **slip only** |  |  | $z$ **slip only** |  |  | $x$ **and** $z$ **slip** |  |  |
|---|---|---|---|---|---|---|---|---|---|
| $b$ | MK4 | Incompact3d | FIK | MK4 | Incompact3d | FIK | MK4 | Incompact3d | FIK |
| 0.002 | 5 | 4.9 | 4.9 | -3 | -2.8 | -2.8 | 1 | 1.1 | 1.2 |
| 0.005 | 10 | 10.0 | 10.0 | -8 | -7.9 | -7.9 | 3 | 3.1 | 3.1 |
| 0.01 | 18 | 17.7 | 17.6 | -16 | -15.8 | -15.9 | 8 | 8.1 | 8.2 |
| 0.02 | 29 | 28.7 | 28.9 | -26 | -25.7 | -25.8 | 17 | 17.2 | 17.2 |

Table B.2: Drag reduction $\mathcal{R}(\%)$ from Min & Kim (2004) (MK4), and computed through Incompact3d and through the FIK identity (B.13).

Table B.2 also shows that the skin-friction coefficient computed by space- and time-averaging the wall-shear stress matches the value computed through the Fukagata-Iwamoto-Kasagi (FIK) identity, which in the case of a hydrophobic surface reads:

$$C_f = \frac{6}{\mathcal{U}_b R_p} - \frac{6}{\mathcal{U}_b^2} \int_0^1 (1-y) uv_{rey} \mathrm{d}y - \frac{6\,\mathcal{U}(0)}{R_p \mathcal{U}_b^2}. \tag{B.13}$$

The identity (B.13) shows that a positive mean wall-slip velocity contributes favourably to a reduction of wall-shear stress and that the drag reduction is linearly proportional to the wall-slip velocity.

In the constant-slip case ($a$=0), the space- and time-averaged wall velocity $\mathcal{U}(0)$ has also been verified to agree with the following

$$\mathcal{U}(0) = \frac{3b}{3b+1} \left[ \mathcal{U}_b - R_p \int_0^1 (1-y) uv_{rey} \mathrm{d}y \right], \tag{B.14}$$

which is found by averaging the wall boundary conditions (5.1) with $a = 0$, and by substitution of (2.13) into (B.13). As expected, $\lim\limits_{b\to\infty} U(0) = \mathcal{U}_b$, i.e. the laminar plug-flow found in §5.2.1 is recovered because the Reynolds stresses vanish slowly as the turbulent production decreases with the mean-flow wall-normal gradient, as shown by Busse & Sandham (2012). The checks in figures B.1 and B.2show that the mean streamwise velocity profile $\mathcal{U}$ and the rms of all the velocity components agree with Min & Kim (2004)'s.

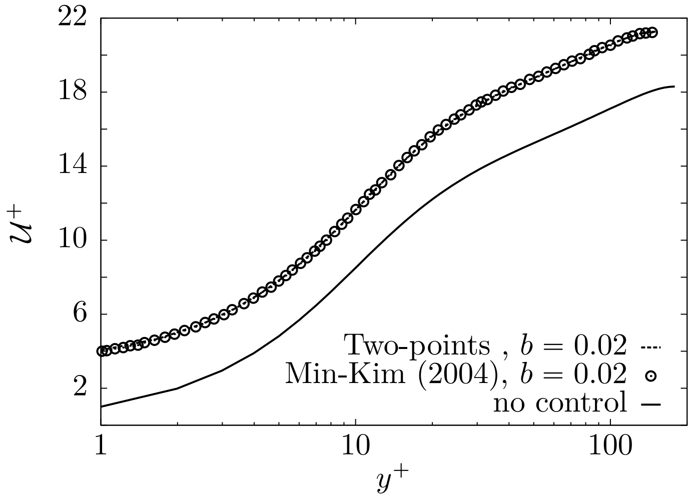


Figure B.1: Profiles of mean streamwise velocity.

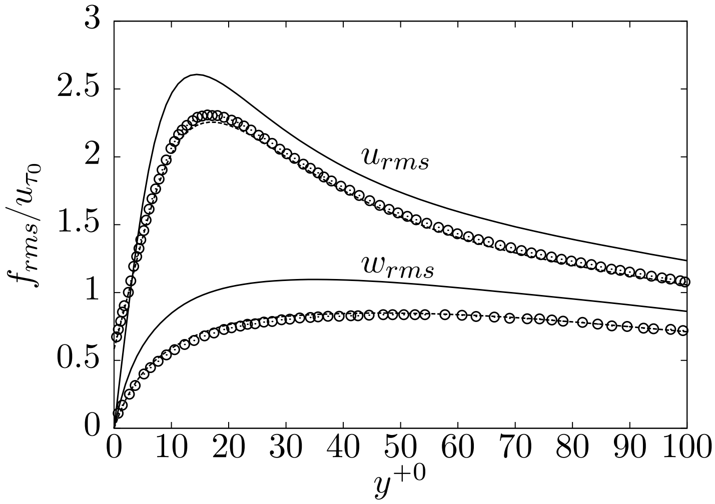


Figure B.2: Rms profiles for the streamwise and spanwise perturbation velocities.

# Bibliography

Airbus 2014 Global market forecast. `http://www.airbus.com/company/market/forecast/`.

Akhavan, R., Jung, W. & Mangiavacchi, N. 1993 Control of wall turbulence by high frequency spanwise oscillations. *AIAA Paper* **93-3282**.

Aljallis, E., Sarshar, M.A., Datla, R., Sikka, V., Jones, A. & Choi, C.-H. 2013 Experimental study of skin friction drag reduction on superhydrophobic flat plates in high reynolds number boundary layer flow. *Phys. Fluids* **25** (2), 025103.

Ball, P. 1999 Engineering shark skin and other solutions. *Nature* **400** (6744), 507–509.

Balogh, A., Aamo, O.M. & Krstic, M. 2005 Optimal mixing enhancement in 3-d pipe flow. *IEEE Trans. Control Syst. Tech.* **13** (1), 27–41.

Balogh, A., Liu, W.J. & Krstic, M. 2001 Stability enhancement by boundary control in 2-D channel flow. *IEEE Trans. Aut. Control* **46** (11), 1696–1711.

Baron, A. & Quadrio, M. 1996 Turbulent drag reduction by spanwise wall oscillations. *Appl. Sc. Res.* **55**, 311–326.

Bazant, M.Z. & Vinogradova, O.I. 2008 Tensorial hydrodynamic slip. *J. Fluid Mech.* **613**, 125–134.

Behr, M. 2004 On the application of slip boundary condition on curved boundaries. *Int. J. Num. Meth. Fluids* **45** (1), 43–51.

Bell, D.J., Lu, T.J., Fleck, N.A. & Spearing, S.M. 2005 Mems actuators and sensors: observations on their performance and selection for purpose. *J. Micromech. Microeng* **15**, S153–64.

Bernard, P.S., Thomas, J.M. & Handler, R.A. 1993*a* Vortex dynamics and the production of Reynolds stress. *J. Fluid Mech.* **253**, 385–419.

Bernard, P.S., Thomas, J.M. & Handler, R. 1993*b* Vortex dynamics in near wall turbulence. *Near-wall turbulent flows* pp. 43–52.

Bernardini, M., Pirozzoli, S. & Orlandi, P. 2014 Velocity statistics in turbulent channel flow up to $Re_\tau = 4,000$. *J. Fluid Mech.* **742**, 171–191.

Betchov, R. 1956 An inequality concerning the production of vorticity in isotropic turbulence. *J. Fluid Mech.* **1**, 497–504.

Bidkar, R. A., Leblanc, L., Kulkarni, A. J., Bahadur, V., Ceccio, S. L. & Perlin, M. 2014 Skin-friction drag reduction in the turbulent regime using random-textured hydrophobic surfaces. *Physics of Fluids (1994-present)* **26** (8), 085108.

Bonaccurso, E., Butt, H-J. & Craig, V.S.J. 2003 Surface roughness and hydrodynamic boundary slip of a Newtonian fluid in a completely wetting system. *Phys. Rev. Lett.* **90**, 144501.

Bowen, M.K. & Smith, R. 2005 Derivative formulae and errors for non-uniformly spaced points. *Proc. R. Soc. A: Math., Phys. and Eng. Sci.* **461**, 1975–1997.

Busse, A., Sandham, N.D., McHale, G. & Newton, M.I. 2013 Change in drag, apparent slip and optimum air layer thickness for laminar flow over an idealised superhydrophobic surface. *J. Fluid Mech.* **727**, 488–508.

Busse, A. & Sandham, N. D. 2012 Influence of an anisotropic slip-length boundary condition on turbulent channel flow. *Phys. Fluids* **24** (5), 055111.

Buxton, O.R.H., Laizet, S. & Ganapathisubramani, B. 2011 The interaction between strain-rate and rotation in shear flow turbulence from inertial range to dissipative length scales. *Phys. Fluids* **23**, 061704.

Cain, A.B., Ferziger, J.H. & Reynolds, W.C. 1984 Discrete orthogonal function expansions for non-uniform grids using the fast Fourier transform. *J. Comp. Phys.* **56**, 272–286.

Canuto, C., Hussaini, M.Y., Quarteroni, A. & Zang, T.A. 2007 *Spectral methods: evolution to complex geometries and applications to fluid dynamics.* Springer.

Choi, C.H. & Kim, C.J. 2006 Large slip of aqueous liquid flow over a nanoengineered superhydrophobic surface. *Phys. Rev. Lett.* **96** (066001).

Choi, C.-H., Westin, K.J.A. & Breuer, K.S. 2003 Apparent slip flows in hydrophilic and hydrophobic microchannels. *Phys. Fluids* **15**, 2897.

Choi, H. & Moin, P. 1994 Effects of the computational time step on numerical solutions of turbulent flow. *J. Comp. Phys.* **113** (1), 1–4.

Choi, H., Moin, P. & Kim, J. 1993 Direct numerical simulation of turbulent flow over riblets. *J. Fluid Mech.* **255**, 503–539.

Choi, H., Moin, P. & Kim, J. 1994 Active turbulence control for drag reduction in wall-bounded flows. *J. Fluid Mech.* **262**, 75–110.

Choi, K.-S., Jukes, T.N., Whalley, R.D., Feng, L., Wang, J., Matsunuma, T. & Segawa, T. 2014 Plasma virtual actuators for flow control. *J. Flow Control, Meas. & Visual.* **3** (01), 22.

Chong, M.S., Soria, J., Perry, A.E., Chacin, J., Cantwell, N.J. & Na, Y. 1998 Turbulence structures of wall-bounded shear flows found using DNS data. *J. Fluid Mech.* **357**, 225–247.

Chorin, A.J. 1968 Numerical solution of the Navier-Stokes equations. *Math. comput.* **22** (104), 745–762.

Chung, Yongmann M & Talha, Tariq 2011 Effectiveness of active flow control for turbulent skin friction drag reduction. *Phys. Fluids (1994-present)* **23** (2), 025102.

Churaev, N.V., Sobolev, V.D. & Somov, A.N. 1984 Slippage of liquids over lyophobic solid surfaces. *J. Coll. Int. Sci.* **97**, 574–581.

Corino, E. R. & Brodkey, R. S. 1977 A visual investigation of the wall region in turbulent flow. *J. Fluid Mech.* **37**, 1–30.

Craig, V.S.J., Neto, C. & Williams, D.R.M. 2001 Shear-dependent boundary slip in an aqueous Newtonian liquid. *Phys. Rev. Lett.* **87**, 054504.

Cucitore, R., Quadrio, M. & Baron, A. 1999 On the effectiveness and limitations of local criteria for the identification of a vortex. *Eur. J. Mech. B / Fluids* **18** (2), 261–282.

Daniello, R.J., Waterhouse, N.E. & Rothstein, J.P. 2009 Drag reduction in turbulent flows over superhydrophobic surfaces. *Phys. Fluids* **21** (085103).

Davis, A.M.J. & Lauga, E. 2009 Geometric transition in friction for flow over a bubble mattress. *Phys. Fluids* **21** (1), 011701.

Dean, R. B. 1978 Reynolds number dependence of the skin friction and other bulk flow variables in two-dimensional rectangular duct flow. *J. Fluids Eng.* **100**, 215–223.

Deville, M.O., Fischer, P.F. & E.H., Mund 2002 *High-order Methods for Incompressible Fluid Flow*. Cambridge University Press.

Dhanak, M.R. & Si, C. 1999 On reduction of turbulent wall friction through spanwise oscillations. *J. Fluid Mech.* **383**, 175–195.

Duggleby, A., Ball, K. S. & Paul, M.R. 2007 The effect of spanwise wall oscillation on turbulent pipe flow structures resulting in drag reduction. *Phys. Fluids* **19** (125107).

Dunn, W. & Shultis, J. 2011 *Exploring Monte Carlo Methods*. Elsevier.

Fasel, H 1976 Investigation of the stability of boundary layers by a finite-difference model of the navier—stokes equations. *J. Fluid Mech.* **78** (02), 355–383.

Fedorov, A.V. & Khokhlov, A. P. 2001 Prehistory of instability in a hypersonic boundary layer. *Th. Comp. Fluid Dyn.* **14**, 359–375.

Fedorov, A., Shiplyuk, A., Maslov, A., Burov, E. & Malmuth, N. 2003 Stabilization of a hypersonic boundary layer using an ultrasonically absorptive coating. *J. Fluid Mech.* **479**, 99–124.

Feng, L., Li, S., Li, Y., Li, H., Zhang, L., Zhai, J., Song, Y., Liu, B., Jiang, L. & Zhu, D. 2002 Super-hydrophobic surfaces: from natural to artificial. *Advanced Materials* **14** (24), 1857–1860.

Frigo, M. & Johnson, S. G. 2005 The design and implementation of FFTW3. *Proceedings of the IEEE* **93**, 216–231.

Fukagata, K., Iwamoto, K. & Kasagi, N. 2002 Contribution of Reynolds stress distribution to the skin friction in wall-bounded flows. *Phys. Fluids* **14** (11), 73–76.

Fukagata, K., Kasagi, N. & Koumoutsakos, P. 2006 A theoretical prediction of friction drag reduction in turbulent flow by superhydrophobic surfaces. *Phys. Fluids* **18**, 051703.

Gad-el-Hak, M. 2001 *The MEMS handbook*. CRC press.

Garcia-Mayoral, R., Seo, J. & Mani, A. 2014 Dynamics of gas-liquid interfaces in turbulent flows over superhydrophobic surfaces. In *Proc. CTR Summer Program*, p. 295.

Gibson, J.F. 2006*a* Channelflow: a spectral Navier-Stokes simulator in C++. *http://www.channelflow.org/*.

Gibson, J.F. 2006*b* Channelflow: a spectral Navier-Stokes simulator in C++. `http://www.channelflow.org/`.

Govardhan, R.N., Srinivas, G.S., Asthana, A. & Bobji, M.S. 2009 Time dependence of effective slip on textured hydrophobic surfaces. *Phys. Fluids* **21** (5), 052001.

Gresho, P.M. & Chan, S.T. 1990 On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 2: Implementation. *Int. J. Num. Meth. Fluids* **11** (5), 621–659.

Grundestam, O, Wallin, S & Johansson, A V 2008 Direct numerical simulations of rotating turbulent channel flow. *J. Fluid Mech.* **598**, 177–199.

Hahn, S., Je, J. & Choi, H. 2002 Direct numerical simulation of turbulent channel flow with permeable walls. *J. Fluid Mech.* **450**, 259–285.

Gad-el Hak, M. 1994 Interactive control of turbulent boundary layers-a futuristic overview. *AIAA journal* **32** (9), 1753–1765.

Gad-el Hak, M. 2000 *Flow control - Passive, Active and Reactive Flow Management*. Cambridge University Press.

Hamilton, J.M., Kim, J. & Waleffe, F. 1995 Regeneration mechanism of near-wall turbulence structures. *J. Fluid Mech.* **287**, 317–348.

Hamlington, P.E., Schumacher, J. & Dahm, W.J.A. 2008 Direct assessment of vorticity alignment with local and nonlocal strain rates in turbulent flows. *Phys. Fluids* **20**, 111703.

Hammond, E.P., Bewley, T.R. & Moin, P. 1998 Observed mechanisms for turbulence attenuation and enhancement in opposition-controlled wall-bounded flows. *Phys. Fluids* **10** (9), 2421–2423.

Harlow, F. H. & Welch, J. E. 1965 Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. of Fluids* **8** (12), 2182.

Hasegawa, Y., Frohnapfel, B. & Kasagi, N. 2011 Effects of spatially varying slip length on friction drag reduction in wall turbulence. In *J. Phys.: Conf. Series*, , vol. 318, p. 022028. IOP Publ.

Hinze, J.O. 1975 *Turbulence*. McGraw Hill, Inc. – Second Edition.

Hodge, W. V. D. 1989 *The theory and applications of harmonic integrals*. CUP Archive.

Hoyas, S. & Jiménez, J. 2006 Scaling of the velocity fluctuations in turbulent channels up to Re$_\tau$=2003. *Phys. Fluids* **18**, 011702.

Hunt, J.C.R., Wray, A.A. & Moin, P. 1988 Eddies, streams, and convergence zones in turbulent flows. *Center for Turbulence Research CTR-S 88* .

Ikeda, K. 2007 Trends in research on turbulence control aiming at reducing friction drag. *Science and Technology Trends* **22**, 98–107.

IPCC 2014 Climate change 2014 synthesis report. `http://www.ipcc.ch/pdf/assessment-report/ar5/syr/SYR_AR5_FINAL_full.pdf`.

J., Kim 2011 Physics and control of wall turbulence for drag reduction. *Phil. Trans. R. Soc.* **369**, 1396–1411.

Jacobi, I., Wexler, J.S. & Stone, H.A. 2015 Overflow cascades in liquid-infused substrates. *Phys. Fluids* **27** (8), 082101.

Jelly, T.O., Jung, S.Y. & Zaki, T.A. 2014 Turbulence and skin friction modification in channel flow with streamwise-aligned superhydrophobic surface texture. *Phys. Fluids* **26** (9), 095102.

Jeong, J. & Hussain, F. 1995 On the identification of a vortex. *J. Fluid Mech.* **285**, 69–94.

Jeong, J., Hussain, F., Schoppa, W. & Kim, J. 1997 Coherent structures near the wall in a turbulent channel flow. *J. Fluid Mech.* **332**, 185–214.

Jiménez, J. 1994 On the structure and control of near wall turbulence. *Phys. Fluids* **6**, 944.

Jiménez, J. & Moin, P. 1991 The minimal flow unit in near-wall turbulence. *J. Fluid Mech.* **225**, 213–240.

Jimenez, J., Uhlmann, M., Pinelli, A. & Kawahara, G. 2001 Turbulent shear flow over active and passive porous surfaces. *J. Fluid Mech.* **442**, 89–117.

Jung, T., Choi, H. & Kim, J. 2016 Effects of the air layer of an idealized superhydrophobic surface on the slip length and skin-friction drag. *J. Fluid Mech.* **790**, R1.

Jung, W.J., Mangiavacchi, N. & Akhavan, R. 1992 Suppression of turbulence in wall-bounded flows by high-frequency spanwise oscillations. *Phys. Fluids A* **4** (8), 1605–1607.

Keefe, L. 1997 A normal vorticity actuator for near-wall modification of turbulent shear flows. *AIAA Paper* **97-0547**.

Keefe, L. 1998 Method and apparatus for reducing the drag of flows over surfaces. *United States Patent* **5,803,409**.

Kim, J. & Moin, P. 1985 Application of a fractional-step method to incompressible Navier-Stokes equations. *J. Comp. Phys.* **59**, 308–323.

Kim, J., Moin, P. & Moser, R. 1987*a* Turbulence statistics in fully developed channel flow at low Reynolds number. *J. Fluid Mech.* **177**, 133–166.

Kim, J., Moin, P. & Moser, R. 1987*b* Turbulence statistics in fully developed channel flow at low Reynolds number. *J. Fluid Mech.* **177**, 133–166.

Kleiser, L. & Schumann, U. 1980 Treatment of incompressibility and boundary conditions in 3-D numerical spectral simulations of plane channel flows. In *Proc. 3rd GAMM*

*Conf. Numerical Methods in Fluid Mechanics* (ed. E. Hirschel), pp. 165–173. GAMM, Vieweg.

Kline, S.J., Reynolds, W.C., Schraub, F.A. & Runstadler, P.W. 1967 The structure of turbulent boundary layers. *J. Fluid Mech.* **30**, 741.

Koumoutsakos, P., Bewley, T. R., Hammond, E. P. & Moin, . 1997 Feedback algorithms for turbulence control–some recent developments. *AIAA paper* (97-2008).

Kravchenko, A.G., Choi, H. & Moin, P. 1993 On the relation of near-wall streamwise vortices to wall skin friction in turbulent boundary layers. *Phys. Fluids* **5**, 3307–3309.

Kravchenko, A. G. & Moin, P. 1997 On the effect of numerical errors in large eddy simulations of turbulent flows. *J. Comp. Phys.* **131**, 310–322.

Laadhari, F., Skandaji, L. & Morel, R. 1994 Turbulence reduction in a boundary layer by local spanwise oscillating surface. *Phys. Fluids* **6** (10), 3218–3220.

Laizet, S. & Lamballais, E. 2009 High-order compact schemes for incompressible flows: A simple and efficient method with quasi-spectral accuracy. *J. Comp. Phys.* **228**, 5989–6015.

Laizet, S, Lamballais, E & Vassilicos, J.C. 2010 A numerical strategy to combine high-order schemes, complex geometry and parallel computing for high resolution dns of fractal generated turbulence. *Computers & fluids* **39** (3), 471–484.

Laizet, S. & Li, N. 2011 Incompact3d: A powerful tool to tackle turbulence problems with up to $\mathcal{O}(10^5)$ computational cores. *Int. J. Num. Meth. Fluids* **67**, 1735–1757.

Laizet, S. & Vassilicos, J.C. 2011 Dns of fractal-generated turbulence. *Flow, Turbulence and Combustion* **87** (4), 673–705.

Lauga, E. & Stone, H.A. 2003 Effective slip in pressure-driven Stokes flow. *J. Fluid Mech.* **489**, 55–77.

Lee, J., Jelly, T.O. & Zaki, T.A. 2015 Effect of reynolds number on turbulent drag reduction by superhydrophobic surface textures. *Flow, Turbulence and Combustion* **95** (2-3), 277–300.

Lee, M. & Moser, R. D. 2015 Direct numerical simulation of turbulent channel flow up to $\mathrm{Re}_\tau \approx 5200$. *J. Fluid Mech.* **774**, 395–415.

Lele, S. K. 1992 Compact finite difference schemes with spectral-like resolution. *J. Comp. Phys.* **103**, 16–42.

Leray, J. 1933 Etudes de diverses équations intégrales non linéaires et de quelques problèmes que pose l'hydrodynamique. *Thèses françaises de l'entre-deux-guerres* **142**, 1–82.

Li, N. 2012 2decomp&fft. `http://www.2decomp.org/`.

list, Top500 2016 Top 500. `http://www.top500.org/`.

Martell, M.B., Perot, J.B. & Rothstein, J.P. 2009 Direct numerical simulations of turbulent flows over superhydrophobic surfaces. *J. Fluid Mech.* **620**, 31–41.

Martell, M.B., Rothstein, J.P. & Perot, J.B. 2010 An analysis of superhydrophobic turbulent drag reduction mechanisms using direct numerical simulation. *Phys. Fluids* **22**, 065102.

Marusic, I., McKeon, B.J., Monkewitz, P.A., Nagib, H.M., Smits, A.J. & Sreenivasan, K.R. 2010 Wall-bounded turbulent flows at high Reynolds numbers: Recent advances and key issues. *Phys. Fluids* **22** (065103).

Min, T. & Kim, J. 2004 Effects of hydrophobic surface on skin-friction drag. *Phys. Fluids* **16** (7).

Min, T. & Kim, J. 2005 Effects of hydrophobic surface on stability and transition. *Phys. Fluids* **17** (108106).

Moin, P. & Mahesh, K. 1998 Direct numerical simulation: a tool in turbulence research. *Annu. Rev. Fluid Mech.* **30**, 539–578.

Morinishi, Y., Lund, T.S., Vasilyev, O.V. & Moin, P. 1998 Fully conservative higher order finite difference schemes for incompressible flow. *J. Comp. Phys.* **143** (1), 90–124.

Moser 2007 Dns data for turbulent channel flow. `http://turbulence.ices.utexas.edu/MKM_1999.html`.

Moser, R., Kim, J. & Mansour, N.N. 1999 Direct numerical simulation of turbulent channel flow up to $Re_\theta = 590$. *Phys. Fluids* **11** (4), 943–945.

Moser, R. D. & Moin, P. 1984 Direct numerical simulation of curved turbulent channel flow. *NASA STI/Recon Technical Report N* **85**, 11316.

Navier, C.L.M.H. 1823 Mémoire sur les lois du mouvement des fluides. *Mémoires de l'Académie Royale des Sciences de l'Institut de France* .

Ou, J., Perot, B. & Rothstein, J.P. 2004 Laminar drag reduction in microchannels using ultrahydrophobic surfaces. *Phys. Fluids* **16**, 4635–4643.

Ou, J. & Rothstein, J.P. 2005 Direct velocity measurements of the flow past drag-reducing ultrahydrophobic surfaces. *Phys. Fluids* **17** (10), 103606–103606.

Pamiès, Mathieu, Garnier, Eric, Merlen, Alain & Sagaut, Pierre 2007 Response of a spatially developing turbulent boundary layer to active control strategies in the framework of opposition control. *Phys. Fluids (1994-present)* **19** (10), 108102.

Pang, J., Choi, K.-S. & Aessopos, A. 2004 Control of near-wall turbulence for drag reduction by spanwise oscillating Lorentz forse. *AIAA Paper* **2004-2117**.

Panton, R.L. 2001 Overview of the self-sustaining mechanisms of wall turbulence. *Progr. Aero. Sc.* **37**, 341–383.

Philip, J.R. 1972 Flows satisfying mixed no-slip and no-shear conditions. *Z. Angew. Math. und Phys.* **23**, 353–372.

Piomelli, U. 2008 Wall-layer models for large-eddy simulations. *Progress in aerospace sciences* **44** (6), 437–446.

Piomelli, U. & Balaras, E. 2002 Wall-layer models for large-eddy simulations. *Ann. Rev. of Fluid Mech.* **34** (1), 349–374.

Pope, S.B. 2000 *Turbulent Flows.* Cambridge University Press.

Quadrio, M. & Ricco, P. 2004 Critical assessment of turbulent drag reduction through spanwise wall oscillations. *J. Fluid Mech.* **521**, 251–271.

Quadrio, M. & Ricco, P. 2011 The laminar generalized Stokes layer and turbulent drag reduction. *J. Fluid Mech.* **667**, 135–157.

Reynolds, O. 1883 An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels. *Phil. Trans. R. Soc.* **35** (224-226), 84–99.

Ricco, P. & Hahn, S. 2013 Turbulent drag reduction through rotating discs. *J. Fluid Mech.* **722**, 267–290.

Robinson, S.K. 1991*a* The kinematics of turbulent boundary layer structure. PhD thesis, NASA TM 103859.

Robinson, S. K. 1991*b* Coherent motions in the turbulent boundary layer. *Annu. Rev. Fluid Mech.* **23**, 601–639.

Roggenkamp, D., Jessen, W., Li, W., Klaas, M. & Schröder, W. 2015 Experimental investigation of turbulent boundary layers over transversal moving surfaces. *CEAS Aeron. J.* pp. 1–14.

Rosenberg, B.J., Van Buren, T., Fu, M.K. & Smits, A.J. 2016 Turbulent drag reduction over air-and liquid-impregnated surfaces. *Phys. Fluids* **28** (1), 015103.

Rothstein, J.P. 2010 Slip on superhydrophobic surfaces. *Ann. Rev. Fluid Mech.* **42**, 89–109.

Rudin, W. 1973 *Functional Analysis.* McGraw-Hill.

Sani, R.L., Shen, J., Pironneau, O. & Gresho, P.M. 2006 Pressure boundary condition for the time-dependent incompressible navier–Stokes equations. *Int. J. Num. Meth. Fluids* **50** (6), 673–682.

Schönecker, C., Baier, T. & Hardt, S. 2014 Influence of the enclosed fluid on the flow over a microstructured surface in the Cassie state. *J. Fluid Mech.* **740**, 168–195.

Schönecker, C. & Hardt, S. 2013 Longitudinal and transverse flow over a cavity containing a second immiscible fluid. *J. Fluid Mech.* **717**, 376–394.

Schönecker, C. & Hardt, S. 2015 Assessment of drag reduction at slippery, topographically structured surfaces. *Microfluidics and Nanofluidics.* pp. 1–9.

Shiplyuk, A.N., Burov, E. V., Maslov, A.A. & Fomin, V. M. 2004 Effect of porous coatings on stability of hypersonic boundary layers. *J. Applied Mech. Tech. Phys.* **45** (2), 286–291.

Stokes, G.G. 1851 *On the effect of the internal friction of fluids on the motion of pendulums*, , vol. 9. Pitt Press.

Stroock, A.D., Dertinger, S.K., Whitesides, G.M. & Ajdari, A. 2002 Patterning flows using grooved surfaces. *Analytical Chemistry* **74** (20), 5306–5312.

Swarztrauber, P.N. 1977 The methods of cyclic reduction, Fourier analysis and the FACR algorithm for the discrete solution of Poisson's equation on a rectangle. *SIAM Rev.* **19**, 490–501.

Temam, R. 1969 Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (ii). *Arch. Rat. Mech. Anal.* **33** (5), 377–385.

Temam, R. 1984 *Navier-Stokes Equations: Theory and Numerical Analysis*. North-Holland.

Temam, R. 1991 Remark on the pressure boundary condition for the projection method. *Theor. Comp. Fluid Dyn.* **3** (3), 181–184.

Thompson, P.A. & Troian, S.M. 1997 A general boundary condition for liquid flow at solid surfaces. *Nature* **389**, 360–362.

Toms, B.A. 1948 Some observations on the flow of linear polymer solutions through straight tubes at large Reynolds numbers. In *Proc. First Int. Cong. Rheol.*, , vol. 2, pp. 135–141. North-Holland, Amsterdam.

Virk, P.S. 1975 Drag reduction fundamentals. *AIchE Journal* **21** (4), 625–656.

Virk, P.S., Merrill, E. W., Mickley, H. S., Smith, K. A. & Mollo-Christensen, E. L. 1967 The Toms phenomenon: turbulent pipe flow of dilute polymer solutions. *J. Fluid Mech.* **30**, 305–328.

Wallace, J. M., Eckelmann, H. & Brodkey, R.S. 1972 The wall region in turbulent shear flow. *J. Fluid Mech.* **54**, 39–48.

Watanabe, K., Udagawa, Y. & Udagawa, H. 1999 Drag reduction of Newtonian fluid in a circular pipe with a highly water-repellent wall. *J. Fluid Mech.* **381**, 225–238.

Wexler, J.S., Jacobi, I. & Stone, H.A. 2015 Shear-driven failure of liquid-infused surfaces. *Phys. Rev. Letters* **114** (16), 168301.

Wilhelmson, R. B. & Ericksen, J.H. 1977 Direct solutions for Poisson's equation in three dimensions. *J. Comp. Phys.* **25**, 319–331.

Willmarth, WW & Lu, SS 1972 Structure of the reynolds stress near the wall. *J. Fluid Mech.* **55** (1), 65–92.

Wilson, P.W., Lu, W., Xu, H., Kim, P., Kreder, M.J., Alvarenga, J. & Aizenberg, J. 2013 Inhibition of ice nucleation by slippery liquid-infused porous surfaces (slips). *Phys. Chemistry Chem. Phys.* **15** (2), 581–585.

Wise, D. J. & Ricco, P. 2014 Turbulent drag reduction through oscillating discs. *J. Fluid Mech.* **746**, 536–564.

Wong, T.-S., Kang, S.H., Tang, S.K.Y., Smythe, E.J., Hatton, B.D., Grinthal, A. & Aizenberg, J. 2011 Bioinspired self-repairing slippery surfaces with pressure-stable omniphobicity. *Nature* **477** (7365), 443–447.

Yakeno, A., Hasegawa, Y. & Kasagi, N. 2014 Modification of quasi-streamwise vortical structure in a drag-reduced turbulent channel flow with spanwise wall oscillation. *Phys. Fluids* **26**, 085109.

Yoshino, T., Suzuki, Y. & Kasagi, N. 2008 Drag reduction of turbulence air channel flow with distributed micro-sensors and actuators. *J. Fluid Sci. Technol.* **3**, 137–148.