EXPERIMENTAL ANALYSIS OF FLOW AND TURBULENCE CHARACTERISTICS IN OSCILLATORY BOUNDARY LAYERS FROM LDV MEASUREMENTS

BY

JOSE M. MIER LOPEZ

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Civil Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Doctoral Committee:

Professor Marcelo H. Garcia, Chair
Professor James L. Best
Professor Kenneth T. Christensen
Professor Gary Parker

ABSTRACT

The oscillatory boundary layer represents a particular case of unsteady wall-bounded flows in which fluid particles follow a periodic sinusoidal motion. Unlike steady boundary layer flows, the flow regime and bed roughness character of oscillatory flows change in time during the oscillation, a characteristic that introduces a high degree of complexity in the analysis. Experimental work in this topic started in the 1960s followed by numerical work in the late 1980s, yet it is not completely understood, particularly in the transitional regimes.

In this work, several oscillatory flow experiments were performed in the Large Oscillatory Water and Sediment Tunnel (LOWST) facility at the Ven Te Chow Hydrosystems Laboratory. A custom PVC floor was built inside the tunnel to obtain a flat and smooth bed. The range of wave Reynolds numbers tested spanned all along the transition regime of the oscillatory boundary layer between the upper limit of the laminar regime and the lower limit of the turbulent regime ($3x10^4 < Re_w < 9x10^5$). A 3D laser Doppler velocimetry (LDV) system was used to measure instantaneous flow velocities with high spatial and temporal resolution, which allowed capturing flow features with great detail inside the boundary layer and even inside the viscous sublayer in some cases. A special set-up was built involving two LDV probes and a refraction-correcting device to be able to measure all three velocity components (u, v, w) simultaneously.

From the velocity measurements, flow characteristics were obtained through the analysis of different variables including mean flow velocities, boundary layer thickness, turbulence intensities, turbulent kinetic energy, viscous and Reynolds stresses, turbulence production, eddy viscosity, quadrant analysis, bed shear stresses, shear velocity, wave friction factor and viscous sublayer thickness. In particular, the results of this work provide detailed evidence of the competition between laminar and turbulent effects taking place in the transition regime of the oscillatory boundary layer as Re_w increased. A surprising behavior was observed in the phase of the peak bed shear stress, which changed dramatically with Re_w: first leading about 40º ahead of the outer flow for low Re_w, then lagging up to 25º behind for the transitional Re_w experiments, and finally returning slightly ahead about 5º for high Re_w. This finding is expected to have

significant implications for the entrainment and transport of sediment near the bed. Also, investigation of the viscous sublayer revealed that the classic steady flow threshold of $z^+ = 5$ doesn't work well for oscillatory flows. A new method was developed to calculate the thickness of the viscous sublayer taking into account the ratio of viscous to turbulent forces near the bed.

These results can be directly applied to better understand sediment transport in the ocean under the action of waves and currents. Furthermore, they will be also useful for a variety of engineering applications related to fluid mechanics including aerospace, biomedical research, engine design, turbines, industrial machinery, pumping systems, pipe transport, marine hydrokinetics, wave dynamics, and river, coastal and estuarine processes.

"Research is what I'm doing when I don't know what I'm doing"

Wernher von Braun

Chief rocket engineer, NASA

ACKNOWLEDGEMENTS

The facilities used to carry out the experiments presented in this work were funded by the Office of Naval Research of the U.S. Navy, through the DURIP program, Award Number N00014-01-1-0540 for the LOWST flume and Award Number N00014-06-1-0661 for the LDV system. The Graduate Research Assistant support received from the ONR's Coastal Geosciences Program, through Award Number N00014-11-1-0293 "Characterization of Bed Morphodynamics Using Multi Beam Echo Sounding (MBES) and Wavelet Transform (WT)", is gratefully acknowledged.

My first words of gratitude go to my advisor Professor Marcelo Garcia for his continued support and valuable guidance through all these years at UIUC. Also, because he believed in this work and encouraged me to pursue this topic despite the lack of direct funding for it. I would also like to thank my former professors in Spain, in particular Jose Revilla, who inspired me to work in the field of hydraulic engineering, and Iñigo Losada, who helped me come to the University of Illinois. I also appreciate the knowledge and guidance provided by the members of my graduate committee: Professors James Best, Kenneth Christensen and Gary Parker, and their willingness to evaluate this work.

During my first years working in the laboratory I gained valuable experience from many former students to whom I am also very thankful. A few of them were particularly influential in the development of this work: Francisco Pedocchi who introduced me to the LOWST and oscillatory flows; Ezequiel Martin who introduced me to working with lasers and optics and inspired me to maintain the highest quality standards when doing experimental work; Blake Landry for his patience sharing his MATLAB® knowledge in the early years, which helped me develop the skills necessary to process and analyze all the experimental data collected; and Mariano Cantero for his advice and mentorship during the years.

I am also thankful to all the people who helped during the preparation and execution of the experiments. In particular Andrew Rehn, Nicholas Moller and Heng Wu, who endured with me

during the most arduous tasks, dredging the sand, building the PVC floor, setting up the LDV and staying in the lab for long nights of measurements. Also Blake Landry, Roberto Fernandez, Davide Motta, Sumit Sinha, Matt Czapiga, David Waterman, Enrica Viparelli and Gianluca Blois who were always available to give a hand in the lab and to contribute their ideas through constructive discussions. Also thanks to our lab manager Andrew Waratuke, who also helped with the sand dredging and was always a valuable resource, and to our very own computer wizard Nils Oberg, who saved the day in many occasions. The CEE shop also provided valuable help with maintenance and repairs in the LOWST and also making the PVC floor.

I would also like to thank all my teachers, mentors, friends and classmates throughout the years who contributed to shape this work in many ways. Also all my officemates, labmates and staff at the Hydrosystems Laboratory who helped create a great working environment and whose friendship I will always remember. A few names deserve particular recognition in this regard: Robin Ray, Mary Pearson, Tatiana Garcia, Viviana Morales and Nam Jeong Choi, for their silent contribution to make the lab an enjoyable place to work.

Finally, I would like to dedicate my kindest words of gratitude to my family and friends in Spain who coped with my absence all these years. In particular to my parents, who managed to provide the best education possible and have always been a true inspiration for me. Together with my brother and my sister, they have supported me unconditionally all these years despite the distance, and have motivated me to achieve this goal (and many more to come!).

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Motivation

Oscillatory flows are a particular case of the more general unsteady flows, characterized by the change in flow conditions over time. In the particular case of oscillatory flows, the conditions are repeated periodically along a cycle with a certain period of oscillation.

These types of flows are found in nature in a variety of physical processes, such as the pulsating flow of blood in the arteries, air flow in lungs, roll-damping of ships, water flow under a sea wave, flow in a piston device, such as an engine cylinder, and any other sort of fluid oscillating under a pressure forcing. Particularly interesting is the case of wave-induced oscillatory flows, with applications in the fields of coastal and off-shore engineering, such as sediment transport, submarine outfalls, drilling platforms, off-shore wind turbines and other marine energy-generating devices, etc.

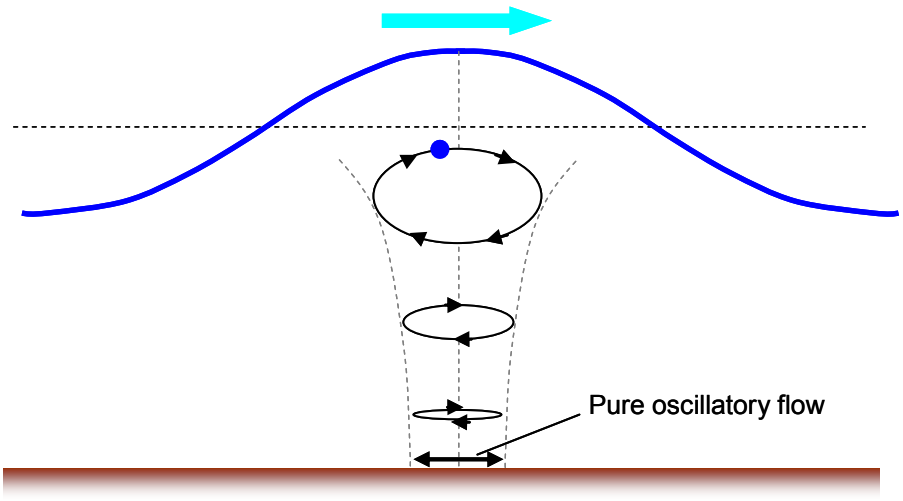


Figure 1. Sketch of water particle trajectories under a surface wave propagating in intermediate or shallow depth. Wave propagating from left to right.

For the purpose of this research work, the focus is going to be on the cases when the oscillatory flow is in the proximity of a boundary, such that the flow interacts with the boundary and vice versa. A direct application of this situation can be found in the bottom of the ocean, where an oscillatory flow is generated under the action of waves on the surface, creating an oscillatory boundary layer (Figure 1).

1.2 Previous work on oscillatory boundary layers

Research in boundary layers has been performed for about a century now, since the early experiments of Prandtl (1905), although the equations of motion for a viscous fluid were established earlier by Navier (1823), Poisson (1831), Saint-Venant (1843) and Stokes (1845). A comprehensive historical review on the development and history of the boundary layer theory can be found in Tani (1977).

In the case of purely oscillatory flow, Jonsson (1963), in Denmark, was among the first to conduct experiments in the turbulent boundary layer. He used a 5 mm micropropeller to measure flow velocities, from which he was able to obtain shear stress distributions and the friction coefficient. Later, it was Kamphuis (1975), in Canada, who expanded the knowledge to the case of rough oscillatory boundary layers, with a special focus on the wave friction factor (f_w). He developed a diagram for a wide range of wave Reynolds numbers (Re_w) and relative roughness (a/k_s) (similar to Nikuradse's (1933) for steady flows), which has been extensively used ever since. However, the behavior and characteristics of the flow inside the oscillatory boundary layer remained still unknown, and specially the mechanisms for the creation and dissipation of turbulence, the role of the viscous sublayer, the effects of roughness, viscosity, bedforms, etc.

In 1980, more experiments by Jonsson (1980) in an oscillating water tunnel over a rough bed showed the existence of a logarithmic overlap layer, similar to the one found in steady flows. He was also able to measure and predict the phase lead of wall shear stress over the free-stream flow. However, the equipment he used didn't allow for measurements of turbulence or near the wall. Later, in Japan, Hino *et al.* (1983) reported an experiment of oscillatory flow in a wind tunnel over a smooth wall. They measured velocities using a laser Doppler velocimetry (LDV)

system, which allowed them to show for the first time the turbulence characteristics associated with this type of flows. The wave Reynolds number for this experiment was $Re_w \approx 3x10^5$, which meant having a transitional oscillatory flow regime. A detailed analysis of mean velocities, turbulence intensities, Reynolds stresses, turbulence energy production, turbulence spectra, ejections, sweeps and fluid-wall interactions was reported, with some surprising conclusions about the accelerating and decelerating stages, and the associated production of turbulent energy. Then, in the United Kingdom in 1987, Sleath (1987) performed a series of experiments over rough beds in an oscillatory water tunnel, using as well a LDV system for the velocity measurements. The rough beds consisted of a layer of sand, gravel or pebbles glued on a flat surface. His study comprised a wide range of wave Reynolds number (Re_w) and relative roughness (a/k_s), and it is probably one of the most complete experimental sets reported. The limitations of his tunnel only allowed him to reach values of $Re_w < 3x10^5$, which made for a difficult analysis, since most of the experiments fell into the transitional oscillatory flow regime, despite his "hopes" for a fully developed turbulence. The analysis performed stayed quite superficial and some conclusions were unclear. He focused on turbulence intensities, Reynolds stresses, eddy viscosity and mixing length, and the effect of increasing roughness in some of these variables. There is, however, a very interesting observation regarding the development of a logarithmic boundary layer in relation with the wave Reynolds number (Re_w) and the phase (ωt), which can be used to understand the changing threshold for the fully developed turbulent regime in oscillatory flows.

Major contributions to the understanding of turbulent oscillatory flows came also from the work by Sumer, Fredsoe and Jensen in Denmark. They performed their experiments in an oscillating water tunnel powered by a pneumatic piston, and a LDV system was used for the measurement of velocities. In Sumer *et al.* (1987), they presented a comparison of the hydraulically smooth and rough oscillatory boundary layers through two experiments with the same wave Reynolds number (Re_w). Mean velocity and shear stress profiles were shown and compared, together with boundary layer thickness evolution over the cycle for both the smooth and rough cases. The authors also observed and discussed about the transfer of momentum from the wall, where turbulence is created, up to the free-stream flow and the relation of this process with the accelerating and decelerating stages of the oscillatory flow.

Later in 1988, Jensen (1988) in his Ph.D. thesis was able to perform a systematic and comprehensive analysis of the turbulent oscillatory boundary layer. He reported fifteen experiments, with wave Reynolds numbers ranging between $7.5\text{x}10^3 < Re_w < 6\text{x}10^6$, although the analysis was focused only on the high Reynolds number experiments in an attempt to achieve fully developed turbulent flow regime conditions. The experiments involved smooth wall and rough wall conditions. Only four of them were performed under rough conditions, which were created by gluing D = 0.35 mm sand paper and D = 1.5 mm sand grains to the flat bed of the tunnel. The relative roughness ranged between $435 < a/k_s < 3700$ for those experiments. His results focused on the effects of the relative roughness (a/k_s) and wave Reynolds number (Re_w) on the most relevant parameters for the characterization of this type of flows, namely, the thickness of the boundary layer (δ), the phase lead of the wall shear stress with respect the free-stream flow velocity ($\Delta\phi_s$), and the wave friction factor (f_w). Similarly to other authors before him, he reported also mean flow and shear stress profiles, turbulence intensities, Reynolds stresses, wall shear stress, etc. In addition, he presented the results in dimensionless variables, both for outer scales and inner scales, the latter being of special interest for the analysis of the oscillatory boundary layer very close to the wall and the quantification of the viscous effects, in a similar way to the steady boundary layers. Although the results are lacking, in some cases, some more experimental evidence to be able to generate strong conclusions about this type of flows, the methodology of the analysis he presented shows how the unsteady oscillatory boundary layer problem may be approached.

More recently, Carstensen *et al.* (2010), also in Denmark, performed several experiments in an oscillatory flow tunnel with smooth bed. The flow conditions in their experiments were such that the amplitude of the oscillation was not constant but it was increasing slowly for every cycle. They covered a range of wave Reynolds numbers between $7\text{x}10^4 < Re_w < 5\text{x}10^6$ to be able to capture the structure of the flow in the transition regime. Their measurements were mainly focused on flow visualization, although bed shear stresses were also measured. They were among the first to obtain videos of vortex tubes and turbulent spots as they were developing in the oscillatory boundary layer with great detail.

In addition to the experimental work described, in recent years there have been a growing number of numerical studies in the literature trying to simulate the oscillatory boundary layer. Among the first was the DNS work of Spalart and Baldwin (1987), which is considered a reference for all subsequent numerical studies. Later came the DNS work of Vittori and Verzicco (1998), which focused on the onset of turbulence in the transition regime. More recently, Pedocchi *et al.* (2011) where among the first to perform a direct numerical simulation (DNS) in the fully turbulent regime (Re_w = 1.4x10^6). Their study focused on the analysis of the turbulent kinetic energy budget, providing good insight on turbulence production and dissipation in the turbulent oscillatory boundary layer. At the same time, Mazzuoli *et al.* (2011) performed a DNS study in which they reproduced two of Carstensen *et al.* (2010) experiments. Their results focused on the formation of flow structures and in particular on the characterization of the turbulent spots observed by Carstensen. After them, Ozdemir *et al.* (2014) followed with more DNS results in the transition regime, focused on the onset of turbulence as well. They also showed flow structures, such as vortex tubes, using the swirling strength concept from Zhou *et al.* (1999). Although the DNS technique has improved significantly in recent years, all these studies are not yet able to capture completely all the features of the oscillatory boundary layer, particularly in the transition regime. In fact, most of them use the experimental data of Hino *et al.* (1983), Sleath (1987) and Jensen (1988) as a reference to compare their results. As such, the results in this work will be compared mainly against other experimental work.

1.3 Objectives

The main objective of this work is to advance the scientific knowledge about the flow and turbulence characteristics of the oscillatory boundary layer. Despite the many efforts from previous researchers, a significant shortage in experimental evidence still exists in this field. Yet experiments can provide the most accurate information to understand a physical phenomenon of such complexity. Consequently, an experimental approach was used during this work.

High spatial and temporal resolution measurements were taken with a laser Doppler velocimetry (LDV) system in a unique oscillatory tunnel such as the Large Oscillating Water-Sediment Tunnel (LOWST) at the Ven Te Chow Hydrosystems Laboratory. The range of flows tested

spanned from the laminar to the turbulent regimes. This way, the existing studies could be complemented and their conclusions revisited and improved. With all the experimental results, a comprehensive analysis was performed, in a similar fashion to the methodology introduced by Jensen (1988). These results, together with existing data from previous researchers on the topic, contributed to build a more complete theory on the characteristics and behavior of the oscillatory boundary layer.

Additionally, the findings of this work are very relevant for the sediment transport phenomenon. Significant implications for initiation of motion, development of bedforms, sediment entrainment, and sediment fluxes inside the oscillatory boundary layer can be derived from the results of this work.

Also, the results obtained will be useful to compare and validate numerical simulations, like the DNS work of Spalart and Baldwin (1987), Vittori and Verzicco (1998), Mazzuoli *et al.* (2011), Pedocchi *et al.* (2011) and Ozdemir *et al.* (2014).

CHAPTER 2

LITERATURE REVIEW: OSCILLATORY BOUNDARY LAYERS

2.1 General concepts about oscillatory flows

An oscillatory flow is a particular type of unsteady flow in which the mean flow repeats itself for every cycle with a constant periodicity, given by the period of the oscillation (T). In a purely oscillatory flow, fluid particles only travel along one direction back and forth, such that the average fluid velocities in the other two orthogonal directions are zero. The sketch in Figure 2 shows the trajectory (x) and streamwise velocity (u) of a fluid particle in this type of flows, which can be represented by sinusoidal expressions of the form:

$$x = a \cdot \sin\left(\omega t + \phi_o\right) \tag{1}$$

$$u = a\omega \cdot \cos\left(\omega t + \phi_o\right) = a\omega \cdot \sin\left(\omega t + \phi_o^{'}\right) \tag{2}$$

where a is the amplitude of the oscillation, such that the particle would travel a distance $2a$ from side to side, ω is the angular frequency, given by $\omega = 2\pi / T$ (in rad/s), T is the period of the oscillation (in s) and ϕ_o is a phase shift (in rad).

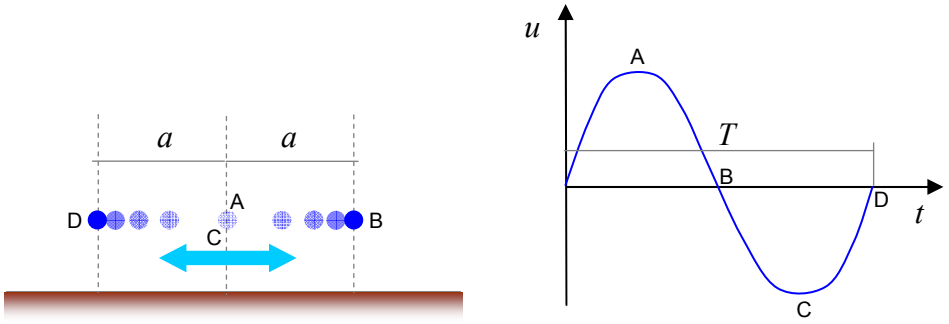


Figure 2. Sketch of fluid particle trajectory (left) and fluid particle streamwise velocity (right) in a purely oscillatory flow.

In order to characterize these flows, it is common practice to use the wave Reynolds number (Re_w) first given by Bagnold (1946). This number is defined in terms of the amplitude (a) and maximum velocity of the oscillation (u_{max}), such that:

$$\mathrm{Re}_w = \frac{a \cdot u_{\max}}{\nu} \tag{3}$$

where ν is the kinematic viscosity of the fluid $\left[\frac{length^2}{time}\right]$, and u_{max} can be calculated as:

$$u_{\max} = \frac{2\pi a}{T} \tag{4}$$

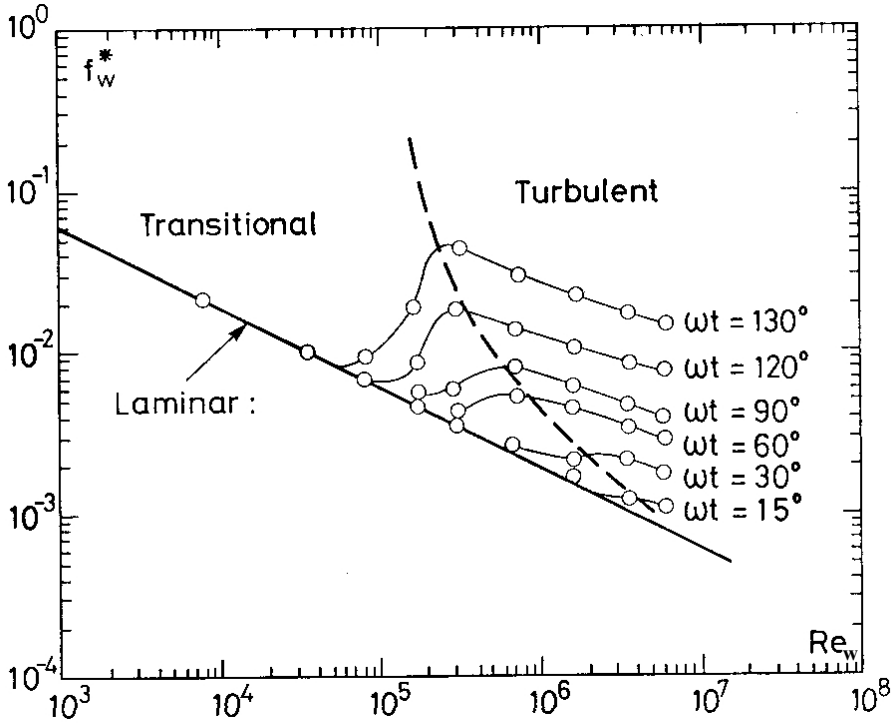


Figure 3. Diagram of normalized wave friction factor (f_w*), based on wave Reynolds number (Re_w) and phase of the oscillation (ωt), showing the different flow regimes for oscillatory flows. Smooth wall. From Jensen (1988).

Unlike steady flows, in unsteady flows the fluid is subject to changes in velocity over time. In the particular case of oscillatory flows, for every half-cycle the flow undergoes one acceleration stage and one deceleration stage, from zero velocity up to maximum streamwise velocity (u_{max}) and back to zero. This implies that oscillatory flows have a different flow regime for every phase (ωt) in the cycle, thus, creating the possibility for laminar, turbulent and laminar-to-turbulent transitional conditions to coexist in sequence, which introduces a high degree of complexity in the analysis of such flows. This complex behavior can be visualized in Figure 3, adapted from Jensen (1988), showing the evolution in phase (ωt) of a characteristic parameter of the flow. For a given experiment (points of constant Re_w) the regime within the experiment can change from laminar to transitional to turbulent conditions.

2.2 The oscillatory boundary layer

When a boundary or wall is found in the proximity of an oscillatory flow, then the flow begins to feel the effect of the wall and two distinct regions appear with different flow characteristics, namely, the outer inviscid region, far from the wall, where the oscillatory flow doesn't feel the wall; and the boundary region or boundary layer, close to the wall, where viscosity effects are important (Figure 4). The velocity field outside the boundary layer is not affected by the presence of the wall and will only depend on the oscillation $u_{out}(\omega t)$. Inside the boundary layer, the velocity field will be affected due to the friction with the wall and will depend both on the oscillation and the distance from the wall $u(z, \omega t)$.

The direct consequence of the existence of the boundary layer is the appearance of shear and turbulence that propagates outward from the wall throughout the entire boundary layer up to the outer free-stream flow. This has been observed in both steady and unsteady flows (see Hino *et al.*, 1983; Sumer *et al.*, 1987; Jensen, 1988).

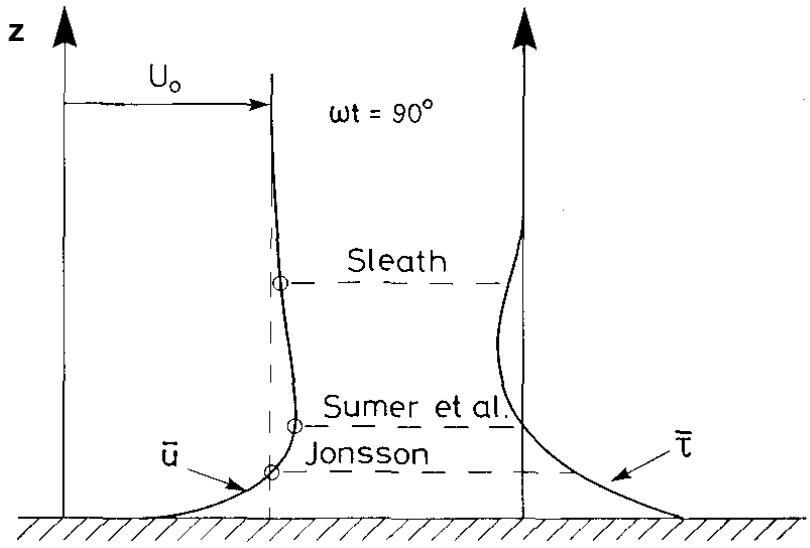


Figure 4. Sketch of the oscillatory boundary layer, showing velocity (u) and shear stress (τ) profiles for phase $\omega t = \pi/2 = 90^o$ of the oscillation, and the definition of the thickness (δ) of the boundary layer for different authors. From Jensen (1988).

In general, the shear stress (τ) in a turbulent flow can be expressed as in equation (5), where the first term represents the viscous contribution due to the vertical velocity gradient and the second term (known as Reynolds shear stress) represents the turbulent contribution due to the velocity fluctuations:

$$\tau = \mu \frac{\partial u}{\partial z} - \rho \overline{u'w'} \tag{5}$$

where μ is the dynamic viscosity $\left[\frac{mass}{length \cdot time}\right]$, given by $\mu = \rho\nu$, and ρ is the fluid density.

As turns out from the unsteadiness of the oscillatory flow, the shear stress is a time-dependent variable, changing its value for every phase (ωt) in the cycle. Particularly interesting is the value of the shear stress at the wall or bed shear stress (τ_b), which is the physical manifestation of the friction between fluid and wall, and is the primary force for the movement of bed sediments.

Using the bed shear stress (τ_b), it is possible to define the shear velocity (u_*) and the friction coefficient (C_f) (see Garcia, 2008):

$$\tau_b = \rho u_*^2 \tag{6}$$

$$\tau_b = \rho C_f u_{out}^2 \tag{7}$$

In wave boundary layers, the friction is usually characterized by the wave friction factor (f_w), which is equivalent to the friction coefficient ($f_w = 2C_f$), and from equations (6) and (7) can be defined as:

$$f_w = \frac{u_*^2}{\frac{1}{2}u_{out}^2} \tag{8}$$

For oscillatory boundary layers, since both u_* and u_{out} are phase-dependent variables, then the wave friction factor will also be dependent on phase $f_w(\omega t)$. This friction is a consequence of the roughness of the wall, due to the existence of irregularities, protuberances or topographic features associated with the wall, the most common of which are usually sediment grains and bedforms. For the case of grain friction, the roughness height (k) is usually taken as the median size (D) of the sediment. Then, the roughness scale is usually taken as Nikuradse's equivalent sand-grain roughness (k_s), which varies between $k_s \approx$ 1-5 D, depending on the authors, with a typical value of $k_s = 2.5D$ (Garcia, 2008). The findings of Kamphuis (1975) and Jensen (1988) for oscillatory boundary layers also agree with that value.

However, the roughness character of the wall depends not only on the roughness height, but also on the flow conditions and the characteristics of the fluid, such that for a particular set of conditions, a flow may feel the wall as hydraulically smooth, rough or in transition. The dimensionless parameter that controls this character is known as the Reynolds shear number or roughness Reynolds number (Re_* or k_s^+) and is obtained from Nikuradse's equivalent roughness (k_s) divided by the inner length scale of the boundary layer (ν/u_*):

$$\mathrm{Re}_* = k_s^+ = \frac{k_s}{\nu / u_*} \tag{9}$$

Threshold values that define the character of the wall for oscillatory flows can be found in the literature (Kamphuis, 1975) and are defined in a similar way as for unidirectional steady flows (Nezu and Nakagawa, 1993; see also Garcia, 2008):

	Unidirectional flow	Oscillatory flow
Hydraulically smooth:	$Re_* < 5$	$Re_{*\,max} < 15$
Hydraulically in transition:	$5 < Re_* < 70$	$15 < Re_{*\,max} < 200$
Hydraulically rough:	$70 < Re_*$	$200 < Re_{*\,max}$

Since the shear velocity (u_*) varies with phase (ωt) over the cycle of the oscillatory flow, then the Reynolds shear number (Re_*) is also a phase-dependent variable, and so the character of the wall roughness with respect to the flow would be constantly changing along the cycle. This fact introduces a high level of complexity for the analysis of rough walls in oscillatory boundary layers, as opposed to steady flows.

This way, the combination of the three different flow regimes and the three different characters of the wall creates the possibility for 9 different behaviors of the boundary layer (Figure 5). These different behaviors can be found one by one for each flow condition in steady boundary layers, however, in unsteady conditions several of them may appear in sequence along with the variation of the flow condition. In the particular case of the oscillatory boundary layer, if the flow regime is fully turbulent and the character of the wall is fully rough for the phase of maximum velocity, then it is guaranteed that the flow will go through transitional and laminar regimes as it tends to zero velocity close to the flow reversal phases, and in doing so, the character of the wall will also evolve through the transitional and smooth regimes even for a small amount of time.

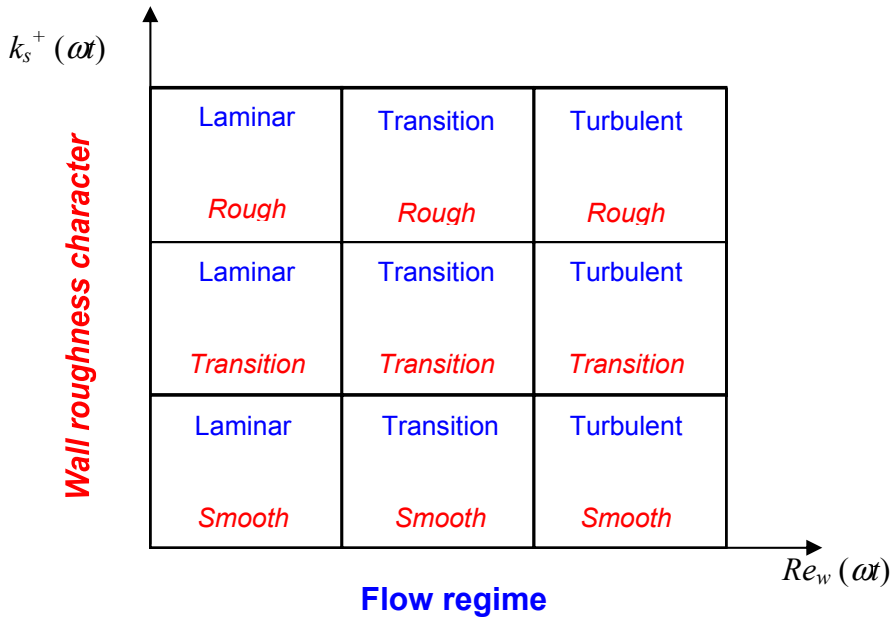


Figure 5. Conceptual diagram of possible boundary layer scenarios according to flow regime and wall roughness character.

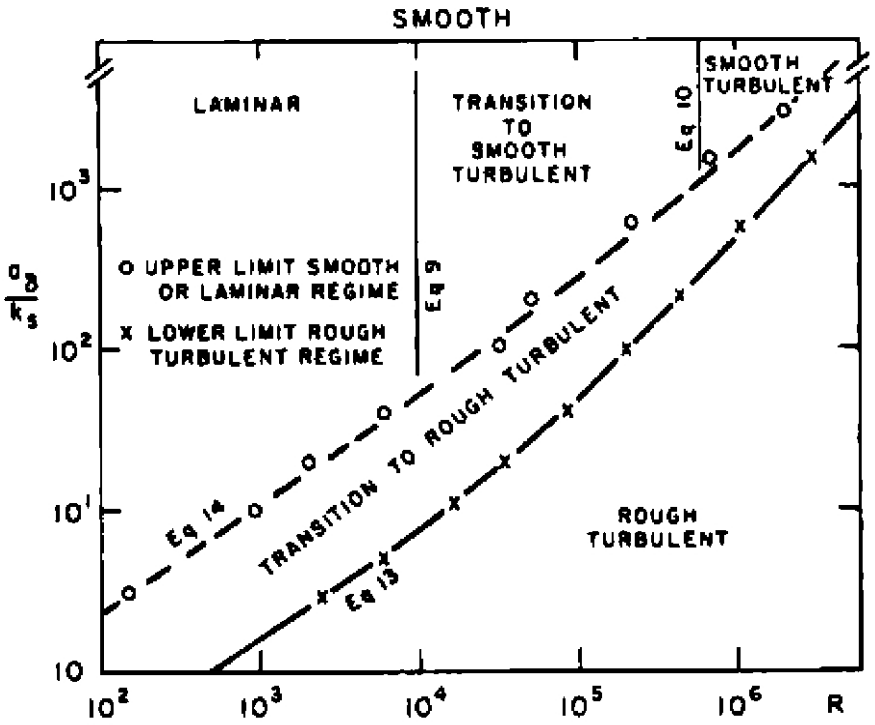


Figure 6. Oscillatory flow regimes diagram. From Kamphuis (1975).

Kamphuis (1975) was able to characterize all these combinations of flow regime and wall roughness character and plotted them into a diagram (Figure 6) showing relative roughness (a/k_s) against wave Reynolds number (Re_w). This diagram was made considering the Re_w calculated with the maximum streamwise velocity of the outer flow (u_{out_max}), which is typically the velocity happening at phase ωt = 90º. Three main regions stand out, namely the laminar-smooth, the turbulent-smooth and the turbulent-rough. Then transition regions occur between those three, mainly the laminar-to-turbulent transition for smooth cases, and a large region covering the smooth-to-rough transition for both laminar and turbulent regimes.

2.3 Governing equations

The oscillatory boundary layer is, in general, a three-dimensional hydrodynamic process, which can be analyzed starting from the general Navier-Stokes equations and applying the particular boundary conditions of the problem. For the sake of clarity, the 2D equations of the oscillatory boundary layer are presented herein, with x the streamwise direction and z the vertical direction.

Conservation of momentum equations:

x-dir: $$\rho\left(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+w\frac{\partial u}{\partial z}\right)=-\frac{\partial p}{\partial x}+\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial z^2}\right)+\rho g_x \tag{10}$$

z-dir: $$\rho\left(\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+w\frac{\partial w}{\partial z}\right)=-\frac{\partial p}{\partial z}+\mu\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial z^2}\right)+\rho g_z \tag{11}$$

Continuity equation:

$$\frac{\partial \rho}{\partial t}+\frac{\partial \rho u}{\partial x}+\frac{\partial \rho w}{\partial z}=0 \tag{12}$$

From the general equations, the Reynolds average procedure can be applied to separate between mean and fluctuating quantities ($\varphi=\overline{\varphi}+\varphi'$). Also, several assumptions about the oscillatory flow are made:

- Incompressible flow: ρ = constant
- Horizontal bed: $g_x = 0;\ g_z = -g$
- Oscillation in x-direction: $\frac{\partial \overline{u}_i}{\partial x} = 0$ (mean velocities constant in *x*-dir)

Then, resolving for the mean flow, from the continuity equation it results that:

$$\frac{\partial \overline{w}}{\partial z} = 0 \tag{13}$$

Which means that $\overline{w}$ is constant in *z*-dir, but since at the wall it has to be zero, then the constant is zero:

$$\overline{w} = 0 \tag{14}$$

And the conservation of momentum equations result as follows:

x-dir: $$\frac{\partial \overline{u}}{\partial t} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x} + \frac{1}{\rho}\frac{\partial}{\partial z}\left(\mu \frac{\partial \overline{u}}{\partial z} - \rho\overline{u'w'}\right) \tag{15}$$

z-dir: $$0 = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial z} + \frac{1}{\rho}\frac{\partial}{\partial z}\left(-\rho\overline{w'w'}\right) - g \tag{16}$$

Now, taking into account that the external forcing of the oscillatory flow is the pressure gradient, which generates the sinusoidal velocity field:

$$\frac{\partial u_{out}}{\partial t} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x} \tag{17}$$

$$u_{out} = \omega a \sin(\omega t) = u_{out\_\max} \sin(\omega t) \tag{18}$$

And, also, taking into account the definition of shear stress (τ):

$$\overline{\tau} = \mu \frac{\partial \overline{u}}{\partial z} - \rho \overline{u'w'} \tag{19}$$

Then, we obtain the equation that describes the mean flow of the oscillatory boundary layer:

$$x\text{-dir:} \quad \frac{\partial(\overline{u} - u_{out})}{\partial t} = \frac{\partial}{\partial z}\left(\frac{\overline{\tau}}{\rho}\right) \tag{20}$$

It is also worth noting that, from the analysis of the z-dir equation, it turns out that the pressure variation in the vertical is not hydrostatic, since a deviatory term remains, related to the turbulence in the vertical velocity component (w'). Jensen (1988) also realized about this fact, but didn't explore any further through his experiments:

$$z\text{-dir:} \quad \frac{\partial \overline{p}}{\partial z} = -\rho\left(g + \frac{\partial \overline{w'^2}}{\partial z}\right) \tag{21}$$

2.4 The laminar regime: analytical solution to the equations

For the particular case where no turbulence is considered, the governing equations of the oscillatory boundary layer can be solved analytically, giving as a result the solution for the laminar oscillatory boundary layer. This solution has been known for many years and is usually called "Stokes second problem" (Stokes, 1845).

In the laminar case, all turbulence terms are neglected, and consequently the shear stress only has the viscous component, so from equation (19) we obtain:

$$\tau = \mu \frac{\partial u}{\partial z} \tag{22}$$

Starting from equation (20) and now using (22) for the shear stress, the following equation is obtained:

$$\frac{\partial(u-u_{out})}{\partial t}=\nu\frac{\partial^2 u}{\partial z^2} \tag{23}$$

where the oscillation is given by $u_{out}=u_{out\_\max}\sin(\omega t)$, as before, and the averaged symbols are not needed any more since no turbulence is considered for the laminar soluton.

A solution for equation (23) can be obtained analytically for the streamwise velocity field inside the laminar boundary layer, resulting:

$$u=u_{out\_\max}\left(\sin(\omega t)-e^{-z/\delta_s}\sin(\omega t-z/\delta_s)\right) \tag{24}$$

where δ_s is called the "Stokes' length" and is defined as $\delta_s=\sqrt{2\nu/\omega}$. This parameter is proportional to the thickness of the laminar boundary layer. This solution can be evaluated for different phases (ωt) and distance from the wall (z) to obtain the distribution of velocities in the cycle (Figure 7).

Particularly interesting is the solution for the shear stress in the laminar regime, which can be obtained from equations (22) and (24). After some trigonometric manipulation, the following expression is obtained (see also Figure 8):

$$\tau=\rho\frac{u_{out\_\max}^2}{\sqrt{\mathrm{Re}_w}}e^{-z/\delta_s}\sin(\omega t-z/\delta_s+\pi/4) \tag{25}$$

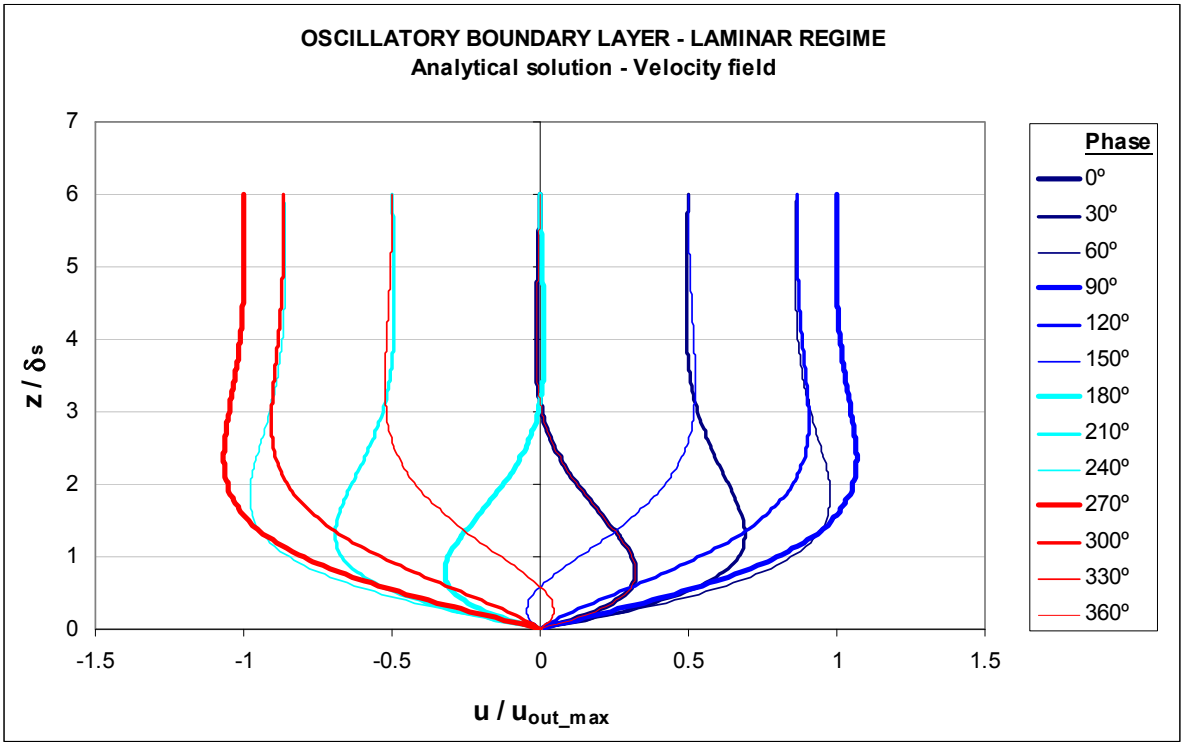


Figure 7. Plot of analytical solution of the flow velocity inside the laminar oscillatory boundary layer for one cycle.

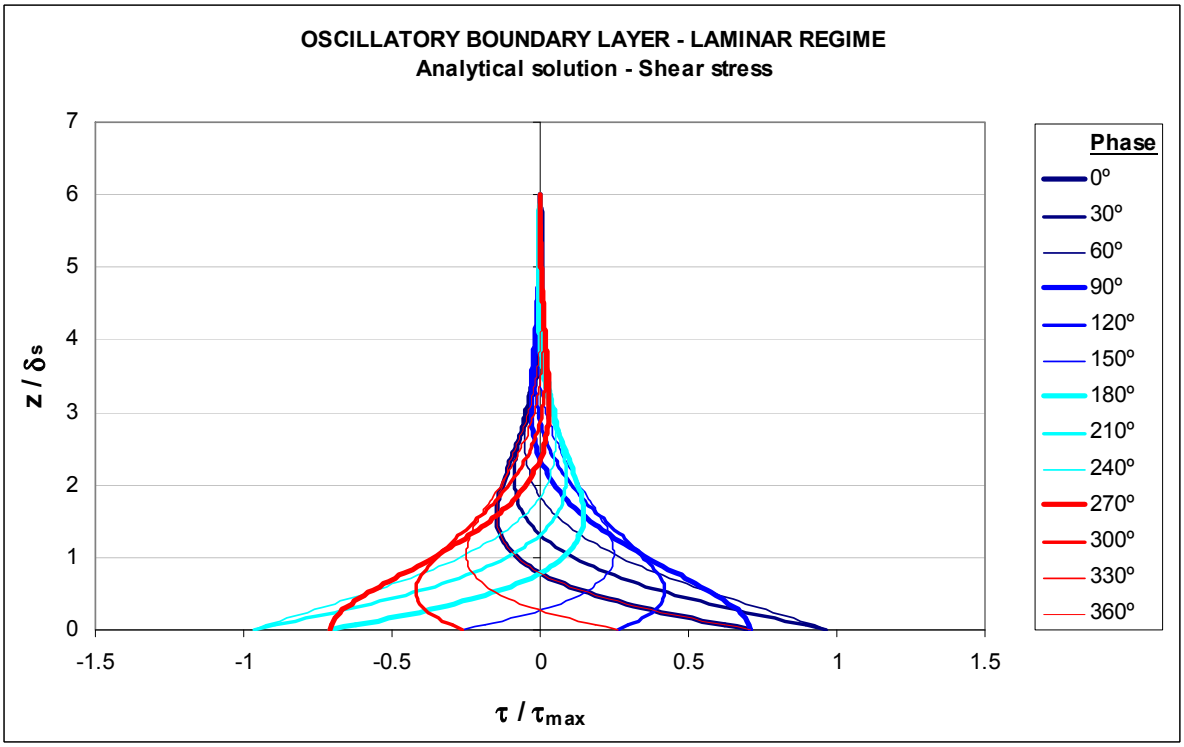


Figure 8. Plot of analytical solution of the shear stress inside the laminar oscillatory boundary layer for one cycle.

Then, from equation (25) the bed shear stress can be obtained for the case when $z = 0$, which gives the following expression:

$$\tau_b = \rho \frac{u_{out\_max}^2}{\sqrt{\mathrm{Re}_w}} \sin(\omega t + \pi/4) \qquad (26)$$

In that equation, the characteristic π/4 or 45º phase lead of the bed shear stress with respect to the outer flow velocity is clearly observed (see also Figure 9).

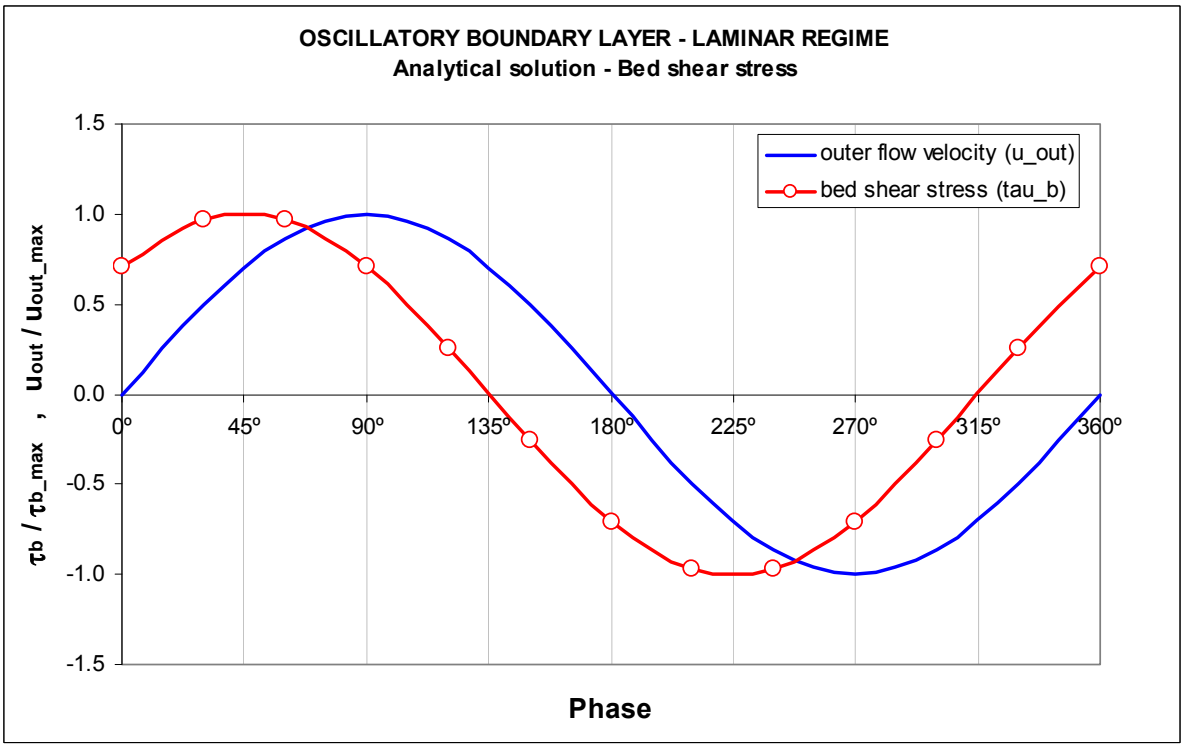


Figure 9. Plot of analytical solution of the bed shear stress and outer flow velocity for the laminar oscillatory boundary layer, showing a 45º phase lead.

2.5 The turbulent regime

In the turbulent regime, the governing equations of the oscillatory boundary layer can't be solved analytically as in the laminar regime. Approximated solutions have been proposed in the literature, being Fredsoe (1984) among the most widely used. However, an experimental

approach is still needed to understand the complex processes taking place in the turbulent oscillatory boundary layer.

As mentioned before, due to the particular nature of oscillatory flows, the turbulent oscillatory boundary layer develops and vanishes for every half-cycle due to the flow reversal. In addition, since the instantaneous flow regime changes for each phase, close to the wall, the boundary layer evolves accordingly, and some particular features characteristic of turbulent oscillatory flows appear.

2.5.1 Mean flow velocity

The streamwise flow velocity (u) inside the boundary layer shows a phase lead and larger peak value with respect to the outer flow. This behavior is expected from the analytical solution of the laminar boundary layer and has also been observed by many authors (Bagnold, 1946; Jonsson, 1980; Hino *et al.*, 1983; Sleath, 1987; Jensen, 1988; Fredsoe *et al.*, 1993; Sumer *et al.*, 1993). See also Figure 10.

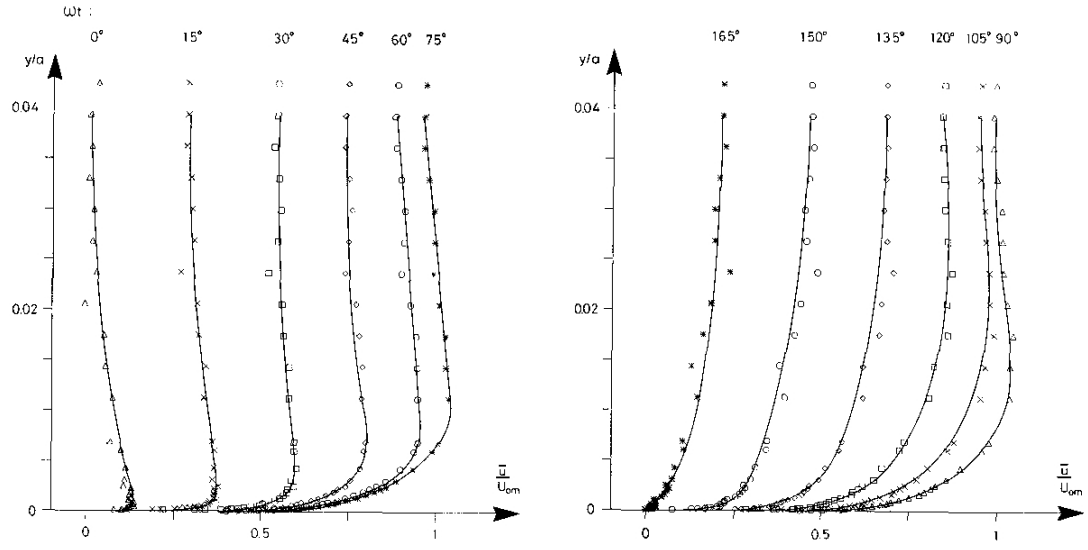


Figure 10. Ensemble-averaged profiles of streamwise velocity (u) of a turbulent oscillatory flow for one half-cycle, showing the accelerating stage (left) and the decelerating stage (right). Smooth bed. $Re_w = 6\text{x}10^6$. From Jensen (1988).

A logarithmic region exists for the phases where the flow inside the boundary layer has reached a fully developed turbulent state, typically far from the flow reversal (Hino *et al.*, 1983; Jensen, 1988), and it is more pronounced for larger wave Reynolds numbers (Re_w) (Sleath, 1987). See Figure 11.

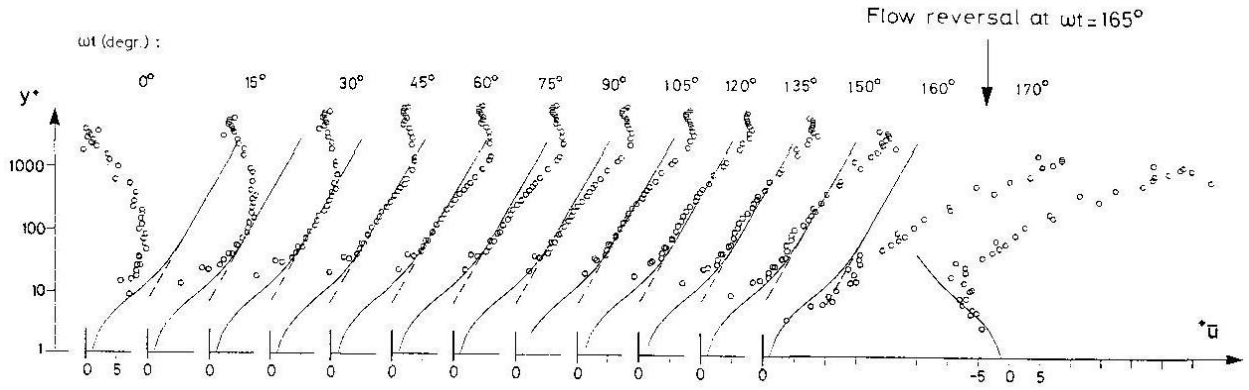


Figure 11. Ensemble-averaged profiles of streamwise velocity (u) for a turbulent oscillatory boundary layer, showing the existence of a logarithmic region for the phases in which the boundary layer is fully developed turbulent. Vertical axis in wall units (dimensionless) in semi-log scale. Smooth bed. $Re_w = 6\text{x}10^6$. From Jensen (1988).

2.5.2 Boundary layer and viscous sublayer

Similarly to the steady boundary layer, the streamwise fluid velocity at the wall is zero ($u|_{z=0} = 0$, i.e. no-slip condition), and a boundary layer develops up to some distance from the wall, with the existence of a viscous sublayer in the inner portion of it (Hino *et al.*, 1983; Jensen, 1988).

The thickness of the boundary layer (δ) grows slowly during the acceleration stage, and then abruptly expands during the deceleration stage (Hino *et al.*, 1983; Jensen, 1988). On the contrary, the thickness of the viscous sublayer (δ_v) grows steadily during the acceleration stage, and then decreases suddenly with the generation of high turbulence about the peak flow, and becomes very thin (Hino *et al.*, 1983). See also Figure 12.

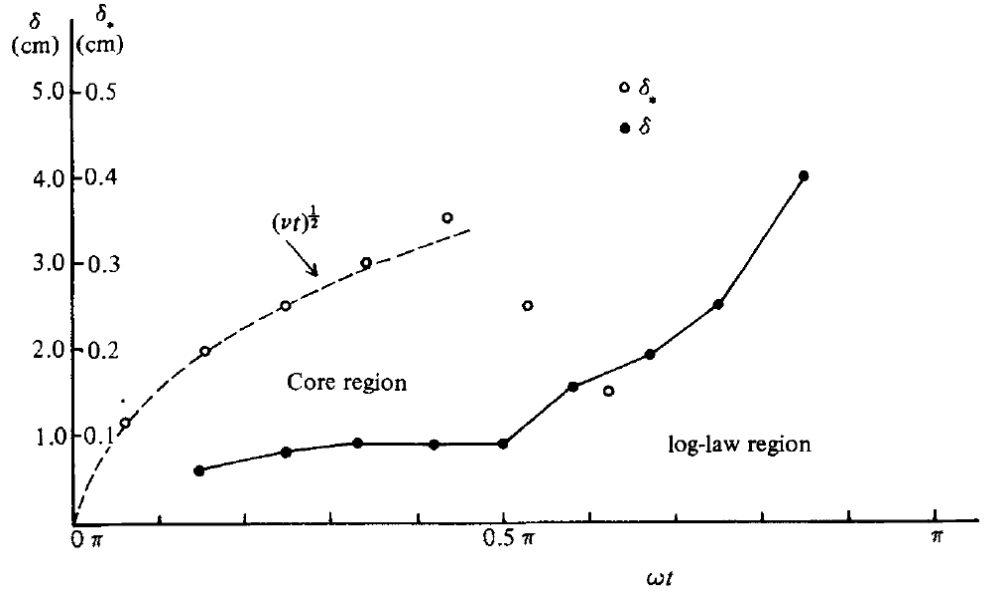


Figure 12. Evolution of boundary layer thickness (denoted δ) and viscous sublayer thickness (denoted δ_*) over one half-cycle in a turbulent oscillatory flow. Smooth bed. $Re_w = 3x10^5$. From Hino *et al.* (1983).

The boundary layer thickness varies with the wave Reynolds number (Re_w) of the oscillatory flow. As Re_w gets larger, the relative thickness of the oscillatory boundary layer (δ/a) gets smaller, according to Jensen (1988). It could be taken as if faster flows (for the same amplitude a) tend to compress the boundary layer and confine it close to the wall. This behavior is similar to the one observed in steady boundary-layer flows (Schlichting, 1979). See Figure 13.

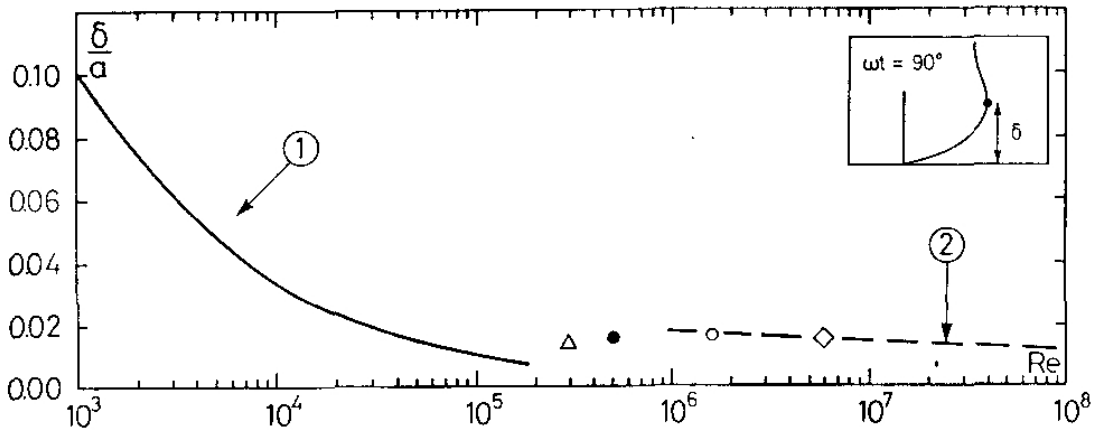


Figure 13. Variation of relative boundary layer thickness (δ/a) with wave Reynolds number (Re_w). Values of δ taken at phase $\omega t = \pi/2 = 90°$. Smooth bed. 1: Laminar solution; 2: Fredsoe's (1984) theoretical solution. From Jensen (1988).

The presence of wall roughness (k_s) also affects the thickness of the oscillatory boundary layer, such that δ/a grows as the relative roughness (a/k_s) decreases (absolute roughness increase) (Jensen, 1988). This effect can be explained from the increase in turbulence that takes places due to the increased roughness, which promotes the transfer of momentum to the upper layers, thus propagating the effect of the wall higher into the flow. See Figure 14.

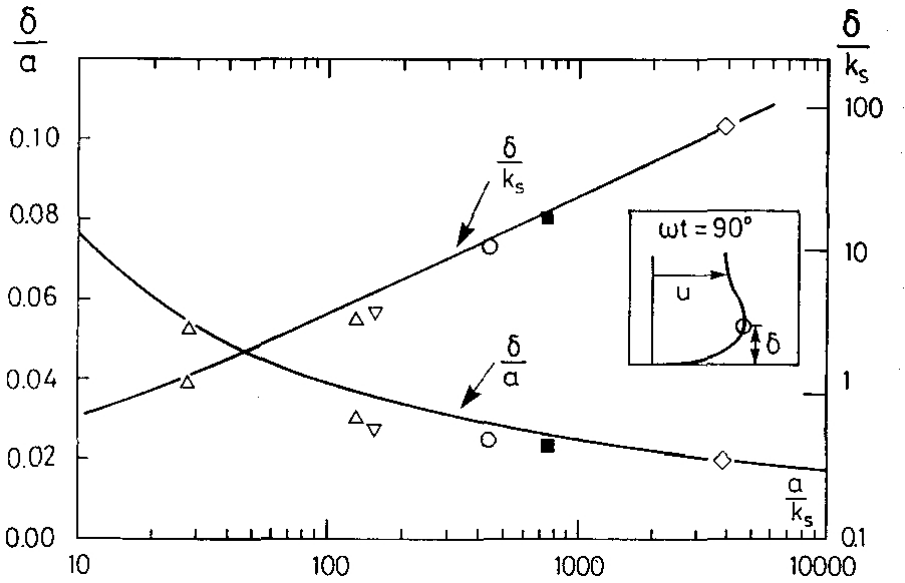


Figure 14. Variation of relative boundary layer thickness (δ/a) with roughness (k_s). Values of δ taken at phase $\omega t = \pi/2 = 90°$. Solid lines: Fredsoe's (1984) theoretical solution. From Jensen (1988).

2.5.3 Bed shear stress

Shear stresses in an oscillatory flow change with the phase (ωt) and so does the wall shear stress. For every half-cycle there is a peak of wall shear stress. According to Hino *et al.* (1983), the magnitude of the peak is a combination of the viscous and the turbulent terms, respectively. See Figure 15.

In addition, the peak of wall shear stresses (τ_{b_max}) happens at a different phase than the peak velocity of the outer flow (u_{out_max}). This effect is usually known as the "phase lead" of the oscillatory boundary layer because typically τ_{b_max} happens in advance of u_{out_max}. However, it is possible that it may not always be the case and that τ_{b_max} may happen after u_{out_max} (causing a phase lag instead) depending on Re_w. This effect was observed by Jensen (1988) and is shown in

Figure 16, however it was not clear what was causing this effect and under what range of Re_w conditions it may happen.

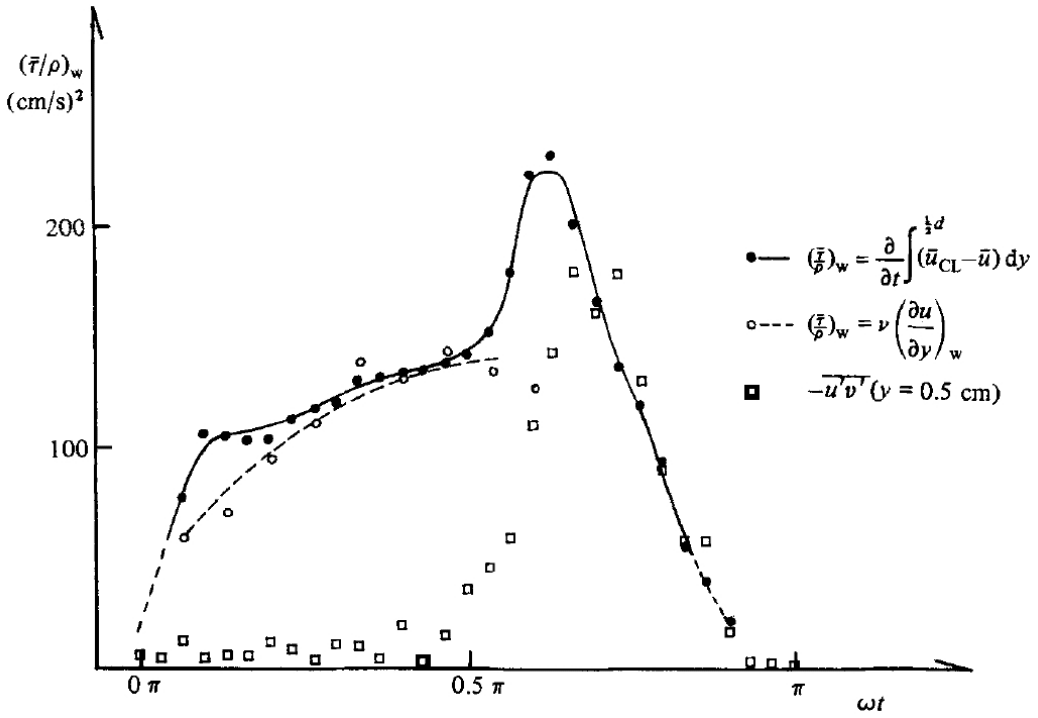


Figure 15. Evolution of wall shear stress (denoted τ_w) over one half-cycle in an oscillatory flow. Smooth bed. Re_w = $3x10^5$. From Hino *et al.* (1983).

Jensen (1988) summarized this effect in a plot relating the phase lead to Re_w (Figure 17). He showed how the phase lead varies with Re_w, with a value of $\pi/4$ or 45º for low Re_w (laminar regime) and decreasing through the transition regime down to about $\pi/32$ or 6º for the turbulent regime. It must be pointed out that the value of 45º for the laminar regime was predicted by the analytical solution of the laminar boundary layer. Other authors (Hino *et al.*, 1983) have provided experimental evidence for this effect as well; however, they showed a phase lag instead, such that the peak of wall shear stress occurs after the peak of outer velocity (see Figure 15). A more detailed investigation is required on this matter, since the direction of the phase difference is contradictory based on the literature. From the experiments by Jensen (1988) it may be argued that this variability may be related to the transition regime of the oscillatory boundary layer.

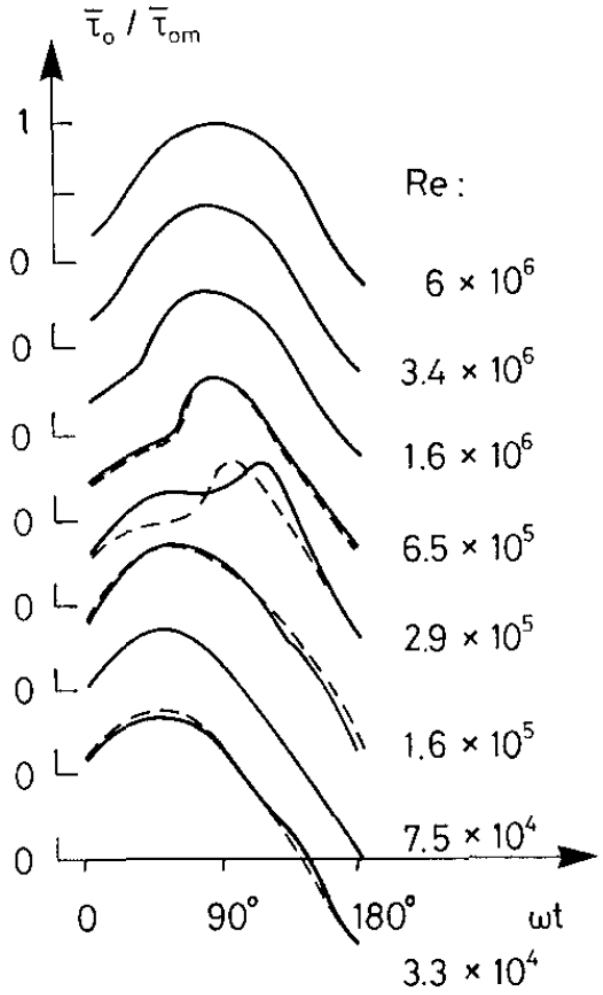


Figure 16. Evolution of normalized wall shear stress over one half-cycle in an oscillatory flow, for different Re_w. Smooth bed. From Jensen (1988).

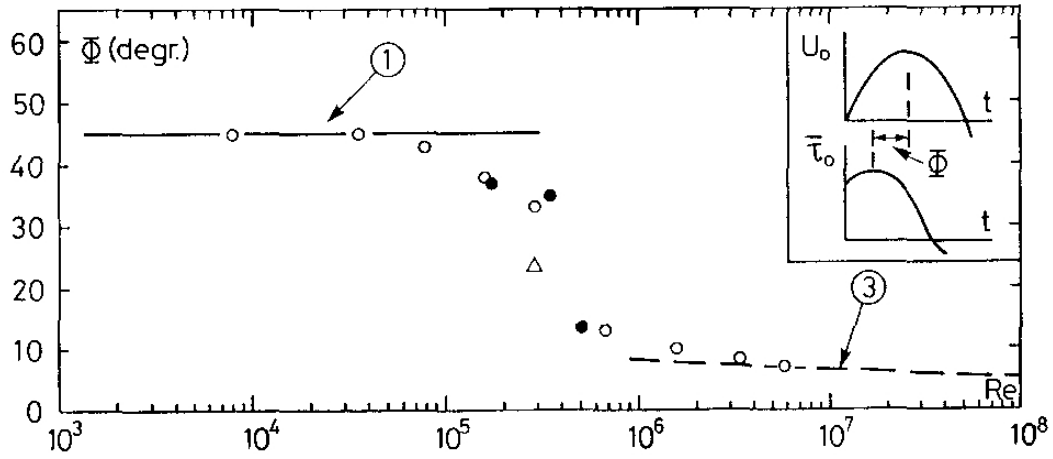


Figure 17. Variation of phase lead of peak wall shear stress (denoted ϕ) with wave Reynolds number (Re_w). Smooth bed. 1: Laminar solution. 3: Fredsoe's (1984) theoretical solution. From Jensen (1988).

Also, Jensen (1988) showed how the phase lead of peak wall shear stress changes with the roughness of the wall (Figure 18). Although more data would be desired, a general tendency can be observed, in which the phase lead becomes larger as the wall gets more rough, assuming that all experiments reported by Jensen (1988) in Figure 18 correspond to the turbulent regime. The tendency in that figure leans toward the value for smooth bed observed in Figure 17 for the turbulent regime. Later, Pedocchi and Garcia (2009) extended this analysis using more data compiled from several authors, showing the phase lead in the smooth-to-rough transition for the turbulent regime.

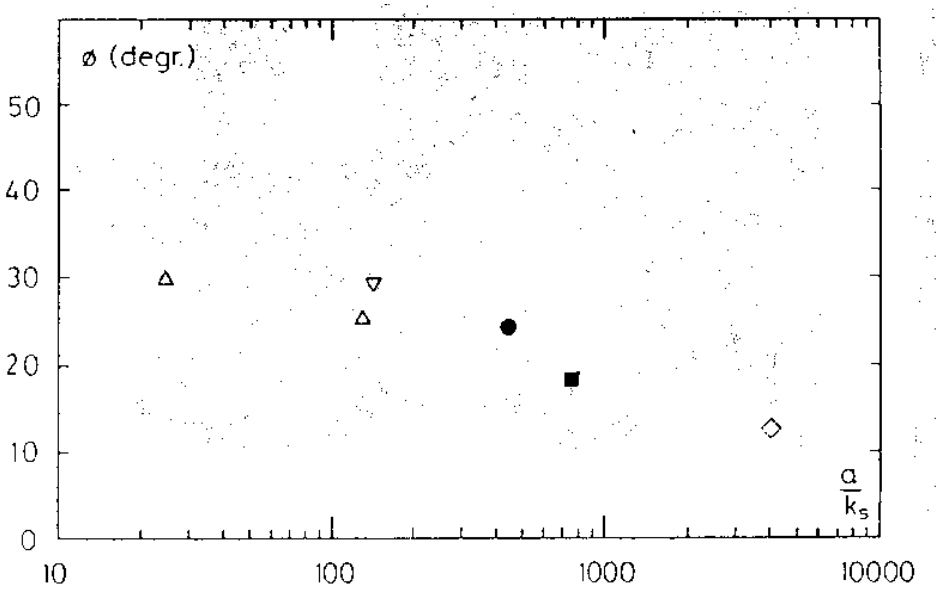


Figure 18. Variation of phase lead of peak wall shear stress (ϕ) with relative roughness (a/k_s). From Jensen (1988).

2.5.4 Wave friction factor

Kamphuis (1975) provided a friction factor diagram for oscillatory flows, similar to Nikuradse's or Moody's diagrams for unidirectional flows. In that diagram (Figure 19) the wave friction factor (f_w) was calculated using u_{out_max} and the maximum shear velocity of the cycle ($u_{*\,max}$). It must be noted that, although u_{out_max} occurs at $\omega t = 90^o$, that is not necessarily the case for $u_{*\,max}$, which typically occurs at a different phase in the cycle depending on Re_w, as will be shown later in the results.

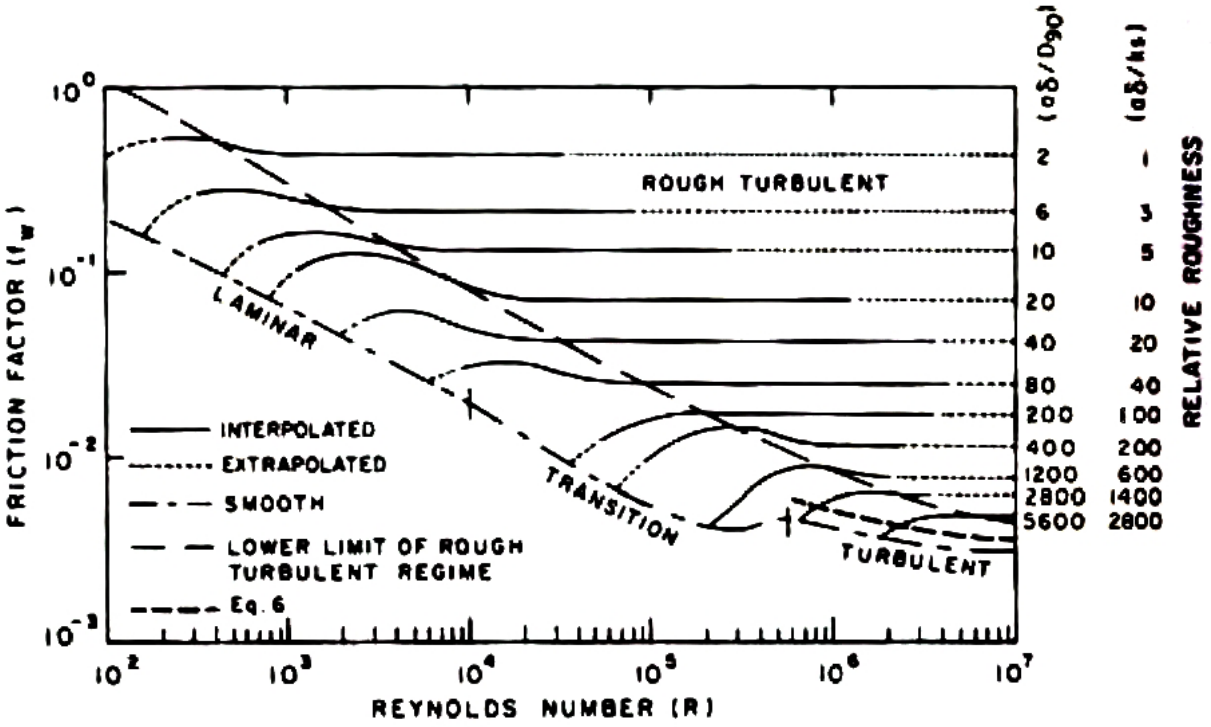


Figure 19. Wave friction factor diagram. From Kamphuis (1975).

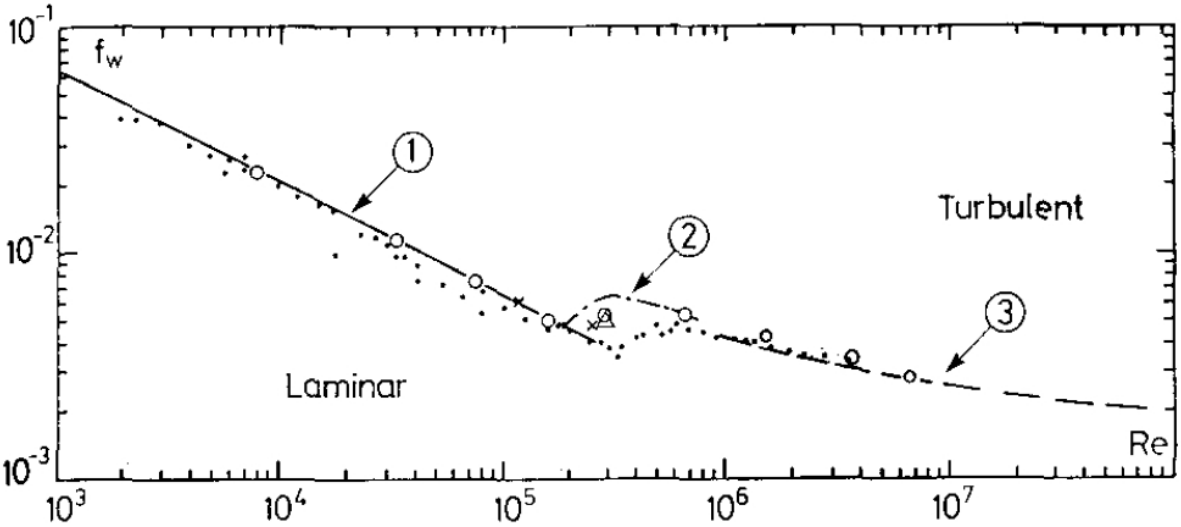


Figure 20. Evolution of wave friction factor (f_w) with wave Reynolds number (Re_w) for smooth bed. Lines: 1- laminar solution, 2- DNS from Spalart and Baldwin (1987), 3- Fredsoe (1984) theoretical solution. Symbols: dots -Kamphuis (1975), triangle - Hino *et al.* (1983), crosses - Sleath (1987) with 0.2 mm sand, hollow circles - Jensen (1988). From Jensen (1988).

Several years later, Jensen (1988) calculated f_w for his experiments and ploted the results in a similar diagram, but only for smooth bed (Figure 20). He also included data from Hino *et al.* (1983), Spalart and Baldwin (1987) and Sleath (1987), finding good agreement in the laminar and turbulent regimes. However, there was some dispersion in the data for the transition regime. More recently, Pedocchi and Garcia (2009) presented a compilation of experimental data from previous studies found in the literature and proposed an empirical expression to calculate the wave friction factor for the smooth-to-rough turbulent regime.

CHAPTER 3

RESEARCH GOALS

3.1 Issues identified from literature review

From the literature review, an important knowledge gap was found for the transition regime of the oscillatory boundary layer, for which very little experimental data is available. This is a complex regime, since flow characteristics change rapidly in a short range of Re_w. However, many geophysical flows of engineering interest are found to belong to the transition regime, so a deeper understanding of the flow behaviour in this regime is necessary.

In the particular case of smooth bed, only Hino *et al.* (1983), Jensen (1988) and Carstensen *et al.* (2010) reported experiments in the transition regime. Hino's data was obtained from one experiment only and using air as fluid, Jensen's measurements in the transition regime didn't include velocity data, only bed shear stress, while Carstensen's measurements were more focused on flow visualization. Also, most of the measurements reported in the literature were obtained for two velocity components only (typically u and w), which limited the ability to achieve a complete understanding of the turbulence characteristics of the flow. This situation evidenced the lack of a consistent data set with 3D velocity measurements covering the whole transition regime in detail.

Furthermore, some contradictory results have been reported in the literature by different authors regarding the occurrence of the bed shear stress peak in the transition regime. In some circumstances, this peak happens in advance of the outer flow velocity maximum (known as phase lead), while in others the peak happens after the outer flow velocity maximum (known as phase lag). The exact circumstances for either condition to occur are not clear in the literature, and so a more detailed set of experiments with time-resolved velocity measurements is necessary to understand this effect.

Another important characteristic of the transition regime is the appearance of turbulence as Re_w increases from the laminar regime to the fully turbulent regime. How turbulence develops in this transition is not clearly understood yet. Visualization experiments like the ones from Carstensen *et al.* (2010) and numerical models have helped in this regard. However, detailed measurements of the velocity fluctuations and turbulent intensities as they grow for increasing Re_w are still missing. Also, these types of measurements would help understand the contribution of turbulence to the changes observed in the mean velocity profiles and bed shear stresses between the laminar and turbulent regimes.

Finally, very little attention is given in the literature to the viscous sublayer inside the oscillatory boundary layer. In the laminar regime a solution for the thickness of the viscous sublayer exists, given by Rayleigh (1911). Also, Hino *et al.* (1983) reported some values from their experiment in the transition regime. However, it is not clear whether or not a viscous sublayer exists in these flows and for what phases of the flow. The main reason for this lack of attention could be the lack of detailed measurements close to the bed with enough temporal and spatial resolution to capture the flow inside the viscous sublayer.

3.2 Research questions

Based on the isses identified from the literature review, the experimental investigation presented in this work was designed to provide answers to the following research questions, in the context of the **transition regime** of the oscillatory boundary layer with **smooth bed**:

1- Understand the evolution of the mean flow characteristics of the oscillatory boundary layer as Re_w increases along the transition regime.

2- Understand the evolution of the turbulence characteristics of the oscillatory boundary layer as Re_w increases along the transition regime.

3- Explain the mechanism of bed shear stress generation and evolution along the oscillation cycle.

4- Clarify the occurrence of the bed shear stress peak during the oscillation cycle, and in particular whether it presents a phase lead or a phase lag with respect the outer flow velocity maximum.

5- Explain the mechanism of turbulence formation and evolution along the oscillation cycle, and investigate the relation with bed shear stresses.

6- Quantify the contribution of viscous shear stresses and turbulent shear stresses to the total shear stress along the oscillation cycle, and their effect on bed shear stress.

7- Analyze whether a viscous sublayer exists in oscillatory flows, and if so, for what phases along the cycle and what would be the thickness.

3.3 Experimental plan

In order to answer those questions an experimental plan was designed based on the information available in the literature and some preliminary tests performed by the author. A summary of the nominal experimental conditions is shown in Table 1.

	Flow conditions				Roughness conditions			
	T	$2a$	u_{out_max}	Re_w	*Bed*	D	k_s	a/k_s
Exp no.	(s)	(m)	(m/s)	(-)	(-)	(mm)	(mm)	(-)
1	10	0.500	0.157	3.9E+04	Fixed	-	-	Smooth
2	10	0.800	0.251	1.0E+05	Fixed	-	-	Smooth
3	10	1.000	0.314	1.6E+05	Fixed	-	-	Smooth
4	10	1.200	0.377	2.3E+05	Fixed	-	-	Smooth
5	10	1.300	0.408	2.7E+05	Fixed	-	-	Smooth
6	10	1.400	0.440	3.1E+05	Fixed	-	-	Smooth
7	10	1.600	0.503	4.0E+05	Fixed	-	-	Smooth
8	10	1.800	0.565	5.1E+05	Fixed	-	-	Smooth
9	10	2.100	0.660	6.9E+05	Fixed	-	-	Smooth
10	10	2.500	0.785	9.8E+05	Fixed	-	-	Smooth

Table 1. Nominal conditions for the experiments in this work.

The oscillatory boundary layer experiments in this work were designed to be carried out over a smooth bed, using water as fluid (kinematic viscosity, $\nu \approx 10^{-6}$). The oscillation period (T) selected was 10 s, and the amplitude of the oscillation ($2a$) varied between 0.5 and 2.5 m. This resulted in a range of wave Reynolds numbers (Re_w) from 4×10^4 to 1×10^6, extending from the laminar regime to the turbulent regime, **covering the entire transition regime in detail**. Ten experiments were planned in this range, uniformly spread around the center of the range to be able to capture the highly non-linear evolution of the boundary layer parameters along the transition.

A plot of wave Reynolds number (Re_w) against relative wall roughness (a/k_s) showing the different regions characterizing the oscillatory boundary layer is shown in Figure 21. This plot was originally proposed by Kamphuis (1975), and then extended and improved by other authors. The ten experiments presented in this work are included in the smooth wall transition region in that plot (labeled "LOWST + LDV"). Also, previous experiments from the literature performed by Jonsson (1980), Hino *et al.* (1983), Sleath (1987) and Jensen (1988) are shown. A few of Jensen's experiments covered this same region, however, only bed shear stress measurements were recorded. As it turns out, no other studies in the literature have reported before such a detailed experimental set of velocity measurements in this important region of the oscillatory boundary layer as the experiments presented in this work.

Additionally, other experiments performed under different experimental conditions by the author (labeled "SOT + LDV") and by the author with other collaborators (labeled "SOT + PIV", see Mujal *et al.*, 2014) are shown in the plot for the sake of completeness; however their results won't be included in this work.

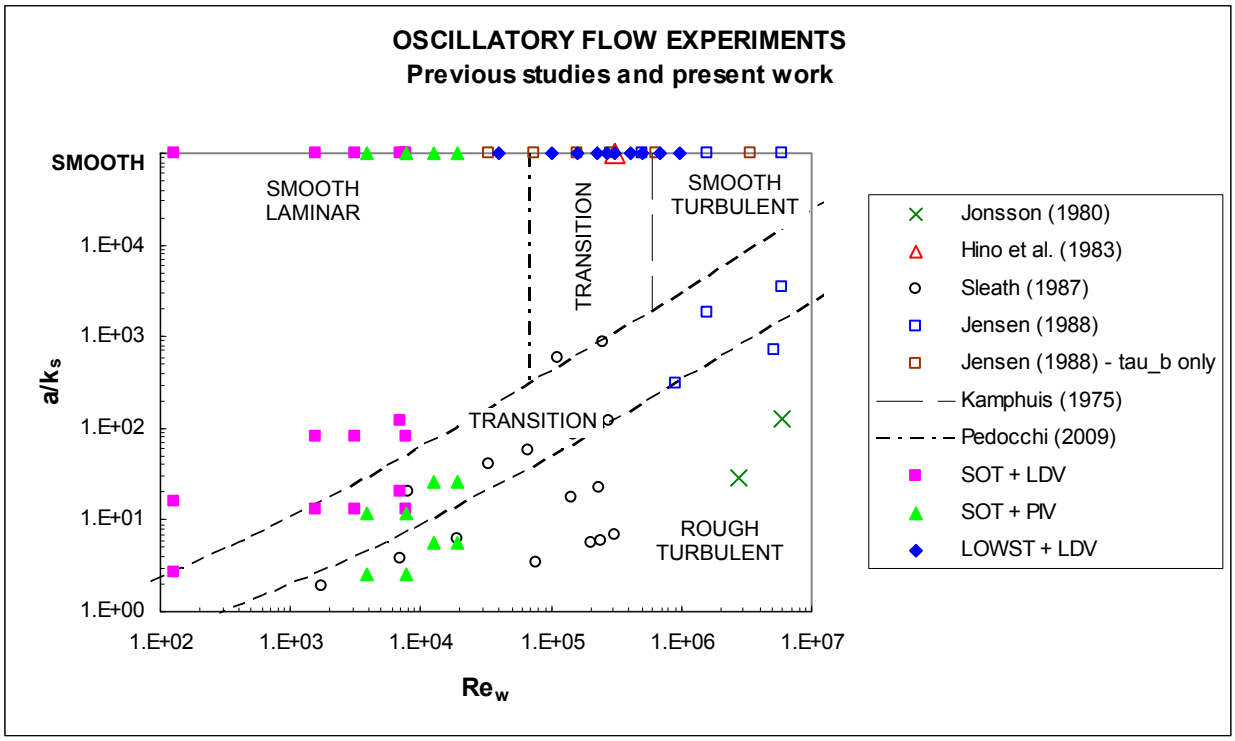


Figure 21. Plot of wave Reynolds number (Re_w) against relative wall roughness (a/k_s), showing previous experiments found in the literature and the experiments performed in this work.

In order to achieve the high Re_w values required, a unique experimental facility capable of high oscillatory flow speeds was used, as is the case of the Large Oscillatory Tunnel (LOWST) in the Ven Te Chow Hydrosystems Laboratory. Also, the use of a state-of-the-art 3D laser Doppler velocimetry (LDV) system allowed all three components of velocity to be measured along a vertical profile inside the boundary layer, with high spatial and temporal resolution. The combination of these factors made possible to obtain the highly-accurate and time-resolved measurements needed to be able to answer the research questions stated.

From the velocity measurements several variables of interest were calculated for all the experiments. These variables provided insightful information about the mean flow and turbulent characteristics of the oscillatory boundary layer and are presented in the results, namely:

- Mean flow velocity
- Boundary layer thickness
- Velocity fluctuations

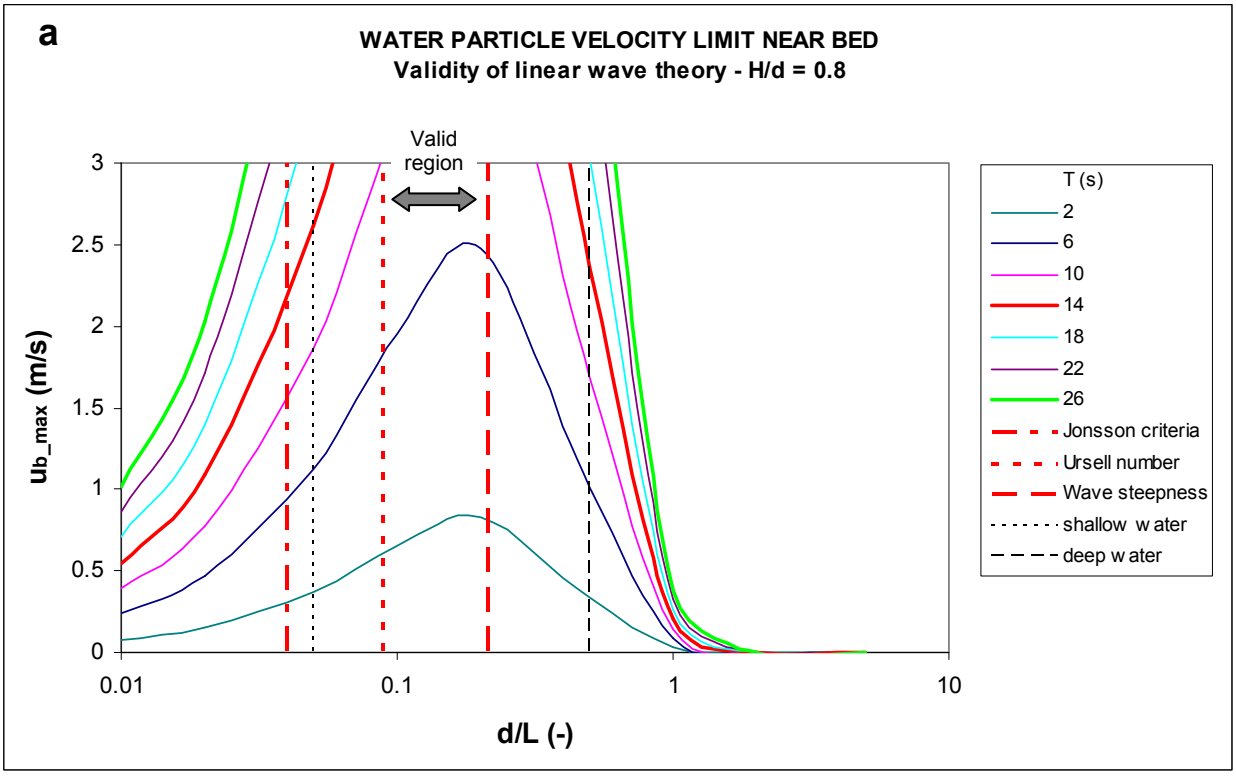
- Turbulence intensities
- Turbulent kinetic energy
- Reynolds shear stress
- Viscous shear stresses
- Total shear stresses
- Turbulence production
- Eddy viscosity
- Skewness of velocity fluctuations
- Quadrant analysis: sweeps and ejections
- Bed shear stress
- Shear velocity
- Wave friction factor
- Viscous sublayer thickness

3.4 Applicability of experiments to coastal environments

The experiments performed can be representative of the conditions occurring in the field, at the bottom of the ocean, for multiple combinations of wave height and depth. Linear wave theory can be used to obtain an expression for the maximum horizontal water particle velocity near the bed (u_{b_max}), which would be equivalent to the outer flow velocity in the oscillatory boundary layer (u_{out_max}):

$$u_{b\,\max} = \frac{\pi H}{T}\left[\frac{1}{\sinh kd}\right] \tag{27}$$

where H is the wave height, T is the wave period, d is depth, and k is the wave number such that $k = 2\pi / L$, where L is the wave length. This equation can be plotted (Figure 22) to obtain lines of constant T that relate u_{b_max} with the relative depth (d/L), for a given ratio of relative wave height (H/d).



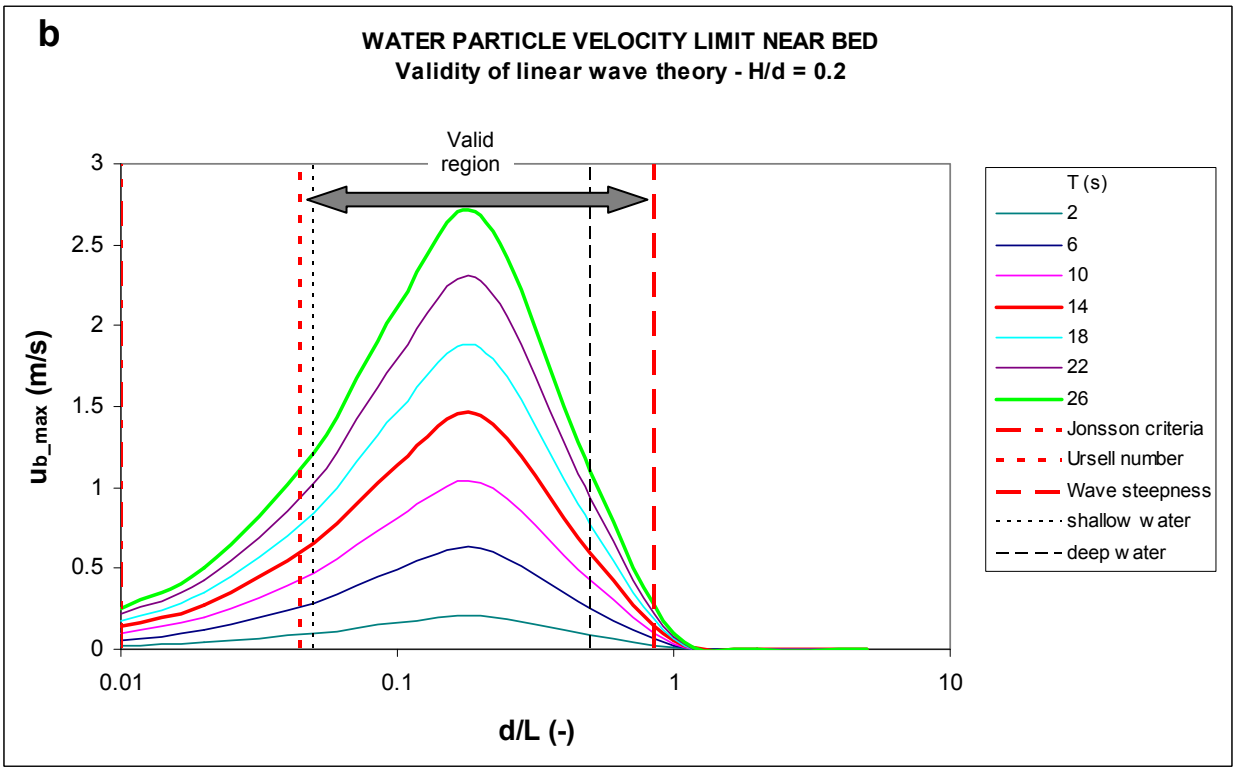


Figure 22. Near-bed maximum velocity (u_{b_max}) as a function of relative depth (d/L), for each period (T), and for the particular cases of relative wave height H/d = 0.8 (a) and H/d = 0.2 (b). Linear wave theory is valid under the curves for each period, and within the limits given by the criteria specified.

In order for linear wave theory to hold valid, several criteria need to be satisfied:

a) Breaking wave limit: specified by a maximum value of relative wave height, beyond which the wave is considered to start breaking: $H/d < 0.8$
b) Wave steepness: specified by the ratio of wave height to wave length, such that $H/L < 0.17$
c) Ursell number (Ursell, 1953): which established a ratio between relative wave height and relative water depth, such that: $HL^2 / d^3 < 100$
d) Jonsson criterion (Jonsson, 1980): which is similar to the Ursell number, but less restrictive: $HL / d^2 < 20$

All these conditions make for a highly complex validity region with a three-dimensional shape in the u_{b_max} - d/L plane. In Figure 22 the valid region is below the curves of constant T, and between the vertical lines corresponding to the Jonsson, Ursell and wave steepness criteria (in fact, the Ursell number is more restrictive than Jonsson's criterion, and so the latter doesn't play a role). One of the extreme, but yet possible, conditions where linear theory is still valid happens at the threshold of all these criteria. In this regard, Figure 22a has been created for the case when $H/d = 0.8$, resulting that from Jonsson $d/L > 0.04$, from Ursell $d/L > 0.09$ and from wave steepness $d/L < 0.21$. If a more common, less extreme, relative wave height is used, such as $H/d = 0.2$, the values of u_{b_max} for each T curve are reduced, while the limits become wider (Figure 22b).

Rearranging equation (27), a dimensionless expression can be obtained that relates u_{b_max} with the relative depth (d/L):

$$\frac{u_{b\max}}{(H/T)} = \frac{\pi}{\sinh 2\pi(d/L)} \tag{28}$$

This equation can be plotted (Figure 23), showing the collapse of all the lines of constant T into a single line, which is also independent of the relative wave height parameter H/d. The criteria for the limits are still dependent on H/d though.

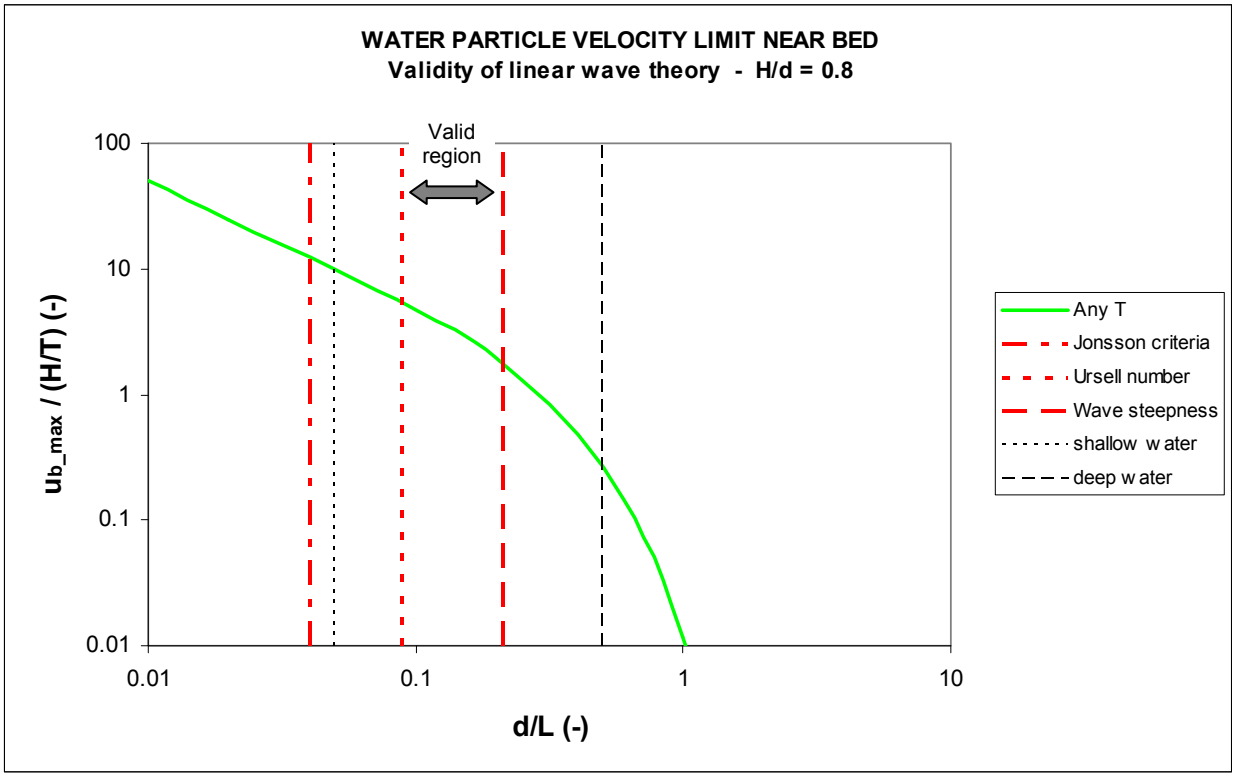


Figure 23. Dimensionless near-bed maximum velocity ($u_{b\_max} / (H/T)$) as a function of relative depth (d/L), for any period (T). Limits are shown for the particular case of relative wave height $H/d = 0.8$. Linear wave theory is valid at the T curve, and within the limits given by the criteria specified.

From this analysis, a more practical plot can be constructed in which the reader can quickly find the combination of wave height (H) and water depth (d) that produce a particular near-bed velocity condition. For that purpose, equation (28) was used in combination with the dispersion relation, which relates water depth with wave length and period:

$$\omega = \sqrt{gk \tanh(kd)} \qquad (29)$$

where ω is the angular frequency, such that $\omega = 2\pi / T$, and g is the acceleration of gravity. Due to the implicit nature of the two equations, there is no explicit solution, so they need to be solved numerically. Figure 24 shows wave height (H) as a function of depth (d), for different values of u_{b_max} (all with dimensions), for a wave period of T = 10 s. Validity limits are also shown, corresponding to each one of the threshold values mentioned before. With this plot it is possible

to find all the possible combinations of H and d that produce a particular u_{b_max} at the bottom of the ocean.

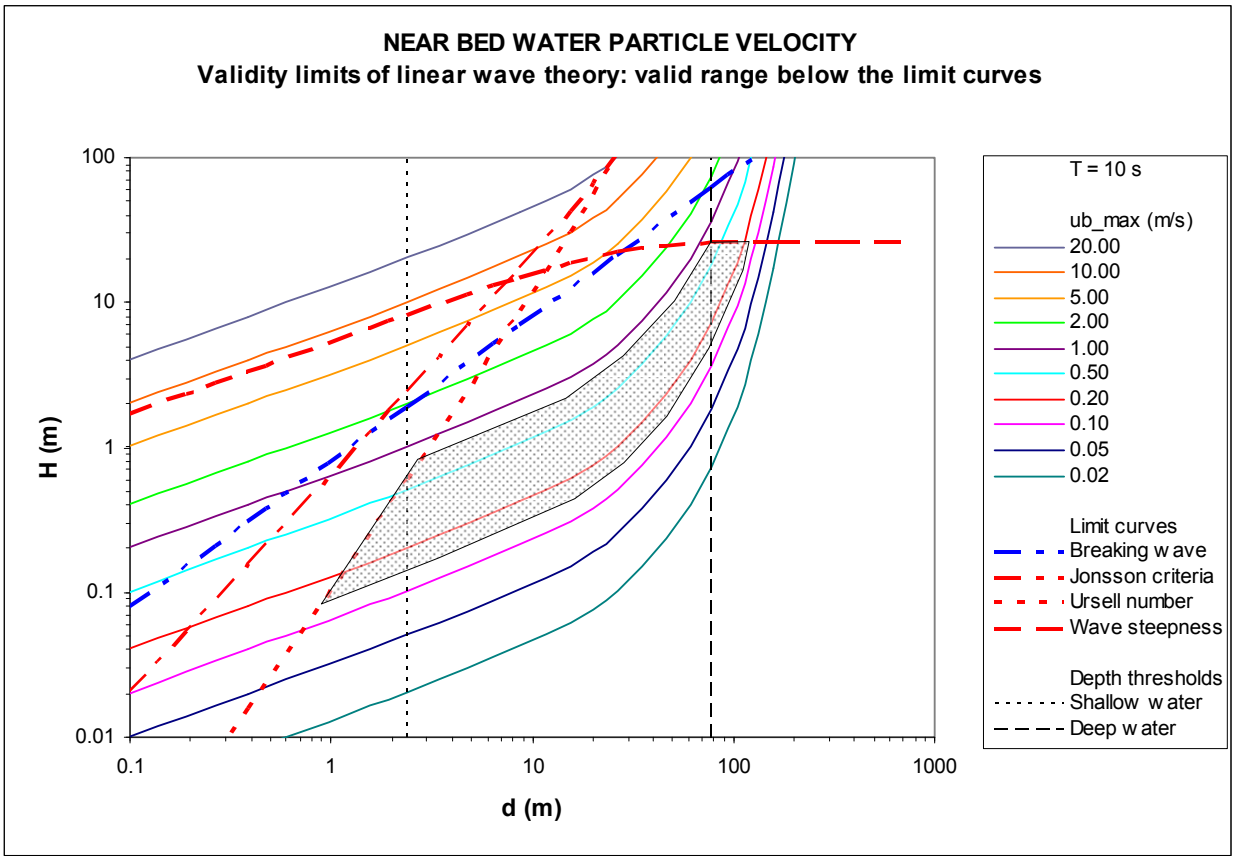


Figure 24. Validity of experiments within linear wave theory. Lines of constant near-bed maximum velocity (u_{b_max}) as a function of wave height (H) and depth (d) in the ocean, for a period of T = 10 s. Shaded region indicates range where the experimental results of this work apply.

The experiments performed in this work stay within a range of maximum oscillatory velocities of u_{out_max} = 0.157 - 0.785 m/s for an oscillation period of T = 10 s. These conditions can be found in the ocean for many different combinations of depth (d) and wave height (H), as shown in Figure 24. This way, one particular experiment is useful to represent the oscillatory boundary layer of any combination of H and d along its corresponding line of constant near-bed maximum velocity (u_{b_max}). For example, a condition with depth d = 9 m and waves of height H = 1 m, will produce the same u_{b_max} in the bottom of the ocean as a condition with depth d = 78 m and waves of height H = 18 m. Note that the plot will change for a different period of oscillation (T). This plot is only valid within the limits of linear wave theory, indicated by the limit curves. A

combination of H and d that falls outside of the limit curves will not be properly reproduced by the experiments.

CHAPTER 4

EXPERIMENTAL FACILITIES AND INSTRUMENTATION

4.1 Large Oscillatory Water-Sediment Tunnel (LOWST)

4.1.1 General characteristics

The Large Oscillatory Water-Sediment Tunnel (LOWST) in the Ven Te Chow Hydrosystems Laboratory was designed to study flow characteristics and sediment transport processes under controlled wave-current flows similar to the ones found in the bottom boundary layer of the ocean.

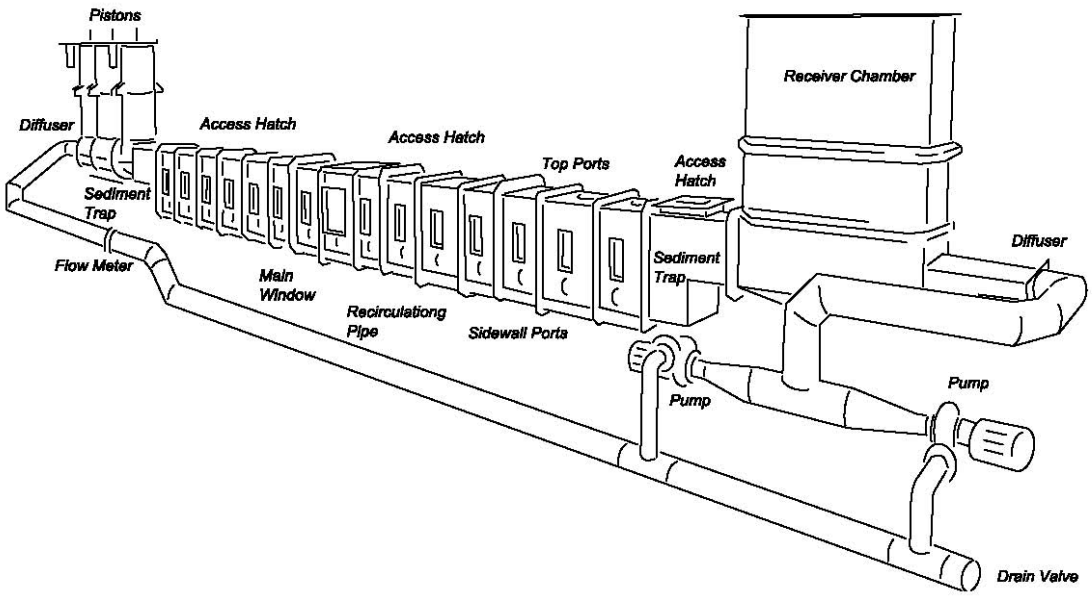


Figure 25. Sketch of the LOWST. From Pedocchi (2009).

The tunnel was made by Engineering Laboratory Design Inc. and MTS Systems Corporation in 2002. The walls are built on a composite sandwich construction of glass reinforced polyester with a wooden core, and structural support is provided by steel beams. Numerous Plexiglas® windows along the side and top walls allow access for instrumentation and for visual observation. Three access hatches on the top allow a person to get in for the preparation of

experiments. The test section is 12 m long and the internal dimensions of the cross-section are 0.8 m wide by 1.2 m high. The tunnel is ready to contain sediment in its bottom half (0.6 m) for movable bed experiments (Figure 26).

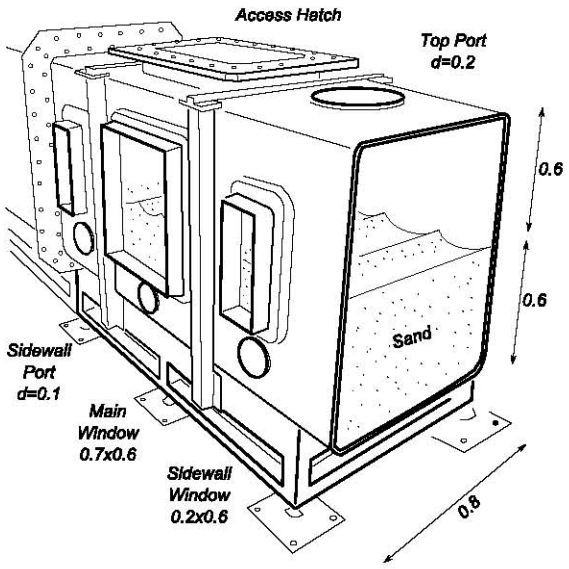


Figure 26. Detail showing a representative cross-section of the LOWST (dimensions in m). From Pedocchi (2009).

The oscillatory motion of the water is driven by three pistons that run inside 0.78 m diameter cylinders with a maximum stroke of 1.37 m. At the opposite end of the tunnel, a 1.0 m by 2.0 m holding tank open to the atmosphere acts as a passive receiver for the water displaced by the pistons. Three servo motors, controlled by a computer, drive the pistons using a screw-gear system. The facility also has two centrifugal pumps that allow for the superposition of a unidirectional current to the oscillatory motion through a pipe recirculation system, up to 0.5 m/s. Flow straighteners and sediment traps are placed at both ends of the main test section.

Thanks to its many access ports and transparent windows, different instruments can be used in the LOWST facility to perform various measurements of the physical processes of interest taking place inside. In particular, the tunnel is equipped with a few onboard instruments: an ultrasonic velocity profiler (UVP) and an acoustic Doppler velocimeter (ADV) for velocity measurements, a pencil beam sonar for bathymetric measurements, and a pressure transducer. A detailed description of these instruments can be found in Pedocchi (2009). However, they were not used

for this work. Instead, a laser Doppler velocimeter (LDV) was used to measure velocities from the outside of the flume, so that the flow was not affected by the instrument.

Figure 27. General view of the LOWST and the LDV system at the Ven Te Chow Hydrosystems Laboratory, UIUC.

4.1.2 Range of operation

The LOWST can be operated for oscillatory flows only, unidirectional flows only or a combination of both. In the oscillatory flow configuration, the maximum amplitude of the oscillation in the test section is $2a$ = 4.1 m, and the maximum velocity is 2 m/s at a period of about T = 6.2 s. The maximum velocity is limited by the amplitude for the long period cases (according to the formula for a sinusoidal wave $u_{max} = 2\pi a / T$), while for the short periods the maximum velocity is limited by the acceleration of the pistons. There is also an absolute speed limit of 2 m/s (see Figure 28). With these values, the maximum wave Reynolds number that could be tested in the LOWST is $Re_w \approx 4\text{x}10^6$.

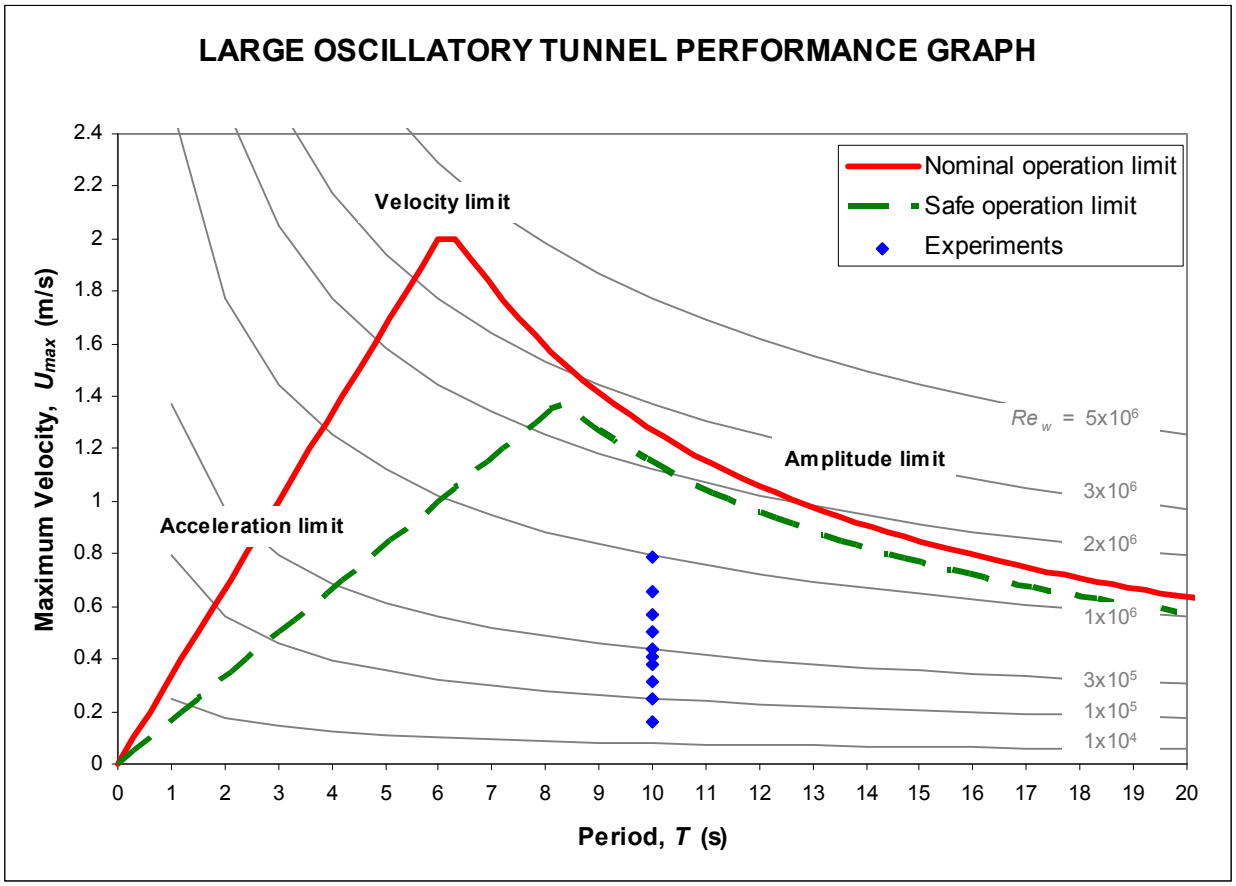


Figure 28. Operational range of the LOWST for oscillatory flows, showing nominal and safe operation limit curves. The conditions used for the experiments presented in this work are also displayed. Lines of constant Re_w are included for reference.

However, these operational limits shouldn't be reached in any circumstance, since damage could occur due to the extreme forces at play. In particular, the acceleration limit is the most critical since fast accelerations introduce very high pressure gradients that could compromise the structural integrity of the tunnel. On the contrary, the amplitude limit is not very critical, since it only represents a physical limitation due to the available stroke of the pistons. The velocity limit is mainly due to the machinery used to move the pistons, and it is hardly ever reached. With these considerations in mind and from the experience acquired while using the LOWST, a more conservative set of limits is provided in Figure 28 to ensure safer operation and longer life of the facility. In this safer situation, the maximum velocity in the measurement section would be about 1.3 m/s for a period of T = 8.4 s with amplitude $2a$ = 3.5 m, and the maximum wave Reynolds

number would be $Re_w \approx 2x10^6$. All the experiments performed in this work fall inside the safe operation limit of the LOWST.

4.2 Laser Doppler velocimetry (LDV) system

Warning: when using LDV systems, safety precautions must be taken to avoid personal injury to the eyes or skin due to the high intensity of the laser light.

4.2.1 Overview of the LDV technique

The following is only an overview of the LDV technique. More information can be found in the following references: Albrecht *et al.* (2003), Tropea *et al.* (2007), TSI Inc. (2005).

4.2.1.1 Introduction

Laser Doppler Velocimetry (LDV) is a technique used to measure instantaneous velocity at a point in fluid flows. It is commonly used for velocity measurements in water and air flows, fuel injection and spray research, and even in the biomedical field for blood flow. LDV is particularly suited for the measurement of fast moving turbulent flows, and for flow measurements near the boundaries. Also, the fact that it is an optical instrument allows for non-intrusive measurements, thus avoiding any disturbances to the flow. Availability of optical access to the flume is required, as well as taking into account possible refraction issues occurring when measuring through windows. Furthermore, it provides very fine spatial resolution (on the order of 0.1 mm), which allows the measurement of velocities very close to the boundaries, and in particular to the bed, which is essential to resolve the velocity gradients and profiles from which shear stress calculations are made. Due to its high temporal resolution (upwards of 10,000 Hz, provided that appropriate seeding is present), the high frequencies of the flow are also preserved, thus allowing for the analysis of turbulent characteristics, especially within the boundary layer. In addition, the fact that its measurement principle is based on geometric considerations results in very accurate velocities, with no need for calibration, since the optical arrangement is factory-set.

Measurements performed with LDV are single-point time series of instantaneous flow velocity. In order to obtain measurements at different points, the measurement volume needs to be traversed inside the flume, which makes this technique very time-consuming. After a complete set is obtained, the data can be analyzed to obtain different flow quantities and statistics. A multi-component LDV system can measure the mean velocity field and the entire turbulence stress tensor at a point. However, spatial correlations or tracking of flow structures is not typically possible with LDV since multiple points are not measured all at the same time.

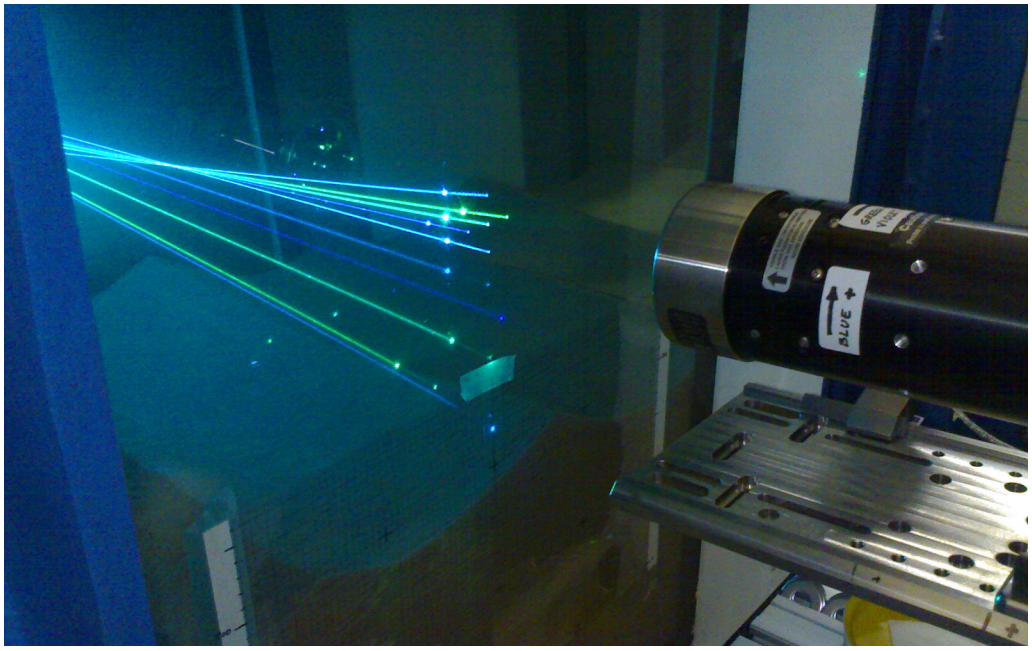


Figure 29. LDV measurements using a 5-beam probe.

4.2.1.2 Working principle

Typical LDV systems consist of a laser, a set of optical components, a probe, a photo detector, and processing electronics. The measurement principle is based on the Doppler effect related to the frequencies of the signals produced by seeding particles when crossing the measurement volume. The measurement volume is located at the intersection of a pair of laser beams, where they interfere with each other, creating a fringe pattern (Figure 30). The spacing of these fringes (Δs) is accurately known from the optical properties of the system. Small seeding particles in the fluid cross those fringes, scattering the laser light with a particular frequency. The scattered light is collected by the probe and then analyzed by the electronics to obtain that frequency ($f = 1/\Delta t$).

From the spacing of the fringes and the frequency of the scattered light, the particle velocity perpendicular to the fringes is computed as $V = \Delta s / \Delta t$. By using multiple pairs of beams at the same time, all components of velocity can be measured at a point.

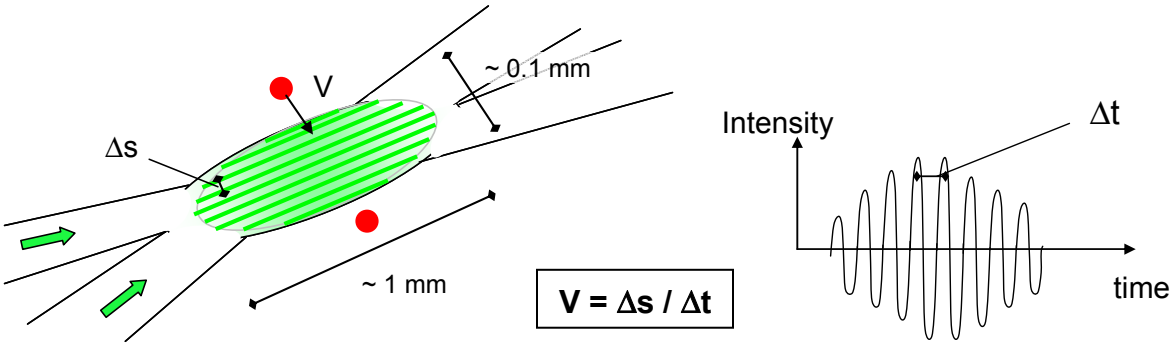


Figure 30. Sketch of the LDV measurement volume and signal generated by the seeding particles, from which velocity is computed.

4.2.1.3 Seeding particles

LDV experiments need the addition of seed particles to the fluid to enhance the light scattering properties and achieve greater data rates. Common types of seeding used in LDV systems are Hollow Glass Spheres (HGS), with average density 1.1 g/cm^3 and mean diameter 11 μm, and Silver-coated Hollow Glass Spheres (S-HGS), with average density of 1.7 g/cm^3 and mean diameter of 14 μm (Potter Industries Inc., 2008). In order for seeding particles to be suitable for LDV measurements, they must have good light reflection properties and, at the same time, they must be able to follow the flow at all turbulent time scales present. In addition, they must be of a suitable size for the optics configuration of the LDV. Ideal candidates for liquid flows are usually neutrally buoyant particles (or close), that are big enough to generate large intensity backscatter signals, and small enough to meet the turbulence criteria.

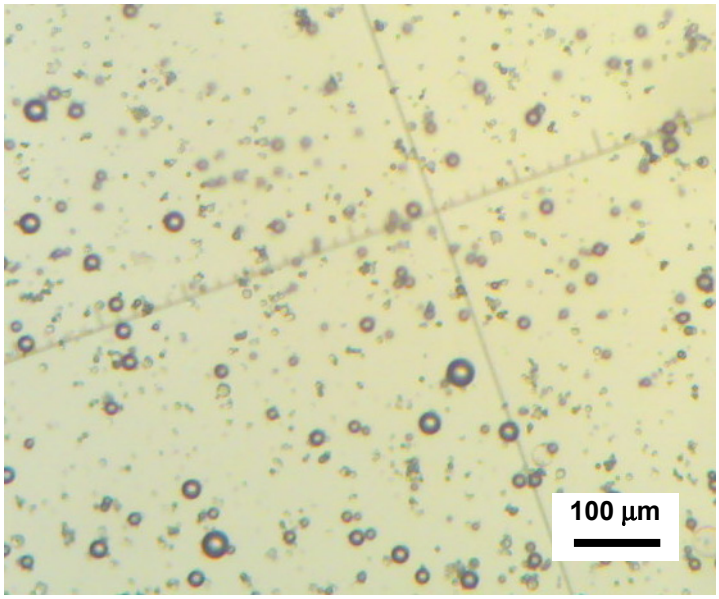


Figure 31. Hollow glass spheres (HGS) as seen under the microscope.

An optimum concentration of seeding in the flume can be found for which the LDV data rate is maximized. In this regard, Mier and Garcia (2009) performed a series of experiments in a water tank using HGS and S-HGS to determine the seeding concentration that results in the maximum possible data rate (samples per second). They observed that the data rate increases linearly with seeding concentration to a peak value, beyond which, additional seeding causes the data rate to decay exponentially. Indeed, linear growth in data rate is expected when seeding is added since more individual particles cross the measurement volume, in agreement with Albrecht *et al.* (2003) and Tropea *et al.* (2007). However, when seeding concentration becomes too high, the medium becomes so turbid that both the incident laser light and the scattered reflections are attenuated as they travel through the medium, causing a decline in data rate. This is shown in Figure 32, which plots normalized data rate against a dimensionless number volume (Nv^*) for the two types of spheres and for various laser beam path lengths. Nv^* represents seeding concentration and is defined by:

$$Nv^* = Nv \cdot e^{\alpha d_w} \tag{30}$$

where Nv is the expected number of particles within the LDV measurement volume, d_w is the distance the laser beams travel through the seeded medium from the flume wall to the

measurement volume (in m), and α is an attenuation coefficient (in m^{-1}), which varies with seeding type. Mier and Garcia (2009) experimentally determined the values of α for their experiments as $\alpha_{HGS} = 7.87\ m^{-1}$ and $\alpha_{S\text{-}HGS} = 5.75\ m^{-1}$.

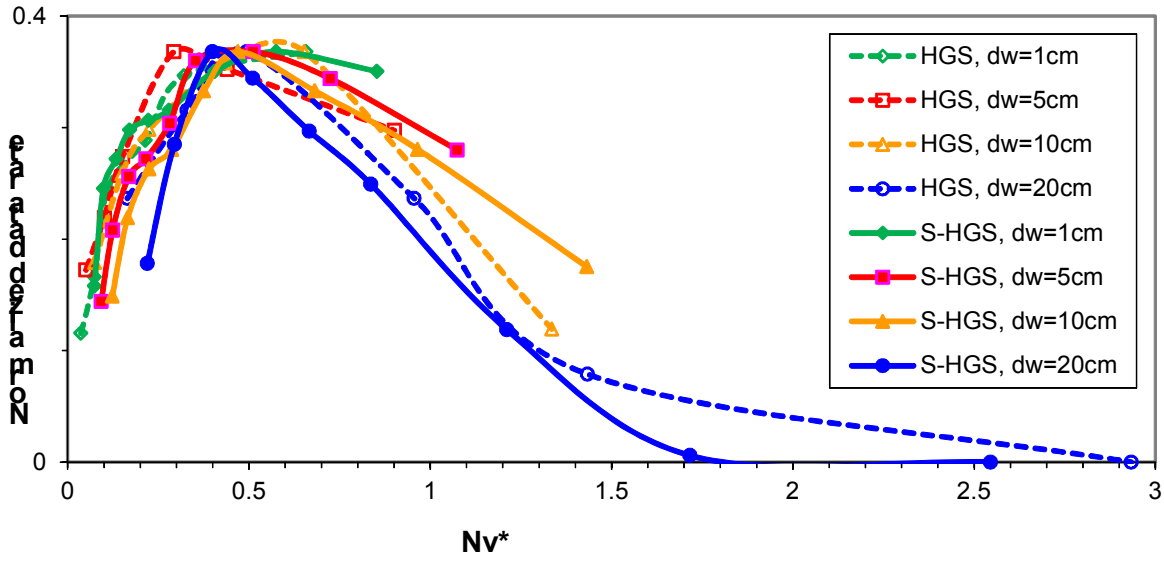


Figure 32. Experimental results of normalized LDV data rate for increasing seeding concentrations (Nv^*) in a water tank, for two different seeding types (HGS and S-HGS) and four distances to the measurement volume inside the seeded medium (d_w = 1, 5, 10, 20 cm). Adapted from Mier and García (2009).

The work of Mier and Garcia (2009) reveals optimal data rates for values of Nv^* in the range of 0.4 to 0.5. This result is independent of seeding type, path length and the optical configuration of the probe since Nv^* accounts for these parameters. Based on their results, they developed a relation to calculate the specific amount, in terms of mass of seeding particles (M_p) needed for a certain experiment involving LDV measurements:

$$M_p = 0.45 \frac{\rho_p V_1 V_T}{V_m \cdot e^{\alpha d_w}} \qquad (31)$$

where ρ_p is the density of the seeding particles, V_1 is the volume of one seeding particle (which can be obtained from the diameter assuming spherical shape), V_m is the LDV measurement volume and V_T is the total volume of water in the flume.

4.2.1.4 Optical refraction

Different lens combinations of the LDV probes allow for longer focal distance (F) in order to perform measurements at farther distances inside the flume. The optical probe usually sits outside of the flume (although there are some applications in which submersible probes are used) and the laser beams travel from the probe through air, through the flume's window, and then through water until they cross each other, creating the measurement volume inside the flume. In order to take the measurements at the right location, the refraction of the beams when traveling through different media (air, window and water) needs to be taken into account using Snell's law:

$$n_1 \cdot \sin k_1 = n_2 \cdot \sin k_2 \tag{32}$$

where n is the refractive index of each medium (air = 1, water = 1.33, glass = 1.52, Plexiglas® = 1.49), and k is the angle of incidence of the beams with respect to the interface perpendicular. This way, a new corrected focal distance (F^*) can be calculated. For the case when the probe axis is perpendicular to the window, the following equation can be used:

$$F^* = F\frac{\tan k_a}{\tan k_w} + d_g\left[1-\frac{\tan k_g}{\tan k_w}\right] + d_a\left[1-\frac{\tan k_a}{\tan k_w}\right] \tag{33}$$

where d is the distance traveled in each medium (a = air, g = window, w = water) such that $F^* = d_a + d_g + d_w$.

An additional degree of complexity is associated with traversing an LDV measurement volume inside a flume (for instance, to obtain a velocity profile). When a probe is moved to a different location, the distances traveled by the beams through the different mediums may change, affecting the refracted focal distance (F^*). For single-probe measurements this issue is quickly resolved by adjusting the traverse displacements to match the desired locations of the measurement volume inside the flume. However, when performing multi-component LDV measurements using two probes, the measurement volume of each probe may experience a

different displacement inside the flume, even if both probes are attached to the same traversing mechanism. A special set-up involving a refraction-correcting device can be used in these situations to compensate the distortions and maintain coincidence in all components (Booij, 2003; Mier and Garcia, 2012).

4.2.1.5 Measurement volume

The LDV measurement volume has an ellipsoid shape, and its dimensions vary with the type of lens used. An average value of the diameter of the measurement volume is 0.1 mm and an average value of its length is about 1 mm, which results in a very small volume indeed (about 0.01 mm^3). Unlike other measurement techniques, the extremely small size of the measuring spot makes LDV an ideal candidate for the measurement of local instantaneous velocities and characterization of flow turbulence.

4.2.1.6 Transformation matrix

Sometimes the directions of measured LDV velocity components ($\vec{V}_1, \vec{V}_2, \vec{V}_3$) do not match the directions of the velocity components in the reference coordinate system ($\vec{u}, \vec{v}, \vec{w}$). For example, when measuring close to the bed, it is convenient to tilt the probe forward a few degrees so that the LDV beams are able to reach the measurement volume inside the flume without interference from bed features. Other times, several probes may be used together at non-orthogonal angles in order to perform multi-component measurements. In such situations, a spatial transformation is required to project the measured velocity components from the instrument coordinate system into the reference coordinate system:

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = M \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} m_{1x} & m_{1y} & m_{1z} \\ m_{2x} & m_{2y} & m_{2z} \\ m_{3x} & m_{3y} & m_{3z} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{34}$$

where M is called the transformation matrix. To obtain the velocity components in the reference coordinate system, equation (34) must be resolved by inverting matrix M. It is important to remember that when the measurement volume is inside a different medium (such as water), the

angles of the measured velocity components ($\vec{V}_1, \vec{V}_2, \vec{V}_3$) may be affected by refraction, and the refracted angles should be used in the transformation matrix. Examples of transformation matrices for different probe configurations can be found in TSI Inc. (2005).

4.2.1.7 Data quality and uncertainty

LDV systems require no calibration, although alignment of the beams is very important to obtain high quality measurements. Strictly speaking, the error associated with LDV measurements is typically very small, in the order of 1 % (Albrecht *et al.*, 2003). However, other factors can also introduce error in LDV measurements, mainly the positioning of the measurement volume, seeding particles, and signal processing. Relative errors within ±2% for higher-order statistics and as low as ±0.3 % for the mean velocities are commonly reported when the measurements are compared with direct numerical simulations, like the experiments in open-channel flows by Niederschulte *et al.* (1990) or in pipe flows by Durst *et al.* (1995) among others.

In LDV systems, however, there is need for proper tuning of the laser sub-system, beam alignment, careful coordinate positioning, optimization of electronics and software settings, and patient operation. In this regard, it must be said that maintaining proper operating procedures and high signal quality standards according to the manufacturer recommendations (see TSI Inc., 2005) ensures low noise readings and high burst efficiencies, preserving the accuracy of the velocity measurements with little operator-dependent influence on the results. This quality usually comes at a cost, though, since the filtering process greatly reduces data rates. For most applications it is typically recommended that the LDV system be operated such that a “high” signal-to-noise ratio is achieved and the appropriate band-pass filter and burst threshold for each channel are selected so as to ensure the highest possible bursting efficiency at each location. Values of bursting efficiency greater than 90 % on average guarantee very good quality of the data collected. In addition, multi-component coincidence with a burst gate overlap of less than 250 % is recommended to ensure good burst correlation between the velocity components. This will further ensure the high quality of the velocity measurements taken.

4.2.1.8 Laser safety

Lasers used in LDV systems are typically classified as “class 4”, meaning very high intensity of light, and can cause severe damage to persons and other equipment. Hazards include mainly the following:

- severe damage to skin and eye tissue can occur from direct exposure to the laser beams, as well as eye damage from specular or diffuse reflections.
- electrical shock or electrocution could result from contact with the high voltage power supplies used in these systems.
- lasers can create a fire when pointed towards or used near flammable materials.

Safety of operation is a priority when using laser equipment.

4.2.2 LDV system characteristics

The LDV system used in this work was manufactured by TSI Inc. in 2007. It has an Ar-ion 6W multiline laser (model Stabilite 2017, from Spectra-Physics), that generates the light beam, which is directed towards a FiberLight™ multicolor beam separator box (model FBL-3). There, the main beam is split in two by means of a Bragg cell, which also introduces a slight frequency shift of 40 MHz on one of the resulting beams. After that, the multi-color spectrum of the beams is divided into each individual color component using a prism, from which only the three main colors are used (green, blue and violet). Each one of these colors corresponds to one velocity component, so that up to 3D velocity measurements can be performed. The individual color beams then get directed and focused into an optical fiber cable by means of a coupler (model CPL2001), carrying the light to a transceiver probe (model TR-60 series), which operates both as a transmitter of incident light and as a receiver of backscattered light. A photo detector module (model PDM 1000) is responsible for converting the bursts of backscattered light into electric analog signals for each one of the velocity components separately. Then, a signal processor (model FSA-3500) takes the analog bursts and converts them into velocity measurements, which are in turn sent to a computer and analyzed with the FlowSizer™ software from TSI Inc. A set of

fiber optic and electronic cables interconnect all these devices to transmit light and electric signals throughout the system (Figure 33).

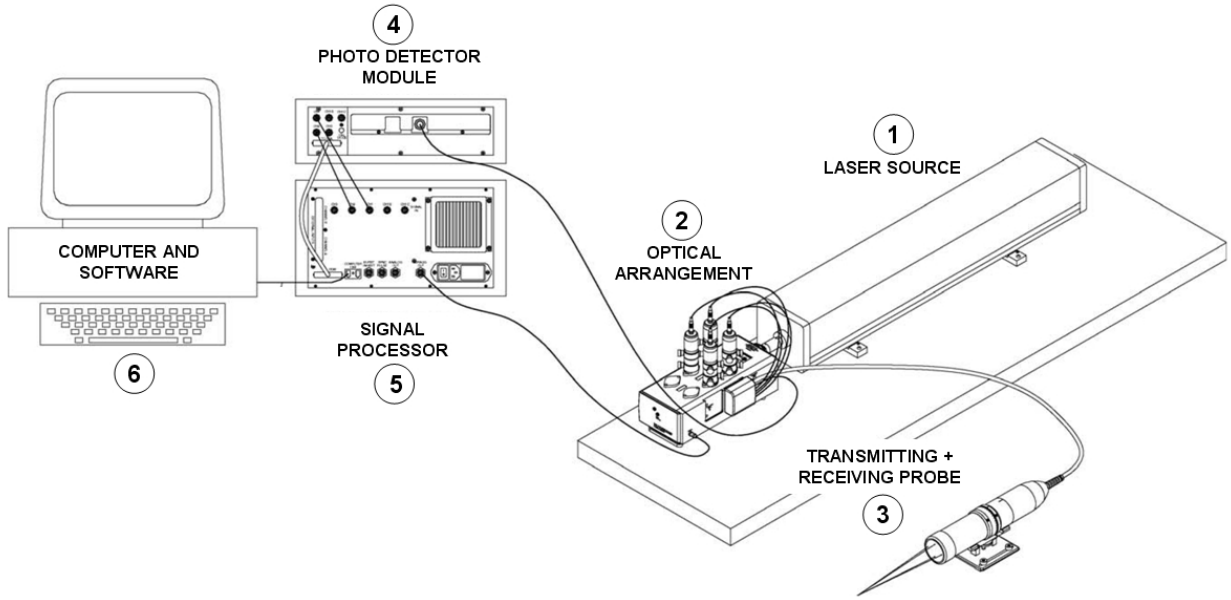


Figure 33. Main components of a typical LDV system (modified from TSI Inc., 2005).

Figure 34. LDV system at the Ven Te Chow Hydrosystems Laboratory, UIUC.

The LDV system at the Ven Te Chow Hydrosystems Laboratory is equipped with three different probes: a 5-beam probe (model TR-360, see Figure 29), a 4-beam submersible probe (model TR-260) and a 2-beam submersible probe (model TR-160, see Figure 35). Each probe can measure one velocity component for each pair of beams. The probes can be used simultaneously in different configurations to measure up to three components combined. The front lens of the probes determines the focal distance, which is the distance from the last surface of the lens to the measurement volume. These lenses can be interchanged to achieve the desired range of focal distance, according to the requirements of a particular experiment. The available lenses provide a range of focal distance from 250 mm to 600 mm in air. In water, the focal distance will increase due to refraction, the magnitude of which depends on the path length of the beams inside the water.

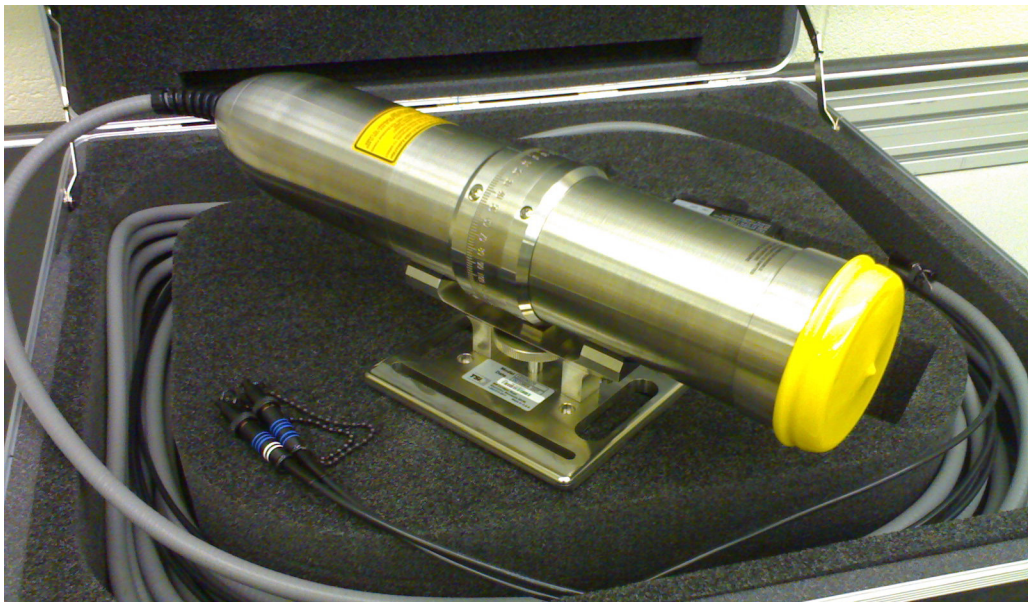


Figure 35. 2-beam submersible probe at the Ven Te Chow Hydrosystems Laboratory, UIUC.

The 5-beam probe (Figure 29), in reality has three color pairs (the central beam is shared by two colors) and can be used for three-component velocity measurements by itself; however the velocity components that share the middle beam won't be orthogonal. In this case, when the transformation is applied to convert the measurements into the orthogonal components, large errors can be produced on the component along the probe axis when the transformation angle is smaller than about 20º. This translates into a limitation on the focal distance that this probe can

be used for 3D measurements, which for the standard lenses would be about 70 mm. This probe is still useful as a two-component probe, and in particular to measure velocities close to the boundaries thanks to the middle beam, which provides some extra clearance to avoid one of the beams from getting blocked before the measurement volume can get close to the boundary. In a regular probe where all four beams come from the sides, one of them will typically get blocked by the flume side wall before the measurement volume gets anywhere close to the bed, unless the probe is tilted. This characteristic was very important for the experiments performed in this work to be able to measure very close to the PVC bed of the LOWST.

4.2.3 Traverse system

The LDV probes were mounted on a 3-axis traverse, driven by a Microstep Controller (model C 142-4.1, from Isel Automation). It is capable of providing a spatial resolution of 0.01 mm in all 3 directions, proving essential for the fine geometric requirements needed inside the boundary layer. The displacement range of the traverse is about 50 cm in all three directions.

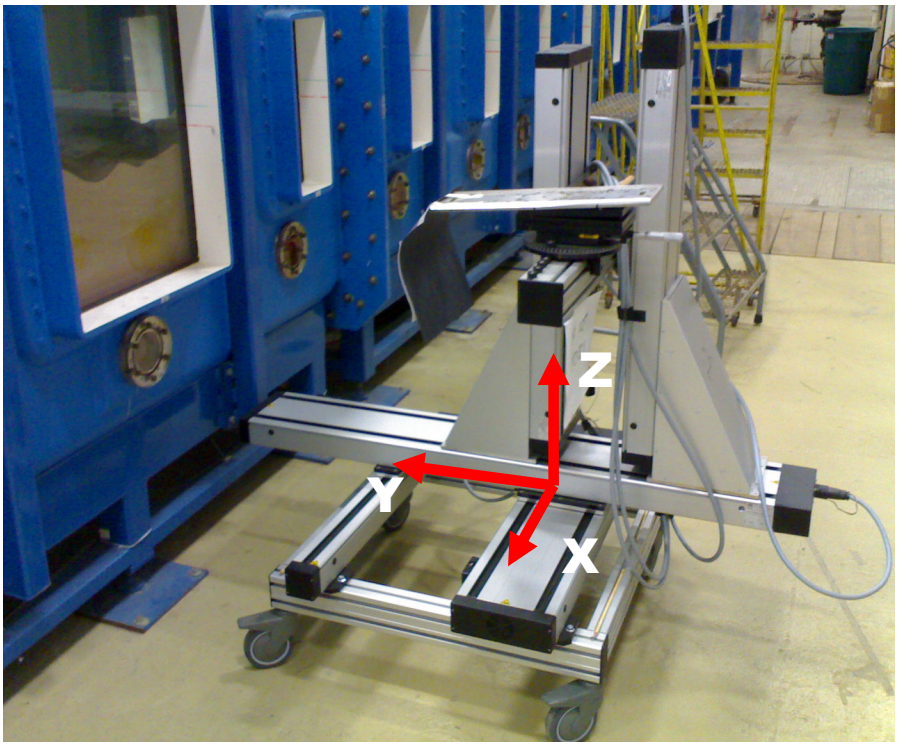


Figure 36. 3-axis traverse system for LDV measurements.

The reference coordinate system used for the traverse is orthogonal, being x the coordinate along the tunnel (streamwise), y the coordinate across the tunnel (spanwise) and z the vertical coordinate (Figure 36). Careful manipulation of the traverse is required in order to accurately position the measurement volume at the desired locations inside the flume, taking into account the effect of refraction in the focal distance of the probes. For measurements close to the bottom, position accuracy is particularly important when setting up the instrument in the vertical coordinate (z). This is due to the large vertical gradients expected for the streamwise velocity component ($\partial u / \partial z$) close to the bed. However, results are not so sensitive to small position inaccuracies in the other two directions (x and y).

4.2.4 Laser enclosure

Experiments in the Large Oscillatory Tunnel required the operation of the LDV system in an open area, so a safety enclosure was built for the purpose of protection to other persons and equipment working in the vicinity (Figure 37).

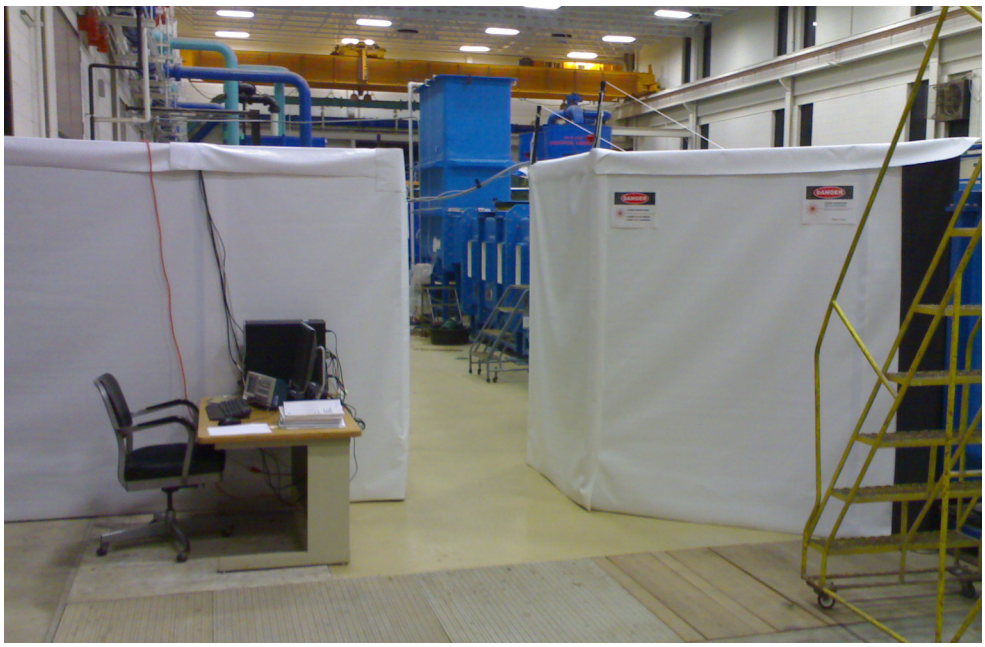


Figure 37. Laser enclosure for the LDV system operating in the main lab.

CHAPTER 5

METHODOLOGY

5.1 LOWST set-up: flat and smooth bed

For the experiments performed in this work, the LOWST was fitted with a fixed bed made of PVC all along the tunnel, resulting in a flat and smooth surface in contact with the flow. In order to build this set-up, first, a 22 cm layer of sand was removed from inside the LOWST all along the tunnel's 12 m long measurement section. Then two layers of a fine-mesh geotextile material where applied to keep the remaining sand in place and avoid any upwards entrainment. Directly on top of them, a coarse-mesh geotextile layer followed, upon which the PVC bed structure was assembled.

Figure 38. Pictures of the sand extraction process, by hand (left) and using a siphoning device (right).

Figure 39. Picture of the inside of LOWST after sand extraction, with the fine-mesh geotextile in place.

Figure 40. Picture of the PVC bed structural cross-section.

The PVC bed cross-section was designed as a combination of I-beams across the width of the tunnel. The thickness of the supporting beams was 1.5 inches (38 mm) for the two in the center and 0.75 inches (19 mm) for the two on the sides. The thickness of the bottom plate was 0.25 inches (6 mm) while the top plate was 0.5 inches (13 mm). This design created a strong yet slender cross-section, which was important to provide enough rigidity to ensure no deformation of the bed under the continuously up-down alternating pressure forces of the oscillatory flow. One additional condition affecting the design was that no holes or attachment points could be made in the composite structure of the LOWST, so the PVC bed had to be self-supporting and able to remain in place under the forces of the flow.

The PVC bed was assembled *in situ*, in short sections due to the size limitations imposed by the tunnel hatches. First the bottom plate and vertical beams were installed. Then, concrete blocks were installed in the three spaces between the beams to provide additional weight to the structure. Finally, the top plates were screwed to the beams using countersunk flat-head screws. Continuity between plates was assured with a tongue-and-grove system. The plates extended from side to side and a neoprene rubber gasket was used in the contact with the walls to provide cushion and close any gaps.

This design ensured that the surface of the bed exposed to the flow was flat and smooth. With the PVC structure in place, the bed elevation inside the tunnel was back to the nominal position (mid height), and the resulting cross-section available for the flow was 0.6 m high by 0.8 m wide.

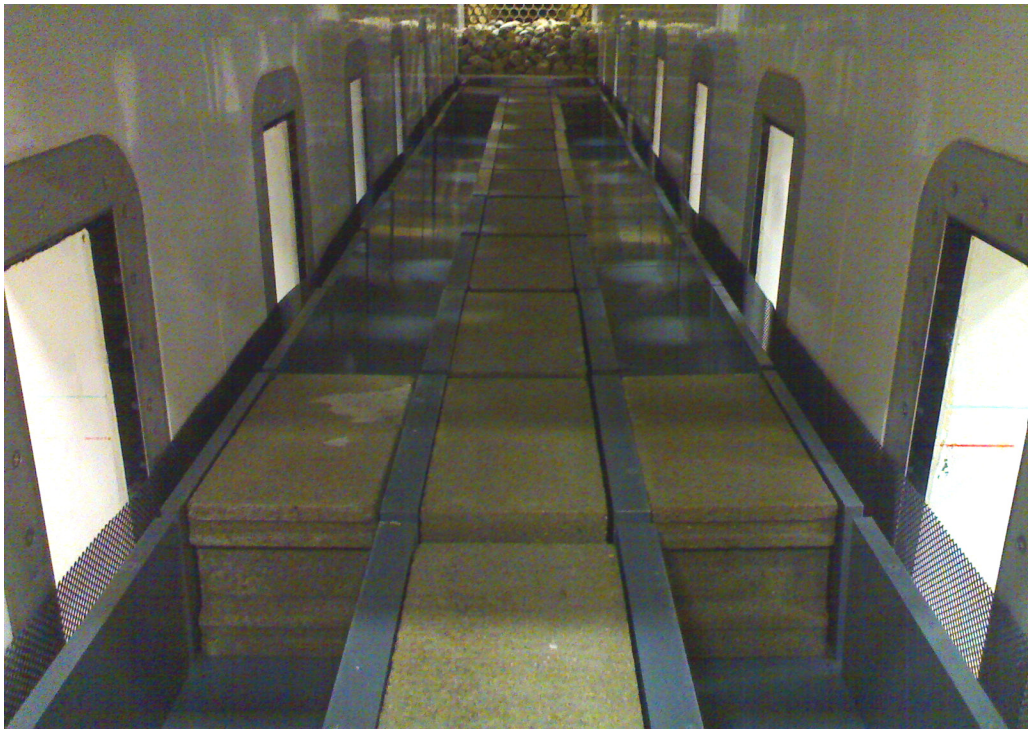


Figure 41. Picture of the inside of LOWST during assembly of the PVC bed. Concrete blocks were placed in between the vertical beams to add weight.

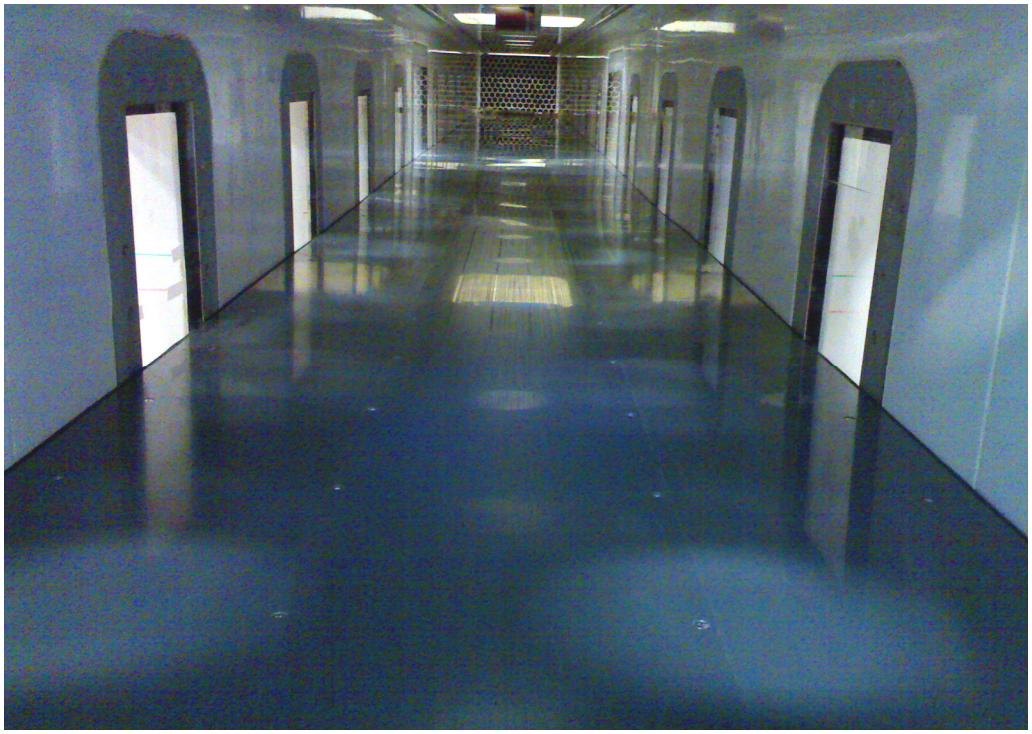


Figure 42. Picture of the inside of LOWST with the PVC bed in place.

5.2 LDV set-up: 3D measurements with 2 probes

The laser Doppler velocimetry (LDV) system at the Ven Te Chow Hydrosystems Laboratory was used to perform the velocity measurements in this work. This technique is particularly suited for the objectives of this research since it offers very fine spatial resolution (on the order of 0.1 mm) and very high temporal resolution (upwards of 10,000 Hz). In addition, the fact that it is an optical technique allows that the measurements be performed from the outside of the flume and, consequently, the flow is not disturbed by the presence of any type of probe, as opposed to mechanical or acoustic techniques. Based on these characteristics, the LDV technique is particularly suitable for the detailed measurement of velocities inside the boundary layer, providing very good and reliable results at the same time.

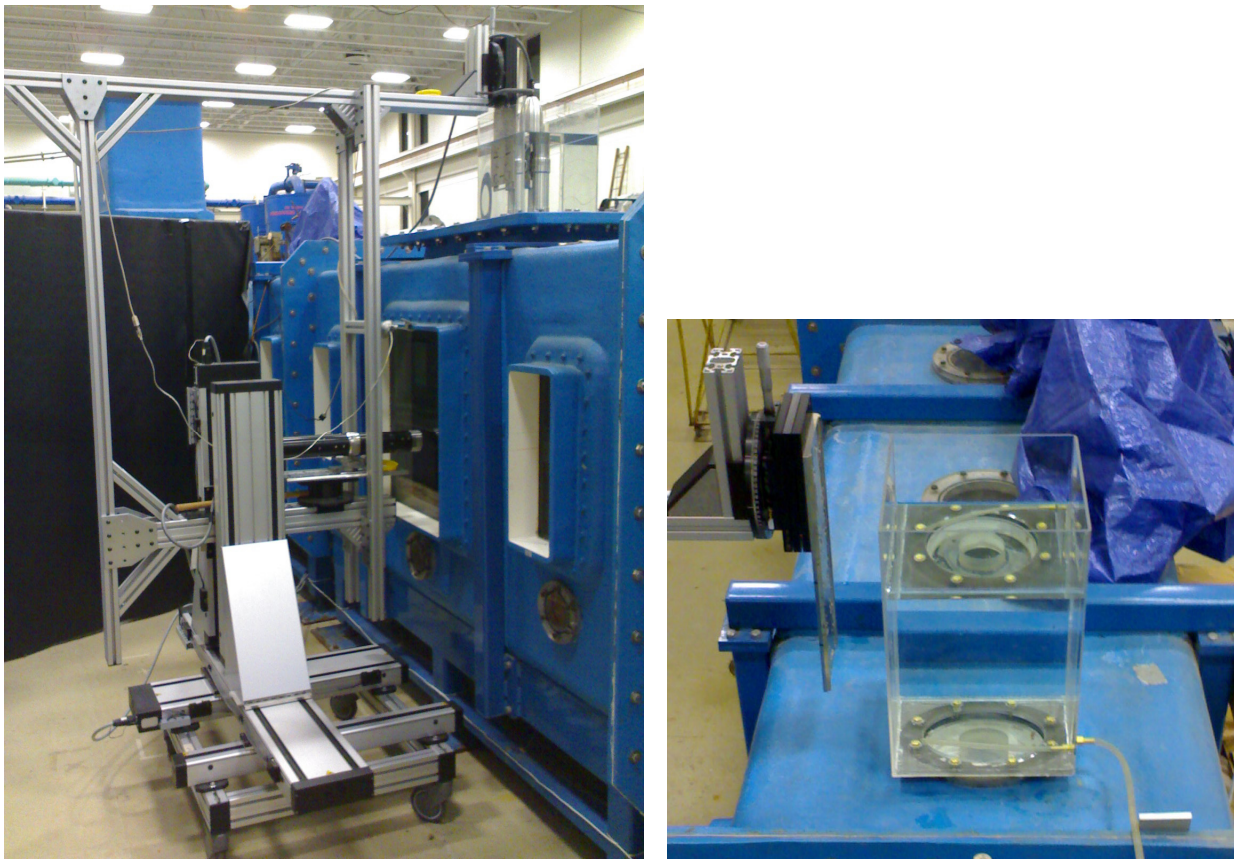


Figure 43. 3D LDV setup in the LOWST showing the 2 probes mounted on the same traverse pointing into the flume (left). Detail of the special water-filled container used to correct refraction of the top probe (right).

A special LDV set-up was built involving 2 probes and a refraction-correcting device, which allowed to obtain measurements of all three velocity components (u, v, w) simultaneously (see also Mier and Garcia, 2012). The 5-beam probe was mounted on the side of the flume with the 360 mm lens to measure two velocity components (u, w) at the centerline of the tunnel. The 2-beam submersible probe was mounted on top of the flume with the 600 mm lens to measure one velocity component (v) at the same location simultaneously, effectively obtaining 3D measurements of velocity at each point. The two probes were mounted on the same traversing system at a nearly 90º configuration, each one accessing the flume through a different window. The top probe was submerged in a special water-filled container such that the focal distance of this probe didn't change when moving the traverse vertically (see Figure 44).

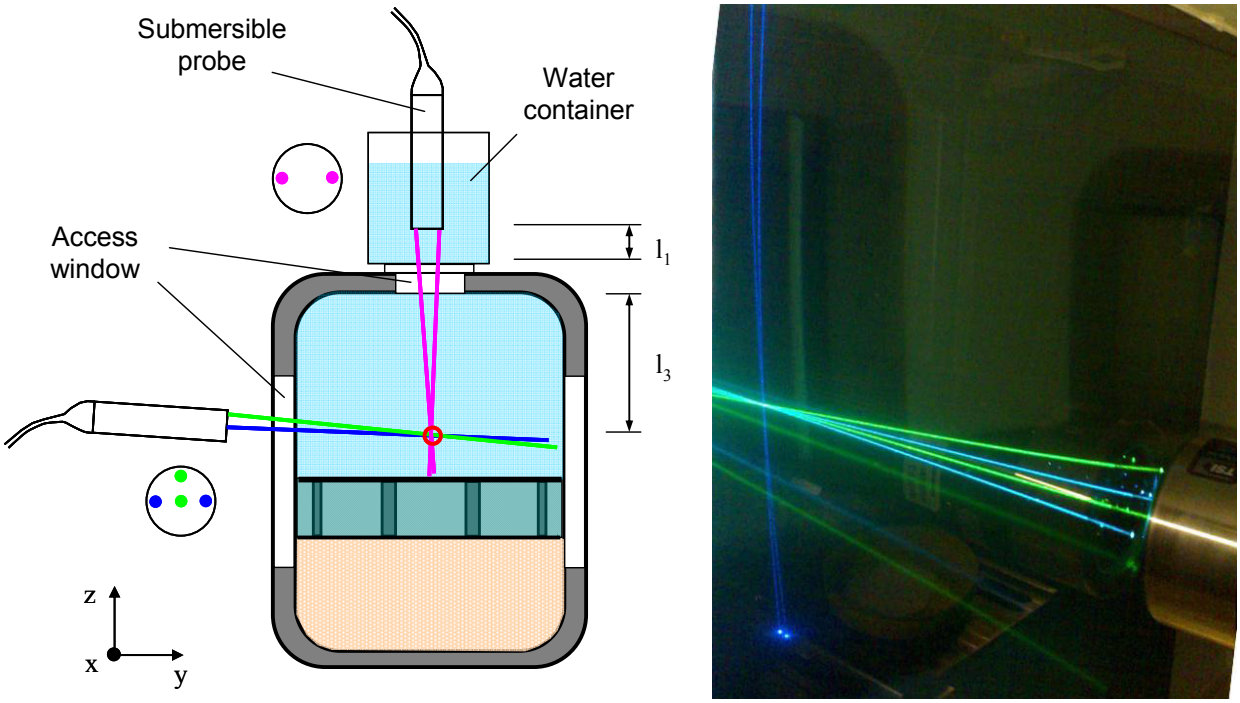


Figure 44. 3D LDV setup using two probes with a refraction correction container for vertical profiling. Schematic (left) and actual image (right) of the laser beams in the LOWST. From Mier and Garcia (2012).

With this setup, after the system was aligned, the measurement volumes of both probes remained coincident when traversing along the z axis. Any reduction in distance traveled by the top probe beams inside the flume (l_3) was compensated with an equivalent gain in distance traveled by the beams inside the external container (l_1). Since the index of refraction of the internal and external mediums was the same (water), the refracted focal length of the top probe did not change, and

consequently the displacement of the measurement volume was the same as the displacement of the probe. Both probes were attached to the same traversing system, and their measurement volumes moved the same distance and remained coincident along the vertical profile.

The 3 components measured where not orthogonal and a transformation was required during the data processing to convert them into the orthogonal reference system (x, y, z). The transformation matrix used for this configuration is shown in Figure 45. It shows the laser beams in the 3D LDV set-up with the side probe tilted forward χ degrees, and with half-angles of the beam pairs in air of α, β and γ, respectively. It is important to note that, since the measurement volume was inside a medium (water) different than air, the angles of the measured velocity components ($\vec{V}_1, \vec{V}_2, \vec{V}_3$) were affected by refraction. Those refracted angles need to be used in the transformation matrix, according to the LDV theory (see also Mier and Garcia, 2012).

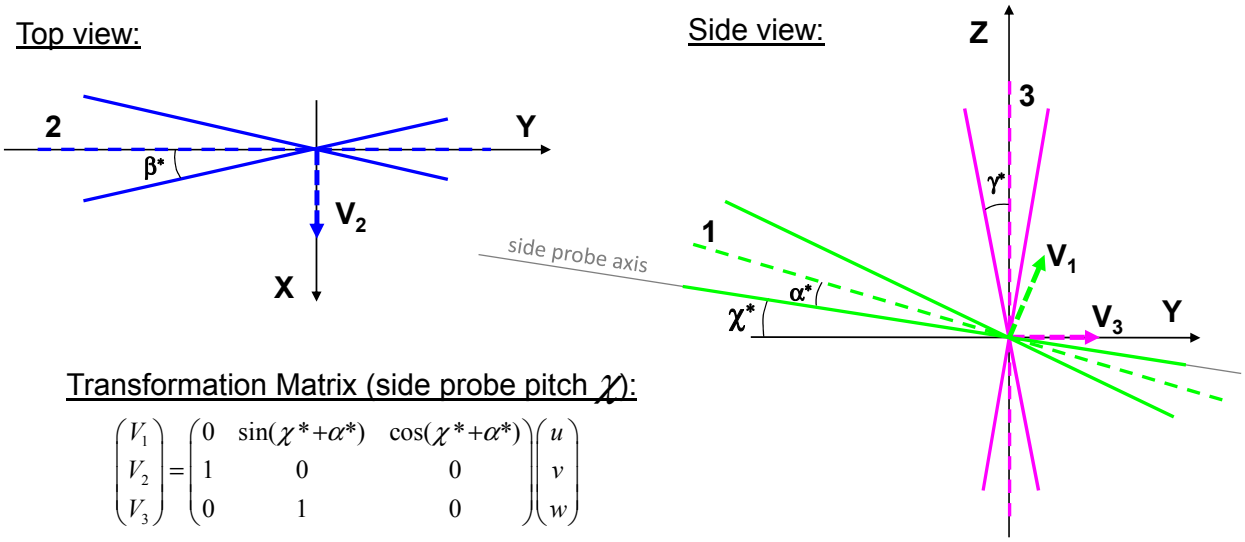


Figure 45. Transformation matrix for the 3D LDV setup used in this work. The side probe is pitched forward χ degrees and the bottom beam of pair #1 comes from the center of the 5-beam probe. Refracted angles were used underwater and are indicated with an asterisk (*).

For the experiments in this work, the side probe was slightly pitched forward to be able to move the measurement volume very close to the bed without interference from any possible obstacle along the beam path. The pitch angle in air was $\chi = 1°$ and the refracted angle in water was $\chi^* = 0.76°$. The angles of the measured velocity components were determined by the optical properties

of the lenses, with values in air $\alpha = 1.98^{\circ}$, $\beta = 3.95^{\circ}$, $\gamma = 2.35^{\circ}$, and corrected for refraction in water $\alpha^{*} = 1.5^{\circ}$, $\beta^{*} = 3.0^{\circ}$, $\gamma^{*} = 1.78^{\circ}$. With these values, the transformation matrix was:

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 0 & 0.0394 & 0.9992 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{35}$$

The inverse matrix was also calculated, resulting:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1.0008 & 0 & -0.0394 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \tag{36}$$

The bottom of the flume was painted black with a matte finish paint to reduce reflections off of the PVC bed. This helped improve the quality of the measurements close to the bed in all 3 components. In particular, for channel 3 (violet color, top probe), corresponding to the spanwise component (v), the measurement of velocities inside the boundary layer was quite challenging because of this issue. Due to the orientation of the probe, strong reflections were being generated from the PVC bed going back straight into the receiving lens of the probe. Without the black paint, the closest point to the bed with good measurements in this channel was at around 20 mm. After the black paint was applied, good measurements in this channel were possible as close as 2 mm from the bed. This distance is still too much in order to resolve the complete boundary layer, but it is the best possible with the set-up used. A better solution would be to have this probe looking into the flume from the side, so that the beams were not perpendicular to the boundary of interest. However, due to window size limitations and angle requirements, that kind of arrangement was difficult in the LOWST. Only channels 1 and 2 (green and blue, respectively) were looking into the flume from the side, which allowed measurements as close as 0.2 mm from the bed for the streamwise and vertical velocity components.

Position accuracy when using the traverse system during this work was within ± 0.05 mm in the vertical coordinate (z). In addition, a correction was applied to every vertical profile during data

processing to account for the uncertainty in the determination of the true $z = 0$ level of the bed. This correction helped minimize errors in the analysis, reducing the uncertainty in the vertical position to about ± 0.01 mm.

5.3 Experimental procedure

For each experiment a velocity profile was measured at the centerline of the tunnel in order to avoid any boundary effects coming from the side walls. From the results obtained for the boundary layer thickness of the bottom wall of the tunnel, it was clear that the boundary layer that could be generated at the side walls would never reach the centerline of the tunnel. Indeed, the maximum thickness of the boundary layer observed for the bottom was around 50 mm. For the side walls it could be assumed to be similar, since they are also smooth. Now considering the centerline is at 400 mm from the side walls, it is clear that it is unlikely that any effects coming from the side walls of the tunnel could have affected the measurements. This way the analysis of the boundary layer parameters can be reduced to a two-dimensional situation in the x-z plane, in which only the bottom wall (bed) needs to be considered.

Velocity profiles were obtained by traversing the LDV probes along the vertical from $z = 0.2$ mm to $z = 200$ mm. Each profile included 54 points, which were unevenly distributed following a logarithmic series in order to obtain higher density of points close to the bottom. The first 12 points were within a distance of 1 mm from the bed and a total of 28 points were in the first 10 mm, which allowed capturing flow features with great detail inside the boundary layer and even inside the viscous sublayer in some cases.

Time series of instantaneous flow velocities for the 3 components (u, v, w) were measured at each location. Measurements were recorded for up to 130 cycles at each location. A high number of cycles were needed in order to obtain good ensemble-averaged quantities during the analysis. This was especially important for the analysis of higher order statistics, such as root-mean-square (RMS) of velocity fluctuations and skewness of velocity fluctuations.

In addition to velocity measurements, also the temperature of the water was measured for each experiment. Readings were taken at the beginning and at the end of each experiment from an analog thermometer installed inside the LOWST. From these readings, the average temperature was computed for each experiment. From the temperature, then it was possible to calculate the density (ρ) and the kinematic viscosity (ν) of the water using the following common formulas (Thiesen *et al.*, 1900):

$$\nu = \frac{1.79 \cdot 10^{-6}}{1 + 0.03368 \cdot Temp + 0.00021 \cdot Temp^2} \tag{37}$$

$$\rho = 1000\left(1 - \frac{(Temp + 288.9414)(Temp - 3.9863)^2}{508929.2(Temp + 68.12963)}\right) \tag{38}$$

where the temperature (*Temp*) is given in degrees Celsius (ºC), the kinematic viscosity (ν) is obtained in m^2/s, and the density (ρ) is obtained in kg/m^3. When calculating these parameters for the experiments in this work, differences with respect to the nominal values under standard conditions were small. However, even though they might be small, the calculation of these parameters is very important, since they play a key role in the analysis of the boundary layer variables.

5.4 Data processing

The outcome of the experiments in this work was a vast set of data sampled at different rates at several locations in the flume. In LDV measurements the data rate is not fixed, but rather variable depending on the seeding conditions and flow velocity, resulting in an uneven number of data points for the locations tested (Nezu and Rodi, 1986; Adrian and Yao, 1987; Mier and Garcia, 2009, 2011). The data needs to be re-sampled at even-time intervals after it is collected in order to avoid any possible velocity bias (Adrian and Yao, 1987; Mier and Garcia, 2013). Additionally, measurements in unsteady periodic flows require a synchronization signal to be

recorded together with the data so that meaningful statistics can be obtained from the analysis (Hino *et al.*, 1983; Jensen *et al.*, 1989; Mier and Garcia, 2013).

In this regard, a systematic methodology for processing LDV data in unsteady periodic flows was developed as part of this work (see also Mier and Garcia, 2013). Given a particular LDV data set from measurements in an oscillatory flow, the goal was to develop a processing algorithm capable of dealing with the special characteristics of this type of data, including: cycle synchronization, filtering of bad data points, even-time re-sampling, coordinate transformation, quality checks, and generation of ensemble statistics; all in one single package and capable of processing all 3 velocity components at the same time.

The processing algorithm was developed using MATLAB® and the computer code is provided in APPENDIX A. It was designed to process the datasets through several steps:

1. First, a cycle synchronization routine was used to detect the time stamps in the data corresponding to the pulses of the synchronization trigger signal. After that, individual cycles were identified and stacked into ensembles.
2. Then, bad data points were identified and removed using a custom routine that allowed user interaction through the filtering process. This included checking for complete cycles that could be out of sync, flocks of bad points generated from photodetector saturation, and spikes and other outliers.
3. With the clean data, now it was possible to perform the re-sampling of the data at even-time intervals, with a specific data rate selected by the user. At this point some statistics of the data set are calculated to optimize the selection of an appropriate data rate without loosing or fabricating information.
4. Until this step, the data was manipulated using 'cell' arrays in MATLAB® due to the existence of locations with different number of samples. However, after even-time sampling the 'cell' arrays were converted to 5-D matrices of velocity for each channel with dimensions (x, y, z, *cycle*, *phase*).
5. Next, a coordinate transformation was applied to convert the velocity data into the Cartesian coordinate system with corresponding velocities (u, v, w). In this step it is important to note

that underwater distortion due to refraction was accounted for in the transformation matrix by using the refracted angles of the measurement directions.

6. A correction in the vertical coordinate (z) of each profile was calculated to account for the uncertainty in the determination of the true $z = 0$ level during the measurements.
7. Finally, a quality check was performed before the data was ready for analysis. Checks included calculating running average and percentage error plots at each measurement location.

After processing the data, clean and tidy datasets were obtained for all 3 components of velocity (u, v, w) at all locations measured. These were used for the subsequent analysis of mean flow and turbulent characteristics of the oscillatory flow.

5.5 Data analysis

The measurements were analyzed in the context of unsteady periodic flows, which requires the use of ensemble averaging of cycles. This type of analysis assumes that the mean flow repeats itself for each cycle, as it is the case of oscillatory flows. This way, average information about the mean flow and turbulence characteristics could be obtained for every phase of the oscillation. However, the analysis of turbulent structures and spatial correlations was limited due to the fact that the LDV technique only provides single-point measurements one location at a time.

From the clean data sets obtained after processing, several variables of interest were analyzed using the ensemble average method. The mathematical expressions of those variables are presented in the equations below. In these equations, the symbol $\langle\ \rangle$ denotes ensemble average, N is the total number of cycles measured at each location, nc is the cycle index such that $nc = 1 \ldots N$, ωt is the phase of the oscillation, and i represents the three dimensions of Cartesian space, so that $i = 1,2,3$ correspond to spatial coordinates x,y,z and velocity components u,v,w respectively. Note that capital letters are used for simplicity to denote ensemble-averaged quantities, such as $U_i = \langle u_i \rangle$. From the instantaneous velocity measurements obtained from the LDV (u_i), the following mean flow and turbulence variables can be calculated:

Instantaneous velocities:

$$u_i = u_i(x, y, z, nc, \omega t) \tag{39}$$

Mean flow velocities:

$$\begin{aligned} U_i &= U_i(x, y, z, \omega t) = \\ &= \left\langle u_i(x, y, z, nc, \omega t) \right\rangle = \\ &= \frac{1}{N} \sum_{nc=1}^{N} u_i(x, y, z, nc, \omega t) = \frac{1}{N} \sum_{nc=1}^{N} u_i \end{aligned} \tag{40}$$

Turbulent velocity fluctuations:

$$\begin{aligned} u_i' &= u_i'(x, y, z, nc, \omega t) = \\ &= u_i(x, y, z, nc, \omega t) - \left\langle u_i\left(x, y, z, nc, \omega t\right) \right\rangle = \\ &= u_i(x, y, z, nc, \omega t) - \frac{1}{N} \sum_{nc=1}^{N} u_i(x, y, z, nc, \omega t) = \\ &= u_i - \frac{1}{N} \sum_{nc=1}^{N} u_i = u_i - U_i \end{aligned} \tag{41}$$

Root-mean-square (RMS) velocity fluctuations:

$$\begin{aligned} U_{i,\,RMS} &= U_{i,\,RMS}(x, y, z, \omega t) = \\ &= \left\langle u_i'^2(x, y, z, nc, \omega t) \right\rangle^{1/2} = \\ &= \left[\frac{1}{N} \sum_{nc=1}^{N} u_i'^2(x, y, z, nc, \omega t) \right]^{1/2} = \left[\frac{1}{N} \sum_{nc=1}^{N} u_i'^2 \right]^{1/2} \end{aligned} \tag{42}$$

Turbulent kinetic energy:

$$\begin{aligned} TKE &= TKE(x,y,z,\omega t) = \\ &= \frac{1}{2}\sum_{i=1}^{i=3}\left(\left\langle u_i{}'^2(x,y,z,nc,\omega t)\right\rangle\right) = \\ &= \frac{1}{2}\sum_i\left(\left\langle u_i{}'^2\right\rangle\right) = \frac{1}{2}\sum_i\left(U_{i,RMS}^2\right) = \frac{1}{2}\left(U_{RMS}^2 + V_{RMS}^2 + W_{RMS}^2\right) \end{aligned} \tag{43}$$

Reynolds shear stresses:

$$\begin{aligned} R_{ij} &= R_{ij}(x,y,z,\omega t) = \\ &= -\rho\left\langle u_i{}'(x,y,z,nc,\omega t)\cdot u_j{}'(x,y,z,nc,\omega t)\right\rangle = \\ &= -\rho\left\langle u_i{}'\cdot u_j{}'\right\rangle = -\rho\frac{1}{N}\sum_{nc=1}^{N}\left[u_i{}'\cdot u_j{}'\right] \qquad for\ i \neq j \end{aligned} \tag{44}$$

Viscous shear stress:

$$\begin{aligned} \tau_{v,ij} &= \tau_{v,ij}(x,y,z,\omega t) = \\ &= \rho\nu\frac{\partial\left\langle u_i(x,y,z,nc,\omega t)\right\rangle}{\partial x_j} = \rho\nu\frac{\partial\left\langle u_i\right\rangle}{\partial x_j} = \rho\nu\frac{\partial U_i}{\partial x_j} \qquad for\ i \neq j \end{aligned} \tag{45}$$

Total shear stresses:

$$\begin{aligned} \tau_{ij} &= \tau_{ij}(x,y,z,\omega t) = \\ &= \tau_{v,ij}(x,y,z,\omega t) + R_{ij}(x,y,z,\omega t) = \rho\nu\frac{\partial\left\langle u_i\right\rangle}{\partial x_j} - \rho\left\langle u_i{}'\cdot u_j{}'\right\rangle = \\ &= \rho\nu\frac{\partial U_i}{\partial x_j} - \rho\frac{1}{N}\sum_{nc=1}^{N}\left[u_i{}'\cdot u_j{}'\right] \qquad for\ i \neq j \end{aligned} \tag{46}$$

Bed shear stress:

$$\begin{aligned}\tau_{b,ij} &= \tau_{b,ij}(x,y,\omega t) = \\ &= \tau_{ij}(x,y,z,\omega t)\big|_{z=z_{bed}} = \tau_{ij}\big|_{z=z_{bed}} \qquad for\ i=1,2 \quad j=3\end{aligned} \tag{47}$$

Turbulence production:

$$\begin{aligned}P_{ij} &= P_{ij}(x,y,z,\omega t) = \\ &= -\left\langle u'_i(x,y,z,nc,\omega t)\cdot u'_j(x,y,z,nc,\omega t)\right\rangle \frac{\partial\left\langle u_i(x,y,z,nc,\omega t)\right\rangle}{\partial x_j} = \\ &= -\left\langle u'_i\cdot u'_j\right\rangle \frac{\partial\left\langle u_i\right\rangle}{\partial x_j} = -\frac{1}{N}\sum_{nc=1}^{N}\left[u'_i\cdot u'_j\right]\frac{\partial U_i}{\partial x_j}\end{aligned} \tag{48}$$

Eddy viscosity:

$$\begin{aligned}\nu_{t,ij} &= \nu_{t,ij}(x,y,z,\omega t) = \\ &= \frac{-\left\langle u'_i(x,y,z,nc,\omega t)\cdot u'_j(x,y,z,nc,\omega t)\right\rangle}{\left(\dfrac{\partial\left\langle u_i(x,y,z,nc,\omega t)\right\rangle}{\partial x_j}+\dfrac{\partial\left\langle u_j(x,y,z,nc,\omega t)\right\rangle}{\partial x_i}\right)} = \\ &= \frac{-\left\langle u'_i\cdot u'_j\right\rangle}{\left(\dfrac{\partial\left\langle u_i\right\rangle}{\partial x_j}+\dfrac{\partial\left\langle u_j\right\rangle}{\partial x_i}\right)} = \frac{-\dfrac{1}{N}\sum_{nc=1}^{N}\left[u'_i\cdot u'_j\right]}{\left(\dfrac{\partial U_i}{\partial x_j}+\dfrac{\partial U_j}{\partial x_i}\right)}\end{aligned} \tag{49}$$

Skewness of velocity fluctuations:

$$
\begin{aligned}
U_{i,\,skew} &= U_{i,\,skew}(x,y,z,\omega t) = \\
&= \frac{\left\langle u_i{}'^3(x,y,z,nc,\omega t)\right\rangle}{\left\langle u_i{}'^2(x,y,z,nc,\omega t)\right\rangle^{3/2}} = \\
&= \frac{\frac{1}{N}\sum_{nc=1}^{N} u_i{}'^3(x,y,z,nc,\omega t)}{\left[\frac{1}{N}\sum_{nc=1}^{N} u_i{}'^2(x,y,z,nc,\omega t)\right]^{3/2}} = \frac{\frac{1}{N}\sum_{nc=1}^{N} u_i{}'^3}{\left[\frac{1}{N}\sum_{nc=1}^{N} u_i{}'^2\right]^{3/2}} = \frac{\frac{1}{N}\sum_{nc=1}^{N} u_i{}'^3}{U_{i,\,RMS}{}^3}
\end{aligned}
\tag{50}
$$

As part of this work, a computer code was developed using MATLAB® to perform all the calculations needed for the analysis of the LDV data following the equations presented. The code is provided in APPENDIX B.

CHAPTER 6

MEAN FLOW RESULTS

6.1 Measured experimental conditions

In Table 2 a summary of the measured experimental conditions is presented for all 10 experiments. It can be observed that measured flow conditions differ slightly from the nominal values presented before in Table 1. However, this issue didn't affect the results of the experiments, since the nominal values were only used as a reference. All the calculations presented in this work have been obtained using the actual measured flow conditions included in this Table 2.

Among the physical parameters, water temperature (*Temp*) was measured for each experiment, which in turn provided a way to calculate water density (ρ) and kinematic viscosity (ν) using the formulas in equations (37) and (38). Among the flow conditions, the oscillation period (T) was set and the maximum velocity of the outer flow (U_{out_max}) was obtained from the measurements. With these two variables, it was then possible to back-calculate the oscillatory amplitude ($2a$) using equation (4) and then to obtain the wave Reynolds number (Re_w) using equation (3).

	Physical parameters (measured)			Flow conditions (measured)				
	Temp	ρ	ν	T	$2a$	U_{out_max}	Re_w	N_{max}
Exp no.	(ºC)	(kg/m^3)	(m^2/s)	(s)	(m)	(m/s)	(-)	(cycles)
1	18.0	998.62	1.07E-06	10	0.468	0.147	3.2E+04	130
2	16.6	998.87	1.11E-06	10	0.761	0.239	8.2E+04	130
3	23.2	997.52	9.45E-07	10	0.958	0.301	1.5E+05	130
4	23.6	997.42	9.36E-07	10	1.159	0.364	2.3E+05	130
5	27.5	996.41	8.59E-07	10	1.261	0.396	2.9E+05	130
6	27.0	996.54	8.68E-07	10	1.362	0.428	3.4E+05	130
7	26.5	996.68	8.77E-07	10	1.566	0.492	4.4E+05	130
8	24.5	997.20	9.17E-07	10	1.770	0.556	5.4E+05	130
9	18.1	998.61	1.07E-06	10	2.069	0.650	6.3E+05	130
10	20.0	998.23	1.02E-06	10	2.368	0.744	8.7E+05	130

Table 2. Physical parameters and flow conditions measured for each experiment.

Given the large amount of information and plots generated from the 10 experiments, only a subset is presented in the results. Notwithstanding, the complete data set for all 10 experiments is included in APPENDIX C. A representative selection of results for 3 of the experiments was made in order to showcase the oscillatory flow behavior across the laminar-to-turbulent transition regime according to wave Reynolds number (Re_w). This way results are shown for one experiment in the lower Re_w range (Experiment no. 1), one in the middle of the range (Experiment no. 5) and one in the upper part of the range (Experiment no. 10).

6.2 Outer flow velocity

The streamwise velocity of the flow far from the bottom was used as a reference in the analysis and presentation of results. This velocity is referred to as outer flow velocity (U_{out}) and represents the flow outside of the boundary layer, thus unaffected by the friction with the bottom. It was calculated as the average of the streamwise velocity of all points measured more than 100 mm away from the bottom. This distance was larger than the boundary layer thickness measured for all phases of the oscillation in all experiments.

Outer flow velocity computed from the measurements is shown in Figure 46 for the 3 experiments selected (blue line). It must be noted that all the measurements in this work are presented such that ωt = 90º was chosen as the phase of the cycle at which U_{out} reaches the maximum (U_{out_max}). Also, a reference sine wave using the nominal parameters for each experiment is shown for comparison (green line).

Taking that into account, small differences between the two lines can be observed. The difference in magnitude is attributed to the difference between nominal and measured parameters, as mentioned before, so it has no influence on the results. However, the shape of the measured wave (blue line) becomes a bit distorted as Re_w increases, creating a small asymmetry between the acceleration and deceleration parts in each half-cycle of the wave. In particular there is a small lag in the decelerating parts of the cycle, evidenced at the zero crossing point, which doesn't happen exactly at ωt = 180º, but rather a few degrees after. This results in the wave not being perfectly sinusoidal, however the difference is very small.

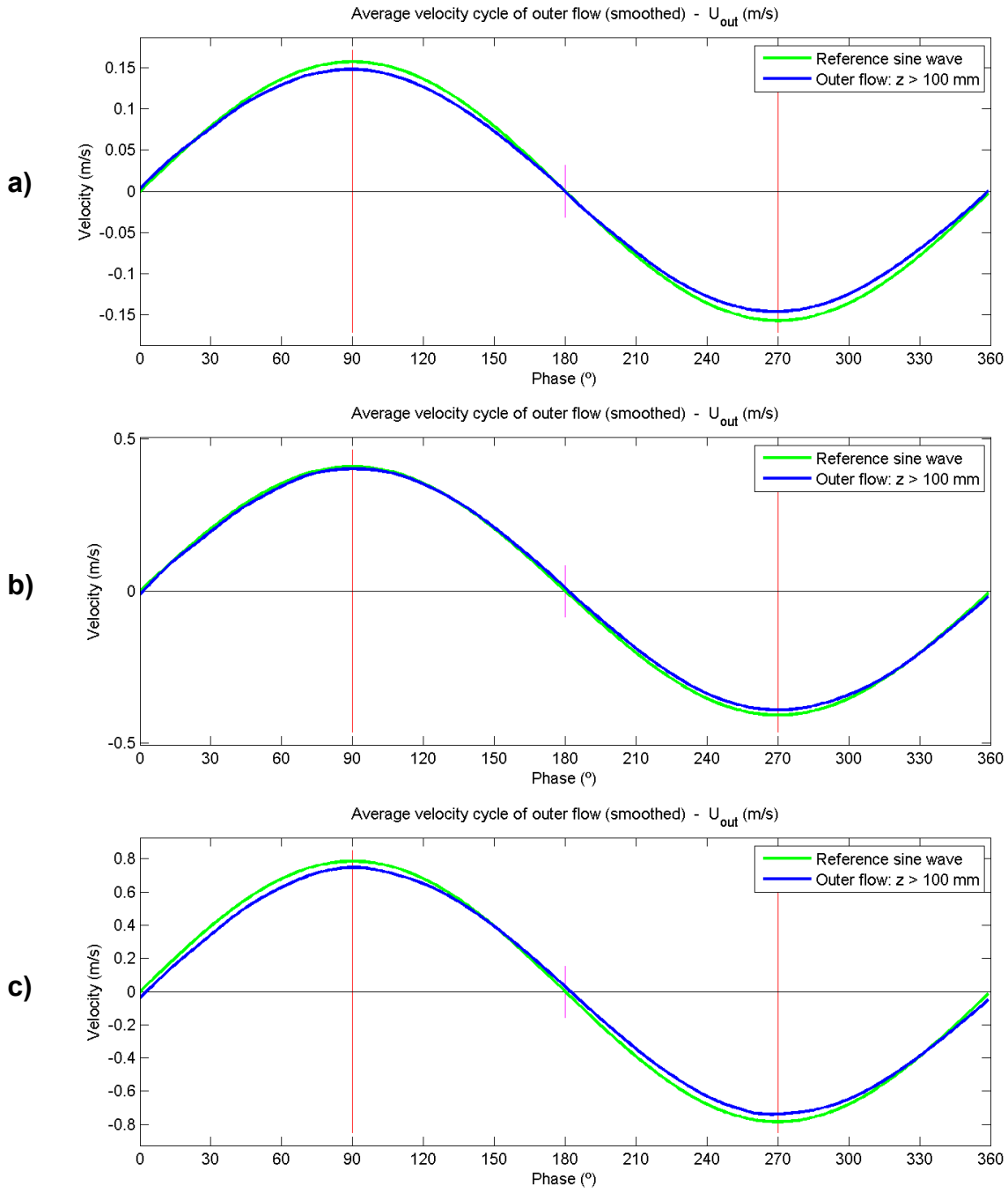


Figure 46. Outer flow velocity for experiments 1 (a), 5 (b) and 10 (c).

Due to the oscillatory nature of the flow, the results obtained for a full cycle can be viewed as two repetitions of each half-cycle. Although not perfectly symmetric, the first half-cycle and the second half-cycle should give very similar results for all the variables presented. Most times, these variables will show opposite signs between the two half-cycles due to the changing flow

direction. For the outer flow, the first half-cycle goes from phases 0º to 180º displaying positive velocities, and the second half-cycle goes from phases 180º to 360º with negative velocities. Negative velocities in this context just means that the flow is going in reverse direction. Within each half-cycle, there is a zone of flow acceleration for the first 90º until the phase of maximum flow velocity, and then the flow decelerates until zero velocity around 180º.

For simplicity during the presentation of results in this work, and given the nearly symmetrical nature of the flow between each half-cycle, phases 0º to 180º will be used to refer to both half-cycles, unless otherwise noted.

6.3 Mean velocities

Mean flow velocity was obtained from the measurements after performing the ensemble averaging of the instantaneous velocities for all cycles, as described before in the methodology. Contour plots, profile plots and cycle evolution plots were generated for all three velocity components (u, v, w), however, only the streamwise component (u) is shown here. The results obtained for the other two components didn't provide any significant information since they exhibited averaged values very close to zero, as it was expected. Plots for v and w can be found in APPENDIX C.

6.3.1 Experiment no. 1

Results for experiment 1 are presented in Figure 47, showcasing the oscillating nature of the flow, with symmetric positive and negative velocities. Both half-cycles present good symmetry as shown in the cycle evolution plot (Figure 47c). A clear distinction in the velocity patterns can be observed between the outer flow and the flow inside the boundary layer. Outer flow is very smooth and symmetric, with good overlap between the equivalent phases of the first and second half-cycles. The velocity magnitude is also very constant in the vertical for all locations outside of the boundary layer. Inside the boundary layer, close to the bottom, the velocities show a different behavior as they are affected by friction with the bed (Figure 47b).

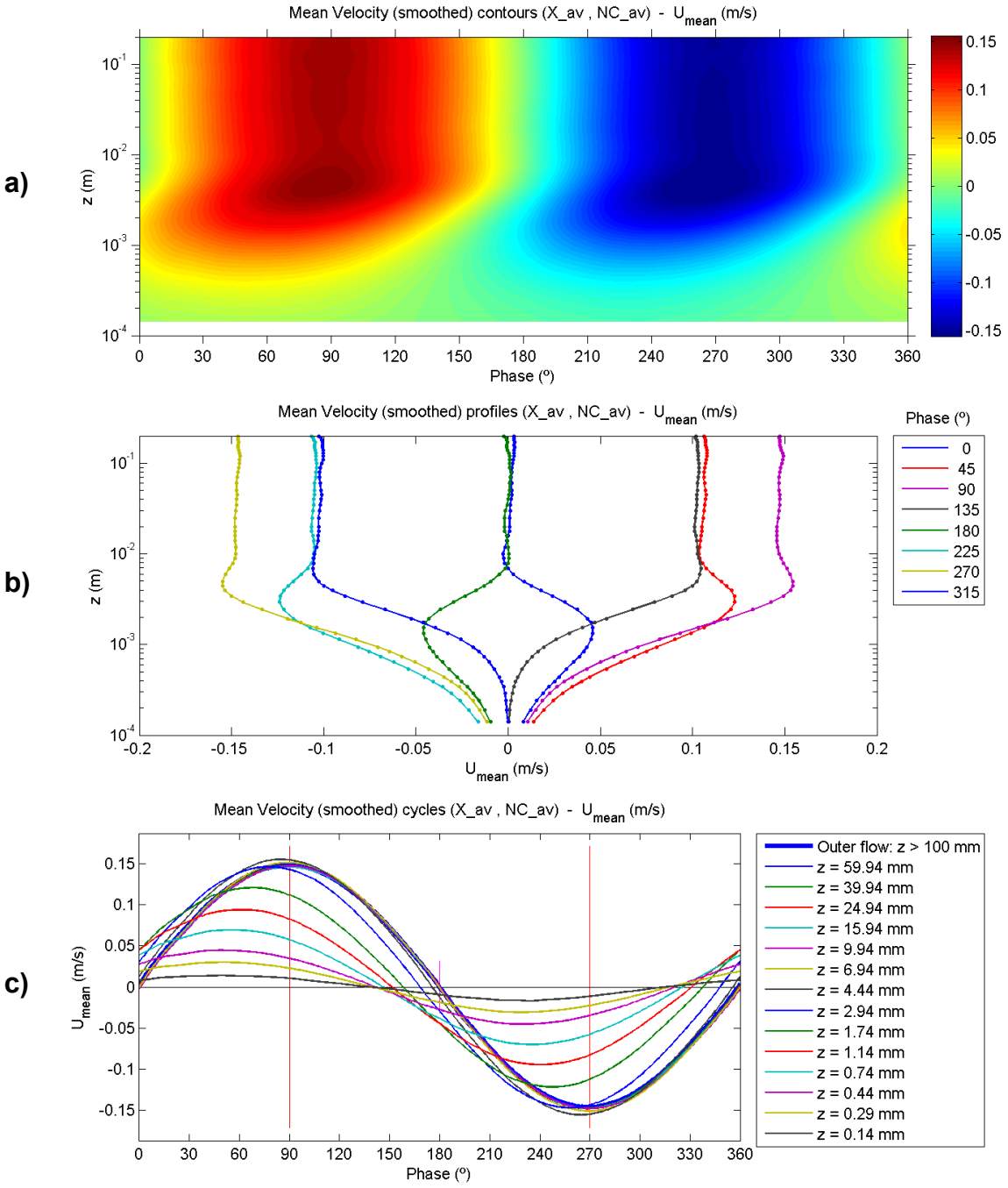


Figure 47. Mean flow velocity contours (a), profiles (b) and cycle evolution (c) for experiment 1.

This experiment falls within the laminar regime and, as such, certain defining characteristics about the laminar oscillatory boundary layer can be observed. In particular, for the phases of maximum outer flow velocity (90º), the velocity peak inside the boundary layer is greater than outside the boundary layer. This effect can be observed in the profile plot (Figure 47b) for the lines corresponding to 90º and 270º. Also, the velocity field inside the boundary layer seems to

act ahead of the velocities outside. This effect can be observed in the contour plot (Figure 47a) by the skewed shape of the contour lines towards the left inside the boundary layer. It is also observed in the profiles (Figure 47b), for instance, at the phases of flow reversal (0º and 180º) when the outer flow is just stopped, but inside the boundary layer the flow is already traveling in the opposite direction. In particular, the velocity near the bed is traveling 45º in advance of the outer flow. This can be easily observed in the cycle evolution plot (Figure 47c), looking at the phase where the velocity maximum occurs for each line. For the first line closer to the bed, this maximum velocity happens at the 45º phase, which is in turn 45º in advance of the outer flow velocity maximum. For higher z locations moving away from the bed, the velocity maximum shifts slowly towards phase 90º until it converges with the outer flow maximum.

6.3.2 Experiment no. 5

Results for experiment 5 are presented in Figure 48. Both half-cycles present good symmetry inside the boundary layer, although outside the boundary layer the flow velocity is slightly skewed towards the positive values, as shown in the cycle evolution plot (Figure 48c). Also, the profiles for phases near flow reversal (180º) don't overlap due to the small shape difference observed between the reference sine wave and the measured outer flow shown before in Figure 46, which was caused by a lag in the decelerating stage of the half-cycle.

This experiment falls within the transition between the laminar and turbulent regimes. Experiments in this transitional regime experience significant changes inside the boundary layer as Re_w increases. In particular, the shape of the velocity field changes significantly in terms of phase of the velocity maximum for each z location. In this experiment the phase of the near bed velocity maximum changed from phase 45º in experiment 1 (laminar regime) to around phase 65º in this transition regime. Additionally, in the contour plot (Figure 48a) a secondary lobe of high velocities appears in the lower part of the boundary layer, near phase 120º. Looking at the cycle evolution plot (Figure 48c), it can be observed that for the first few z locations near the bed the velocity profile experiences two maximums, one ahead of phase 90º (outer flow velocity maximum) and this one behind. This new velocity maximum becomes less evident for higher z locations and eventually there is only one maximum, ahead of phase 90º.

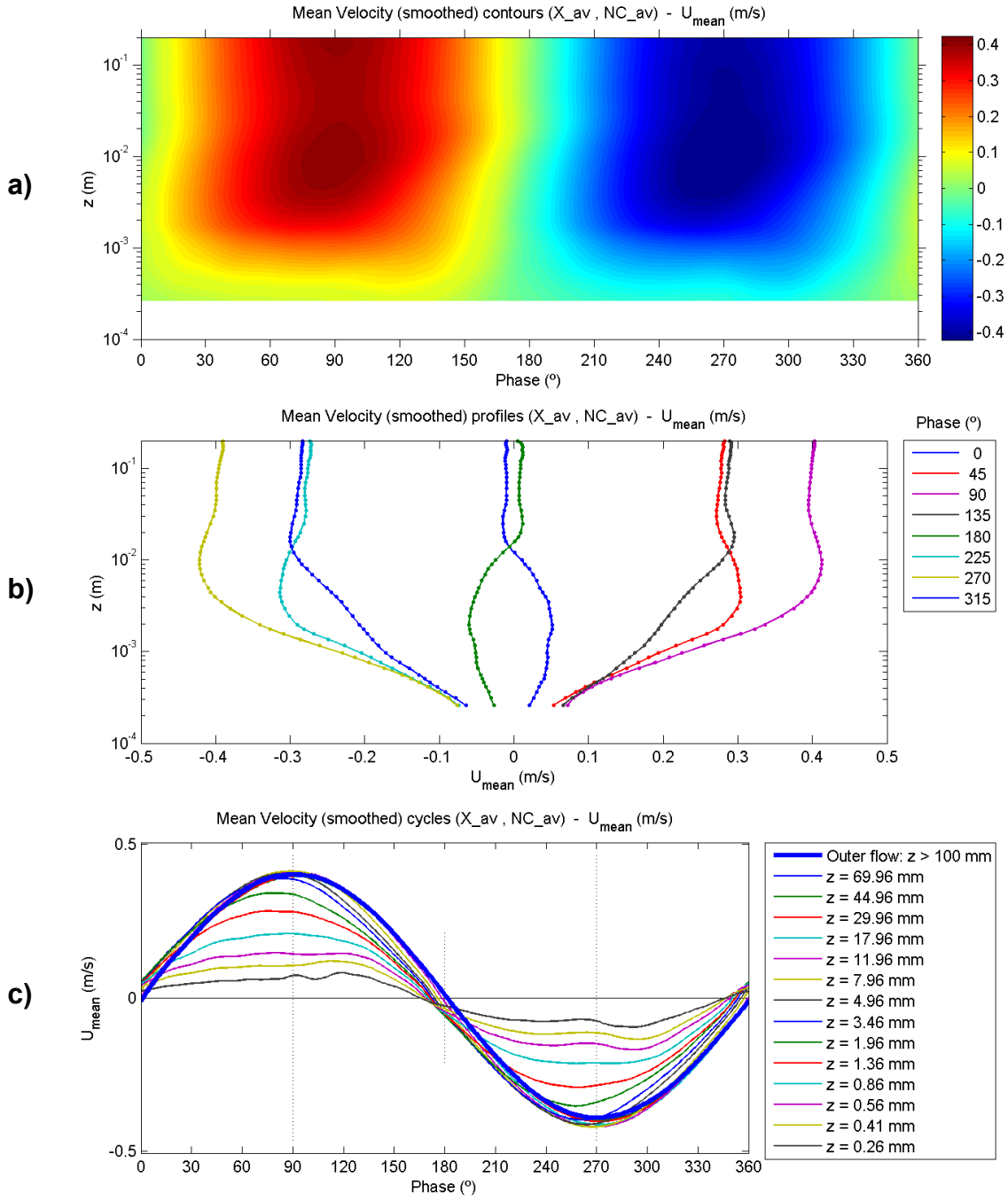


Figure 48. Mean flow velocity contours (a), profiles (b) and cycle evolution (c) for experiment 5.

This situation having two velocity maximums near the bed will have important implications for bed shear stress as will be shown later. The appearance of this second velocity maximum happens gradually from experiment 3 (Re_w = 1.5x10^5) until experiment 8 (Re_w = 5.4x10^5) being most evident for this experiment 5 shown here. It must be pointed out that this effect could only be observed thanks to the high temporal and spatial resolution of the LDV measurements

performed in this work, and in particular to being able to measure so close to the bed, since this phenomenon was only visible for the locations within 1 mm from the bed.

Also, the velocity field inside the boundary layer continues to act ahead of the velocities outside, but the phase lead is smaller than it was in experiment 1. This effect can still be observed in the contour plot (Figure 48a) by the skewed shape of the contour lines towards the left inside the boundary layer. It is also observed in the profiles (Figure 48b), for instance, at the phases of flow reversal (0º and 180º).

6.3.3 Experiment no. 10

Results for experiment 10 are presented in Figure 49. Both half-cycles present good symmetry inside and outside the boundary layer. The profiles for phases near flow reversal (180º) don't overlap due to the small shape difference observed between the reference sine wave and the measured outer flow shown before in Figure 46, which was caused by a lag in the decelerating stage of the half-cycle.

This experiment represents a condition near the fully turbulent regime. As Re_w approaches the fully turbulent regime, all the changes in shape inside the boundary layer described before for the transitional regime tend to converge towards a unified shape. In particular, from the two velocity maximums observed near the bed in experiment 5, the one behind phase 90º gradually moves towards phase 90º and eventually takes over the other maximum and dominates at a phase around 85º. This effect can be observed in the contour plot in Figure 49a, where only one big lobe remains and it is located near that phase. Additionally, a noticeable "step" shape appears for phases between 30º and 60º, also visible in the cycle evolution plot (Figure 49c).

Also, the velocity field inside the boundary layer continues to act ahead of the velocities outside but the phase lead is even smaller than it was in experiment 5. This effect can still be observed in the contour plot (Figure 49a) by the skewed shape of the contour lines towards the left inside the boundary layer. It is also observed in the profiles (Figure 49b), for instance, at the phases of flow reversal (0º and 180º). Furthermore, the profiles reveal the appearance of a logarithmic layer (shown linear in the semi-log plot in Figure 49b) for some part of the profile from about $z = 0.5$

mm to 10 mm and for a few of the phases only, from about 70º to 150º. Sleath (1987) also observed this possibility.

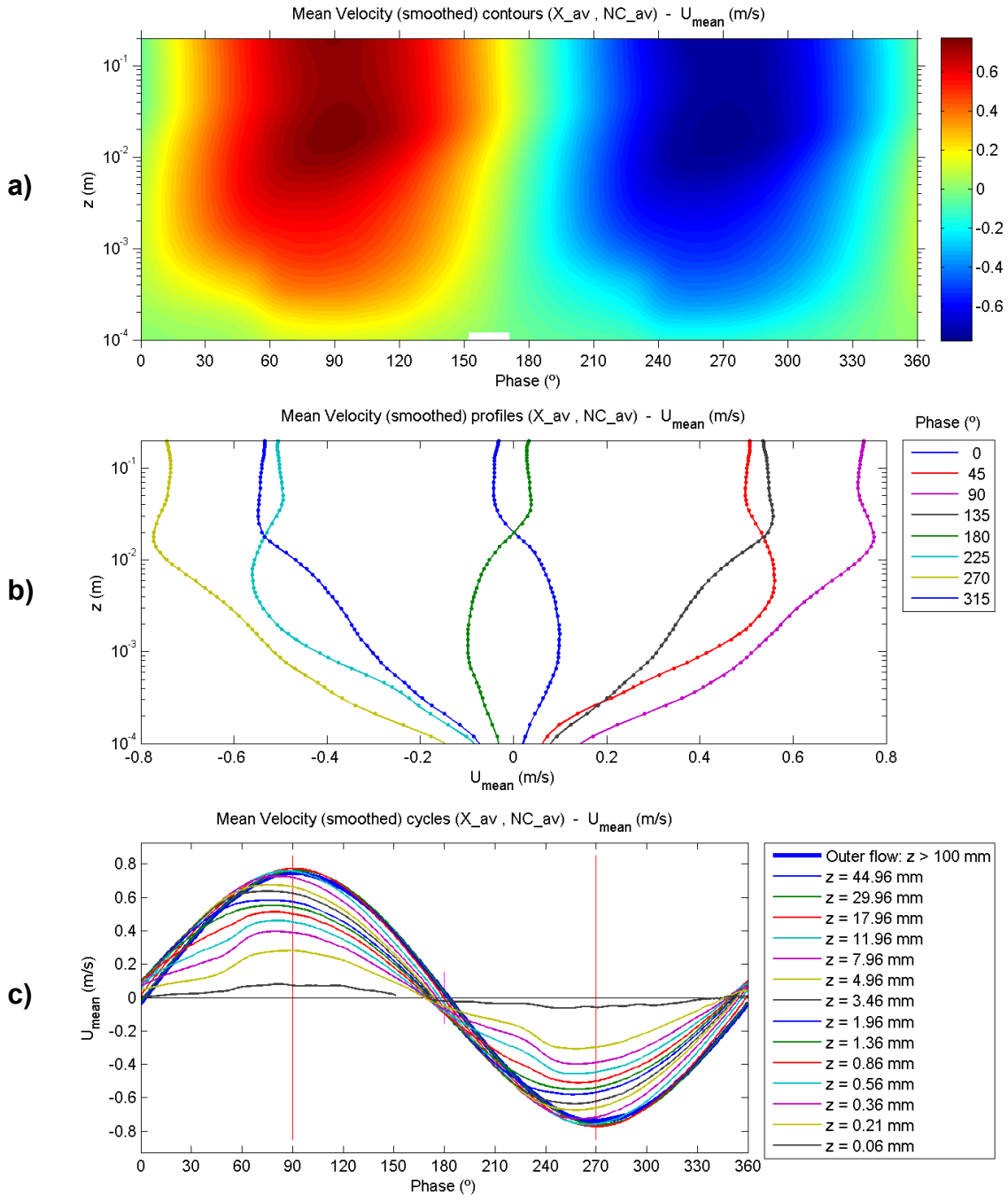


Figure 49. Mean flow velocity contours (a), profiles (b) and cycle evolution (c) for experiment 10.

6.3.4 Comparison for all experiments

Mean velocity profiles for phase 90º were compared for all 10 experiments and are presented in Figure 50. Note that velocities have been normalized by U_{out_max} and the vertical coordinate has been made dimensionless with the boundary layer thickness (δ_{90}) to facilitate comparison. The analytical solution for the laminar regime is included in the plot together with experiment 1 (laminar regime). As it can be observed in the plot, the results exhibited good agreement with the analytical solution since both profiles match quite well. In addition, the experimental data of Hino *et al.* (1983) is displayed together with experiment 5 for comparison, since they both have the same Re_w. In this case the profiles deviate significantly from each other for the lower part of the boundary layer ($z/\delta_{90} < 0.2$). On the contrary, for the upper part of the boundary layer the results match quite closely. The reason for this disagreement is unknown, but it could be due to the fact that they used air as fluid instead of water. Data from Jensen (1988) was also included for comparison with the high Re_w experiments (turbulent regime). Jensen (1988) didn't measure velocities in the transition regime and the smallest Re_w he reported was $Re_w = 1.6\text{x}10^6$, which is larger than Re_w for experiment 10 ($Re_w = 8.7\text{x}10^5$). Despite the Re_w difference, Jensen's data was ploted together with the profile for experiment 10, showing good agreement. No other studies were included since, unfortunately, there is very little experimental data available in the transition regime of the oscillatory flow. In this regard, the contribution from this work will be very useful to achieve a better understanding of the oscillatory boundary layer in the transition regime.

Looking at the shape of the mean velocity profiles in Figure 50 and their evolution as Re_w increased, it can be observed how the bottom part of the profile evolved from a linear relation (curve in semi-log plot) in the laminar regime to a logarithmic relation (straight line in semi-log plot) in the turbulent regime. This is an important observation, since self-similarity exists for the profiles in the laminar and the turbulent regimes, but it doesn't exist in the transition regime. The only way to know the shape of the profiles in the transition regime is through experiments like the ones presented in this work. This fact highlights the important contribution of this work to expand the knowledge in this complicated regime of the oscillatory boundary layer and can be used later on for the calibration of numerical models.

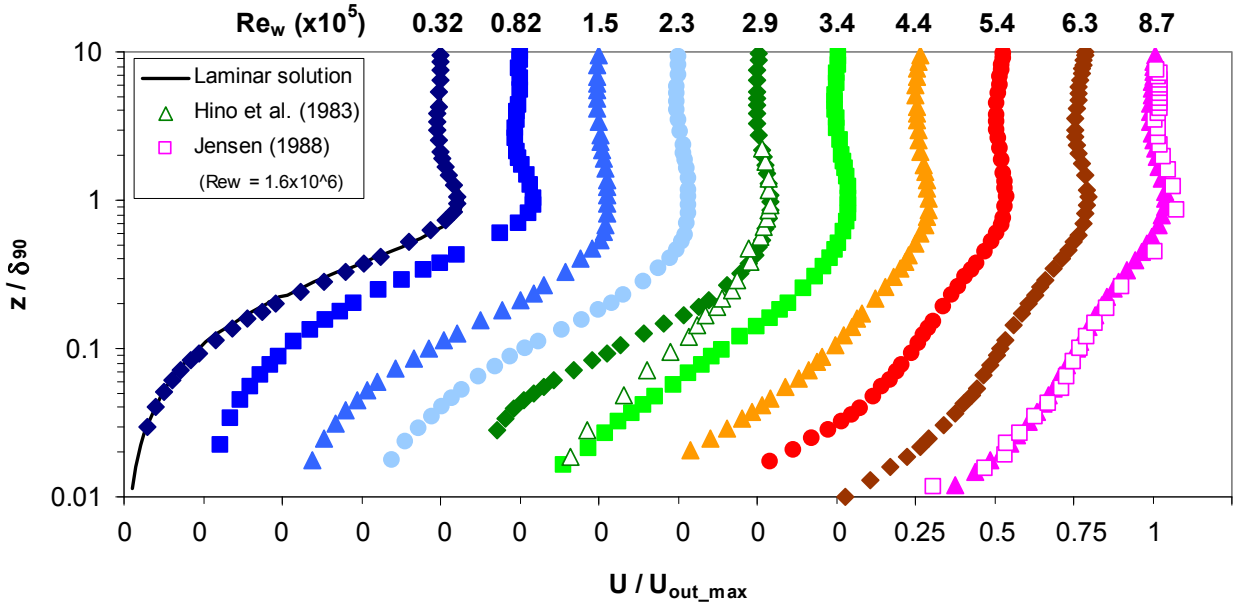


Figure 50. Normalized mean velocity profiles at phase 90º for all experiments in this work (solid symbols). The laminar solution (solid line) is ploted together with the profile for experiment 1 (laminar regime). Experimental data from Hino *et al.* (1983) is for $Re_w = 2.9\text{x}10^5$, similar to experiment 5. Experimental data for Jensen (1988) is for $Re_w = 1.6\text{x}10^6$ and is ploted together with the profile for experiment 10 (turbulent regime).

The mean flow velocities and particularly the changes experienced by the velocity field inside the boundary layer during the Re_w laminar-to-turbulent transition regime have a strong influence on the evolution of all the other variables presented in this work.

6.4 Boundary layer thickness

Boundary layer thickness (δ) was calculated in a similar way to the method used by Sumer *et al.* (1987), which was also used by Jensen (1988), and is illustrated in Figure 4. Upon this method, δ is defined as the z location of maximum streamwise velocity. This location can be easily identified from the mean velocity profiles shown before. As Jensen (1988) pointed out, this z location is coincident with the location at which the shear stresses experience the first zero crossing starting from the bed. The definition of δ in this way is similar to the steady boundary layer definition. The main difference in the case of oscillatory flows is that the location of the velocity maximum changes during the cycle, growing from the bottom up to the outer flow, where the true boundary layer threshold is (δ_{top}). In order to simplify this issue and provide a

reference value for δ, Sumer *et al.* (1987) used the boundary layer thickness calculated at phase ωt = 90º (δ_{90}). For the experiments in this work, δ was calculated for all phases of the oscillation and cycle evolution plots were generated. Only the plots for experiments 1, 5 and 10 are shown here, the others can be found in APPENDIX C.

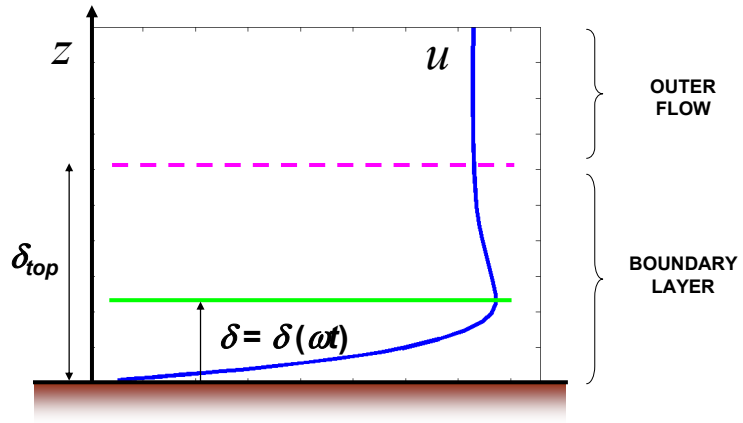


Figure 51. Definitions of oscillatory boundary layer thickness used for the results in this work.

The plots in Figure 52 show how δ grows continuously for each half-cycle, as it was expected from the theory and also from previous work on oscillatory flows. Looking at the plots in detail, it can be observed how the boundary layers from both half-cycles coexist for some time during the phases of flow reversal (around 0º and 180º). Indeed, when the flow changes direction, the flow closer to the bed changes before and ahead of the flow far from the bed, as it was observed from the mean velocity profiles. This part of the flow is already moving in the opposite direction (left) and so a new δ_{left} starts to grow close to the bed while higher up in the boundary layer the flow is still traveling in the original direction (right) and δ_{right} is still present.

The growth rate of δ during the half-cycle seemed to be quite constant for all the experiments. However, a more detailed look at the plots revealed a slight increase in the growth rate during the deceleration stage of the half-cycle for the higher Re_w experiments. This change is particularly noticeable for experiment 5, happening at around phase 60º (Figure 52b). This effect could be related to the increase in turbulence taking place during the deceleration stage, as will be shown later.

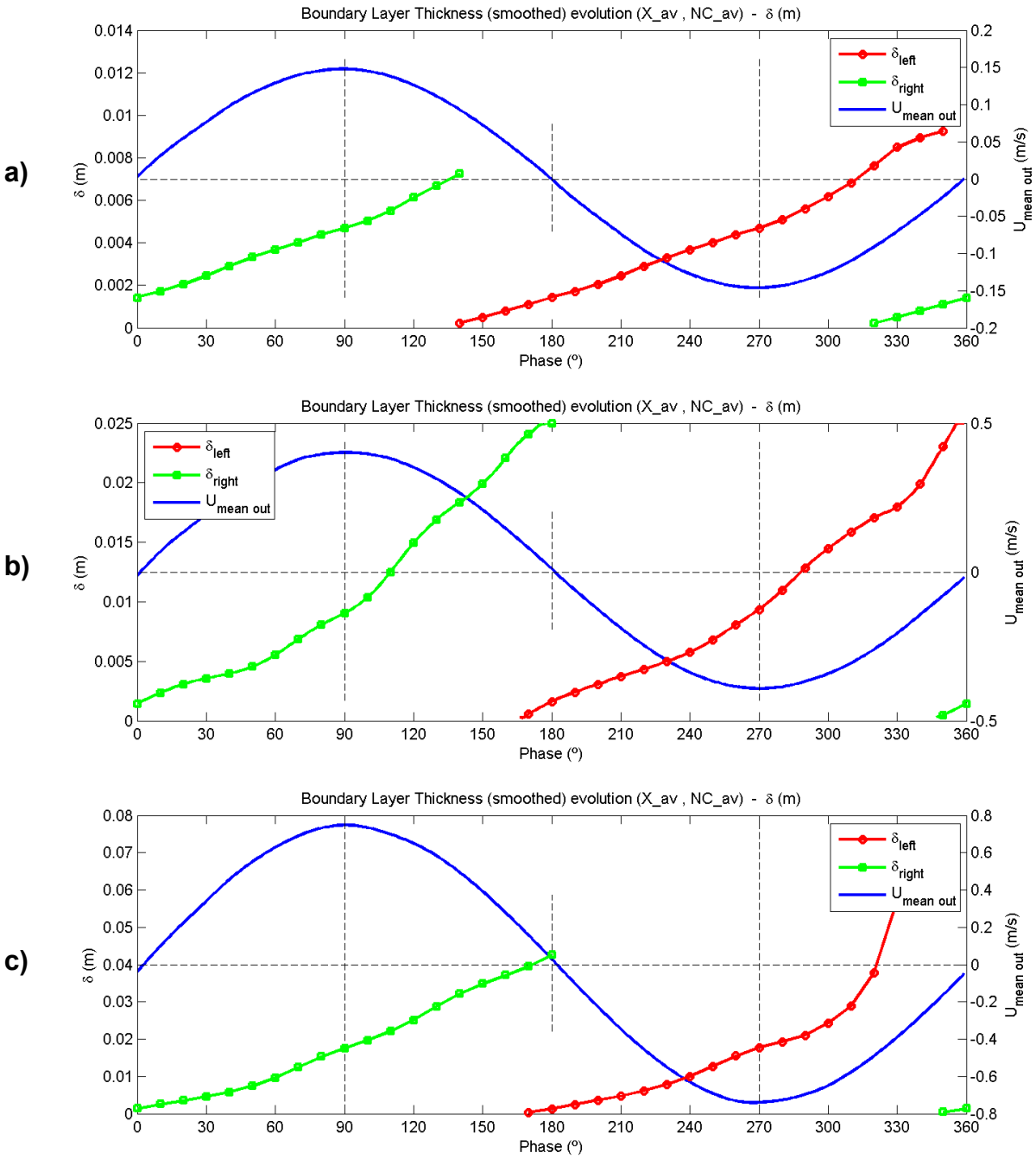


Figure 52. Boundary layer thickness cycle evolution for experiments 1 (a), 5 (b) and 10 (c).

Jensen (1988) showed similar results for the two experiments reported in the fully turbulent regime, although the growth rate in his plots seemed to increase slowly and evenly along the half-cycle. On the contrary, Hino *et al.* (1983), in their transition regime experiment, reported that δ grew slowly during the acceleration stage of the half-cycle and then it suddenly changed to a faster growth rate during the deceleration stage. This would be similar to the effect observed in

Figure 52b, although for the case of Hino *et al.* (1983) the change was more dramatic. Hino's definition of the boundary layer was slightly different though, since they used the location of the lower end of the log-law in the velocity profiles to determine δ.

Looking at the magnitude, Figure 53 shows boundary layer thickness evolution with Re_w for all experiments in this work. It can be observed that δ was larger as Re_w increased. Both δ_{top} and δ_{90} are included in that plot for comparison. In particular, for experiment 1 (laminar regime) δ_{top} reached a maximum value of around 9 mm on average between the two half-cycles. For experiment 5 (transition regime), this value was around 25 mm, while for experiment 10 (turbulent regime) it was around 45 mm. δ_{90} exhibited a similar behaviour as δ_{top} and the magnitude was about half of δ_{top}. Trend lines for the transition regime experiments are also included in Figure 53 and a straight line (in log-log scale) can be observed. Fitting of the trend lines to a power law relation revealed δ was roughly proportional to $Re_w^{1/2}$.

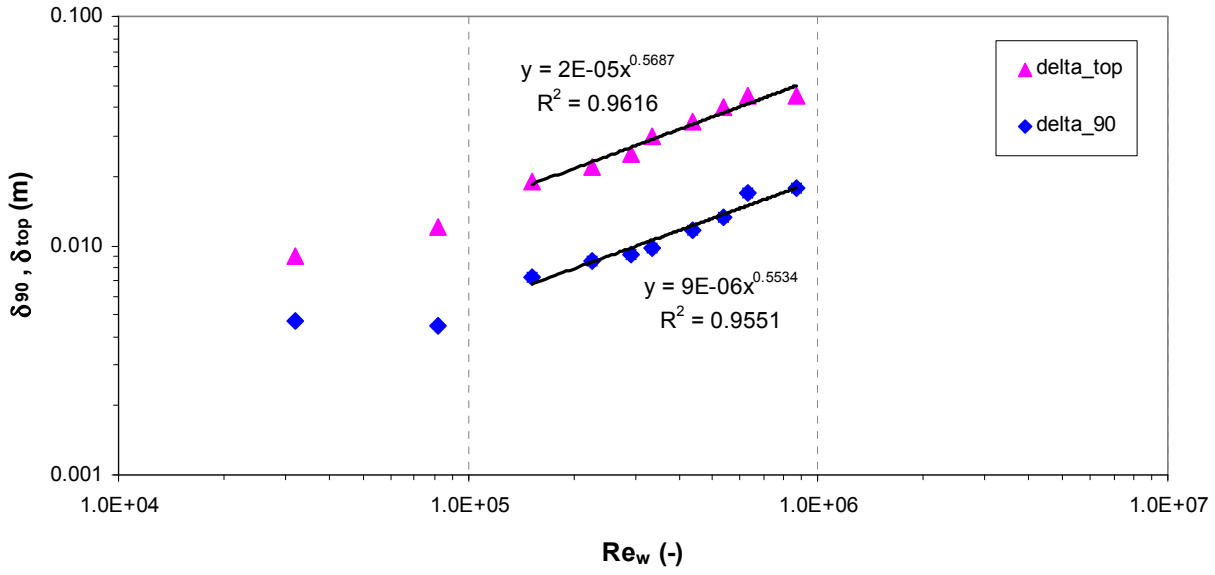


Figure 53. Evolution of boundary layer thickness (δ_{90} and δ_{top}) with wave Reynolds number (Re_w) for all experiments. Trend lines for the transition regime range are also displayed.

Boundary layer thickness results for δ_{90} were compared with previous studies from the literature and are shown in Figure 54. In order to facilitate comparison, the relative boundary layer thickness (δ_{90} / a) was used, which is a dimensionless parameter. Lines for the laminar and

turbulent regimes are also included. In the laminar regime, the line corresponds to the analytical solution to the equations, given by $\delta_{90} / a = 6\pi / \sqrt{32\,\mathrm{Re}_w}$, and is proportional to the Stokes length. In the turbulent regime, the line corresponds to Fredsoe (1984) approximation. Experimental results from Hino *et al.* (1983) and Jensen (1988) are included, as well as results from the DNS simulation of Spalart and Baldwin (1987). The results of this work compare quite well with the previous studies, and provide a significant contribution to the state of the art given the lack of experimental data available in the transition regime.

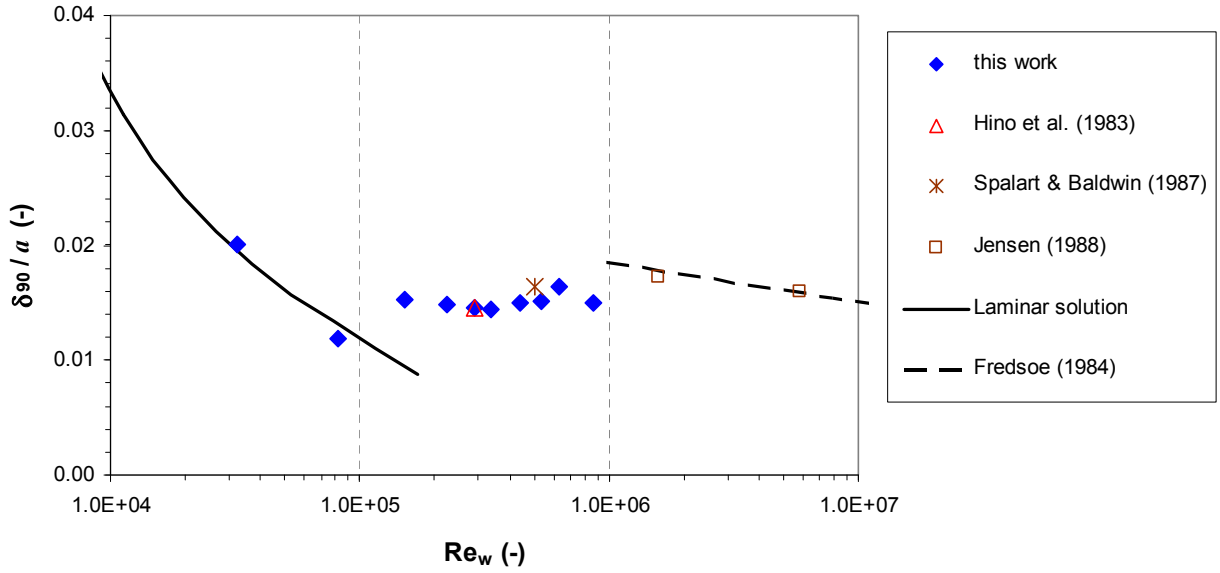


Figure 54. Evolution of relative boundary layer thickness (δ_{90} / a) with wave Reynolds number (Re_w).

CHAPTER 7

TURBULENCE CHARACTERISTICS

7.1 RMS velocity fluctuations

RMS velocities were obtained from the measurements after performing the root-mean-square of the instantaneous velocity fluctuations for all cycles, as described before in the methodology. This variable provides good information on the turbulent fluctuations of the flow. Contour plots, profile plots and cycle evolution plots were generated for all three velocity components (U_{RMS}, V_{RMS}, W_{RMS}). Only the contour plots are shown here, the others can be found in APPENDIX C.

Results for the streamwise component (U_{RMS}) are presented in Figure 55 for the three experiments selected 1, 5 and 10. In the plot corresponding to experiment 1 (Figure 55a) very low values of U_{RMS} are observed inside the boundary layer. This behavior was expected since this experiment falls in the laminar regime. Higher values are observed for the outer flow, and a clear divide between the two zones can be identified, most likely signaling the thickness of the boundary layer for that experiment. On the contrary, for experiment 5 (Figure 55b) regions of high U_{RMS} were found inside the boundary layer, mainly for the phases of decelerating flow, with the peak centered at around phase 110º and location z = 0.6 mm. The magnitude of the peak is around 0.057 m/s, which represents about 14 % of U_{out_max}. It is worth noting that outside the boundary layer U_{RMS} values are still very similar to experiment 1, however the colors in the plots are different due only to the different scale used. For experiment 10 (Figure 55c) with a Re_w near the fully turbulent condition, it can be observed how U_{RMS} increases in magnitude and also in terms of the phase range, with the high intensity zone now covering from about phase 50º to 130º, indicating widespread turbulence inside the boundary layer both before and after U_{out_max} (90º). The peak is now happening at phase 60º and location z = 0.4 mm, which is lower and in advance with respect to experiment 5. The magnitude is close to 0.084 m/s, which represents about 11 % of U_{out_max}.

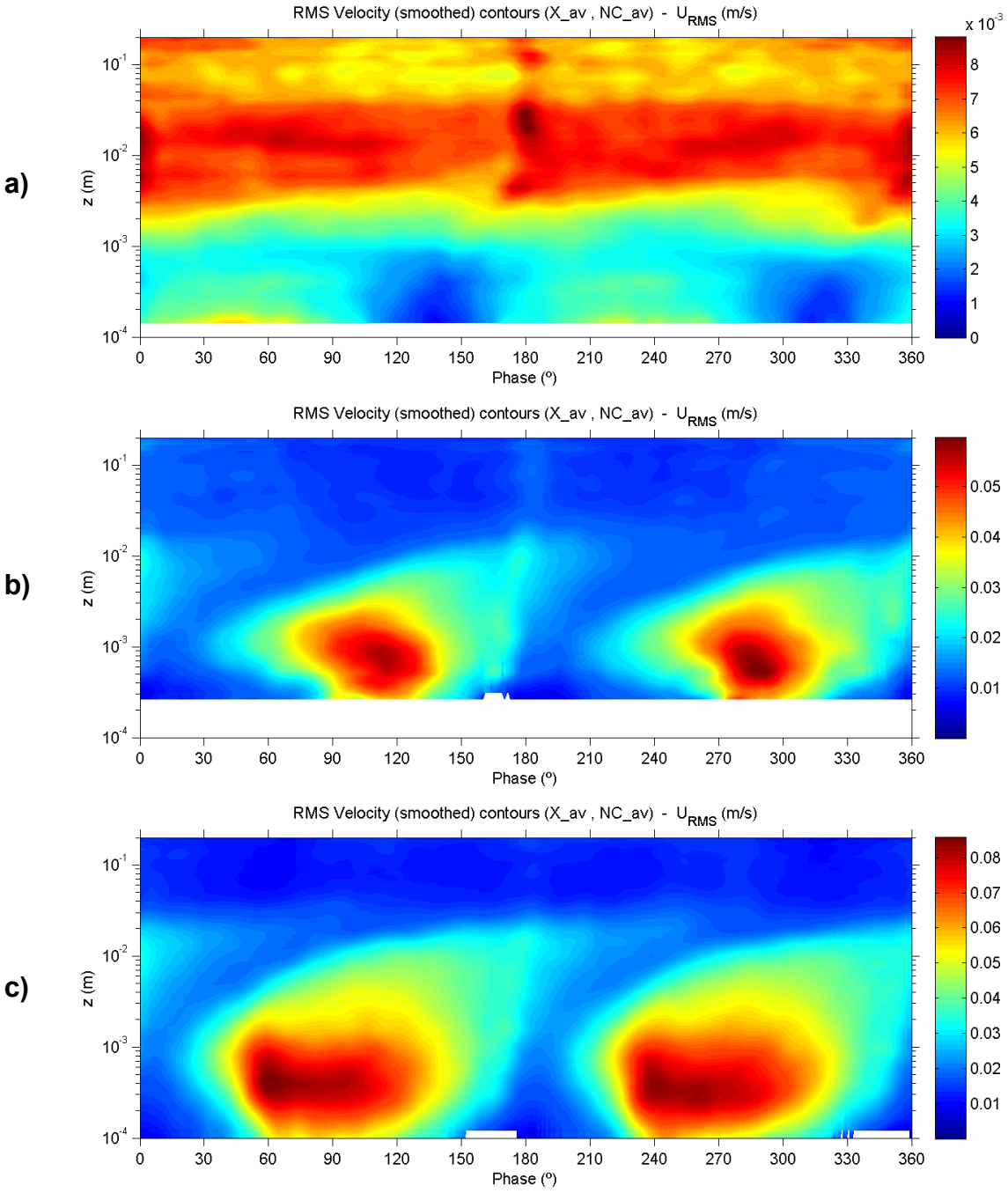


Figure 55. U_{RMS} velocity contours for experiments 1 (a), 5 (b) and 10 (c).

Results for the spanwise component (V_{RMS}) are presented in Figure 56. As it was mentioned in the methodology section, it was not possible to obtain measurements below z = 2 mm in any experiment for this velocity component. As such, some information was lost and the plots are not complete all the way down to the viscous sublayer. However some information was still obtained for some portion of the boundary layer.

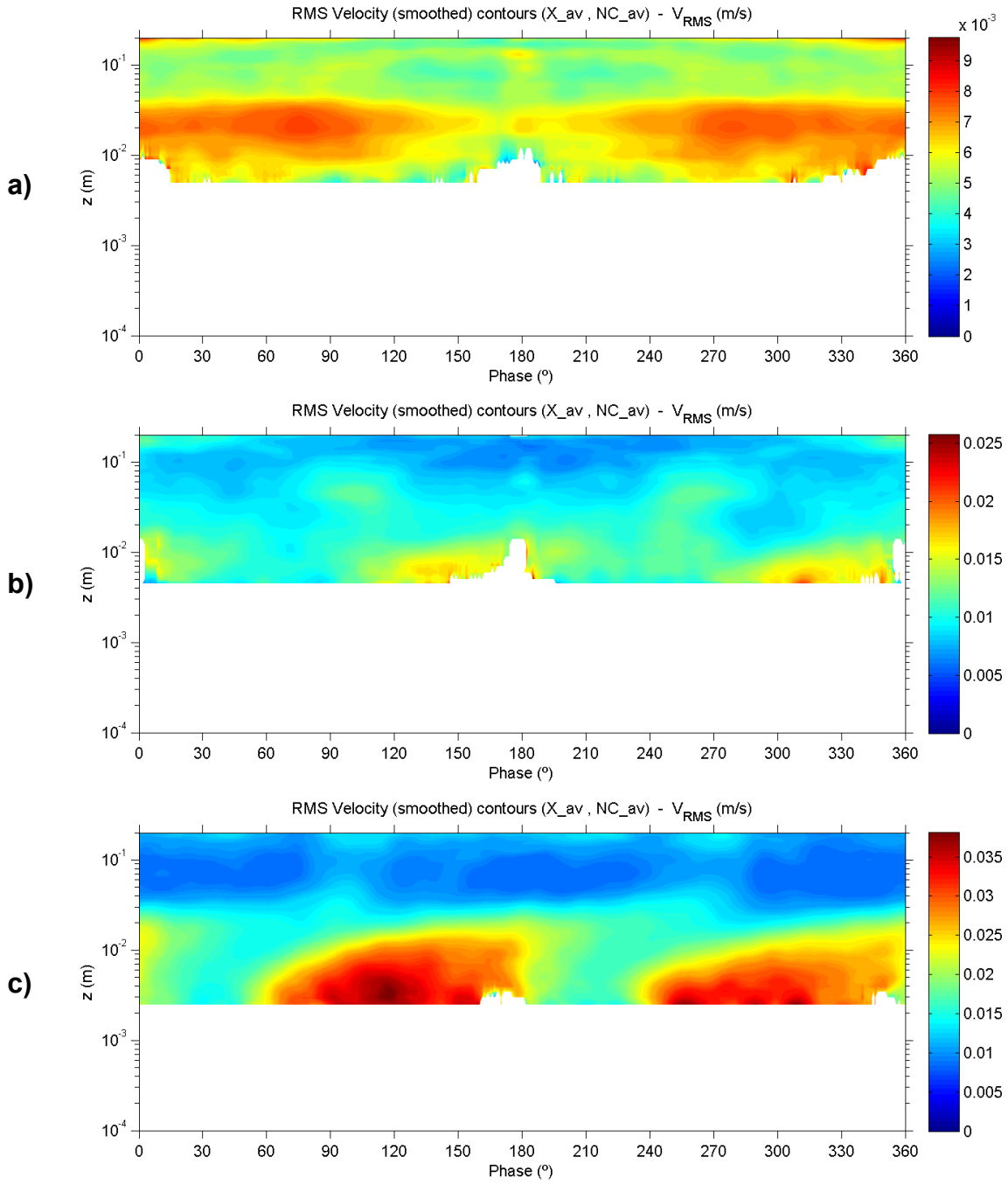


Figure 56. V_{RMS} velocity contours for experiments 1 (a), 5 (b) and 10 (c).

As it was observed before for the streamwise component, the residual turbulence for the outer flow remains almost constant regardless of the Re_w, having values around 0.01 m/s for all three experiments. In experiment 1 (Figure 56a) only the outer flow was captured. For experiment 5 (Figure 56b), some part of the turbulence inside the boundary layer is observed above z = 4.5

mm, while for experiment 10 (Figure 56c), a good portion of the high V_{RMS} zone was captured above z = 2.5 mm, extending from phases 60º to almost 180º. The magnitude of V_{RMS} in that zone was about 30 % less than U_{RMS}. It is not easy to determine the location or magnitude of the peak in this case due to the missing data for the bottom part of the boundary layer, however, looking at the shape of the turbulent zone, it could be argued that the peak could certainly be around that location at z = 2.5 mm.

Results for the vertical component (W_{RMS}) are presented in Figure 57. Again, this case highlights the clear divide between the outer flow and the boundary layer as far as the presence of turbulence is concerned. For the outer flow, values around 0.01 m/s are observed again, and remain almost constant with increasing Re_w for the three experiments. This value is very similar to the values obtained for U_{RMS} and V_{RMS} for the outer flow, which indicates that turbulence in the outer flow zone is very homogenous.

Inside the boundary layer, for experiment 1 (Figure 57a) the plot indicates there is absolutely no turbulence in the vertical direction for the laminar regime. For experiment 5 (Figure 57b), a zone of vertical turbulence is present mainly for the decelerating part of the oscillation and the peak has a value around 0.017 m/s, which is about 70 % smaller than the peak of U_{RMS}. It is located at z = 3 mm and around phase 120º, which is much higher and slightly behind the peak of U_{RMS}. For experiment 10 (Figure 57c), the zone of high turbulence extends from phases 60º to 160º and its vertical location changes upward with the oscillation, starting at around z = 1 mm and moving up to almost z = 10 mm. This effect is already present in experiment 5 as well, but becomes more evident as Re_w increases. This indicates upwards propagation of turbulence as the cycle develops, mainly happening during the deceleration stage of the oscillation. The magnitude of the peak reaches values around 0.03 m/s, showing that W_{RMS} grows as Re_w increases, as it was expected. This value is about 20 % smaller than V_{RMS}, and about 45 % smaller than U_{RMS}. It is also behind the phases of peak U_{RMS} for the same experiment.

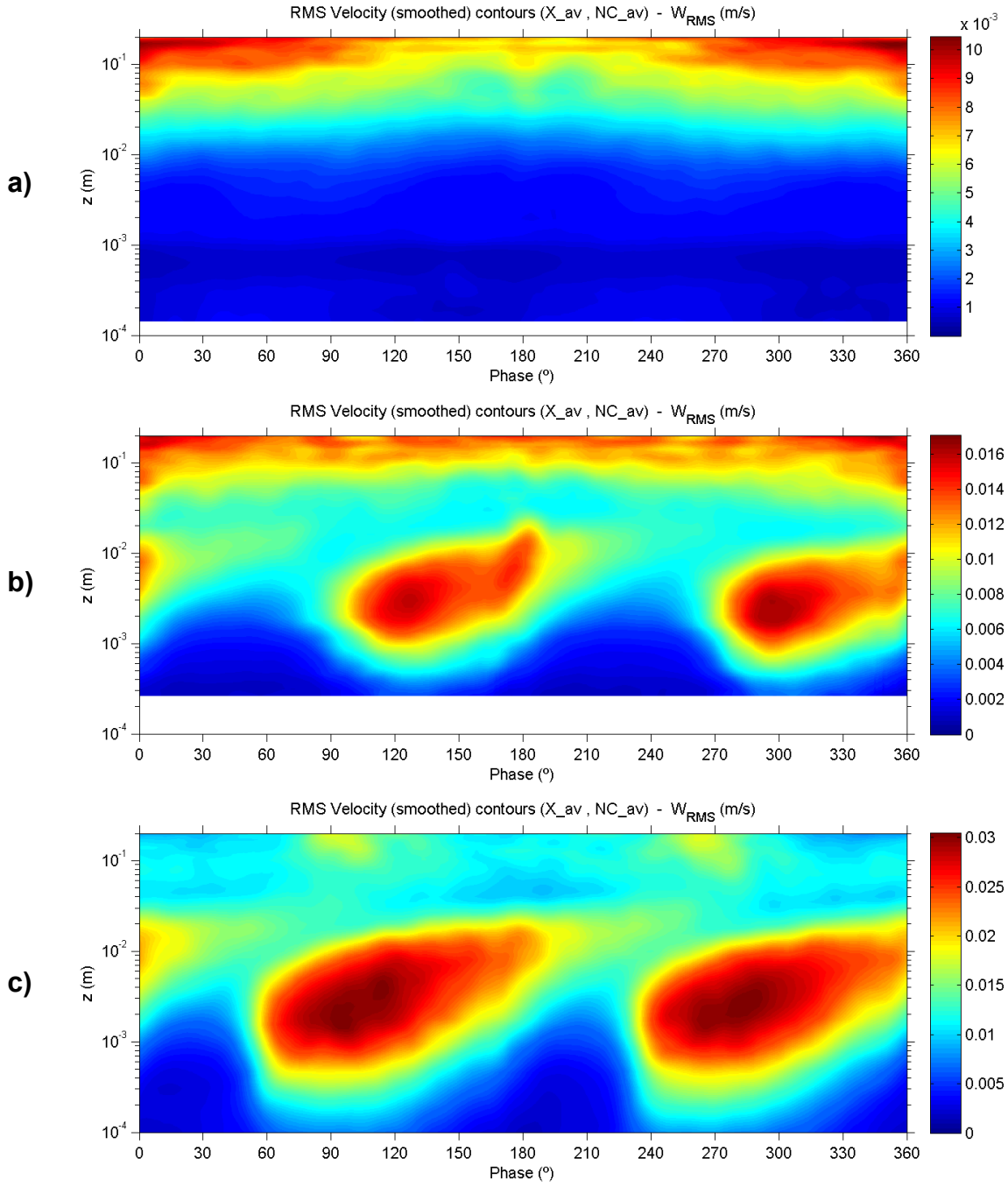


Figure 57. W_{RMS} velocity contours for experiments 1 (a), 5 (b) and 10 (c).

Overall, it was observed that the intensity of turbulence increased as Re_w increased, as it was expected. Also, turbulence in the streamwise direction (U_{RMS}) is dominant with respect the other two components and the magnitude is more than double of peak V_{RMS} and nearly triple of peak W_{RMS}. Regarding vertical location, the turbulence gets closer to the bottom as Re_w increases. In particular, U_{RMS} is more intense near the bottom starting as close as z = 0.2 mm in experiment 10,

and remains high up to z = 1 mm, then progressively reduces all the way to about z = 20 mm where it meets the magnitudes of V_{RMS} and W_{RMS}. On the contrary W_{RMS} is very small close to the bottom, and begins to increase at around z = 1 mm, extending upwards to z = 20 mm, together with V_{RMS}. This indicates it is the streamwise component the one responsible for the most turbulence in the lower part of the boundary layer.

7.2 Turbulent kinetic energy

Values of turbulent kinetic energy (*TKE*) were obtained from the velocity fluctuations, as described before in the methodology. This variable provides good information on the overall presence of turbulence in the flow as it combines the information from the three components. Contour plots, profile plots and cycle evolution plots were generated. Only the contour plots are shown here, the others can be found in APPENDIX C.

The plots in Figure 58 reaffirm some of the characteristics observed before from the RMS velocity fluctuations. In particular, for experiment 1 (laminar regime), no turbulence is observed inside the boundary layer, but a clear divide is present between this layer and the outer flow. Then as Re_w increases, values of *TKE* increase as well. Since the streamwise component was dominant in terms of turbulence magnitude, the zones of high *TKE* follow quite closely the same patterns as U_{RMS} before. It must be mentioned that since it was not possible to obtain velocity measurements of the spanwise component (v) below a certain location z, the results of *TKE* are lacking that contribution as well. This could be misleading in the interpretation of these plots, since *TKE* values would be slightly higher than shown in Figure 58 if the missing contribution from V_{RMS} was present. This is mostly felt in the zone between z = 1 mm and z = 3 mm.

Regardless of this shortcoming, the peak of *TKE* is very much located at the same spot as U_{RMS}. It can be clearly observed how the peak moves from around phase 110º (behind 90º) to around phase 60º (before phase 90º) as Re_w increases, and also how it gets closer to the bottom as well. As the oscillation progresses, turbulence moves higher up and is significant until about z = 30 mm. It starts to lose strength from about phase 120º until it vanishes at the phase of flow

reversal. After that it dissipates completely, and then everything happens again for the next half-cycle.

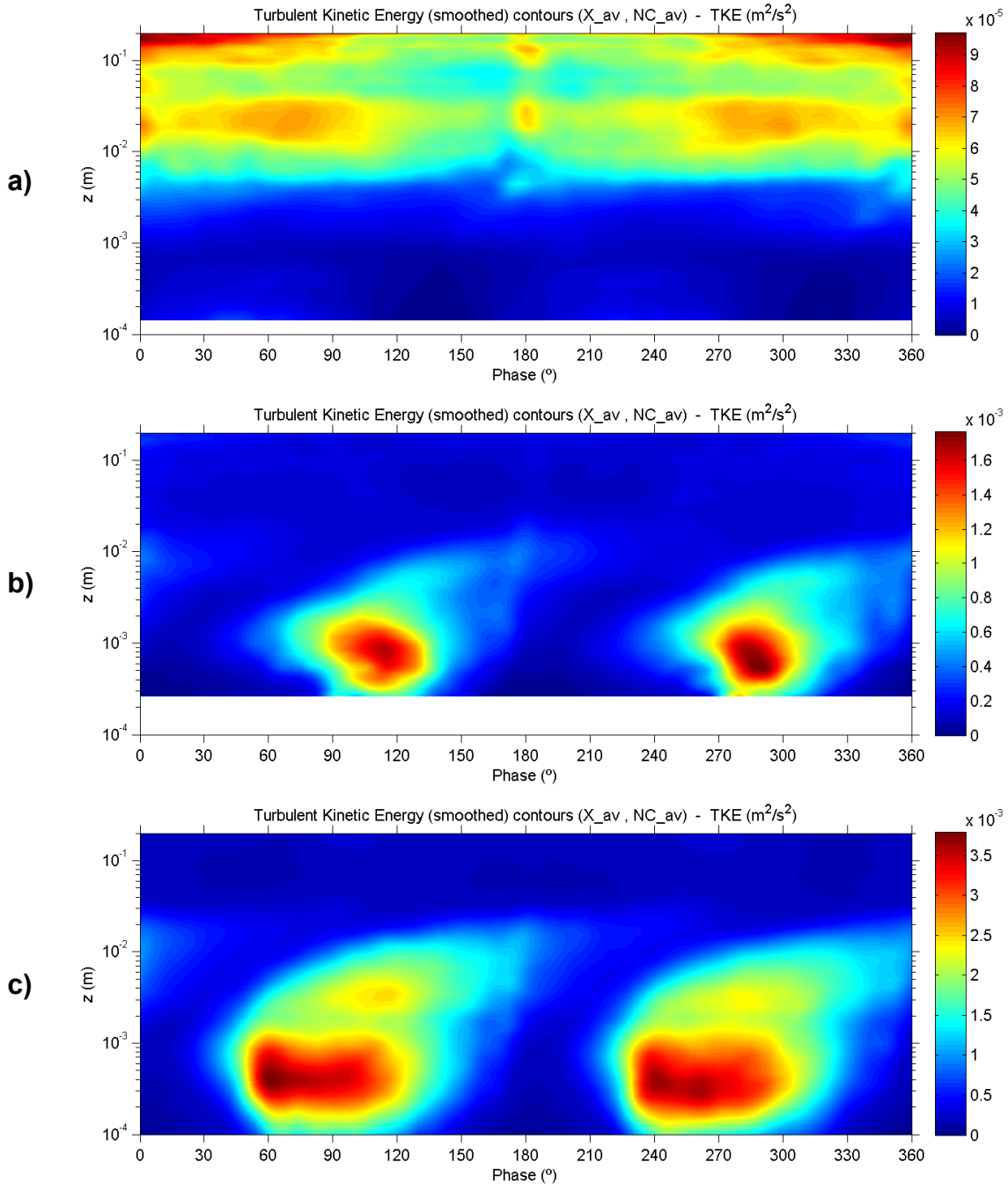


Figure 58. Turbulent kinetic energy contours for experiments 1 (a), 5 (b) and 10 (c).

In fact, the zone of minimum *TKE* between the two half-cycles corresponds to the growth of the new boundary layer for the reverse flow. This observation is in line with other variables which

indicate how independent the flow inside the boundary layer is from one half-cycle to the next, and how the complete boundary layer is reworked for every new half-cycle, with no remaining turbulence present from the previous one.

7.3 Turbulent shear stress (Reynolds shear stress)

Reynolds shear stresses were obtained from the velocity fluctuations, as described before in the methodology. This variable provides information on the presence of shear in the flow induced by turbulence. Three components of Reynolds shear stress can be computed from the measurements depending on the shear planes of interest. This way R_{xz} is obtained from velocity fluctuations in the streamwise (u') and vertical (w') directions, R_{xy} from the streamwise (u') and spanwise (v') directions and R_{yz} from the spanwise (v') and vertical (w') directions. Only the R_{xz} component is shown since it is the most interesting of the three for the purposes of this analysis and also because the lack of good v' measurements inside the boundary layer limited the availability of results for R_{xy} and R_{yz} in that region. Contour plots, profile plots and cycle evolution plots were generated. Only the contour plots are shown here, the others can be found in APPENDIX C.

Results are presented in Figure 59 for the three experiments selected 1, 5 and 10. In the plot corresponding to experiment 1 (Figure 59a), very low values of R_{xz} are observed inside the boundary layer. Again, this behavior was expected since this experiment falls in the laminar regime. As Re_w increases, R_{xz} becomes more intense, particularly around the phase of flow reversal (180º). This can be observed for experiment 5 (Figure 59b) where a high intensity zone occurs for phases between 160º and 190º. This effect is even more clear for experiment 10 (Figure 59c) in which the highest intensity is still near the phase of flow reversal, but the zone of high R_{xz} has expanded from phases 110º to 230º. The vertical location of the R_{xz} peak seems to remain constant regardless of Re_w, at a distance between 5 mm and 20 mm from the bottom. The wider zone of high R_{xz} seems to expand upwards as Re_w increases, reaching the top of the boundary layer in each experiment.

Values of R_{xz} in the plots change sign for every half-cycle. This is only due to the changing flow direction. However, it also helps to visualize the divide between the turbulent shear stresses

generated in each one of the two half-cycles. This indicates that these stresses disappear completely from one half-cycle to the next, and that they grow new again with every half-cycle.

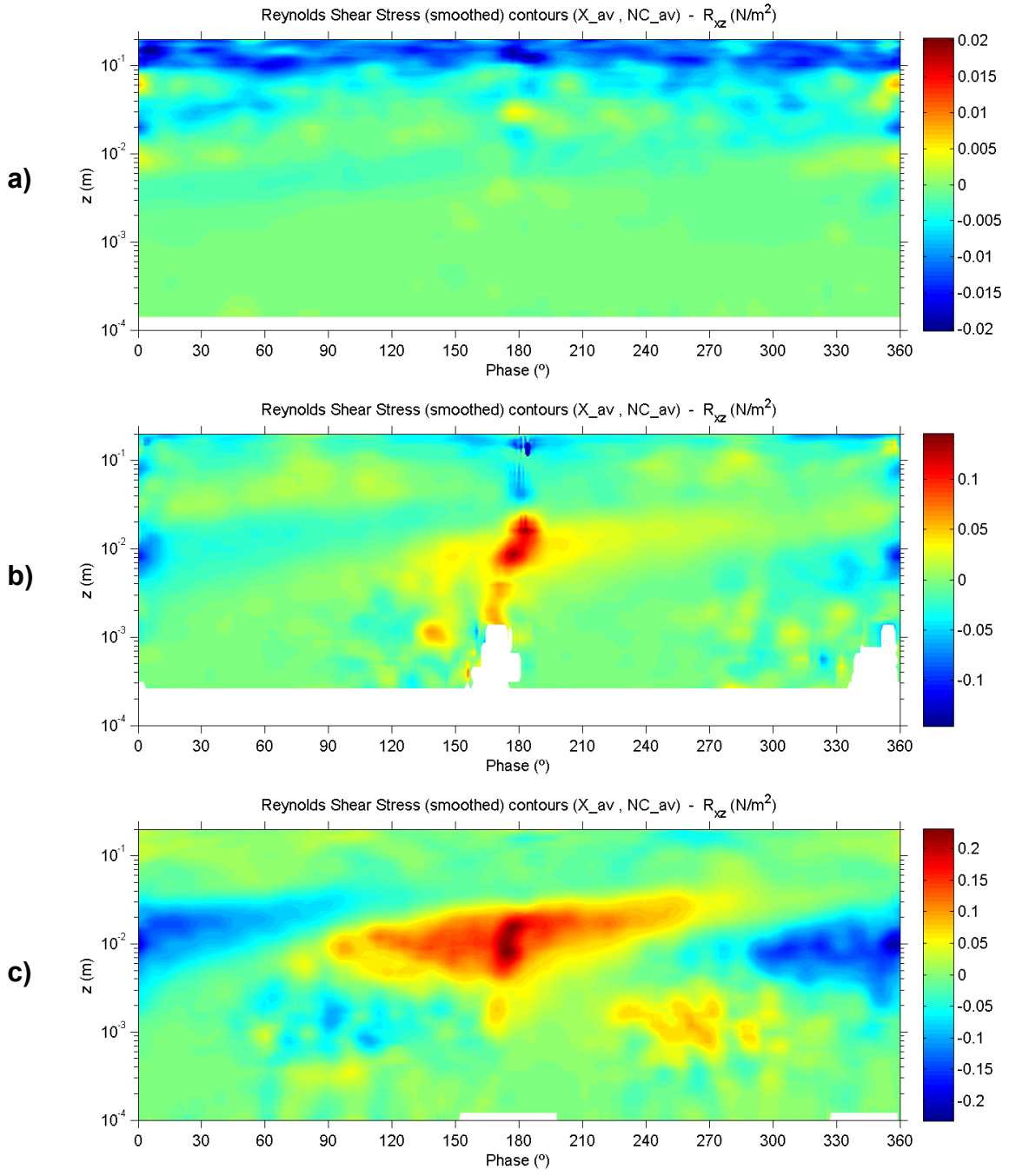


Figure 59. Reynolds shear stress contours for experiments 1 (a), 5 (b) and 10 (c).

Looking at the phases and vertical location where the Reynolds shear stresses occur, and together with other findings of this work and some of the previous knowledge in oscillatory boundary

layers, we could relate the presence of high R_{xz} in that zone to other features of these flows. In particular, it is likely that the occurrence of high R_{xz} is associated with the presence of a shear layer generated by the opposing mean flow happening near the phases of flow reversal. Indeed, as observed before in the mean velocity analysis, the flow in the lower part of the boundary layer reverses direction before the flow in the upper part of the boundary layer. Furthermore, this effect could also be associated with the presence of structures in the flow such as vortex tubes or turbulent spots found to appear around those phases and distance from the bottom (Carstensen *et al.*, 2010).

7.4 Viscous shear stress

Viscous shear stresses were obtained from the gradient of mean flow velocities, as described before in the methodology. This variable provides information on the presence of shear in the flow due to viscous effects and mean velocity gradients. Six components of viscous shear stress can be computed from the measurements depending on the shear directions of interest. Due to the nature of the oscillatory flow and that the measurements were performed at the centerline of the flume, gradients in the streamwise or spanwise directions are negligible and not relevant for the analysis. The main gradients happen in the vertical direction due to the presence of the bed, and that is the direction of boundary layer growth, which is the focus of this work. This way τ_{v_xz} was obtained from streamwise velocity gradients in the vertical direction (*du/dz*). Contour plots, profile plots and cycle evolution plots were generated. Only the contour plots and profile plots are shown here, the others can be found in APPENDIX C.

Viscous shear stress is typically present in areas of low turbulence, where viscous effects have a dominant role over turbulent effects. This is particularly evident in low Re_w experiments and in the lower part of the boundary layer near the bed. This zone within the boundary layer is typically called the viscous sublayer. The thickness of this sublayer is very small, typically in the order of microns for a water flow, and it was in no case larger than 1 mm for the transition experiments of this work. Thus, measurements inside this zone are very difficult to obtain. The high spatial resolution of the measurements in this work provided unprecedented measurements inside this viscous sublayer with great detail, and the results are shown in the following plots.

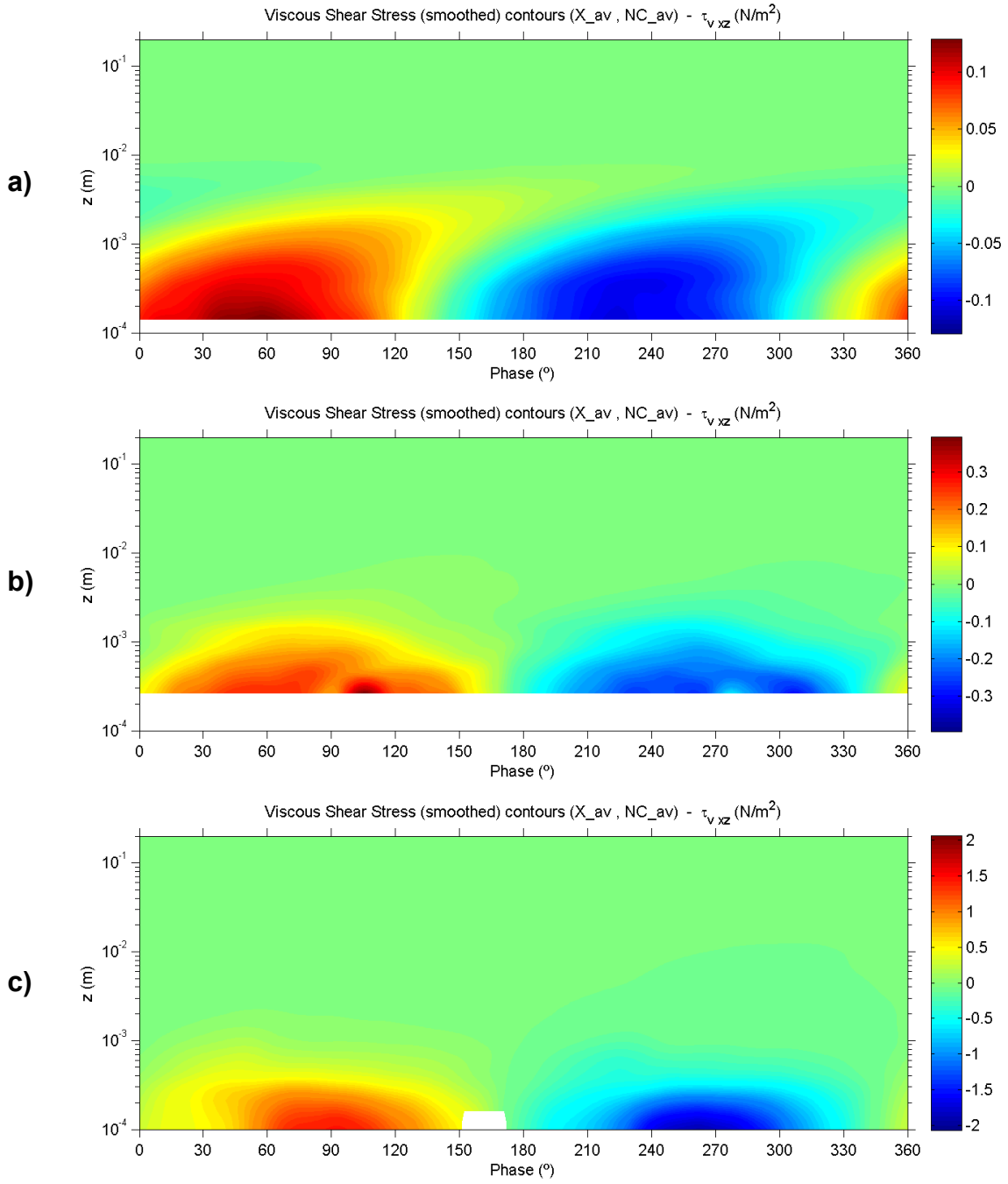


Figure 60. Viscous shear stress contours for experiments 1 (a), 5 (b) and 10 (c).

Figure 60 shows contour plots of viscous shear stress for the three experiments selected 1, 5 and 10. It can be observed how the highest intensity is found near the bed and it quickly vanishes far from the bed, as it was expected. This behavior is opposite to the behavior of the Reynolds shear stress presented before, which was higher far from the bed. Also, the highest values of viscous

shear stress are found around the phases of maximum streamwise velocity. This is also the opposite as what was found for the Reynolds shear stresses, which happen near the phases of flow reversal. In addition, as Re_w increases, the zone of high viscous stress gets compressed even closer to the bed, in agreement with the thickness of the viscous sublayer. In the plot corresponding to experiment 1 (Figure 60a), viscous stresses are observed up to z = 4 mm, with the peak near the bed at phase 45º and then following a similar upwards development in phase as the mean flow velocity. This behavior agrees well with the theory for the laminar oscillatory boundary layer presented earlier, as it was expected since this experiment falls in the laminar regime. From the plots corresponding to experiments 5 (Figure 60b) and 10 (Figure 60c), it is observed how the intensity of the viscous shear stress increases quite dramatically as Re_w increases, reaching values around 1.5 N/m^2 for experiment 10. Also, the zone of high viscous shear stress shifts back in phase with respect the laminar case, moving to phases around 90º.

Profile plots are also presented in Figure 61. In these plots, the vertical distribution of the viscous shear stress is highlighted. Particularly interesting is the shape of the profiles near the bed. As the measurements got closer to the bed it was found that the profiles reached a point where the intensity of the viscous shear stress no longer increased, but remained more or less constant. This can be observed for the last few points of each profile. This effect is a direct consequence of the shape of the mean velocity profiles themselves, such that when the velocity profiles become linear, the gradient becomes constant. This behavior meaning that those points with a constant viscous shear stress are truly inside the viscous sublayer. The distance at which this behavior is found changes for every phase in the cycle, and is also different for each experiment as it depends on Re_w, as explained before. However, having been able to measure this close as to find the viscous sublayer is a remarkable achievement for this work. This set of profile plots represents clear evidence to support this finding.

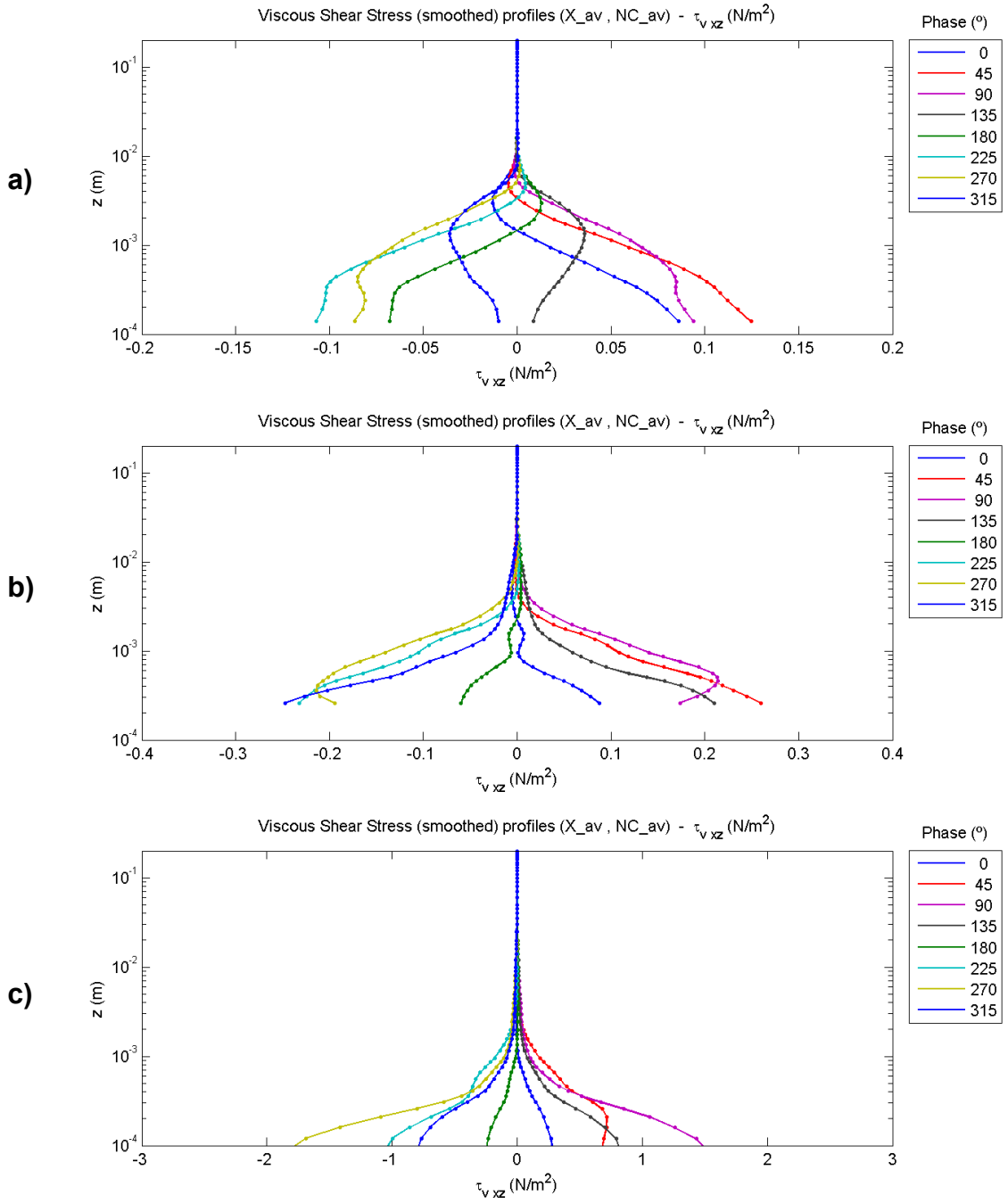


Figure 61. Viscous shear stress profiles for experiments 1 (a), 5 (b) and 10 (c).

7.5 Total shear stress

Total shear stresses were obtained from the addition of viscous and turbulent shear stresses, as described before in the methodology. This variable provides information on the combined presence of shear due to viscous and turbulent effects. Also, this variable was used to obtain bed

shear stress later on in the analysis. Only the total shear stress in the *x*-*z* plane was obtained (τ_{xz}). Contour plots, profile plots and cycle evolution plots were generated and are shown below for experiments 1, 5 and 10. The other experiments can be found in APPENDIX C.

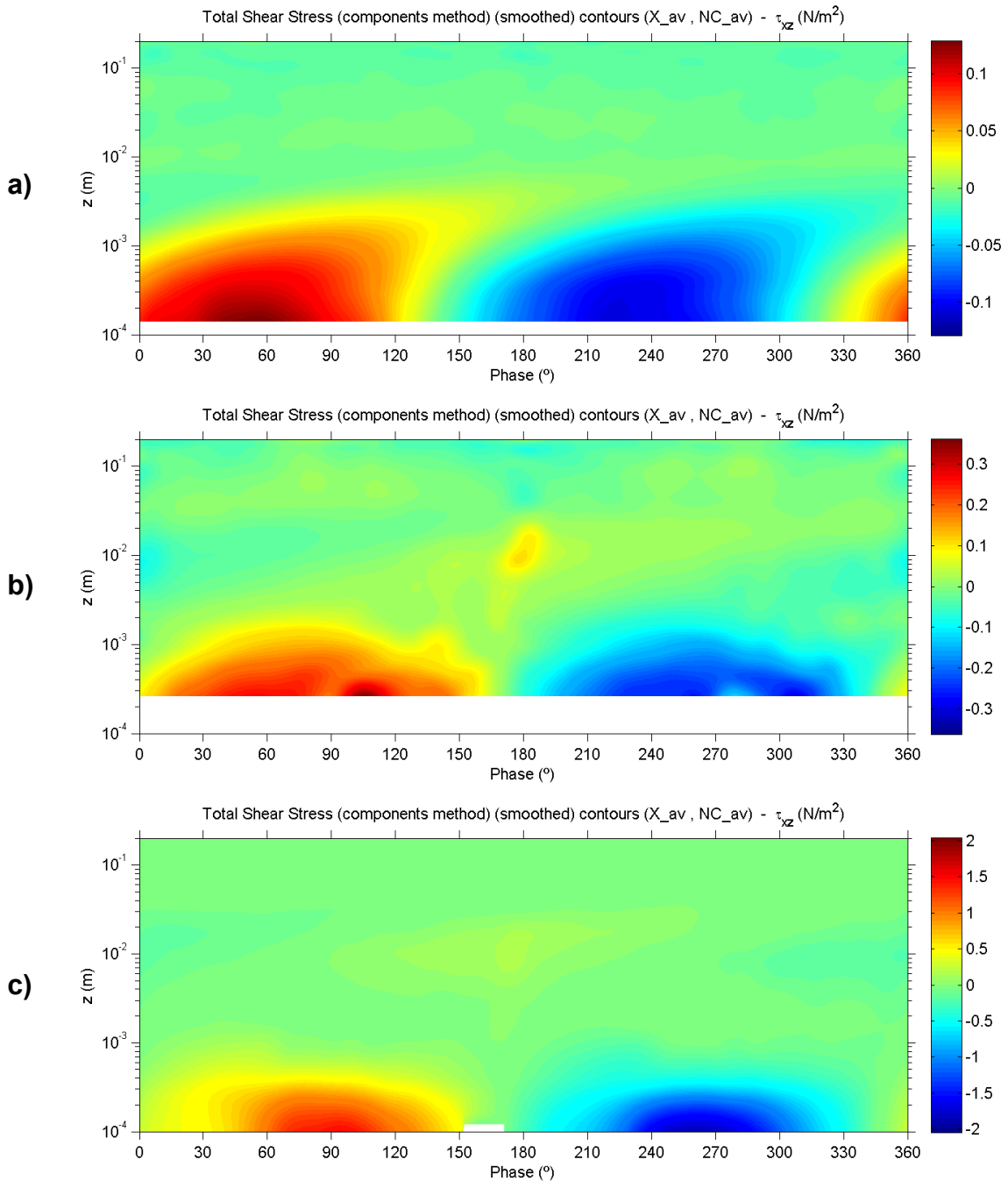


Figure 62. Total shear stress contours for experiments 1 (a), 5 (b) and 10 (c).

From the contour plots in Figure 62 the contribution of turbulence and viscous shear stresses can be evaluated for the different zones in the boundary layer. In particular, for experiment 1 in the laminar regime (Figure 62a), all stresses come from the viscous component. As Re_w increases, for experiment 5 in the transition regime (Figure 62b) the turbulent contribution becomes more important and shows values comparable to those of the viscous stresses. For experiment 10 (Figure 62c) near fully turbulent regime, it seems like the turbulent component losses strength relative to the viscous contribution. This behavior was not expected and constitutes a surprising finding, which deserves further investigation.

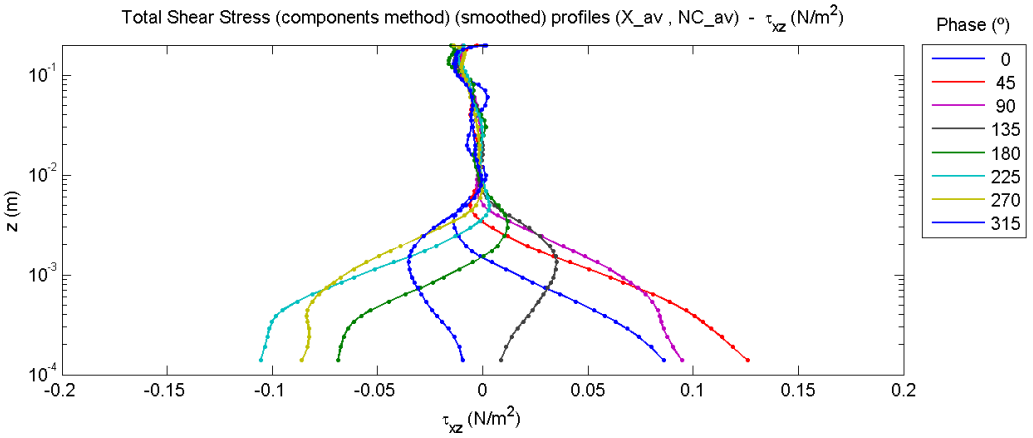
Additionally, the phase and vertical location of the areas of viscous and turbulent shear stresses are clearly distinct from these plots. Viscous shear stresses are more intense near the bed for phases around the mean velocity maximum, while turbulent shear stresses are high far from the bed for phases around the flow reversal. Furthermore, from the positive and negative signs obtained due to the different half-cycles, it can be stated that the shear stresses generated in one half-cycle are dissipated upwards and do not affect the new shear stresses generated for the next half-cycle.

From the profile plots in Figure 63, again the main features described before are clearly observed. In particular the relative magnitude of viscous versus turbulent shear stresses, and also the distinct divide in terms of vertical location between the two components.

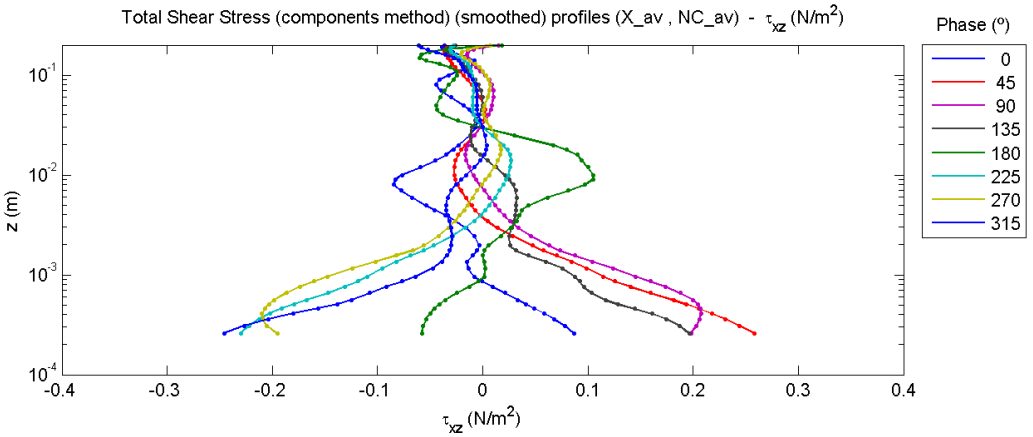
Cycle evolution plots for the total shear stress are presented in Figure 64. These plots mainly show the total shear stress near the bed and at a few vertical locations farther from the bed. It can be observed how the total shear stress near the bed changes in direction for every half-cycle, as it was expected. Also, an important effect can be observed from these plots in terms of the phase of the total shear stress peak for the different vertical locations. In particular, for the plot corresponding to experiment 1 (Figure 64a), the phase of the peak is around 45º near the bed, and quickly shifts back to values beyond 90º for z = 2 mm. As Re_w increases, interesting effects appear in the shape of the total shear stress near the bed. In particular, for experiment 5 (Figure 64b), a double peak appears near the bed, one situated before and one after phase 90º. In fact, the stronger of the two peaks is the one happening after 90º. Just a few microns farther from the wall

this effect is not felt as much and it quickly disappears beyond z = 1 mm. In the case of experiment 10 (Figure 64c), the total shear stress shows a very strong peak around phase 90º, slightly in advance, which again is completely vanished beyond z = 0.5 mm.

a)



b)



c)

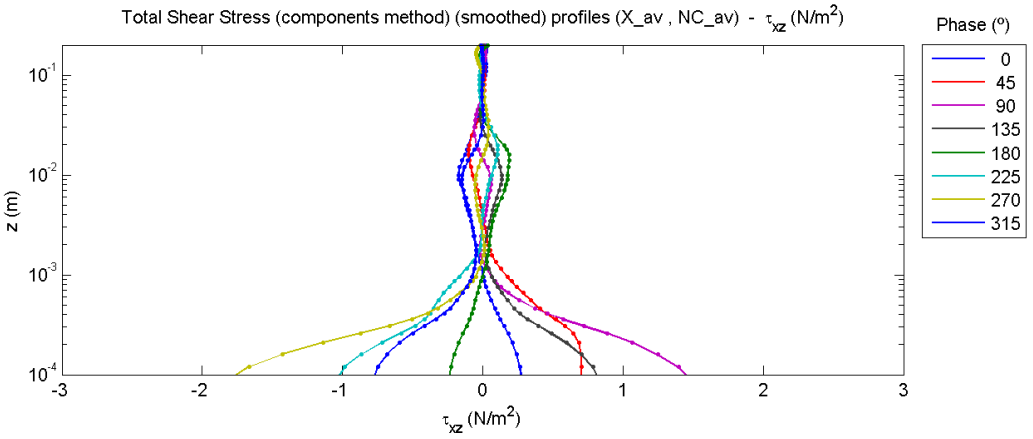


Figure 63. Total shear stress profiles for experiments 1 (a), 5 (b) and 10 (c).

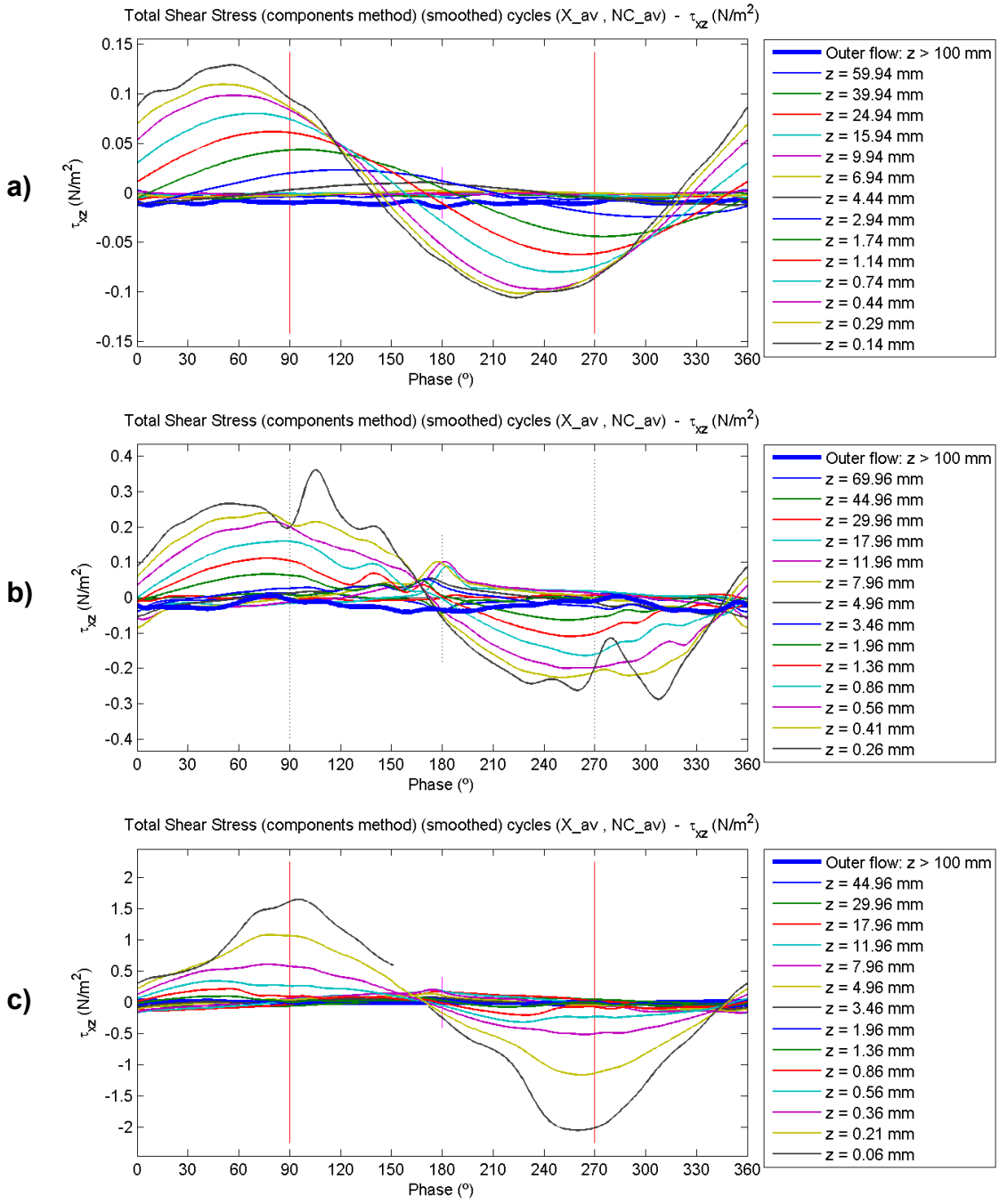


Figure 64. Total shear stress cycle evolution for experiments 1 (a), 5 (b) and 10 (c).

These observations highlight the importance of having measurements so close to the bed, which where only possible with the high spatial resolution provided by the LDV in combination with the traverse system. A different instrument that was not capable of measuring inside the first millimeter close to the bed would have completely misrepresented these effects and others regarding the behavior inside the oscillatory boundary layer.

7.6 Turbulence production

Turbulence production in the *x*-*z* plane (P_{xz}) was calculated as the product of the turbulent vertical flux of streamwise fluctuations (*u'w'*, present in the Reynolds shear stress) and the vertical gradient of streamwise velocity (*du/dz*, present in the viscous shear stress), as described before in the methodology. This variable provides information on the phases and vertical locations in the flow where turbulent kinetic energy is being produced. Typically, in turbulent shear flows, the flux is negative in zones of positive velocity gradient and, conversely, the flux is positive in zones of negative velocity gradient. The result of the product would give a negative amount in both situations. Thus, a negative sign is introduced in the calculation of turbulence production such that it becomes a positive term in the turbulent kinetic energy balance.

In the particular case of oscillatory flows, the shape of the streamwise velocity profile exhibits a positive gradient zone below the velocity peak, and a negative gradient zone above the velocity peak, so production can be expected to happen in both zones. However, the gradient closer to the bottom is typically much stronger and so most of the production should be concentrated near the bottom.

Contour plots were generated for all experiments. Only experiments 1, 5 and 10 are shown here, the others can be found in APPENDIX C. Results are presented in Figure 65. The magnitude of production in experiment 1 (Figure 65a) was very small, as it was expected for the laminar regime. The gradients were relatively strong but turbulence was negligible, so the production was not significant. For experiments 5 (Figure 65b) and 10 (Figure 65c), it can be observed that most of the production was located near the bottom, below z = 1 mm, for phases between 90º and 150º in experiment 5, and between 50º and 140º in experiment 10. This observation agrees with the behaviour exhibited by the *RMS* velocities and *TKE*, which showed how turbulence becomes stronger for earlier phases and gets closer to the bottom as Re_w increases. Also, a zone of low values of production was present in experiment 10 between z = 3 mm and z = 20 mm, for phases from 70º to 180º. This zone corresponds to the middle part of the velocity profile, just below the velocity peak, where the gradient is still positive.

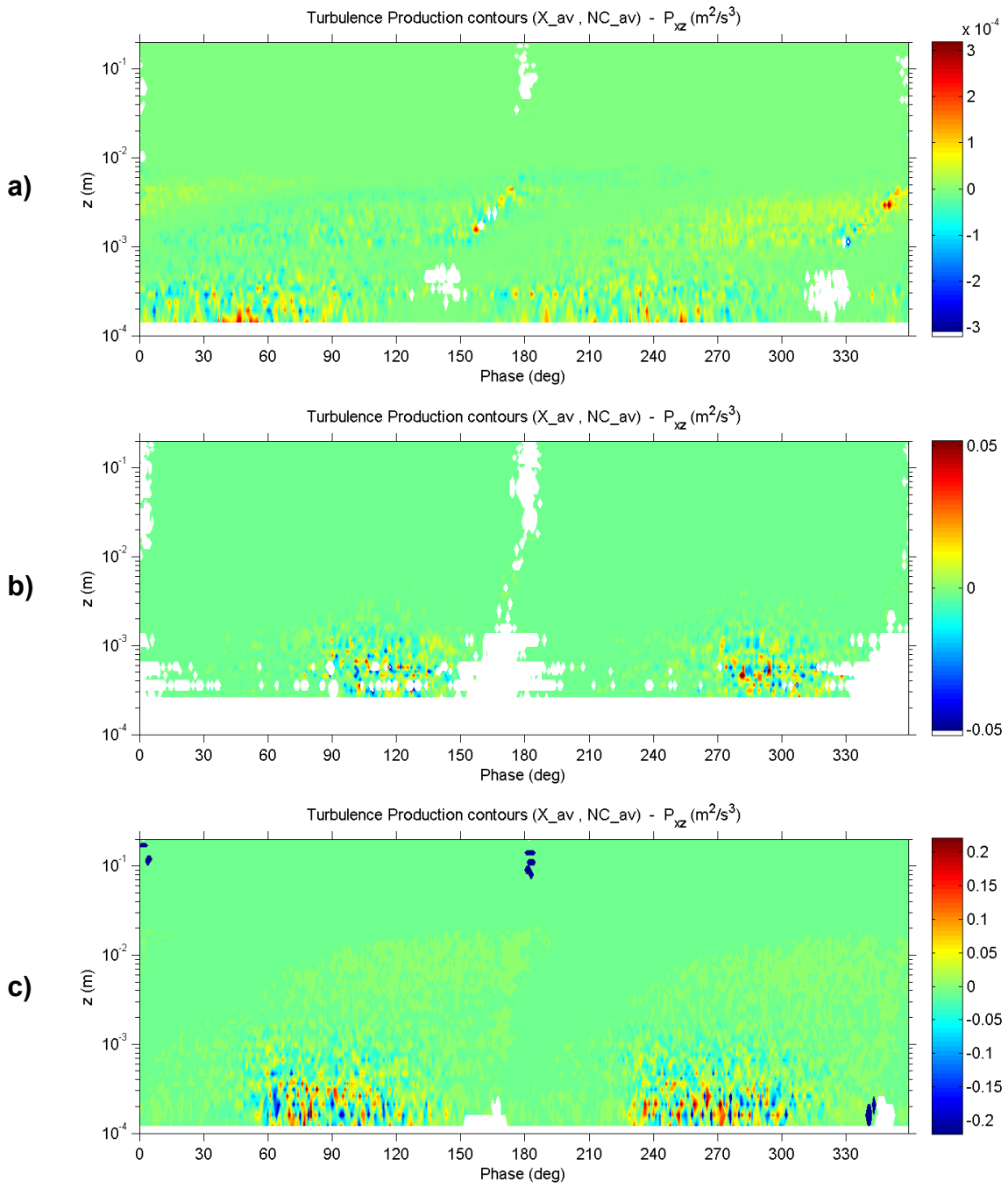


Figure 65. Turbulence production contours for experiments 1 (a), 5 (b) and 10 (c).

Additionally, in the zone of high turbulence production near the bottom (below z = 1 mm), spots of high positive production and high negative production were found next to each other for multiple phases and vertical locations, which was not expected. This type of behaviour is strange since production is typically a positive magnitude. The reason for this result is unknown and further investigation is needed on this regard. However, there is evidence in the literature of

flows with negative production (Gayen and Sarkar, 2011) and this kind of behaviour is typically associated with the presence of some type of flow structure. No measurements of flow structures were performed in this work, but that could be a possible explanation for this phenomenon. It is also possible that it was due to a problem with the velocity measurements, or during data processing, however that is unlikely.

7.7 Turbulent viscosity (Eddy viscosity)

Eddy viscosity (ν_t) is a synthetic parameter commonly used in the modeling of fluids that relates the small scale turbulence to the mean flow. This way the transfer of momentum by turbulent eddies can be accounted for without modeling the eddies themselves (this is known as the Boussinesq approximation). Eddy viscosity in the *x*-*z* plane (ν_{t_xz}) was calculated as the ratio between the turbulent vertical flux of streamwise fluctuations (*u'w'*, present in the Reynolds shear stress) and the vertical gradient of streamwise velocity (*du/dz*, present in the viscous shear stress), as described before in the methodology. Zones of very small gradients have been excluded from the calculation in order to avoid divisions by zero or by very small numbers, which would have resulted in unrealistically large values of ν_{t_xz}. In the particular case of the oscillatory flow, the streamwise velocity profile outside of the boundary layer is nearly constant, and so gradients are very small. As a consequence, ν_{t_xz} results were only obtained inside the oscillatory boundary layer. Regarding the sign, eddy viscosity is positive in zones where the flux and the gradient have opposite signs, which is the case of most turbulent shear flows. Negative values of eddy viscosity are rare, but possible (Liberzon *et al.*, 2007). They may happen when the flux and the gradient have the same signs, and can be found in zones of negative turbulence production.

Contour plots of eddy viscosity were generated for all experiments. Only experiments 1, 5 and 10 are shown here, the others can be found in APPENDIX C. Results are presented in Figure 66. Values of ν_{t_xz} for experiment 1 (Figure 66a) were very small since there was no significant turbulence. For experiments 5 (Figure 66b) and 10 (Figure 66c), zones of high ν_{t_xz} were found in the upper part of the boundary layer above z = 5 mm, and for phases between 0º and 120º. That zone corresponds to the part of the streamwise velocity profile above the velocity peak and with

negative velocity gradient. The magnitude of ν_{t_xz} in that zone was about $2x10^{-5}$ for experiment 5 and about $6x10^{-5}$ in experiment 10.

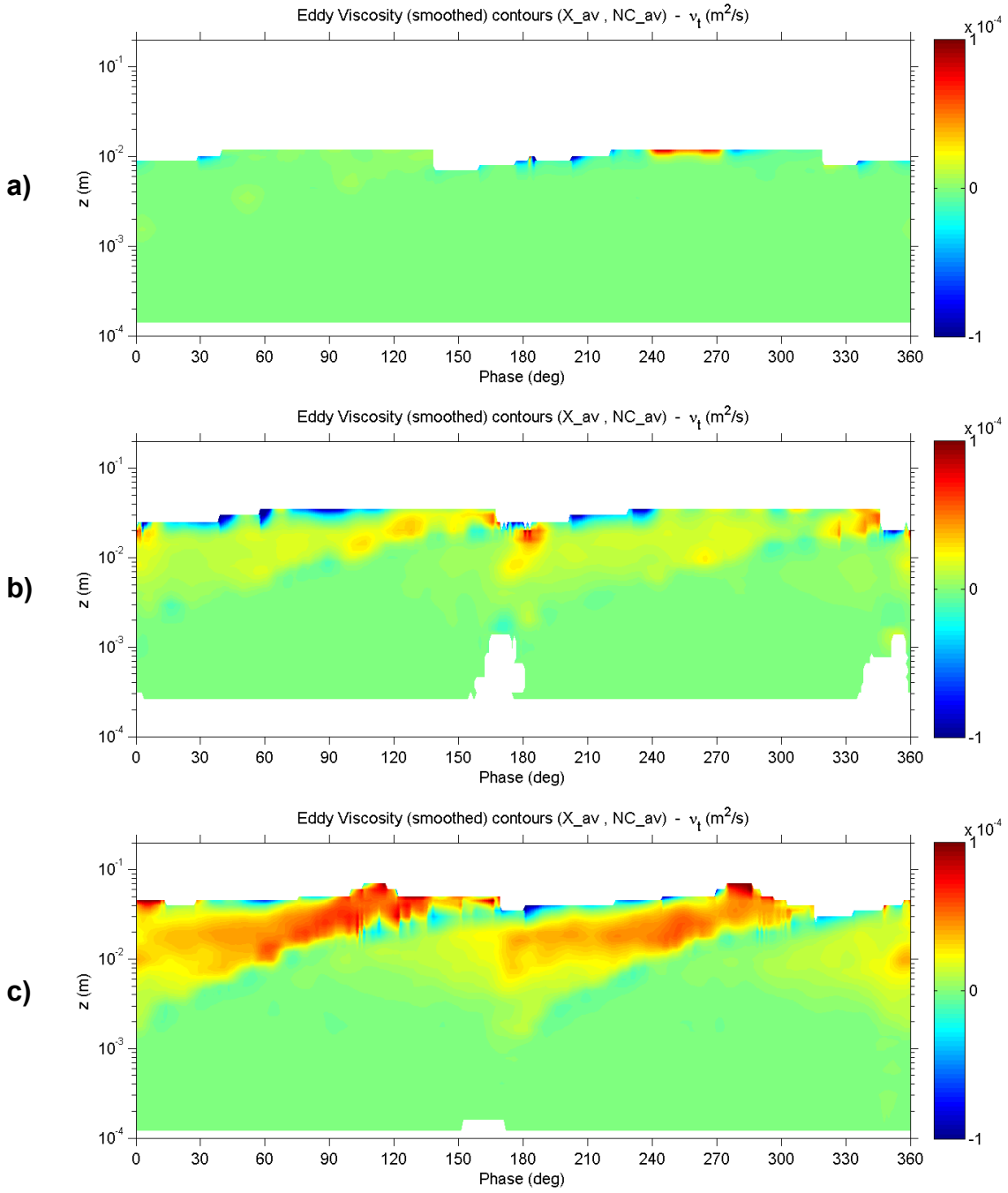


Figure 66. Eddy viscosity contours for experiments 1 (a), 5 (b) and 10 (c).

7.8 Skewness of velocity fluctuations

Skewness was obtained from the measurements by calculating the third order moment of the instantaneous velocity fluctuations for all cycles, as described before in the methodology. This variable provides information on whether the positive or the negative fluctuations are more important at a particular phase and z location in the flow. Skewness is a dimensionless number, and the scale has no direct physical translation. A skewness of 0 would correspond to a perfectly symmetrical distribution of positive and negative fluctuations around the mean velocity. Positive skewness means the positive fluctuations are stronger than the negative fluctuations, and the opposite would be true for negative skewness. When skewness is greater than 0.5, it is considered a significant value. When it is greater than 1, it means there is a strong skewness. In this work, skewness was calculated for the fluctuations of the three velocity components (u', v', w'), however only the streamwise (u') and vertical (w') components are shown since they are the most relevant for the analysis presented. Contour plots were generated for all experiments. Only experiments 1, 5 and 10 are shown here, the others can be found in APPENDIX C.

Results for the streamwise component (u') are presented in Figure 67 for the three experiments selected 1, 5 and 10. A very different behavior is observed between experiment 1 (laminar regime) and experiments 5 and 10. Indeed, it was expected that the skewness would follow a mirror pattern with opposite signs between the first and second half-cycles. This is because of the sign change of the mean flow velocity due to the oscillating nature of the flow. In fact, negative skewness in the second half cycle would mean that the stronger fluctuations are going with the mean flow. However, that is not the case for experiment 1 (Figure 67a), where skewness in the upper zone of the boundary layer (z = 2 mm to 20 mm) remains negative for the most part of both half-cycles. This means that for the first half-cycle, the stronger fluctuations go against the mean flow, while for the second half-cycle they go with the mean flow. Closer to the bed (z < 1 mm) the skewness shows some alternating patterns following the change in mean velocity, which is the behavior expected. This observation will be useful to better understand the quadrant analysis patterns later on. It must be noted that skewness is a relative variable, and so, even though it may show high values for experiment 1, the absolute magnitude of the turbulence in that experiment was very small, as shown in the RMS velocity fluctuations before. For that

reason, any interpretation of turbulent features for that experiment should be considered carefully.

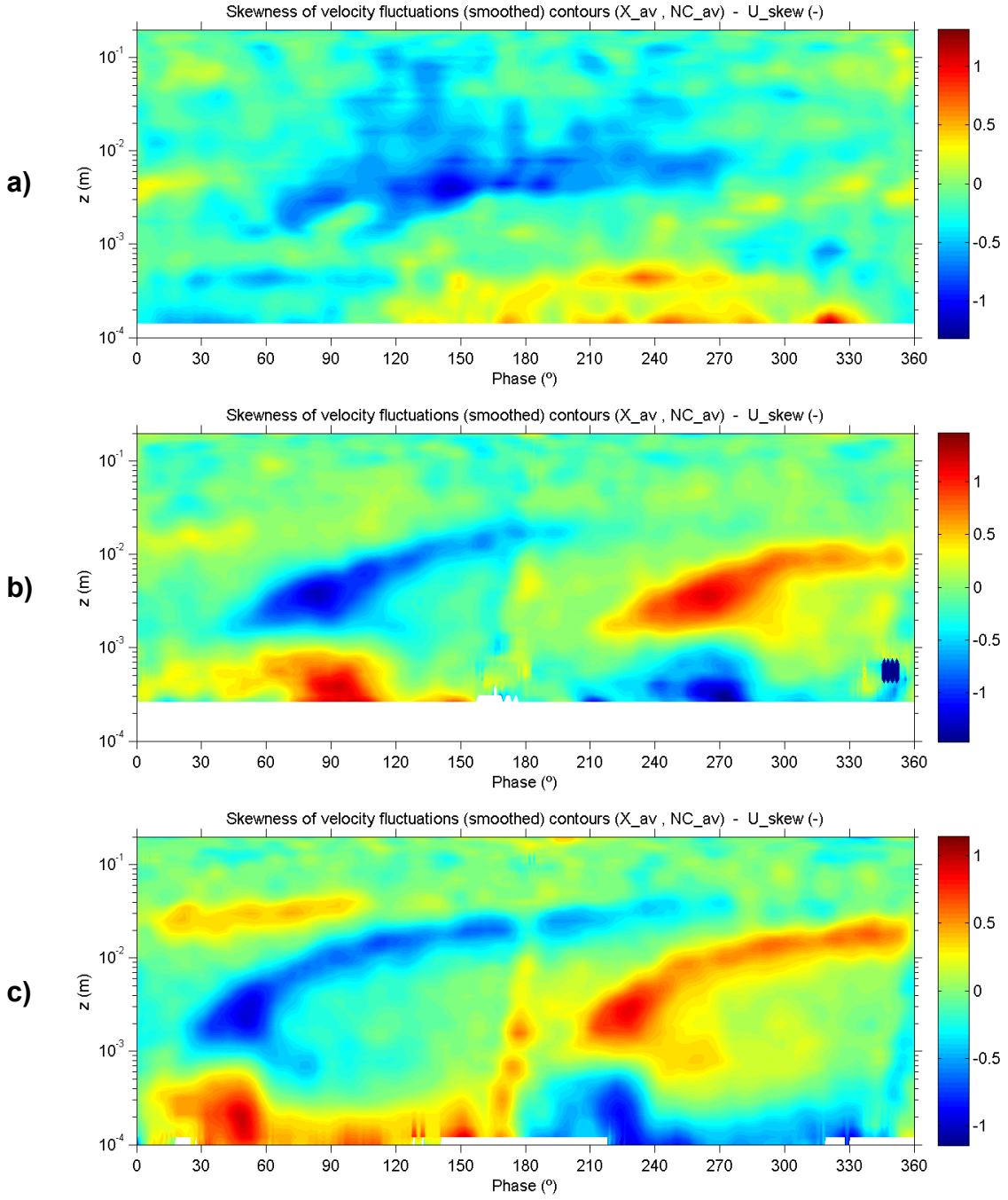


Figure 67. Skewness of *u'* contours for experiments 1 (a), 5 (b) and 10 (c).

Experiments 5 (Figure 67b) and 10 (Figure 67c) show some strong skewness patterns at certain phases in the oscillation. In particular, for the upper part of the boundary layer (z = 2 mm to 30

mm) a zone of negative skewness in the first half-cycle (and positive in the second half-cycle) indicates that the stronger fluctuations are going against the mean flow, specially for phases 45º to around 120º, moving upwards along the cycle. In the case of experiment 10 this effect happens earlier in the cycle, for phases between 30º and 60º. In the lower part of the boundary layer, closer to the bed (z < 1 mm), and inside the viscous sublayer, the skewness pattern is completely the opposite, with the stronger streamwise fluctuations going with the mean flow.

Results for the vertical component (w') are presented in Figure 68 for the three experiments selected 1, 5 and 10. Again, the skewness pattern observed for experiment 1 (laminar regime) is quite different from experiments 5 and 10. However, there is a major difference with respect the streamwise component, since now the mean flow in the vertical is very close to zero for both half-cycles. As a consequence, positive skewness would indicate positive fluctuations in both half-cycles, and the opposite is true for negative skewness. Taking that consideration into account, it seems that the skewness pattern observed for experiment 1 (Figure 68a) now follows the expected behavior. In this case, skewness values are small, although still a general trend can be observed. In particular, for the upper part of the boundary layer (z = 6 mm to 20 mm) skewness is mostly negative, which would indicate that negative fluctuations are dominant in that zone. Below that, zones of mild positive and negative skewness are intermixed, with no dominant direction except for a few isolated spots.

Experiments 5 (Figure 68b) and 10 (Figure 68c) show similar skewness patterns for the vertical component as well. In particular, for the upper part of the boundary layer (z = 3 mm to 30 mm) a zone of positive skewness in both half-cycles indicates that positive fluctuations are stronger, starting around phase 90º for experiment 5 and earlier at phase 60º for experiment 10, and then moving upwards along the cycle. In the lower part of the boundary layer, closer to the bed (z < 1 mm), and inside the viscous sublayer, zones of mild positive and negative skewness are intermixed, with no dominant direction except for a few isolated spots, like in experiment 1.

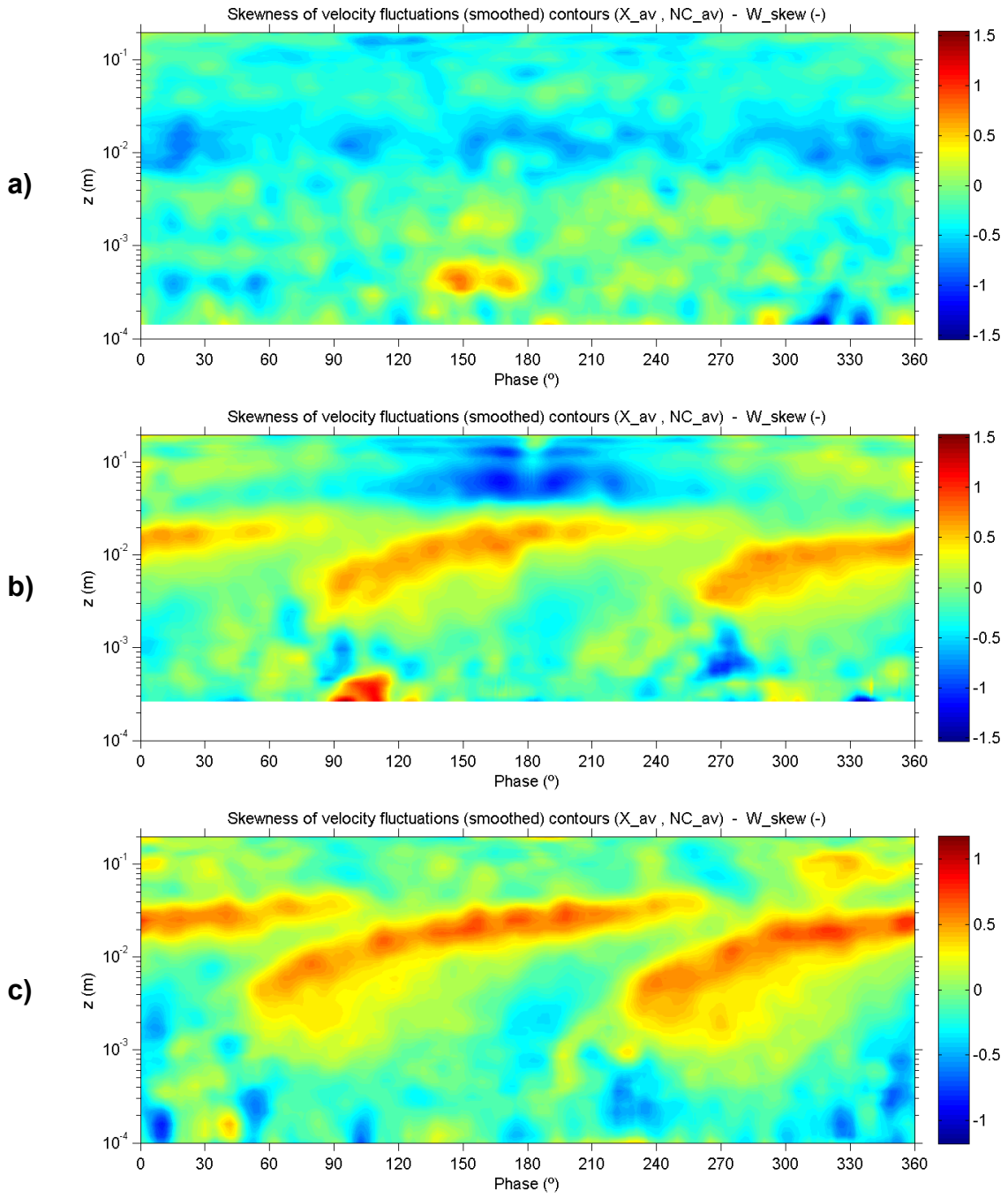


Figure 68. Skewness of *w'* contours for experiments 1 (a), 5 (b) and 10 (c).

7.9 Quadrant analysis: sweeps and ejections

Instantaneous velocity fluctuations in the streamwise (*u'*) and vertical (*w'*) components were analyzed to classify them in quadrants according to their interactions with the mean flow. This

technique was used to get some information on turbulent structures in the oscillatory boundary layer, in particular to identify sweeps and ejections.

Four different types of interactions can be observed according to the relation of the velocity fluctuations with respect to the mean flow. These are classified in 4 regions or quadrants as follows (see also Figure 69):

Q1:	$u' > 0$	& $w' > 0$	(outward interaction)
Q2:	$u' < 0$	& $w' > 0$	(ejection)
Q3:	$u' < 0$	& $w' < 0$	(inward interaction)
Q4:	$u' > 0$	& $w' < 0$	(sweep)

Ejections and sweeps contribute to positive shear stress and represent mixing of momentum that works against the pattern of the velocity profile. A threshold H is typically applied to distinguish between signals of interest and smaller interactions, such that:

$$\left|u' \cdot w'\right| > H \cdot \left\langle \left|u' \cdot w'\right| \right\rangle \tag{51}$$

where H is a suitable threshold coefficient, typically around 2 or 3, the symbol $\left| \right|$ means absolute value and the symbol $\left\langle \right\rangle$ means ensemble average.

It must be mentioned that, for the interpretation of the quadrant analysis in the case of oscillatory flows, it is necessary to take into account that the mean flow can be positive or negative depending on the phase of the flow. As such, during the phases of returning flow, a negative fluctuation ($u' < 0$) will in fact go with the mean flow ($U < 0$) and so the interactions will be a Q1 or Q4, whereas a positive fluctuation ($u' > 0$) will go against the mean flow ($U < 0$), and so the interactions will be Q2 or Q3. Taking this into account, the quadrant patterns should be very similar between the first and second half-cycles.

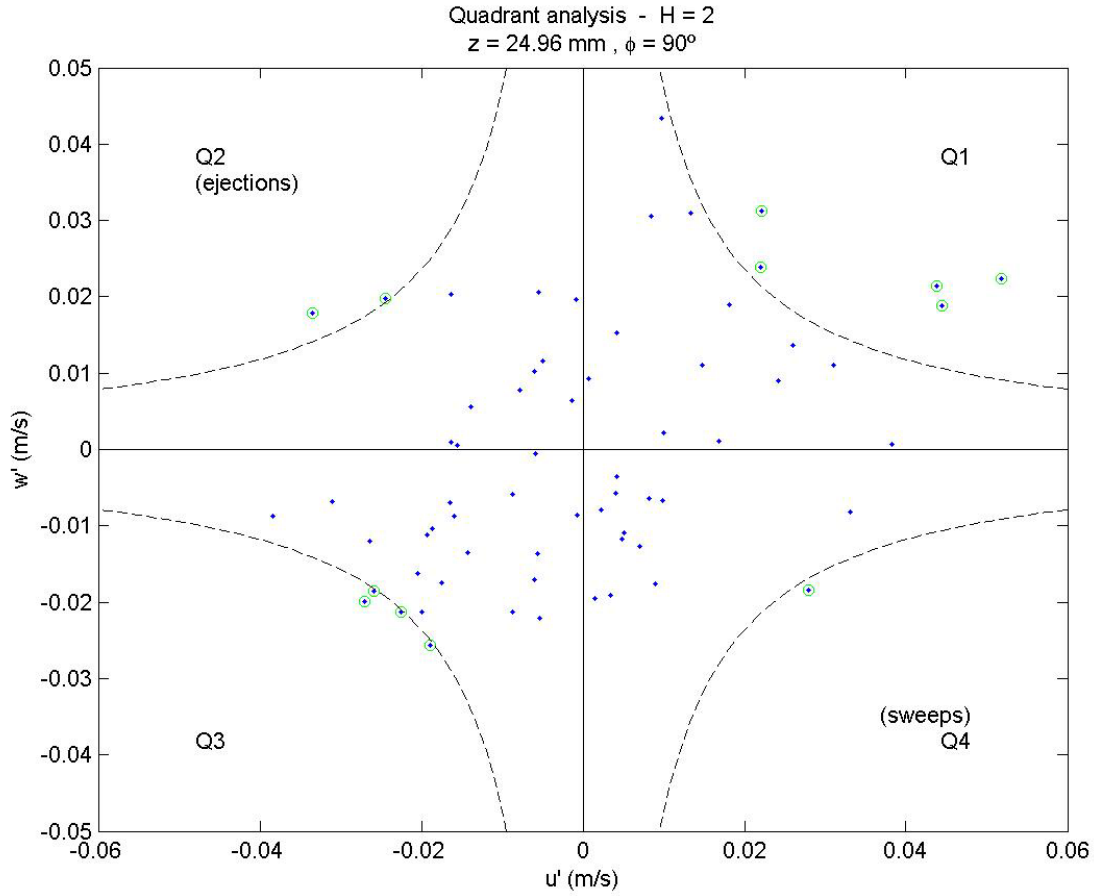


Figure 69. Quadrant analysis for experiment 10, at location z = 24.96 mm and phase 90º. Dashed lines represent the threshold given by H = 2. Highlighted points are above the threshold and are the only ones that count towards the quadrant analysis. For this particular phase and location the dominant quadrant is Q1.

Contour plots identifying each one of the four types of quadrant interactions were generated for all experiments. Results are presented in Figure 70 for experiments 1, 5 and 10. The other experiments can be found in APPENDIX C.

Despite some scatter in the plots, some general patterns can be observed. First, it is worth noting the big difference existing between the patterns of experiment 1 (laminar regime) and the patterns in experiments 5 and 10. In experiment 1 (Figure 70a), there seems to be some effect causing the quadrant patterns to be different between the first and second half-cycles. The reason for this effect can be found in the skewness patterns for the streamwise velocity fluctuations presented before in Figure 67. As it was mentioned then, the streamwise velocity fluctuations show negative skewness along both half-cycles in the upper part of the oscillatory boundary

layer, giving as a result that the fluctuations go with the flow for the second half-cycle, and thus the quadrant becomes Q4 from a Q3 in the first half-cycle. For the lower part of the boundary layer, the quadrant pattern is quite mixed without any general trend. In any case, it must be mentioned that being that the absolute magnitude of turbulence in this experiment was so small, the existence of turbulence interactions, sweeps and ejections, may be questionable, so the quadrant analysis in this case should be interpreted carefully.

For experiment 5 (Figure 70b), several zones can be distinguished from the analysis. In particular, a zone of outward interactions (Q1) was found in the upper part of the boundary layer (z = 10 mm to 30 mm), which seems to act as a divide between the boundary layer flow and the outer flow, essentially sending turbulence out of the boundary layer. In the middle part of the boundary layer (z = 1 mm to 10 mm) two distinct zones were identified: first, a zone of inward interactions (Q3), mainly dominant from phases 0º to 60º, during flow acceleration; second, a zone of ejections (Q2), mainly dominant from phases 90º to 180º, during flow deceleration. In between, both zones coexist during phases 60º to 90º. All these zones seem to be moving upwards as the cycle progresses. Closer to the bed and inside the viscous sublayer (z < 1 mm), zones of outward interactions (Q1) and sweeps (Q4) alternate each other without a dominant pattern.

For experiment 10 (Figure 70c), a similar quadrant pattern was observed as in experiment 5. In this case, however, the zones are more clearly delimited. In particular, the upper zone of outward interactions is still there but extends further up to about z = 60 mm. The zones of inward interactions (Q3) and ejections (Q2) in the middle part of the boundary layer are more clearly separated now, with Q3 remaining in phases 0º to 60º and Q2 taking over the zone between phases 60º and 90º and extending up to 180º. In the lower part of the boundary layer and in the viscous sublayer, zones of outward interactions (Q1) and sweeps (Q4) coexist again, however, it seems that the presence of sweeps has increased slightly with respect to experiment 5 for the later phases of the oscillation between 150º and 180º.

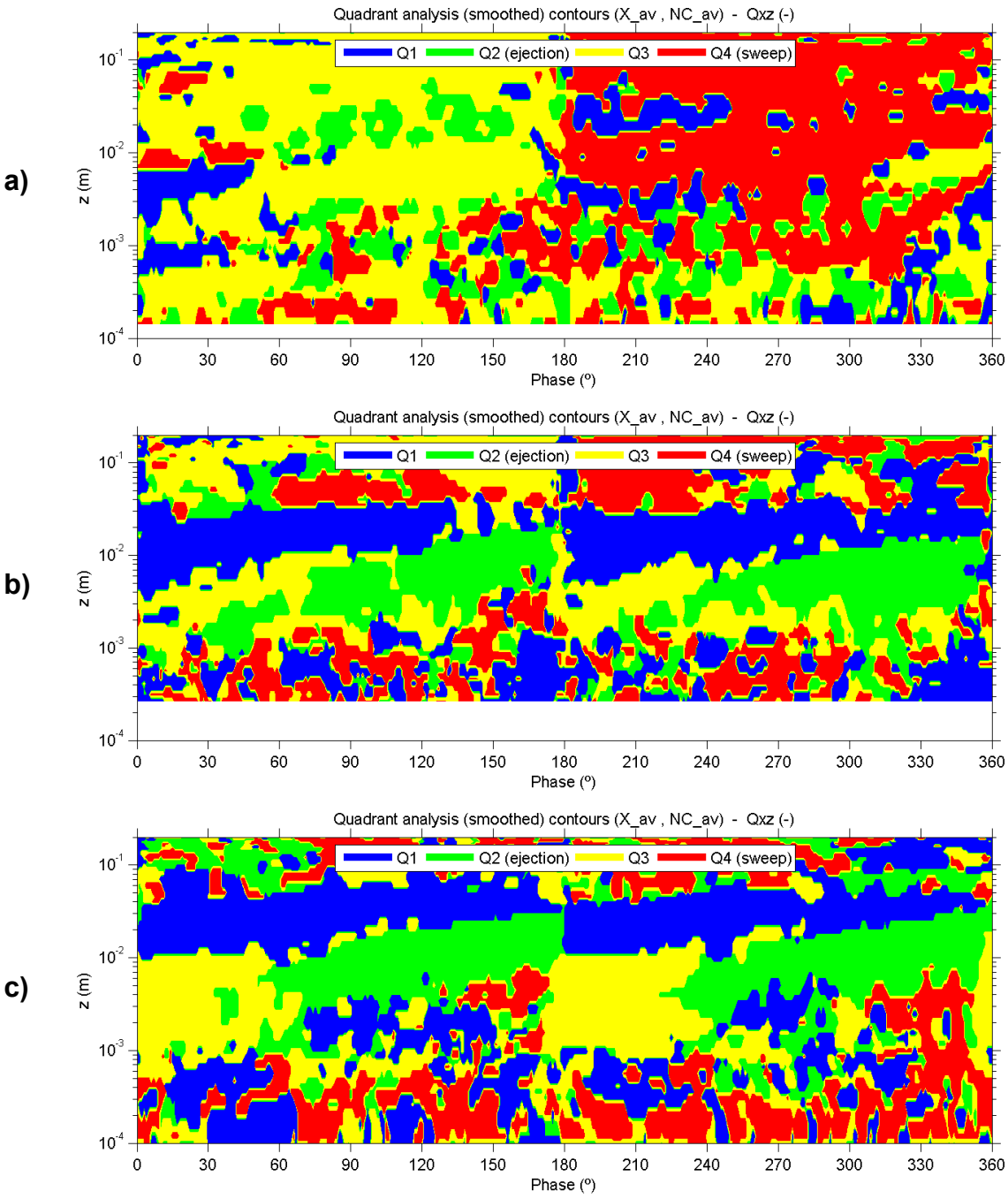


Figure 70. Quadrant analysis contours for experiments 1 (a), 5 (b) and 10 (c).

The quadrant patterns found in this work will be useful for future work related with the analysis of fluxes and transport patterns in oscillatory flows, and in particular in the context of sediment particles, sediment transport, suspended sediment concentrations and sediment fluxes.

CHAPTER 8

VISCOUS SUBLAYER ANALYSIS

8.1 Bed shear stress

Bed shear stress (τ_b) along the cycle was obtained from the total shear stress near the bottom, which includes both the viscous and turbulent components, as described in the methodology. Inside the viscous sublayer the total shear stress reaches a nearly constant value, and so a good estimate of τ_b can be calculated as the average of the total shear stress for the z locations inside the viscous sublayer. This is only possible when measurements are available very close to the bottom and inside the viscous sublayer, as it was the case in these experiments. Otherwise, it would not be possible to get τ_b from the total shear stress, and most likely τ_b values would be underestimated.

In addition, two other methods were tested for comparison: the linear fit and the logarithmic fit. Results from the linear fit method were obtained by fitting a straight line to the linear part of the velocity profiles inside the viscous sublayer, then obtaining the value of shear velocity and bed shear stress, as described in the methodology. This method gave very similar results as the total shear stress method. Indeed, the linear fit method is equivalent to calculating the viscous shear stress near the bottom. And since most of the total shear stress is due to the viscous component near the bottom, both results are expected to be very close. Results for the logarithmic fit method were obtained by fitting a logarithmic equation to the logarithmic part of the profiles (which become a straight line in a semi-log plot), then obtaining the value of shear velocity and bed shear stress, as described in the methodology. Results from this method are conditioned to the presence of a logarithmic layer in the streamwise velocity profile, which is not always the case in oscillatory flows.

Using these three methods bed shear stress plots were generated for all experiments. Only experiments 1, 5 and 10 are shown here (Figure 71), the others can be found in APPENDIX C.

From these plots it can be observed that the magnitude of the maximum bed shear stress (τ_{b_max}) increases with Re_w, as it was expected. Indeed, faster flows are expected to generate more shear on the bottom. A value of 0.11 N/m^2 was obtained for experiment 1 (laminar regime), which agrees quite well with the theoretical value of 0.12 N/m^2 predicted by the laminar solution. For experiment 5 (transition regime) this value was 0.32 N/m^2, while for experiment 10 (turbulent regime) it was 1.5 N/m^2.

Also, two important characteristics of oscillatory flows can be observed from these plots. First, due to the unsteady nature of the oscillatory boundary layer, bed shear stress is not a constant, but changes for every phase along the cycle. Also, the shape of the bed shear stress evolution along the cycle is not constant, but changes depending on Re_w, particularly during the transition regime. This way, for low Re_w in the laminar regime ($Re_w < 6x10^4$), bed shear stress typically follows a sinusoidal curve, with the maximum around phase 45º, as it is the case of experiment 1 (Figure 71a). This result agrees with the analytical solution for the laminar regime presented before in Figure 9. As Re_w increases, the shape of the curve changes and the phase of the maximum shifts closer to phase 90º. For high Re_w in the turbulent regime ($Re_w > 6x10^5$) as in experiment 10 (Figure 71c), the shape stabilizes, although is not a perfect sinusoidal, and the maximum stays around phase 85º. For intermediate values of Re_w in the transition regime ($6x10^4 < Re_w < 6x10^5$), the shape of the bed shear stress evolution along the cycle changes dramatically, even showing a double peak, one in the acceleration zone and one in the deceleration zone of each half-cycle, as it can be observed in experiment 5 (Figure 71b). In this transition regime, the maximum of the bed shear stress shifts from phase 45º near the laminar regime to phases around 120º in the middle of the transition, and then back to around 85º as Re_w gets closer to the fully turbulent regime. These observations for the bed shear stress in the transition regime constitute one of the main findings of this work and are discussed in more detail in the next section.

From the plots in Figure 71, it can also be observed that the linear fit method (green line) gives very similar results to the total shear stress method (red line) as it was expected. On the contrary, the logarithmic fit method is only able to produce good results for the high Re_w experiments, close to the fully turbulent regime, and even so, only for a few phases along the cycle.

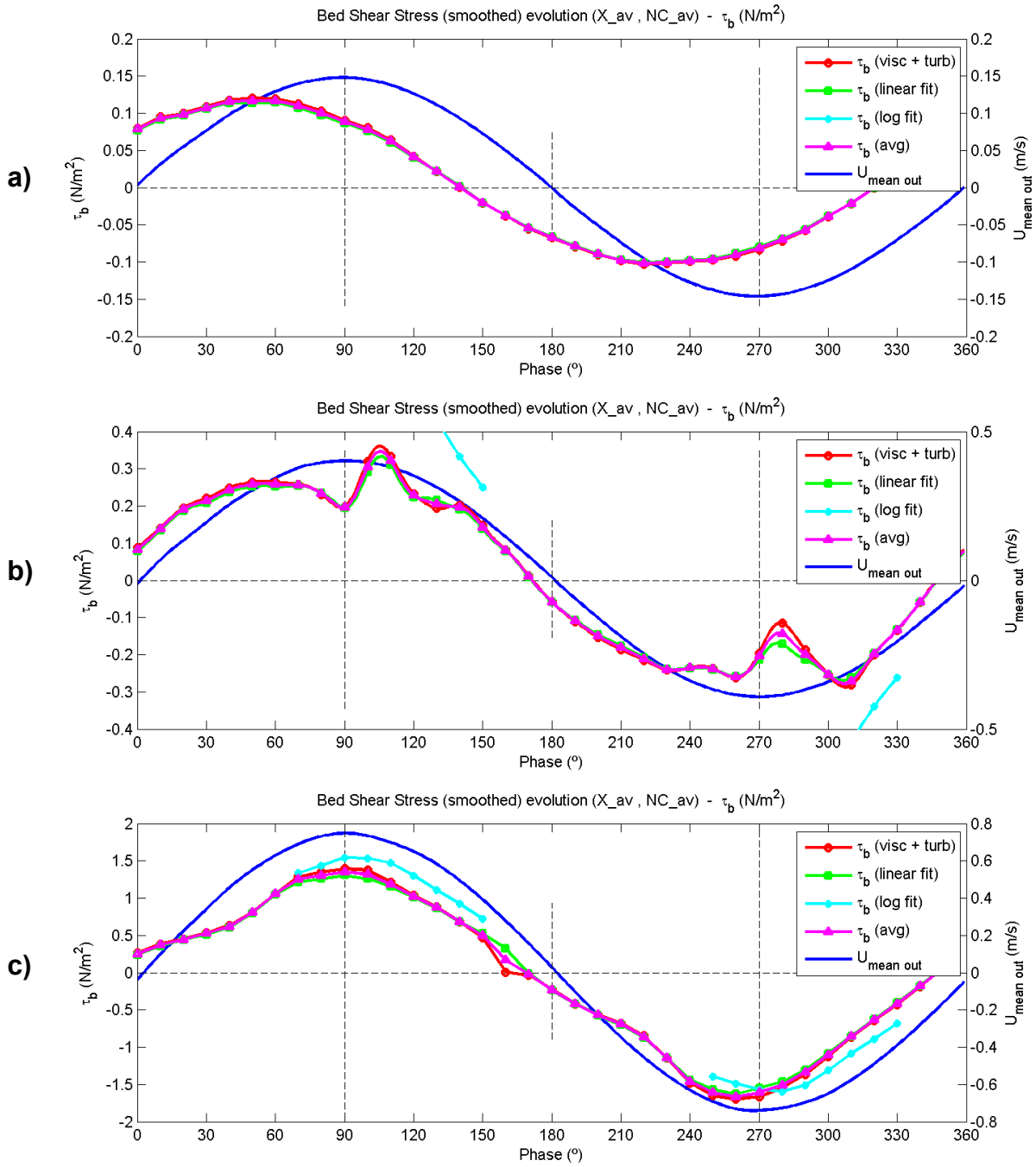


Figure 71. Bed shear stress cycle evolution for experiments 1 (a), 5 (b) and 10 (c).

Indeed, as it was shown by Sleath (1987), the velocity profile only becomes logarithmic for a few phases and vertical locations in the flow. The range of phases with a logarithmic velocity profile increases with Re_w, typically expanding to cover all the phases of decelerating flow for high Re_w, as it was also shown by Jensen (1988). For the experiments presented in this work, a logarithmic zone was first observed for experiment 4 with $Re_w = 2.3\text{x}10^5$, for a few phases

around 130º. However, the value of τ_b was overestimated when using this method for experiments in the transition regime. Only from experiment 8 (Re_w = 5.4x10^5) and beyond, the values of τ_b from the logarithmic fit gave reasonable results compared to the other two methods. For experiment 10 (Re_w = 8.7x10^5), the logarithmic zone was present for phases between 70º and 150º and τ_b values were very similar to those obtained from the other two methods. However, the phase at which the maximum occurs was not properly captured by the logarithmic fit method, as it can be observed in Figure 71c. Also, a major disadvantage of this method is that it is unable to provide information of τ_b outside of those phases where a logarithmic profile exists.

8.2 Phase difference of bed shear stress

From the bed shear stress results it was observed that the phase of the maximum bed shear stress (τ_{b_max}) experienced a dramatic change as Re_w increased. First, τ_{b_max} was ahead of the outer flow velocity maximum (U_{out_max}) for the low Re_w experiments. Then it shifted to phases later than U_{out_max} for some of the intermediate Re_w experiments. And finally, τ_{b_max} returned slightly ahead of U_{out_max} for the high Re_w experiments. Bed shear stress in this Re_w range had been reported before by other authors (Hino *et al.*, 1983; Jensen, 1988), however no explanation was provided for the existence of this effect by any of them. Furthermore, when Jensen (1988) reported the phase difference ($\Delta\phi$) between τ_{b_max} and U_{out_max} (see Figure 17), he ignored this effect and only considered positive values of $\Delta\phi$, meaning that τ_{b_max} could only be ahead of U_{out_max}. As it will be shown in this section, that plot was misleading, since $\Delta\phi$ can indeed exhibit negative values for a significant range of Re_w in the transition regime.

In order to understand this effect, the experiments in this work were specifically designed to cover the transition regime using small increments in Re_w. This allowed to capture the evolution of $\Delta\phi$ along the transition regime with great detail. To better illustrate this effect, a plot of bed shear stress evolution for one half-cycle is shown in Figure 72 for all experiments together in the same plot. This plot is similar to the one presented by Jensen (1988), included in Figure 16. Note that bed shear stress values have been normalized for each experiment to allow comparison between them. This way the focus was on the shapes of the lines and not on the specific values,

and so the phases where bed shear stress peaks occur were easier to identify. Also, an offset has been applied to each experiment in the vertical axis to be able to stack all lines together in the same plot, which helped to compare visually between experiments. In addition, lines corresponding to the experiments of Hino *et al.* (1983) and Jensen (1988) are shown for values of Re_w that were similar in their experiments. From this plot, it can be observed that the bed shear stress along the transition regime experiences two peaks of high intensity. One of the peaks was always found in the acceleration stage of the half-cycle (phases 0º to 90º) and is indicated in green, while the other was mainly found in the deceleration stage (90º to 180º) and is indicated in red. The higher of the two peaks for each experiment is indicated with a blue circle. It can be observed that the green peak was dominant for the low Re_w experiments up until $Re_w = 2.3 \times 10^5$. On the contrary, the red peak starts to appear from $Re_w = 1.5 \times 10^5$ and then became dominant for $Re_w \geq 2.9 \times 10^5$. The green peak still existed for a few more experiments and then it vanished for $Re_w \geq 5.4 \times 10^5$.

From these observations and taking into account the results obtained from the analysis of the turbulence characteristics of the oscillatory boundary layer, it is possible to look into the origin of each one of the peaks. In particular, it could be argued that the green peak was related to the laminar regime of the flow. Indeed, this was the only peak existing for the low Re_w experiments, corresponding to the laminar regime. As Re_w increased, the flow in the boundary layer transitioned to a more turbulent regime, however, for the first part of the half-cycle, the laminar behavior still prevailed even for high Re_w, which explains why the green peak can still exist at $Re_w = 5.4 \times 10^5$, which is the upper end of the transition. On the other hand, it could be argued that the red peak was related to the turbulent regime. Indeed this peak was non-existing for the low Re_w. Only when Re_w increased the red peak appeared, and it did so in the later part of the half-cycle, where turbulence perturbations start to reach the bottom. This peak became stronger as Re_w increased, and at the same time the phase shifted slowly towards the earlier part of the half-cycle. This indeed corresponds to the behavior observed in the turbulent intensities, which got stronger and closet to the bed for the earlier phases of the half-cycle as Re_w increased. Finally, for experiment 10 with Re_w in the turbulent regime, the red peak was the only peak existing and the shape of the bed shear stress evolution became more sinusoidal again.

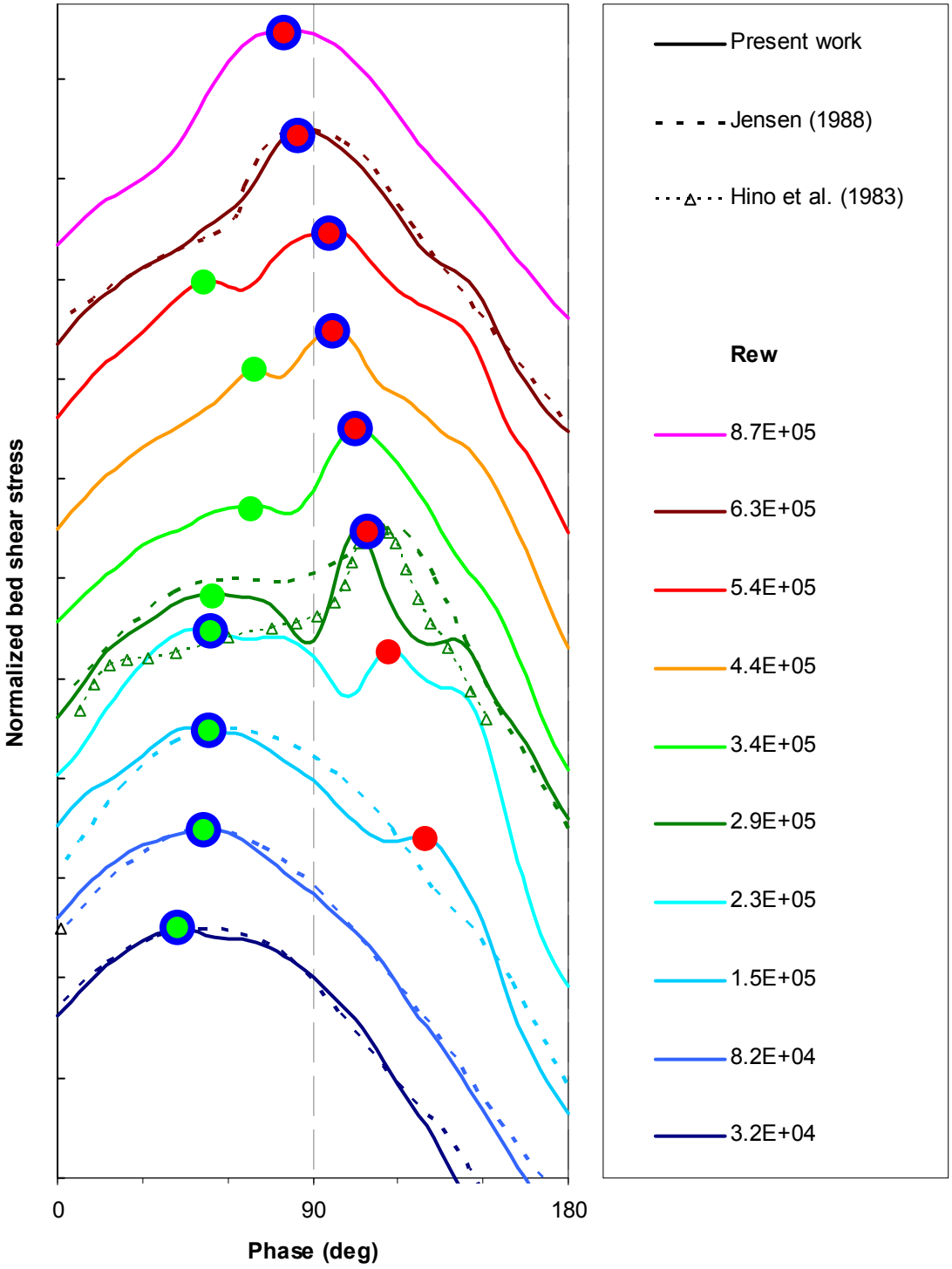


Figure 72. Normalized bed shear stress evolution during one half-cycle for experiments 1 to 10. Also, results from Hino *et al.* (1983) and Jensen (1988) for similar values of Re_w are shown for comparison.

In addition, it was observed in the mean flow analysis that the mean velocity near the bed experienced a significant increase during the deceleration stage of the half-cycle as Re_w increased. That velocity increase was visible in the "lobes" appearing in the contour plots of Figure 48a and Figure 49a, and is very much related to the increase in bed shear stress and the appearance of the red peak. Altogether, this analysis shows how turbulent effects compete with laminar effects during the transition regime, and how eventually the turbulence effects dominate close to the bottom as Re_w increases.

From the previous analysis, the phase difference ($\Delta\phi$) can be calculated between the maximum bed shear stress (which is given at the green or red peaks) and the maximum outer flow velocity (which corresponds to phase 90º) for all the experiments. Results are shown in Figure 73, with the green and red peaks having the same meaning as in Figure 72. Circled in blue are the peaks corresponding to the maximum bed shear stress. It can be observed that $\Delta\phi$ was positive (meaning there was a phase lead) for the low Re_w experiments, as it was expected. This agrees with the theory, which predicts a 45º phase lead in the laminar regime. As Re_w increased, $\Delta\phi$ reduced slowly to values around 30º. For $Re_w = 2.9 \times 10^5$, $\Delta\phi$ became negative (meaning there was a phase lag) and this was due to the red peak becoming dominant. This indicated that turbulent effects were getting stronger and eventually created higher bed shear stress than the laminar effects, as explained before. In this new situation with the red peak dominant, as Re_w increased, the phase of the peak got closer to phase 90º and this in turn made $\Delta\phi$ smaller, although still negative. Eventually, for $Re_w = 6.3 \times 10^5$, $\Delta\phi$ became positive again, and then remained at values around 5º to 10º in the turbulent regime.

A plot similar to Figure 73 was made by Jensen (1988) and is included in Figure 17. However, in Jensen's plot $\Delta\phi$ was always positive. He completely overlooked the effect of the phase lag during the transition regime, despite having one experiment in this condition. Also, the symbol corresponding to Hino *et al.* (1983) experiment was placed in the positive $\Delta\phi$ zone, when in fact corresponds also to a situation with phase lag, as it is now included in Figure 73. The plot of Jensen (1988) although very valuable at the time, can be misleading in the interpretation of the phase lead and phase lag of the maximum bed shear stress.

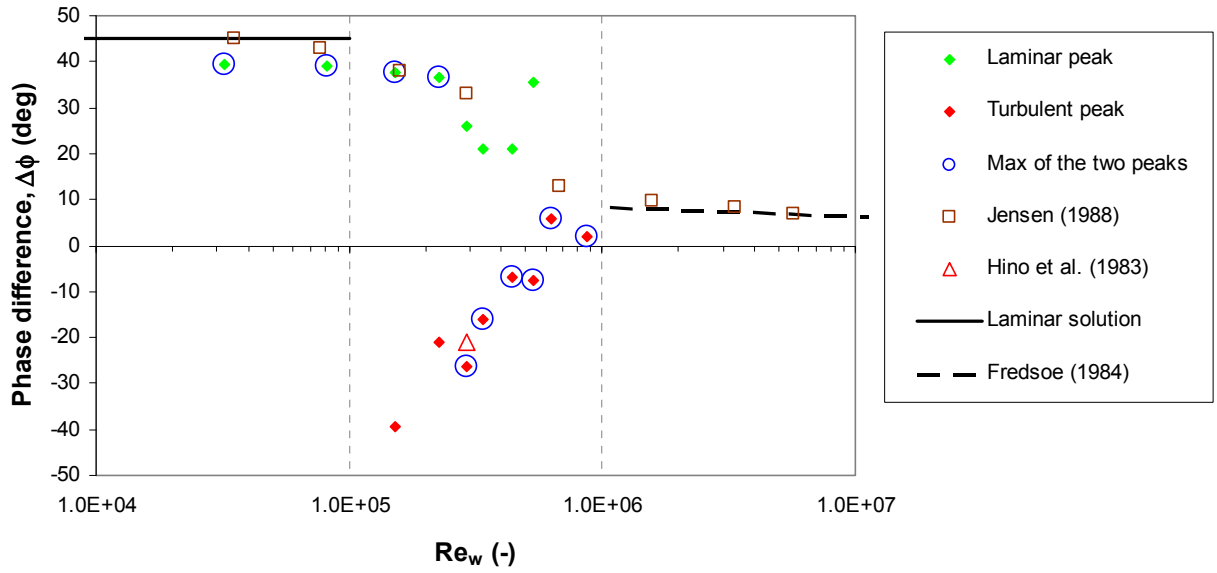


Figure 73. Phase difference ($\Delta\phi$) of bed shear stress peak (τ_{b_max}) with respect to maximum outer flow velocity (U_{out_max}).

The consequences of these observations about the behavior of $\Delta\phi$ during the transition regime could be very significant for all the processes related to sediment transport, and in particular to sediment entrainment in suspension. Indeed, the fact that the maximum bed shear stress could happen either some time before or after the maximum outer flow velocity, would significantly change the time when sediments are entrained in suspension. This in turn would affect to which part of the flow in the oscillation cycle the suspended sediments are exposed. If they were suspended at the deceleration stage of the first half-cycle, it is possible that they could be subject to the current created by the second half-cycle for longer time. This effect would transport the suspended particles in the opposite direction to the flow that put them in suspension in the first place. Furthermore, under non-symmetrical wave conditions or with symmetrical waves but in the presence of a small current superimposed, this effect could create situations where sediments could be traveling in the direction opposite to the net current in the area. This could have a dramatic effect on the estimation of sediment fluxes. The analysis of these processes, however, falls outside of the scope of this work, and further analysis is required to better understand them.

8.3 Shear Velocity

Shear velocity (U_*) along the cycle was obtained from the bed shear stress as $U_* = \sqrt{\tau_b / \rho}$. Results are presented in Figure 74 for experiments 1, 5 and 10. Plots for the other experiments can be found in APPENDIX C.

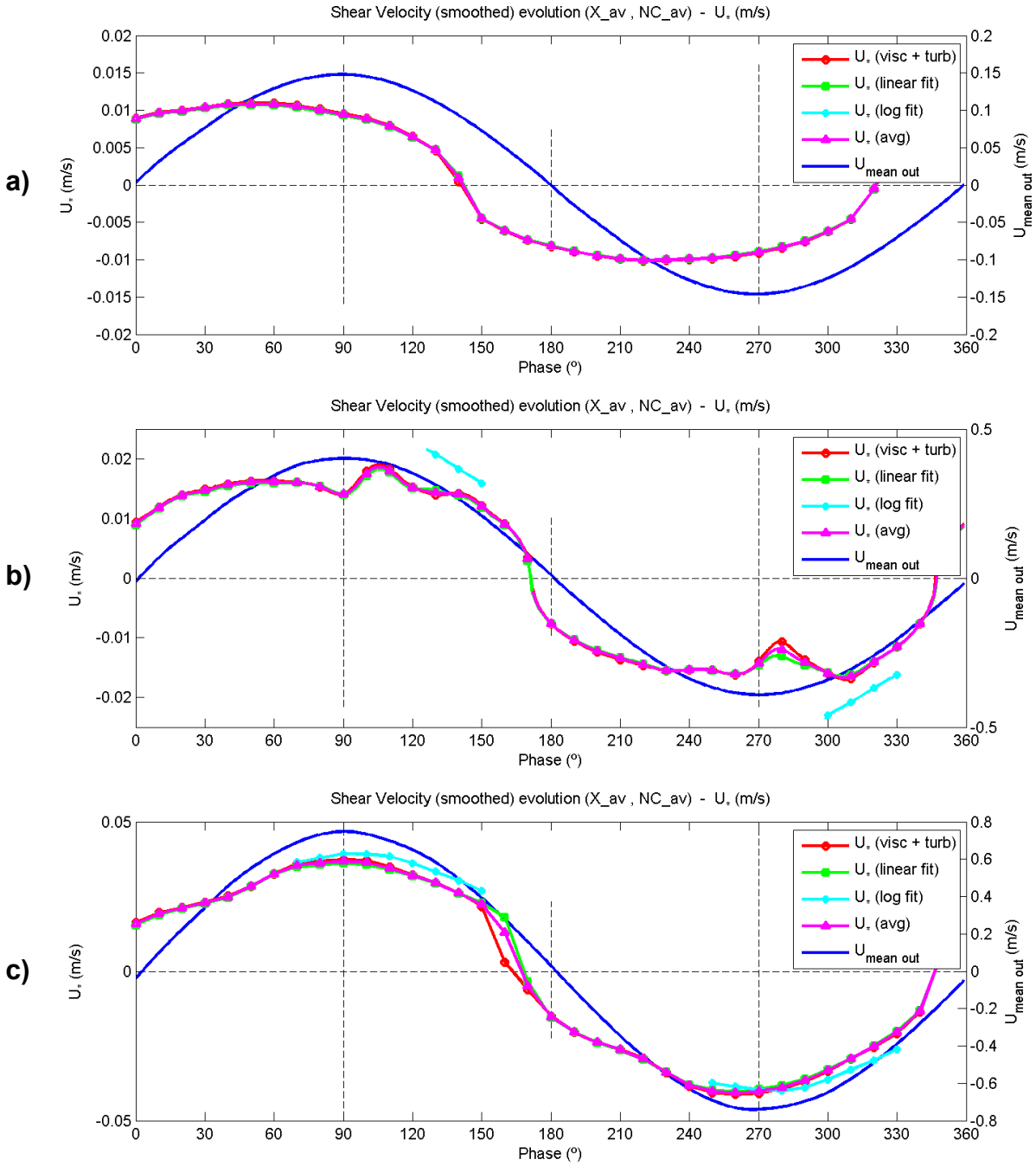


Figure 74. Shear velocity cycle evolution for experiments 1 (a), 5 (b) and 10 (c).

Shear velocity was calculated for all three methods used in τ_b: from the total shear stress at the bottom, from the linear fit of streamwise velocity profiles near the bottom, and from the logarithmic fit of streamwise velocity profiles inside the boundary layer. In fact, for the two fitting methods, U_* was obtained directly from the fit, then τ_b was calculated from U_*.

As a consequence, U_* shows the same behavior as it was described for τ_b before, both in terms of the evolution along the cycle and also the change in shape with increasing Re_w. The main difference would be the magnitude, since $U_* \sim \tau_b^{1/2}$. This relation makes the shape of U_* observed in the plots (Figure 74) look more smooth than the shape of τ_b., except near the phases of flow reversal, where the change becomes more sharp. Still, the double peak effect and the phase shift of the shear velocity maximum ($U_{*\,max}$) with increasing Re_w along the transition regime are also present, as it was the case in τ_b plots.

Despite the fact that the value of U_* changes for every phase along the cycle, it is necessary to keep in mind that the maximum ($U_{*\,max}$) is commonly used in the literature as a reference to compare between different experiments and to use in subsequent calculations, regardless of the phase in the flow where it might occur.

8.4 Wave friction factor

Wave friction factor (f_w) along the cycle was obtained from the shear velocity (U_*) and the outer flow velocity (U_{out}), using the relation given in equation (8). Results of f_w along the cycle for experiments 1, 5 and 10 are presented in the plots in Figure 75. Plots for the other experiments can be found in APPENDIX C. Included in the plots are also U_* and U_{out} for reference. In order to avoid unrealistic values of f_w due to a division by a very small number, results are only shown for the phases where $U_{out} > U_{out\_max} / 2$.

Similarly to U_*, it was expected that the value of f_w change for every phase along the cycle ($f_w = f_w(\omega t)$), since it depends on both U_* and U_{out}, and these in turn are also dependent on phase. Indeed, when looking at the plots in Figure 75, it can be clearly observed how the friction factor

for oscillatory flows changes along the cycle. This fact, although already mentioned in the literature (Jensen, 1988), it had never been shown before, and constitutes an important contribution from this work.

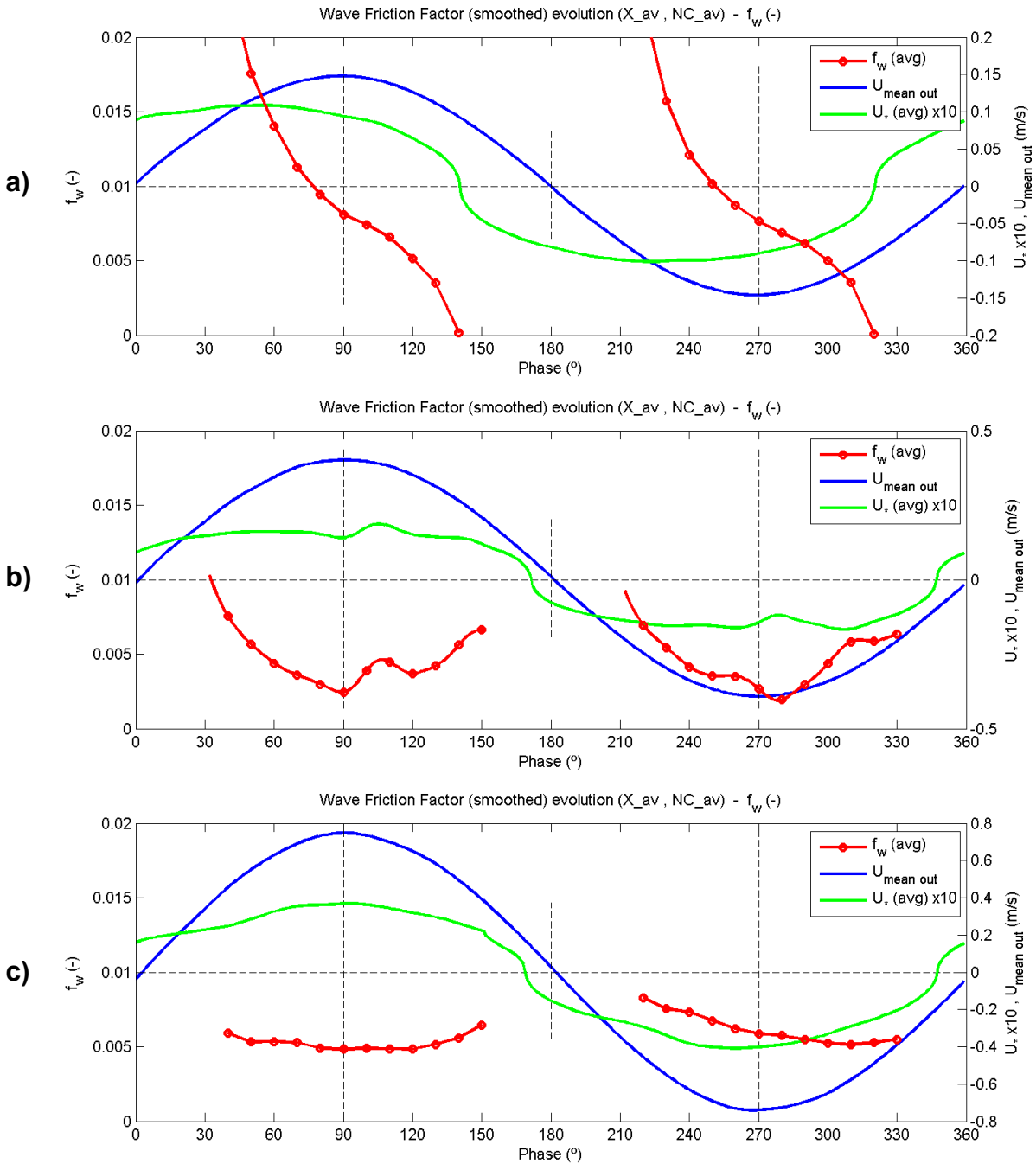


Figure 75. Wave friction factor cycle evolution for experiments 1 (a), 5 (b) and 10 (c).

From the plots, it can also be observed how the relation between U_* and U_{out} at different phases influences f_w to become larger or smaller depending on the phase. In particular, for experiment 1 (Figure 75a), the difference in phase between the curves for U_* and U_{out}, with U_* being around 45º ahead of U_{out}, makes f_w to show a great deal of variability along the cycle, from very high values near phase 45º to zero near phase 140º. For experiment 5 (Figure 75b), with U_* being now more flat, and closer in phase with U_{out}, the shape of f_w becomes more sinusoidal, following the influence of U_{out}. A few bumps are observed due to the peaks experienced by U_*. As Re_w increases, the shape of f_w becomes more flat. This can be observed in experiment 10 (Figure 75c), where both U_* and U_{out} show a similar sinusoidal shape, nearly in phase, making the value of f_w almost constant for all phases along the cycle.

It must be mentioned that it is common practice to find in the literature that the wave friction factor be computed using the maximum value of the shear velocity ($U_{*\,max}$) and the maximum value of the outer flow velocity (U_{out_max}), regardless of the phase where they happen. In this work, the value obtained this way will be named the "reference wave friction factor" (f_{w_ref}). As shown from the results in this work, that method could be valid to characterize the flow for high Re_w, in the fully turbulent regime, since f_w is almost constant along the cycle. However, in the laminar or transition regimes, with lower Re_w, the phases of $U_{*\,max}$ and U_{ou_maxt} can be very different, and so reporting the wave friction factor using that method could be misleading.

A plot showing f_{w_ref} evolution with Re_w along the transition regime with smooth wall is presented in Figure 76, including the results obtained in this work. This plot is similar to the diagrams presented before in Figure 19 and Figure 20 from Kamphuis (1975) and Jensen (1988), respectively. For comparison purposes the experimental results from Kamphuis (1975), Hino *et al.* (1983) and Jensen (1988), as well as the DNS results from Spalart and Baldwin (1987) are also included. In addition, analytical solutions are delineated for the laminar regime (obtained from the equations) and for the turbulent regime (obtained from Fredsoe (1984) approximation). As it can be observed from this plot, f_{w_ref} follows a straight line (in log-log scale) in the laminar and turbulent regimes. However, there is no clear trend in the transition regime and consequently some dispersion was observed when comparing the results from the literature. The results of this work compared reasonably well with previous studies, although it seemed experiments 5, 7 and 8

could be underestimating the value of f_{w_ref}. This could be due to the fact that in those experiments the first measurement location close to the bottom was at z = 0.26 mm, z = 0.24 mm and z = 0.23 mm respectively, compared to an average of z = 0.13 mm for the other experiments. Having the first measurement location higher could have caused underestimation of the bed shear stresses and U_* in those experiments and, consequently, f_{w_ref} would have been affected.

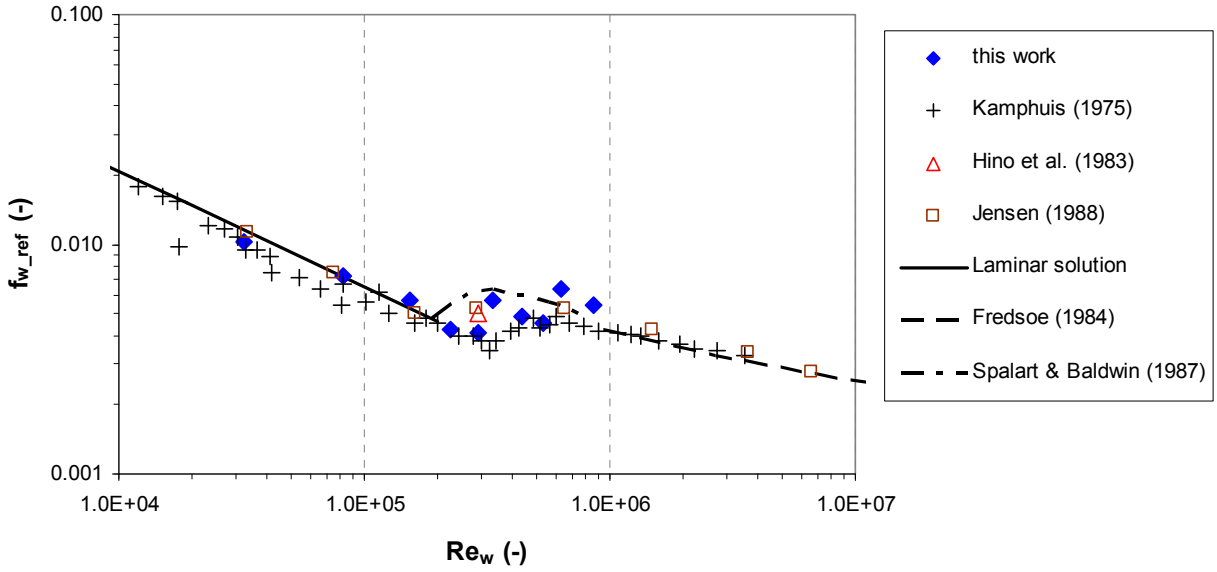


Figure 76. Reference wave friction factor (f_{w_ref}) evolution with Re_w for smooth wall. Solid symbols correspond to the experiments in this work.

8.5 Viscous sublayer thickness

The viscous sublayer is defined as the zone inside the boundary layer where viscous effects are important compared to turbulent effects. This sublayer is typically very thin for turbulent water flows (less than 1 mm) and is found close to the wall, where velocities and turbulence fluctuations are small. For steady flows, the divide between the viscous sublayer and the rest of the boundary layer on top is typically marked by the threshold between the linear part of the velocity profile near the wall and the logarithmic part of the velocity profile far from the wall. This threshold is known to be proportional to the viscous length scale ($z_v = \nu/u_*$), which relates the viscosity of the fluid (ν) and the shear velocity of the flow (u_*), and it can be calculated as $\delta_v = 11.6 \cdot \nu / u_*$. However, only the zone where the velocity profile closely follows a linear

relation is considered as the strictly viscous zone, which is found at $\delta_v = 5 \cdot \nu / u_*$ (Schlichting, 1979; Nezu and Rodi, 1986).

For unsteady flows, and in particular for oscillatory flows, the thickness of the viscous sublayer (δ_v) is expected to change along the cycle, since the shear velocity (U_*) also changes along the cycle. The evolution of δ_v along the cycle is also expected to be affected by the acceleration and deceleration of the flow and also by the flow reversal, since these conditions create significant changes in the vertical distribution of turbulence, and in particular near the bottom.

For the experiments in this work, the thickness of the viscous sublayer (δ_v) was difficult to determine. When the classic relation for steady flows was used, it was found that it didn't work well for certain phases along the cycle, and also depending on Re_w. For instance, in the high Re_w experiments, zones of high turbulence were found below the threshold and inside the viscous sublayer for several phases along the cycle, while in the low Re_w experiments, zones with high viscous stresses and very low turbulence where left outside above the threshold. This issue has not been previously reported in the literature. In fact, very little experimental evidence was found about the viscous sublayer in oscillatory flows over a smooth wall. Only the paper by Hino *et al.* (1983) shows the thickness of the viscous sublayer for their experiment in the transition regime, however they didn't mention how it was calculated. For the laminar regime, an analytical solution was provided by Rayleigh (1911), who investigated deeper into the Stokes second problem, and found that δ_v is proportional to $(\nu t)^{1/2}$. No reference was found in this regard for the turbulent regime.

As a consequence, a new method was developed as part of this work to obtain a reasonable estimation of the thickness of the viscous sublayer (δ_v) along the cycle. The method takes a step back from the steady flow relation and goes back to the very definition of the viscous sublayer, trying to find the zone of the flow near the wall where viscous effects are significant compared to turbulent effects. In order to do this, a ratio of viscous forces to turbulent forces was devised in the following way:

$$ratio\_VT = \frac{viscous\ forces}{turbulent\ forces} = \frac{\tau_v}{e} \tag{52}$$

where the viscous shear stress (τ_v) was used as the characteristic viscous force and the turbulent kinetic energy ($e = \rho * TKE$) as the characteristic turbulent force. Note that both variables have the same units (N/m^2), thus *ratio_VT* is a dimensionless variable. Due to limitations in the available data near the bottom, only the streamwise contribution to the turbulent kinetic energy is considered for this analysis. This way, equation (52) can be rewritten as follows:

$$ratio\_VT = \frac{\tau_v}{e_u} = \frac{\tau_v}{\frac{1}{2}\rho U_{RMS}^2} \tag{53}$$

The question remains as to find a suitable value of *ratio_VT* that defines the threshold of the viscous sublayer. In this regard a simplification can be made to equation (53) in order to obtain a working relation near the bottom. In particular, near the bottom, the viscous shear stress can be approximated by $\tau_v \approx \rho (U_*)^2$, resulting in:

$$\frac{\tau_v}{\frac{1}{2}\rho U_{RMS}^2} \approx \frac{\rho U_*^2}{\frac{1}{2}\rho U_{RMS}^2} = 2\left(\frac{U_*}{U_{RMS}}\right)^2 \tag{54}$$

Also, it is worth looking at the equivalent situation for the case of steady flows. As mentioned before, it is commonly accepted for steady flows that the limit of the purely viscous zone is found at $z^+ = 5$. Looking at experimental data reported in the literature (Nezu and Nakagawa, 1993), it is found that at that distance the relation between RMS velocity and shear velocity is $u_{RMS} / u_* \approx 1.8$. This relation gives an idea of the level of turbulence in relation to the viscous forces that is commonly accepted to be found in the limit of the viscous sublayer. Indeed, it indicates that a great deal of turbulence is still happening at the limit of $z^+ = 5$, and even though viscous forces are dominant, turbulence forces are still present as well in the viscous sublayer. Now, we can use the same criterion for the case of oscillatory flows, and apply it to equation (54). This way, a value of *ratio_VT* = 0.62 is found as the threshold for the viscous sublayer in

oscillatory flows. In fact, this threshold should hold valid for both steady and unsteady turbulent flows, since it is a general criterion based on the intrinsic definition of the viscous sublayer, without any particularization regarding the type of flow.

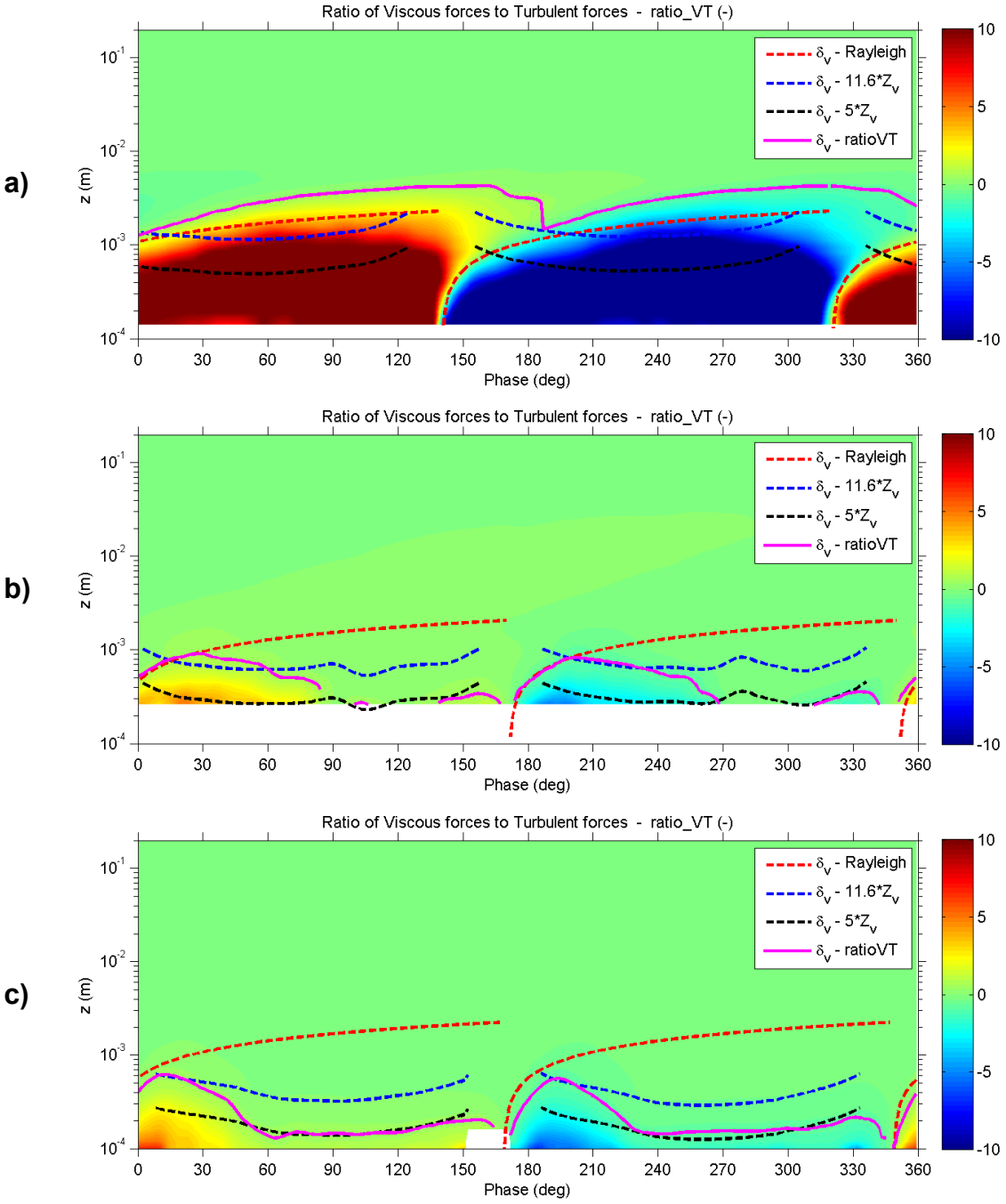


Figure 77. Contours plots showing the ratio of viscous forces to turbulent forces for experiments 1 (a), 5 (b) and 10 (c). Also, lines of viscous sublayer thickness obtained from different methods are superimposed.

Contour plots of *ratio_VT* were created based on the definition given before and are shown in Figure 77 for experiments 1, 5 and 10. The change in sign just means the different flow direction along the cycle. Also, superimposed lines of viscous sublayer thickness (δ_v) are shown, calculated using the different methods explained before: Rayleigh relation for laminar regime, steady flow criterion for $z^+ = 11.6$ (general threshold of viscous sublayer) and $z^+ = 5$ (purely viscous zone) and the proposed method using *ratio_VT* = 0.62. For experiment 1 (Figure 77a), it can be observed that viscous forces clearly dominate up to a distance of 4 mm from the bed. Also, these viscous forces are present for the entire duration of each half-cycle. This indicates the viscous sublayer is rather thick for all phases of the oscillation in the laminar regime. Indeed, in this regime almost the entire boundary layer is expected to show a strong viscous influence due to the low amount of turbulence present. From the theory, the viscous sublayer in this regime should grow continuously, starting at the phase of flow reversal near the bottom. It can be observed in the plot that the Rayleigh line captures this behavior quite well, as it was expected, since it represents the analytical solution specific for the laminar regime. On the contrary, the lines of $z^+ = 5$ and $z^+ = 11.6$ completely fail to capture the behavior in this regime. This can be explained because these lines were conceived for developed turbulent flow, and so they struggle to predict δ_v in laminar conditions. They work well when the streamwise velocity profile is fully developed and shows distinct linear and logarithmic regions. Finally, the line obtained using the *ratio_VT* method shows a similar shape as the Rayleigh line, and the magnitude is larger by a factor of about 2. This is not necessarily wrong since the viscous sublayer in this laminar regime is expected to extend to almost the entire boundary layer, which for this experiment was about 10 mm high. This method seems to be able to provide a good estimate of the overall shape of the viscous sublayer in the laminar regime.

As Re_w increases, the behavior of the viscous sublayer becomes more complex, since the flow can experience both laminar and turbulent conditions depending on the phase along the cycle. This can be observed in experiment 5 (Figure 77b), where viscous forces are dominant near the bottom for the acceleration stage (phases 0º to 90º) of the half-cycle, but then turbulence becomes stronger near the bottom for the deceleration stage (phases 90º to 180º). Under those conditions, the viscous sublayer is expected to behave in a laminar way at the beginning of the

half-cycle, then transitioning towards a turbulent behavior. This can be observed in the plot, where *ratio_VT* is high for phases -10º to 30º near the bottom in a way that suggests a laminar behavior. In this zone, the Rayleigh line would be applicable to predict δ_v. As the oscillation progresses, there is a transition, where viscous forces decrease from about phases 30º to 80º, then turbulent forces dominate afterwards from phases 80º to 170º. In this region dominated by turbulence, it is expected that the turbulent lines from steady flow would be able to predict δ_v. As a result, the thickness of the viscous sublayer first grows, then slowly decreases, and eventually becomes nearly constant in the turbulent zone. Now, looking at the line obtained from the *ratio_VT* method, it can be observed how it predicts quite well this behavior, following the Rayleigh line at the beginning of the half-cycle, then it decreases along the transition until it meets the turbulent line corresponding to $z^+ = 5$ for the deceleration stage of the half-cycle.

Looking at experiment 10 (Figure 77c), the same behavior is observed as in experiment 5, with a laminar zone close to the bed at the beginning of the half-cycle, then a transition and then a zone where turbulence dominates closer to the bed. The main difference in this case is that the turbulent zone close to the bed has expanded, now starting earlier in the half-cycle, covering phases from about 50º to 170º. As a consequence, the transition has shortened, now covering phases from 15º to 50º, and also the laminar zone has shortened, now covering phases from -10º to 15º. It can be observed how the thickness of the viscous sublayer (δ_v) calculated from the *ratio_VT* method was able to predict this behavior quite well, following the Rayleigh line at the beginning of the half-cycle, then converging towards the turbulent line of $z^+ = 5$.

Using the *ratio_VT* method described before, viscous sublayer thickness (δ_v) was calculated for all the experiments. This method was mainly necessary for the experiments in the turbulent regime, however, it also applies to the transition and laminar regimes as observed before. Results of δ_v along the cycle for experiments 1, 5 and 10 are presented in the plots in Figure 78. Plots for the other experiments can be found in APPENDIX C. Outer flow velocity (U_{out}) is also included in these plots for reference.

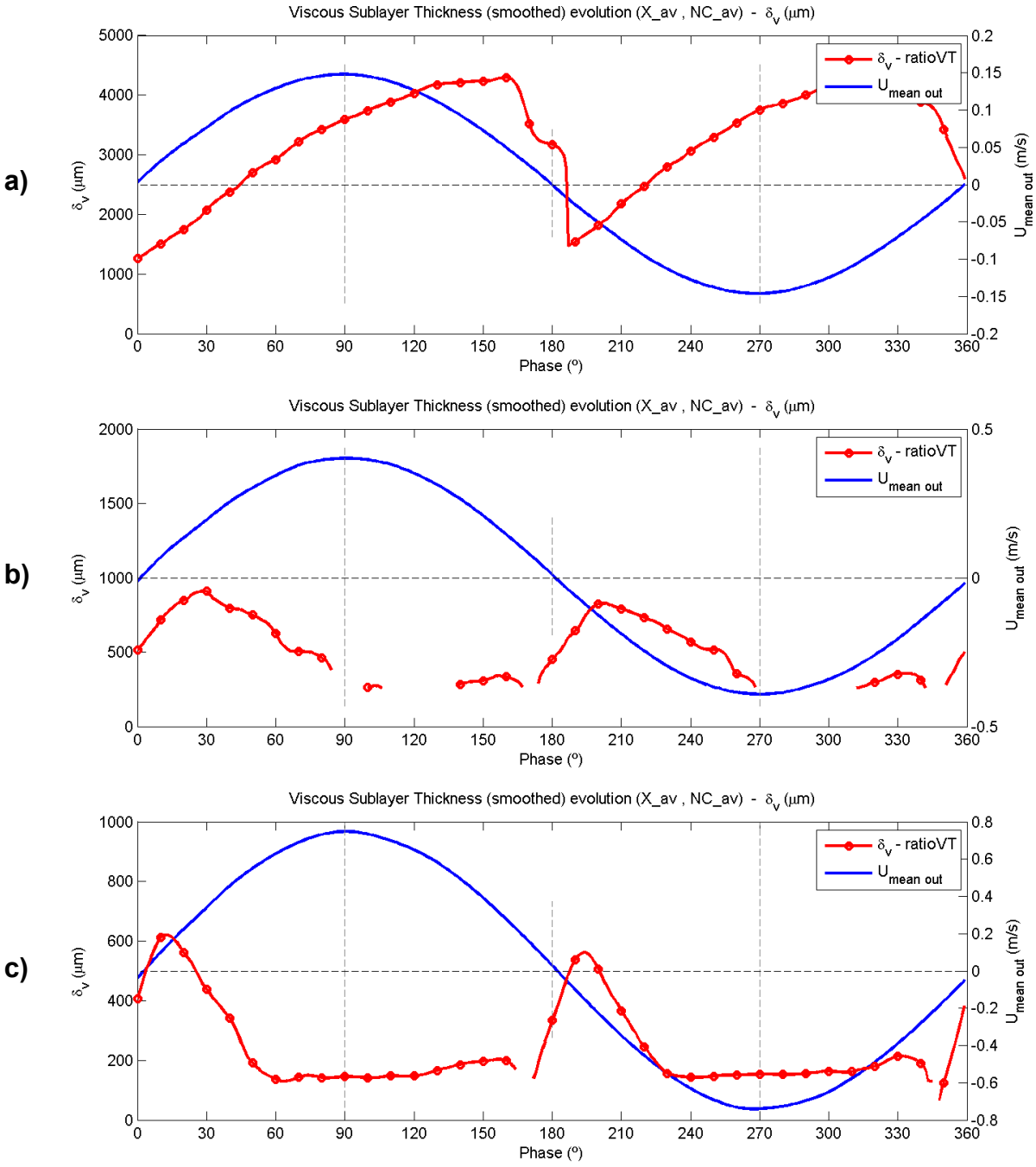


Figure 78. Viscous sublayer thickness cycle evolution for experiments 1 (a), 5 (b) and 10 (c).

In the plots in Figure 78, it can be observed how the viscous sublayer thickness is smaller as Re_w increases. In particular, for the laminar regime experiment (Figure 78a) δ_v grows along the entire half-cycle, reaching a value of around 4 mm. On the contrary, for the higher Re_w experiments (Figure 78b,c) the shape of δ_v changes along the half-cycle, showing the different regimes

happening near the bed. First the flow starts laminar, and δ_v grows, reaching values of around 0.9 mm for experiment 5 and 0.6 mm for experiment 10. Then the flow experiences a transition when turbulence effects start to be noticeable, which makes δ_v decrease. And for the final part of the half-cycle where turbulence dominates near the bed, δ_v becomes nearly constant with a value similar to that of $z^+ = 5$ as shown before, which for experiment 5 is around 0.3 mm and for experiment 10 is about 0.15 mm.

8.6 Summary of viscous sublayer parameters

In Table 3 a summary of the calculated viscous sublayer parameters is presented for all 10 experiments. These include maximum shear velocity ($U_{*\,max}$), maximum bed shear stress (τ_{b_max}), phase difference ($\Delta\phi$) of maximum bed shear stress with respect maximum outer flow velocity, reference wave friction factor (f_{w_ref}) calculated with the maximum shear velocity and the maximum outer flow velocity, reference viscous length scale ($Z_{v\_min} = \nu / U_{*\,max}$) and viscous sublayer thickness at phase 90º (δ_{v_90}). For completeness, boundary layer thickness at phase 90º (δ_{90}) and maximum boundary layer thickness (δ_{top}) are also included. Negative values of $\Delta\phi$ indicate that τ_{b_max} happens after the outer flow velocity maximum (phase 90º typically). For experiments 5, 7 and 8 it was not possible to calculate δ_{v_90} because it was smaller than the first data point in the profile. Instead, an approximate value of $z^+ = 5$ is provided.

Exp no.	$U_{*\,max}$ (m/s)	τ_{b_max} (N/m^2)	$\Delta\phi$ (deg)	f_{w_ref} (-)	Z_{v_min} (mm)	δ_{v_90} (mm)	δ_{90} (mm)	δ_{top} (mm)
1	0.011	0.11	39.5	0.0102	0.102	3.672	4.7	9
2	0.014	0.21	39.0	0.0073	0.077	2.655	4.5	12
3	0.016	0.26	37.5	0.0057	0.059	0.693	7.3	19
4	0.017	0.28	36.5	0.0043	0.056	0.326	8.6	22
5	0.018	0.32	-26.5	0.0041	0.048	0.238	9.2	25
6	0.023	0.52	-16.0	0.0057	0.038	0.172	9.8	30
7	0.024	0.58	-7.0	0.0048	0.036	0.181	11.7	35
8	0.026	0.70	-7.5	0.0045	0.035	0.173	13.4	40
9	0.037	1.34	6.0	0.0063	0.029	0.169	17.0	45
10	0.039	1.51	2.0	0.0054	0.026	0.150	17.7	45

Table 3. Viscous sublayer parameters calculated for each experiment.

CHAPTER 9

UNCERTAINTY ANALYSIS

Uncertainty in experimental measurements comes from different sources, the most typical being instrument accuracy, instrument bias, positioning accuracy, facility variability, human error and sampling error. Each one of these sources of uncertainty was taken into account during this work. In particular, sampling error was analyzed in detail.

9.1 Instrument errors

Regarding instrument accuracy, LDV systems are one of the most accurate flow measurement devices nowadays, and the error associated is typically very small, in the order of 1 % (Albrecht *et al.*, 2003). Instrument bias can be important in unsteady flow measurements with LDV due to varying data rates depending on the velocity of the flow (Adrian and Yao, 1987). In order to avoid that bias in this work, an even-time sampling routine was developed and applied during data processing and it is described in the methodology in more detail.

9.2 Positioning errors

Positioning accuracy during the measurements was influenced by two effects: first, the position of the measurement volume at the beginning of each profile, which was always referenced to the PVC floor. This initial positioning had an uncertainty of ± 0.05 mm in z, and later it was further improved during data processing by adjusting the $z = 0$ level using the near-bed velocity profiles. The second effect related to position accuracy was the displacement of the measurement volume inside the tunnel. In this regard, the traverse system was capable of displacements as small as 0.01 mm in all 3 directions, providing very good accuracy for all locations in the profile.

9.3 Facility variability

The LOWST facility is equipped with an electro-mechanic system by MTS to control the movement of the pistons and generate the flow. The timing of this system was very accurate (in the order of 0.01 s), which ensured proper repetition of the oscillatory flow cycles. In addition, a trigger system was used to synchronize the LDV measurements with the motion of the pistons. Furthermore, the phase of the oscillation was fine-tuned during data processing, achieving an accuracy of ± 1º.

9.4 Sampling error

Sampling error is associated with the duration of a time series collected at a particular location and, in particular, with the number of data points (N) collected. When N is too small, the uncertainty of flow statistics increases, regardless of the instrument. Typically $N > 1000$ independent samples are enough to compute high-order turbulence statistics at a particular location with small sampling errors.

In the oscillatory flow experiments performed in this work, measurements were collected at each location for a certain number of cycles. During data processing, the time series were divided in cycles and all the cycles were ensembled to obtain mean flow and turbulence statistics. Because of this ensemble-averaging process, to compute the flow statistics for a particular phase at each location there was only one data point available per cycle. This is a major inconvenience when measuring unsteady flows and made for very time-intensive experiments. In addition, at certain locations (such as close to boundaries and in areas of very low velocities) it might be more difficult to obtain good data, and information may not be available for some of the cycles. In order to account for this eventuality, at those locations it was necessary to measure for longer time in order to collect some extra data.

Once the data was collected, the standard error (*S.E.*) associated with the mean value of a particular flow statistic ($\overline{X}$) was calculated according to the following equation:

$$S.E.(\overline{X}) = \frac{\sigma_X}{\sqrt{N}} \tag{55}$$

where σ_X is the standard deviation of the variable X and N is the number of data points available. Expressions for the standard deviation of common turbulence statistics were obtained from Benedict and Gould (1996). Percent error (*P.E.*) was then calculated as:

$$P.E.(\overline{X}) = \frac{S.E.(\overline{X})}{\overline{X}} = \frac{\sigma_X}{\overline{X}\sqrt{N}} \tag{56}$$

9.4.1 Number of valid data points

During the experiments, the goal was to measure for 130 cycles at the locations close to the bottom were turbulence was more intense in order to obtain better quality statistics. Far from the bottom, outside of the boundary layer, measurements were collected for 65 cycles, since turbulence levels were expected to be lower. This strategy was a good compromise to help reduce the total duration of each experiment by a significant amount of time with a small increase in the uncertainty for the upper locations. Despite the efforts, the number of available data points for certain phases of the oscillation was lower than the goal, especially close to the bottom and near the phases of flow reversal because flow velocities were very low.

Contour plots showing the number of valid data points for all vertical locations (z) and phases were generated for all experiments. Results are presented in the following figures for experiments 1, 5 and 10. The other experiments can be found in APPENDIX C.

Figure 79 corresponds to the streamwise velocity component (u), Figure 80 to the spanwise component (v) and Figure 81 to the vertical component (w). The red color corresponds to zones with the highest data points (up to 130), the green color to zones with intermediate number of points (around 65), and the blue colors to zones with low number of points (around 20 to 30). The blue zones are mainly found near the bottom and the phases of flow reversal, which were the most complicated zones to collect measurements from and consequently the drop-out rate was the highest. In Figure 80, for the spanwise component (v) a large zone of dark blue color can be

observed below z = 3 ~ 5 mm depending on the experiment, indicating no valid data points for those locations. From all three velocity components, the vertical one (w) was the one with the best data, since it exhibited the lowest drop-out rate.

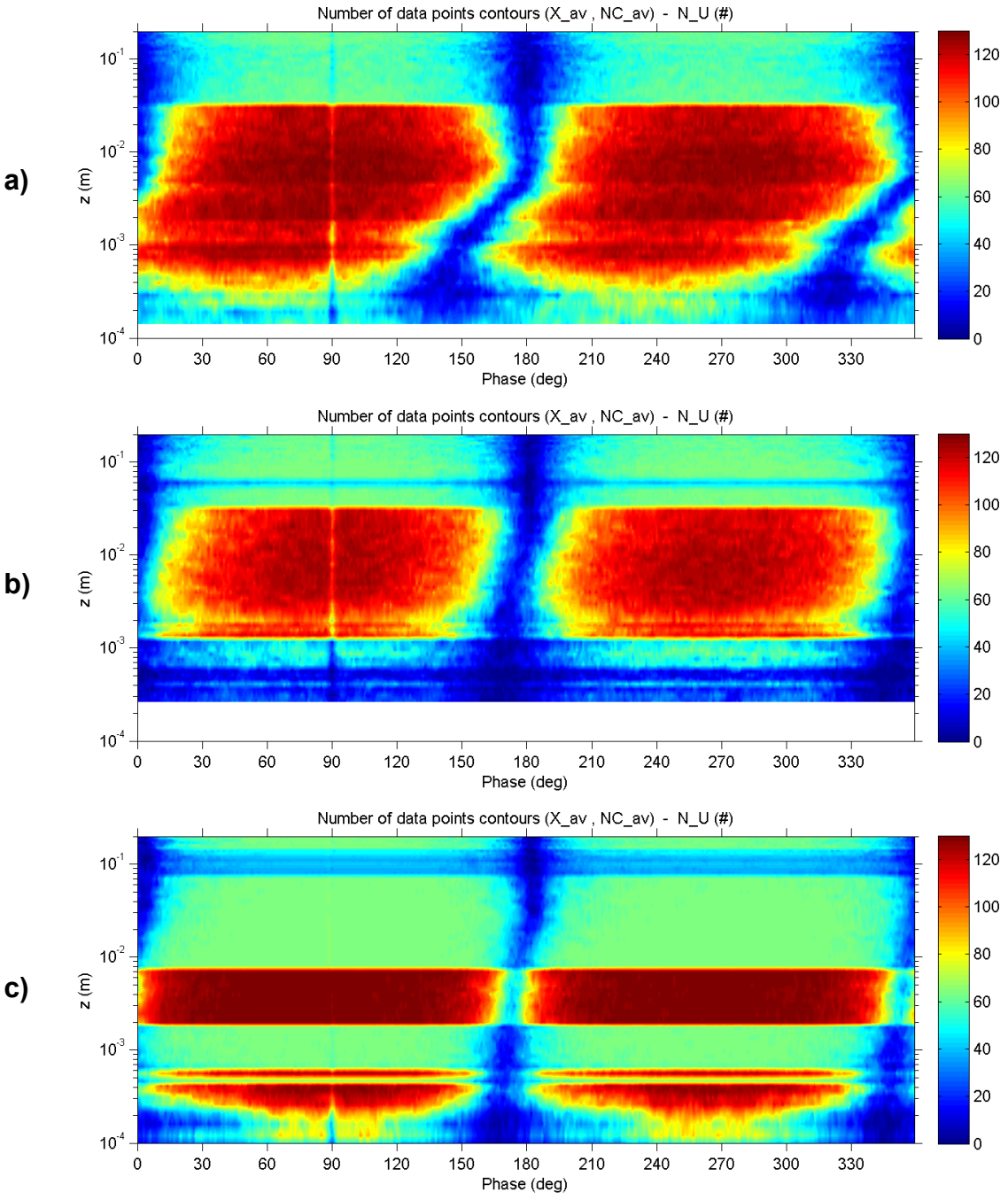


Figure 79. Contour plots showing the number of data points available in the streamwise component (u) for experiments 1 (a), 5 (b) and 10 (c).

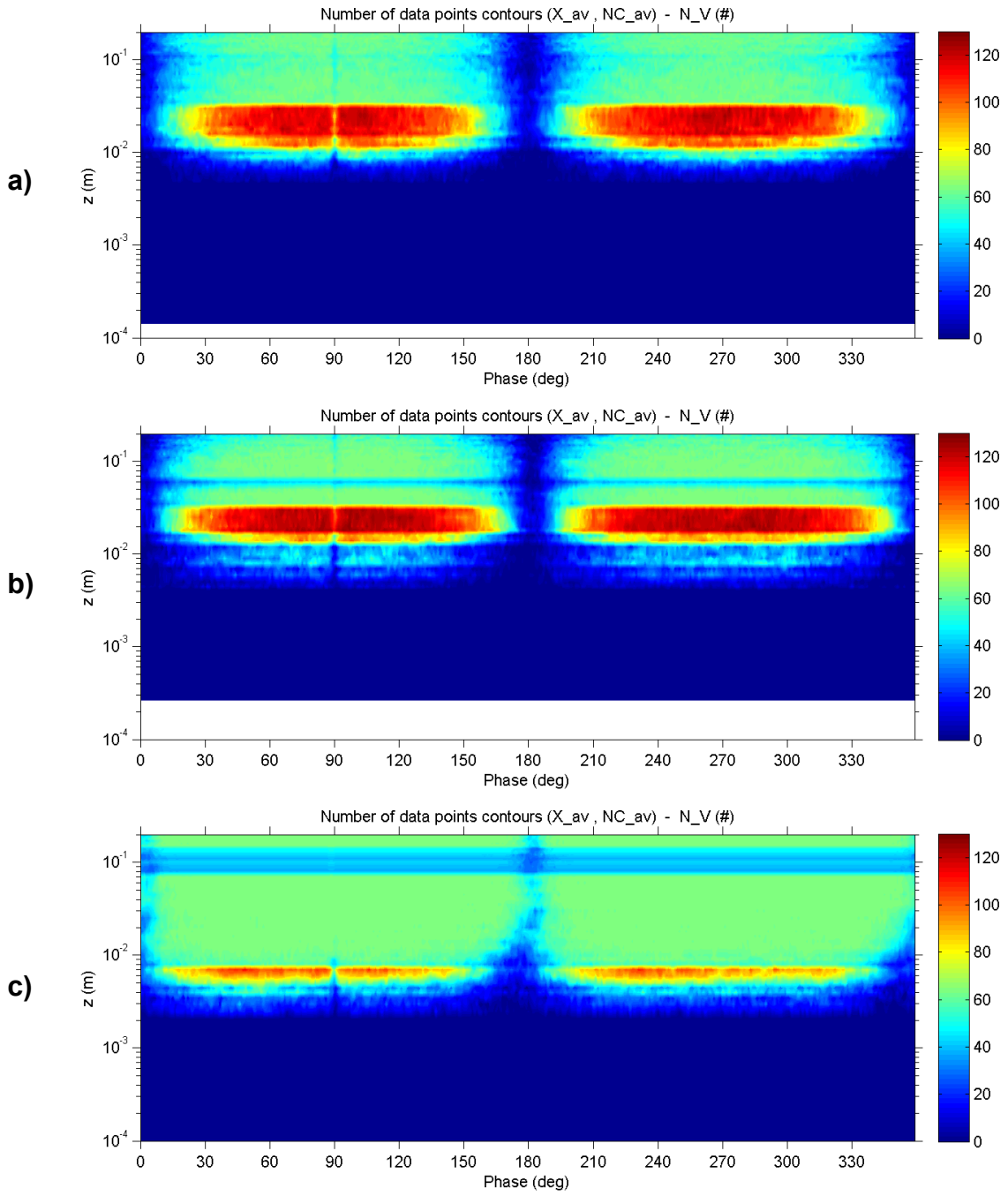


Figure 80. Contour plots showing the number of data points available in the spanwise component (*v*) for experiments 1 (a), 5 (b) and 10 (c).

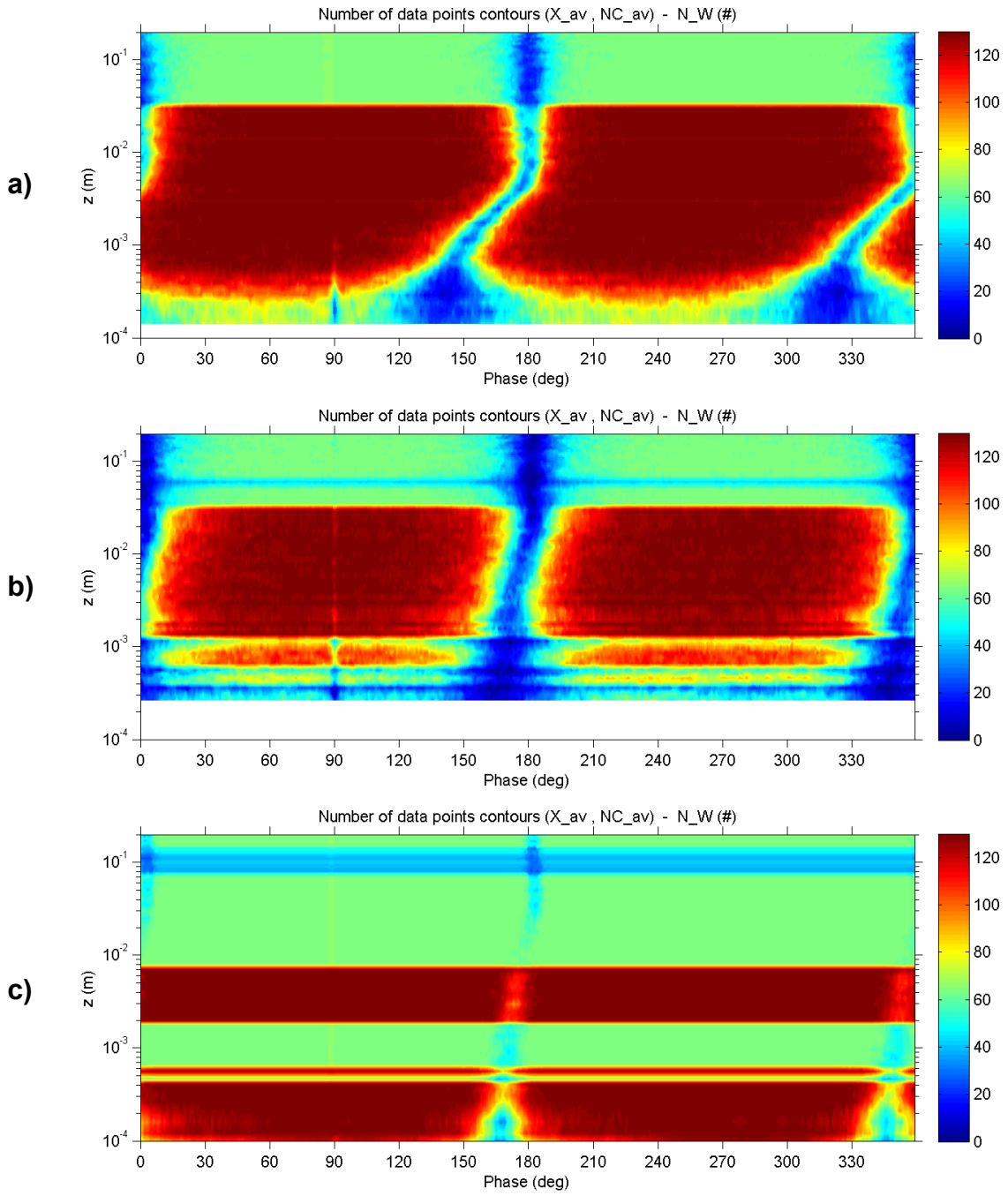


Figure 81. Contour plots showing the number of data points available in the vertical component (w) for experiments 1 (a), 5 (b) and 10 (c).

9.4.2 Percent error of mean velocity

Percent error of mean velocity for the streamwise component (U) was calculated using equation (56) and the expression for the standard deviation given by Benedict and Gould (1996):

$$\sigma_X = \sqrt{\langle u'^2 \rangle} \tag{57}$$

Contour plots of percent error of U were generated for all experiments. Results are presented in Figure 82 for experiments 1, 5 and 10. The other experiments can be found in APPENDIX C.

It can be observed that the error for the mean velocity was very small, with values below 1 % for most locations and phases of the oscillation (zones of dark blue color), increasing up to 5 % near the bottom. Only for the phases near flow reversal the error was higher, with values around 20 % and up to 40 % for some locations. This was mainly due to mean velocities getting close to zero at those locations, giving unrealistic high values of the percent error when dividing in equation (56) by a very small number.

9.4.3 Percent error of RMS velocity fluctuations

Percent error of the RMS velocities was calculated for the three components (U_{RMS}, V_{RMS}, W_{RMS}) using equation (56) and the expression for the standard deviation given by Benedict and Gould (1996):

$$\sigma_X = \sqrt{\frac{\langle u'^4 \rangle - \langle u'^2 \rangle^2}{4\langle u'^2 \rangle}} \tag{58}$$

Contour plots of percent error of U_{RMS} , V_{RMS} and W_{RMS} were generated for all experiments. Results are presented in the following figures for experiments 1, 5 and 10. The other experiments can be found in APPENDIX C.

Figure 83 shows percent error for U_{RMS} , Figure 84 corresponds to V_{RMS} , and Figure 85 corresponds to W_{RMS} . All three components show very similar values for all the experiments, with percent errors between 5 % and 10 % for most locations, and increasing up to 20 % (zones of light blue color) for a few locations near the bottom and near the phases of flow reversal.

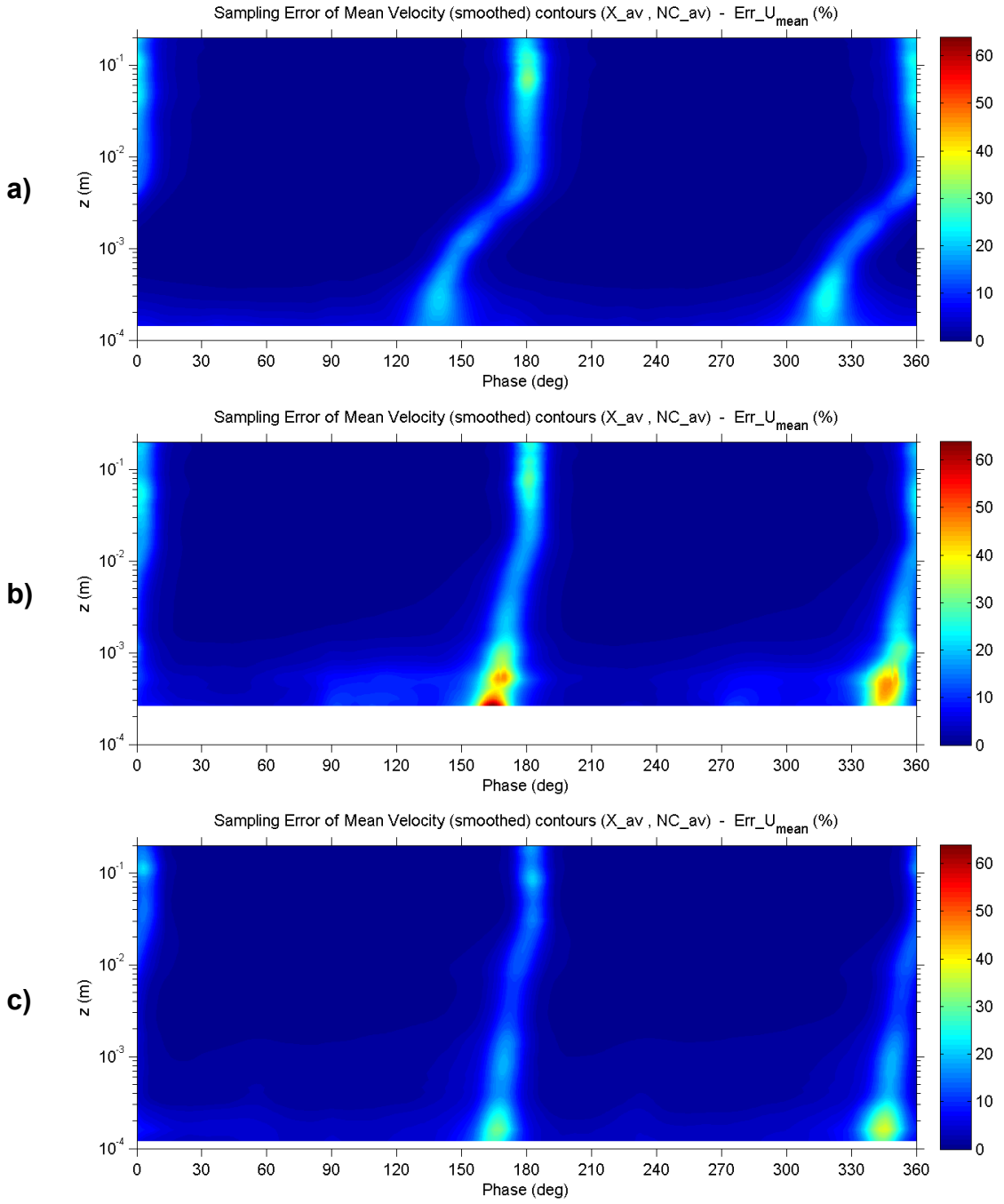


Figure 82. Contour plots showing percent error of mean velocity for the streamwise component (*U*) for experiments 1 (a), 5 (b) and 10 (c).

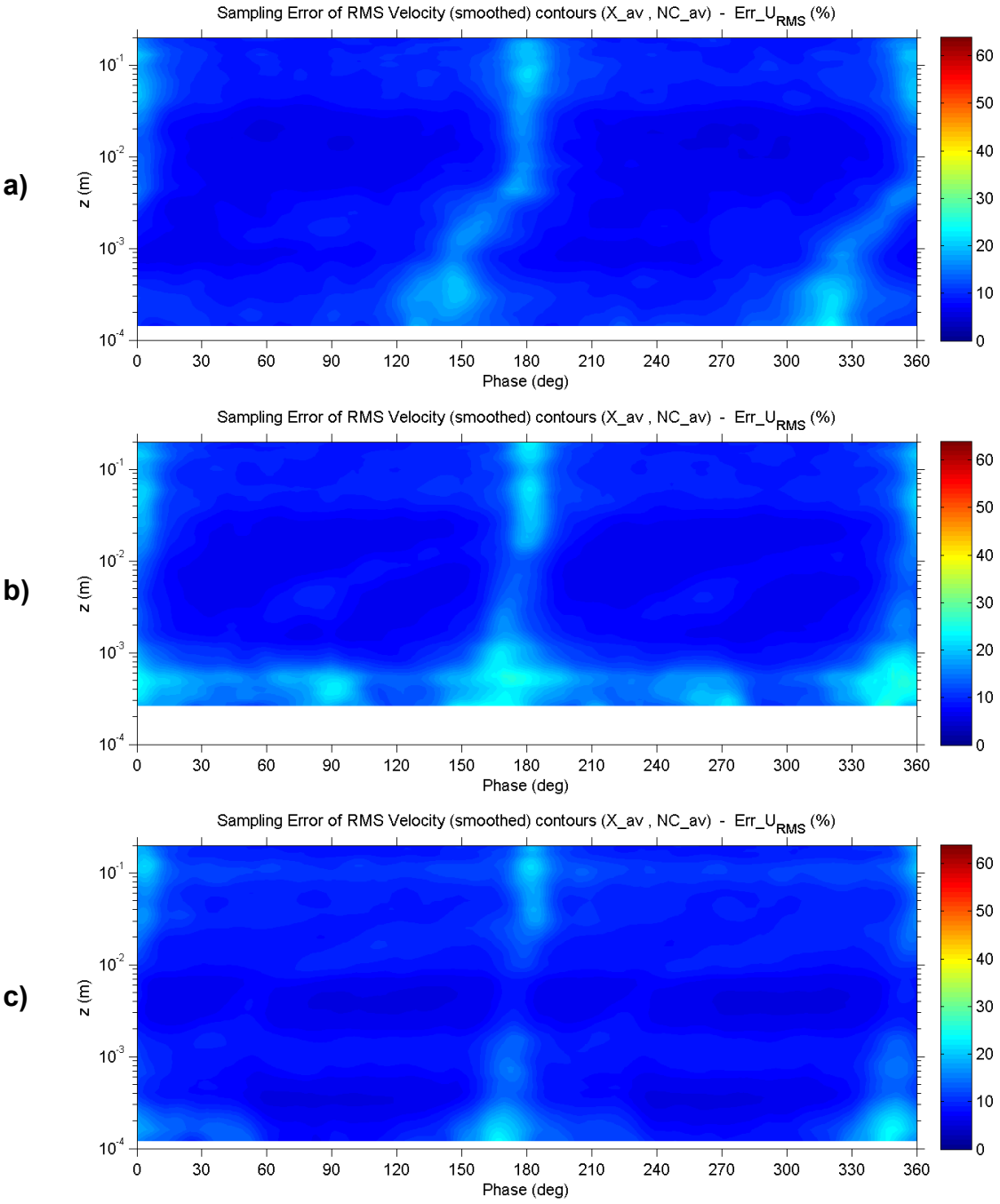


Figure 83. Contour plots showing percent error of U_{RMS} for experiments 1 (a), 5 (b) and 10 (c).

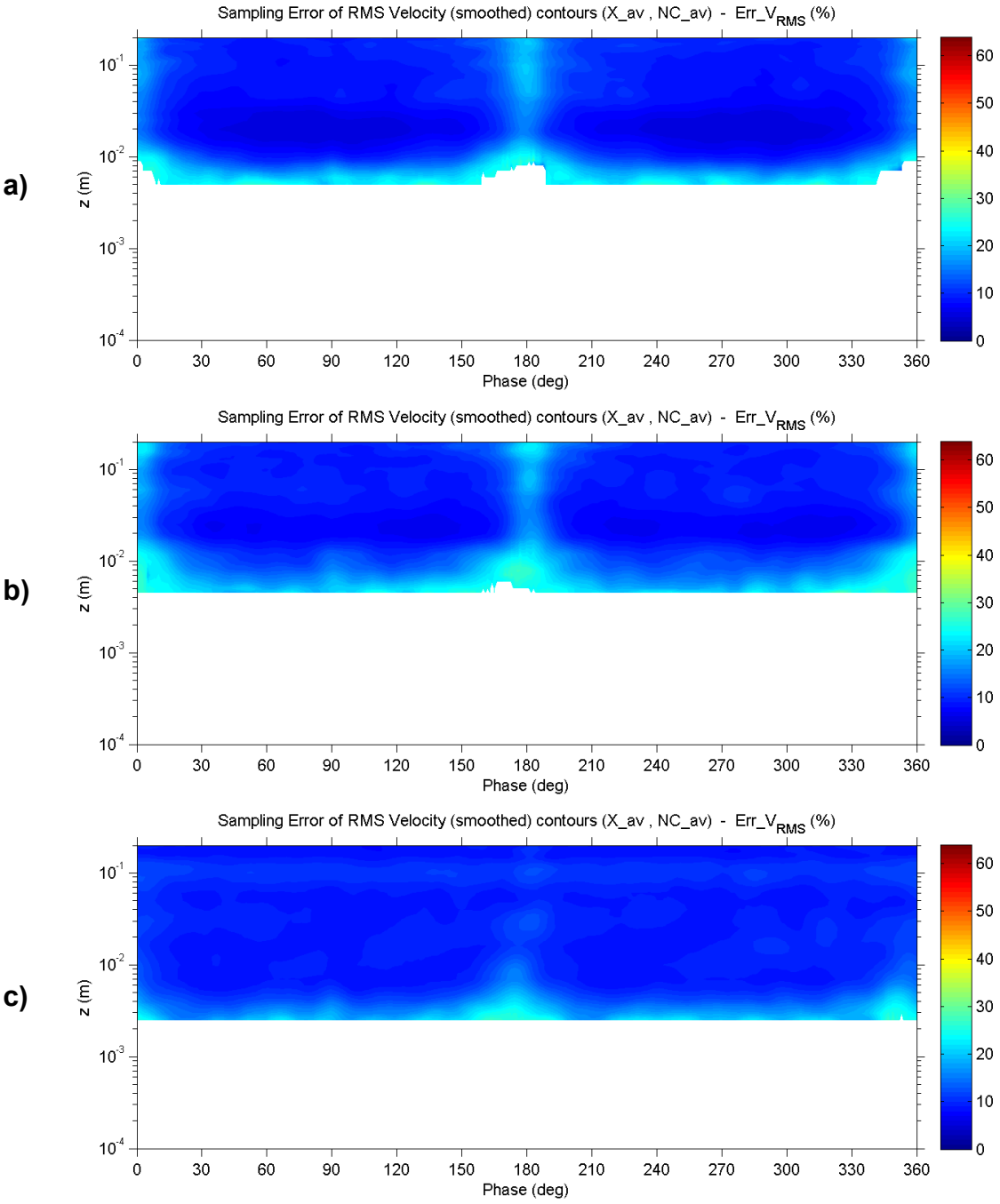


Figure 84. Contour plots showing percent error of V_{RMS} for experiments 1 (a), 5 (b) and 10 (c).

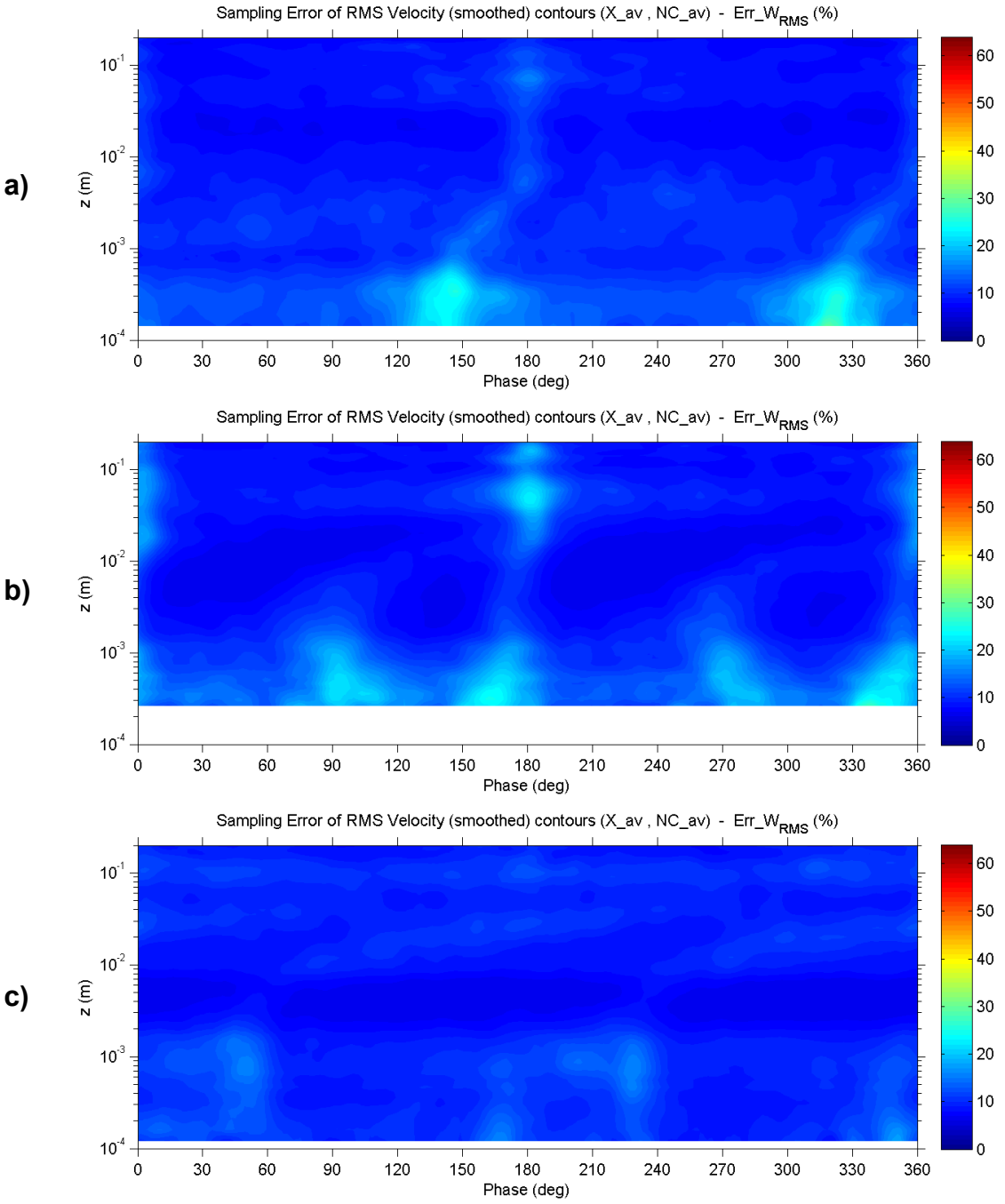


Figure 85. Contour plots showing percent error of W_{RMS} for experiments 1 (a), 5 (b) and 10 (c).

9.4.4 Percent error of turbulent kinetic energy

In order to calculate percent error of *TKE*, the rules of error propagation were applied to the expression of *TKE* given in equation (43), yielding:

$$S.E.(TKE) = \frac{1}{2}\left[S.E.\left(\left\langle u'^2 \right\rangle\right) + S.E.\left(\left\langle v'^2 \right\rangle\right) + S.E.\left(\left\langle w'^2 \right\rangle\right)\right] \tag{59}$$

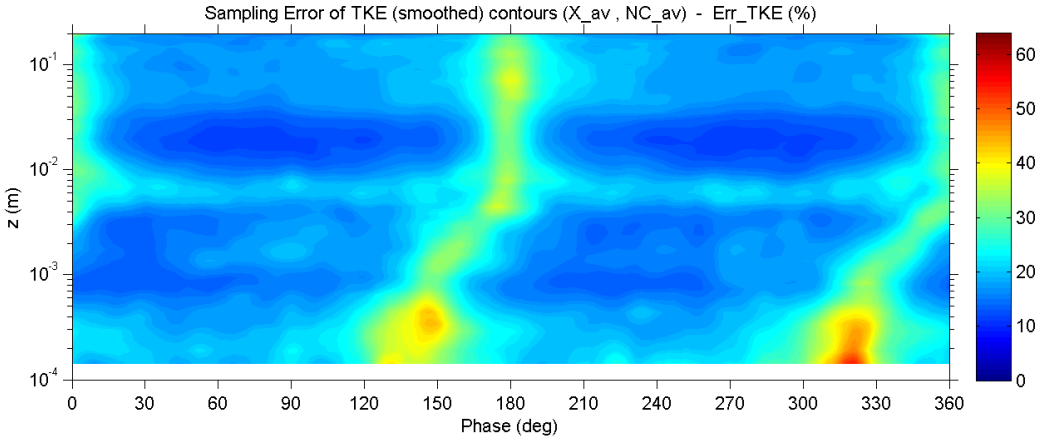
In this equation, the standard errors (*S.E.*) of the mean-squared variables for the three velocity components $\left\langle u'^2 \right\rangle, \left\langle v'^2 \right\rangle, \left\langle w'^2 \right\rangle$ were calculated using equation (55) and the expression for the standard deviation given by Benedict and Gould (1996):

$$\sigma_X = \sqrt{\left\langle u'^4 \right\rangle - \left\langle u'^2 \right\rangle^2} \tag{60}$$

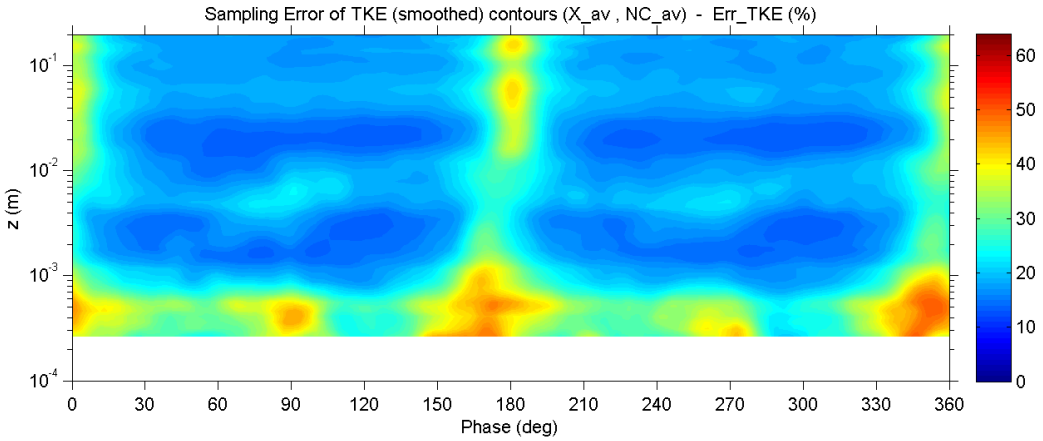
Finally, equation (56) was used to calculate the percent error of *TKE*. Contour plots were generated for all experiments. Results are presented in Figure 86 for experiments 1, 5 and 10. The other experiments can be found in APPENDIX C.

The values obtained were similar in all the experiments, with percent errors between 10 % and 20 % for most locations, and increasing up to 40 % (zones of yellow color) for a few locations near the bottom and near the phases of flow reversal.

a)



b)



c)

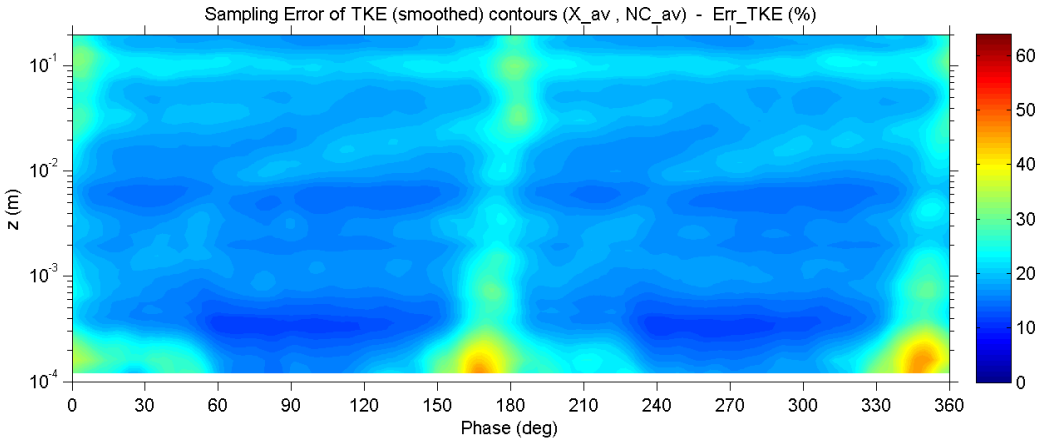


Figure 86. Contour plots showing percent error of *TKE* for experiments 1 (a), 5 (b) and 10 (c).

9.4.5 Percent error of Reynolds shear stress

Percent error of Reynolds shear stress was calculated for the *x-z* component $\langle u'w' \rangle$ using equation (56) and the expression for the standard deviation given by Benedict and Gould (1996):

$$\sigma_X = \sqrt{\langle u'^2 \, w'^2 \rangle - \langle u'w' \rangle^2} \tag{61}$$

In this case, using $\langle u'w' \rangle$ in the denominator of equation (56) resulted in divisions by a very small number for certain locations, giving unrealistic high values of percent error, so the denominator was replaced with the product U_{RMS} x W_{RMS} .

Contour plots were generated for all experiments. Results are presented in Figure 87 for experiments 1, 5 and 10. The other experiments can be found in APPENDIX C.

The values obtained were similar in all the experiments, with percent errors between 10 % and 15 % for most locations, and increasing to 30 % (zones of light blue and green color) for the locations near the bottom and up to 40 % (zones of yellow and orange color) near the phases of flow reversal. It must be noted that the percent error of this variable was substantially higher for the locations near the bottom compared to the rest of locations in the vertical. The reason for this can be explained from the fact that the number of valid data points *N* for the combination of *u'w'* was smaller than for each of the individual components *u'* or *w'* alone. The results were more sensitive to this effect near the bottom were *N* of the individual components was already small. This effect can be more clearly observed in the plots of experiments 1 (Figure 87a) and 5 (Figure 87b).

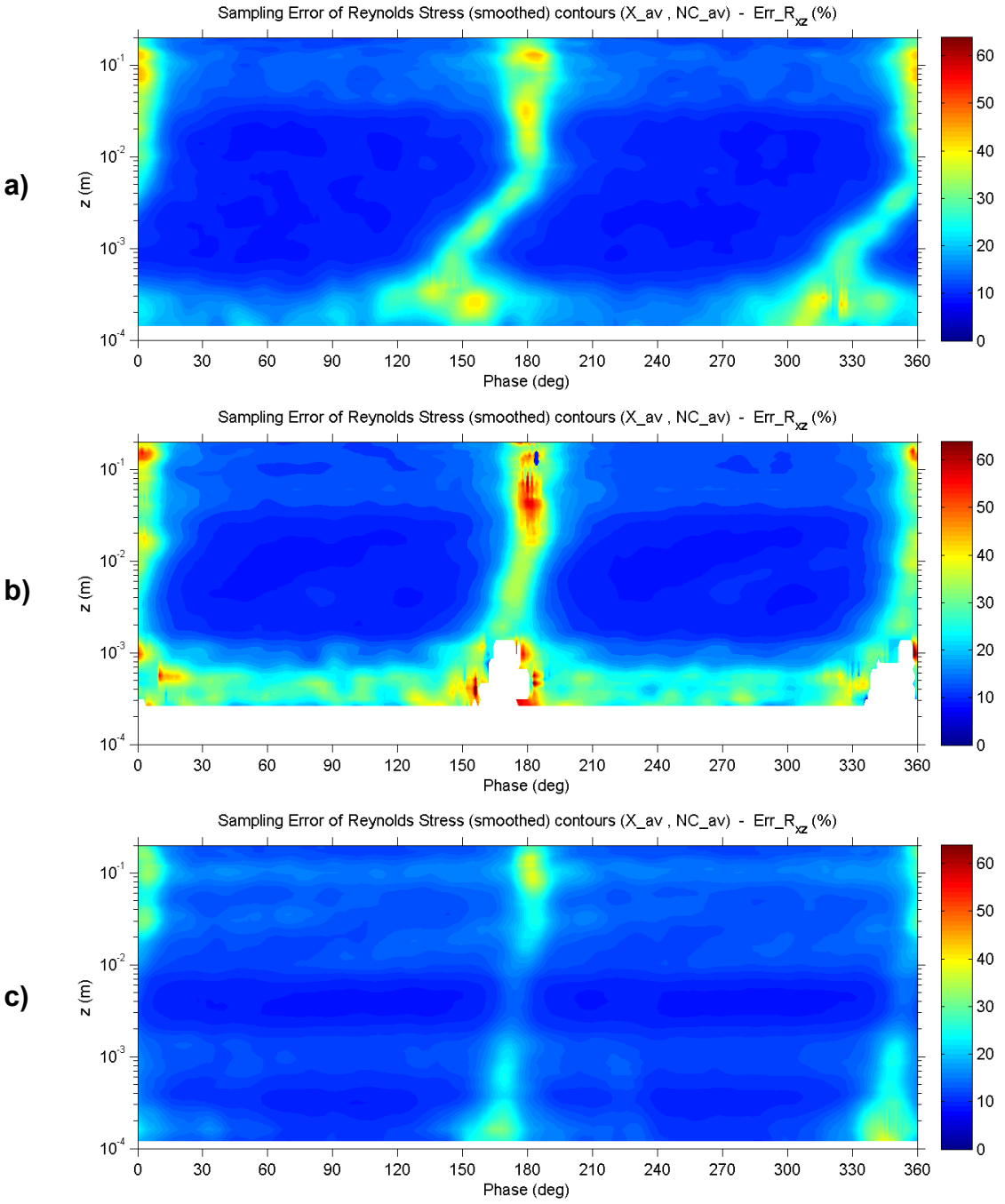


Figure 87. Contour plots showing percent error of *TKE* for experiments 1 (a), 5 (b) and 10 (c).

CHAPTER 10

CONCLUSIONS

10.1 General conclusions

A series of experiments were performed involving the measurement of velocities inside the boundary layer of oscillatory flows over a flat and smooth bed. The experiments were designed to cover the complete range of wave Reynolds numbers in the transition regime, between the upper limit of the laminar regime and the lower limit of the turbulent regime. The main focus was to analyze in detail the mean flow and turbulence characteristics of the oscillatory boundary layer in the transition regime. The key findings of this work are summarized below.

Mean velocities exhibited the characteristic behavior of oscillatory flows in which a phase lead exists for the velocities inside the boundary layer with respect to the outer flow. This phase lead was around 45º for low Re_w (laminar regime) and reduced to about 10º for higher Re_w (turbulent regime). The boundary layer was well-defined and the thickness ranged between 10 mm for experiment 1 (low Re_w) and 50 mm for experiment 10 (high Re_w). No influence from turbulence in the outer flow was observed inside the boundary layer. Also, the two half-cycles of the oscillation seemed to be quite independent from each other, as observed from the contour plots in many of the turbulence variables.

For the experiments in the transition regime, an interesting effect was observed in the mean velocity for the phases of decelerating flow very close to the bed (less than 1 mm). In that zone, a "lobe" of higher velocities appeared in the contour plots, extending all the way down to the bed. First, it was located at around phase 120º for the lower Re_w experiments, and then it extended towards phase 90º as Re_w increased. This zone of higher mean velocities could be related to the presence of turbulence that finds its way down to the bed in the decelerating stage of the half-cycle, bringing higher momentum flow to the lower part of the boundary layer.

Turbulence results support that conclusion. The zone of high turbulence intensity inside the boundary layer got closer to the bed as Re_w increased, reaching all the way down to 0.1 mm in the higher Re_w experiment. Also, this zone of high turbulence was located around phase 120º for the low Re_w experiments and progressively moved ahead towards phase 60º in experiment 10. In terms of magnitude, turbulence increased with Re_w as expected. Also, turbulence in the streamwise direction (U_{RMS}) was dominant with respect to the other two components and the magnitude was more than double of peak V_{RMS} and nearly triple of peak W_{RMS}. Regarding vertical location, U_{RMS} was more intense in the lower part of the boundary layer within 2 mm from the bed, while V_{RMS} and W_{RMS} seemed to be stronger in the upper part between 2 mm and 10 mm from the bed. Also, W_{RMS} showed upwards propagation of turbulence as the cycle developed, mainly happening during the deceleration part of the oscillation. Outside of the boundary layer, results showed very similar values of turbulence for the three components, indicating homogenous turbulence in the outer flow.

Reynolds shear stresses followed a different pattern, showing the highest intensities near the phase of flow reversal (180º) and located far from the bed at a distance between 5 and 20 mm. This distance seemed to be quite independent of Re_w. The magnitude increased with Re_w, as expected. It is likely that the occurrence of high Reynolds shear stress was associated with the presence of a shear layer generated by the opposing mean flow happening near the phases of flow reversal. This could explain the existence of vortex tubes and other coherent structures observed around those phases and distances from the bottom by other authors (Carstensen *et al.*, 2010).

Viscous shear stresses dominated near the bed and they quickly vanished far from the bed, as it was expected. The highest values were found around the phases of maximum streamwise velocity. Viscous shear stress profiles revealed a zone of nearly constant intensities very close to the bed, which suggested the presence of a viscous sublayer. The distance at which this behavior was found changed for every phase in the cycle, and also with Re_w.

Total shear stresses were calculated from the viscous and Reynolds contributions. The two components were clearly differentiated inside the boundary layer: the viscous component was

more intense near the bed for phases around the mean velocity maximum, while the Reynolds component was more intense far from the bed for phases around the flow reversal. In terms of the relative magnitude, the viscous component dominated for the low Re_w experiments in the laminar regime, as expected. However, for higher Re_w, the viscous component was still dominant, such that it was three times larger than the Reynolds component for experiment 5 (in the transition regime) and five times larger for experiment 10 (in the turbulent regime), which was completely unexpected.

Quadrant analysis for the experiments in the transition and fully turbulent regimes revealed a generalized zone of ejections in the decelerating stage of the half-cycle at a distance between 2 mm and 20 mm from the bed. Also, zones of sweeps were found within 1 mm from the bed scattered along the entire half-cycle.

The results of this work also provided evidence about the strong phase-dependent nature of several variables related to the viscous sublayer (bed shear stress, shear velocity, wave friction factor and viscous sublayer thickness), which was typically neglected or omitted in the literature before. In particular, bed shear stress and shear velocity exhibited a sinusoidal behavior for all Re_w. The friction factor showed strong phase dependence for low Re_w, while for high Re_w it became almost constant. Viscous sublayer thickness didn't follow any of those patterns and was more complicated to calculate.

Bed shear stresses increased with Re_w, as expected. A surprising behavior was observed in the phase of the maximum bed shear stress, which changed dramatically with Re_w: first leading about 40º ahead of the outer flow for low Re_w, then lagging up to 25º behind for the transitional Re_w experiments, and finally returning slightly ahead about 5º for high Re_w. Further investigation into this effect revealed the presence of two distinct peaks of bed shear stress in the half-cycle for Re_w in the transition regime. One peak was located in the acceleration stage and was related to the laminar behavior of the boundary layer. The other peak was found in the decelerating stage and was related to the turbulent behavior of the boundary layer. For low Re_w, the laminar peak dominated, while for high Re_w the turbulent peak dominated. This turbulent peak slowly shifted from around phase 120º for low Re_w towards phase 85º for the higher Re_w. Results in the

literature had never highlighted this effect before. This finding is expected to have significant implications for the entrainment and transport of sediment near the bed in unsteady flows.

Investigation of the viscous sublayer revealed that the classic steady flow threshold of $z^+ = 5$ doesn't work well for oscillatory flows. A new method was developed to calculate the thickness of the viscous sublayer taking into account the ratio of viscous to turbulent forces near the bed. The results revealed the existence of a laminar zone in the early phases of the half-cycle where viscous sublayer thickness grows, followed by a transition, before turbulence from the upper part of the boundary layer gets near the bed compressing the viscous sublayer down. The turbulent zone extended towards the earlier phases of the half-cycle as Re_w increased. For the highest Re_w experiment, the thickness of the viscous sublayer in the turbulent zone was about 0.15 mm.

The viscous sublayer analysis provided indirect evidence of the velocity profile becoming fully developed turbulent in the decelerating part of the half-cycle, in agreement with the observations made in the mean velocity analysis. Altogether, the results of this work show how turbulence effects compete with viscous effects during the transition regime, and how eventually the turbulence effects dominate close to the bottom as Re_w increases.

Finally, all these observations highlight the importance of having measurements so close to the bed, which where only possible with the high spatial resolution provided by the LDV in combination with the traverse system. A different instrument that was not capable of measuring within 1 mm from the bed would have completely misrepresented most of the key findings observed inside the oscillatory boundary layer.

10.2 Future work

A vast amount of data was collected during the 10 experiments in this work, however the analysis presented was not exhaustive and could be extended further. Of particular interest would be the analysis of the turbulent kinetic energy budget, taking into account all the terms in the equation including production, transport and dissipation of turbulence. This could be useful to provide a better understanding of the complete energy balance in oscillatory flows.

In addition, a good complement to this analysis would be to perform numerical simulations of oscillatory boundary layer flows using a CFD environment (such as OpenFoam or any other commercial package), in a similar way to Pedocchi *et al.* (2011). The results of this work could be very useful in the validation of such simulations. They could be performed for different scales, such as LES, RANS and DNS, and will help understand some of the more complicated effects that were difficult to capture during the experiments.

Also, several authors have shown the existence of coherent structures in the oscillatory boundary layer, such as vortex tubes or turbulent spots. Due to the limitations of the LDV technique, it was not possible to capture these structures, although indirect evidence can be seen in the turbulence analysis performed. Further experiments using PIV could be used to capture these effects in full detail.

Finally, it would be very interesting to explore in detail the implications of these results for sediment transport. In this regard, a Lagrangean model for tracking sediment particles could be used, in which the flow and shear stress characteristics could be fed from the experimental results of this work or directly from CFD results. Qualitative and also quantitative analyses could be done in order to understand the relation of the flow characteristics observed in these experiments with sediment transport and bed morphology under oscillatory flows.

REFERENCES

Literature cited

Adrian, J., and Yao, C.S. (1987). "Power spectra of fluid velocities measured by laser Doppler velocimetry". *Experiments in Fluids*, vol. 5, pp. 17-28.

Albrecht, H., Borys, M., Damaschke, N., and Tropea, C. (2003). "Laser Doppler and Phase Doppler Measurement Techniques", Eds. R. Adrian, M. Gharib, W. Merzkirch, D. Rockwell, and J. Whitelaw. Springer-Verlag, Berlin.

Bagnold, R.A. (1966). "An approach to the sediment transport problem from general physics". *U.S. Geological Survey Professional Paper 422-I*, USGS, Washington D.C., USA. 42 pp.

Benedict, L.H., and Gould, R.D. (1996). "Towards better uncertainty estimates for turbulence statistics". *Experiments in Fluids*, vol. 22, pp. 129-136.

Carstensen, S., Sumer, B.M., and Fredsoe, J. (2010). "Coherent structures in wave boundary layers. Part 1. Oscillatory motion". *Journal of Fluid Mechanics*, vol. 646, pp. 169-206.

Durst, F., Jovanovic, J., and Sender, J. (1995). "LDA measurements in the near-wall region of a turbulent pipe flow". *Journal of Fluid Mechanics*, vol. 295, pp.305-335.

Fredsoe, J. (1984). "Turbulent boundary layer in wave-current motion". *Journal of Hydraulic Engineering*, ASCE, vol. 110, no. 8, pp. 1103-1120.

Garcia, M.H., ed. (2008). "Chapter 2: Sediment Transport and Morphodynamics". *Sedimentation Engineering*, ASCE Manuals and Reports on Engineering Practice No. 110, ASCE/EWRI, Reston, Virginia.

Gayen, B., and Sarkar, S. (2011). "Negative turbulent production during flow reversal in a stratified oscillating boundary layer on a sloping bottom". *Physics of Fluids*, vol. 23 (101703).

Hino, M., Kashiwayanagi, M., Nakayama, A., and Hara, T. (1983). "Experiments on the turbulence statistics and the structure of a reciprocating oscillatory flow". *Journal of Fluid Mechanics*, vol. 131, pp. 363-400.

Jensen, B.L. (1988). "Experimental Investigation of Turbulent Oscillatory Boundary Layers". *Ph.D. Thesis*, Institute of Hydrodynamics and Hydraulic Engineering, Technical University of Denmark.

Jensen, B.L., Sumer, B.M., and Fredsoe, J. (1989). "Turbulent oscillatory boundary layers at high Reynolds numbers". *Journal of Fluid Mechanics*, vol. 206, pp. 265-297.

Jonsson, I.G. (1963). "Measurements in the turbulent wave boundary layer". *Proceedings of the 10th IAHR Congress*, London, vol. 1, pp. 85-92.

Jonsson, I.G. (1980). "A new approach to oscillatory rough turbulent boundary layers". *Ocean Engineering*, vol. 7, pp. 109-152.

Kamphuis, J.W. (1975). "Friction factor under oscillatory waves". *Journal of the Waterways, Harbors and Coastal Engineering Division*, ASCE, vol. 101, no. 2, pp. 135-144.

Liberzon, A., Luthi, B., Guala, M., Kinzelbach, W., and Tsinober, A. (2007). "On Anisotropy of Turbulent Flows in Regions of Negative Eddy Viscosity". *Springer Proceedings in Physics*, vol. 109, pp. 85-88.

Mazzuoli, M., Vittori, G., and Blondeaux, P. (2011). "Turbulent spots in oscillatory boundary layers". *Journal of Fluid Mechanics*, vol. 685, pp. 365-376.

Mier, J.M., and Garcia, M.H. (2009). “Challenges to LDV measurements in large experimental facilities: optimum seeding concentration”. *33rd IAHR Congress*, Vancouver (Canada).

Mier, J.M., and Garcia, M.H. (2011). “Particle size discrimination from single-probe backscatter laser Doppler velocimetry (LDV) measurements”. *2nd MEH Symposium*, Bahia Blanca (Argentina).

Mier, J.M., and Garcia, M.H. (2012). “3D LDV Measurements in Oscillatory Boundary Layers”. *AGU Fall Meeting*, San Francisco (USA).

Mier, J.M., and Garcia, M.H. (2013). “Data processing with MATLAB® for LDV measurements in unsteady periodic flows”. *3rd MEH Symposium*, Santa Fe (Argentina).

Mujal, A., Mier, J.M., Christensen, K.T., Bateman, A., and Garcia, M.H. (2014). "PIV Experiments in Rough-Wall, Laminar to Turbulent, Oscillatory Boundary Layers". *Experiments in Fluids*, vol. 55, no. 1, art. 1633.

Navier, C.L.M.H. (1823). “Memoire sur les lois du mouvement des fluids”. *Mem. Acad. R. Sci.*, vol. 6, pp. 389-441.

Niederschulte, M.A., Adrian, R.J., and Hanratty, T.J. (1990). “Measurements of turbulent flow in a channel at low Reynolds numbers”. *Experiments in fluids*, vol. 9, pp. 222-230.

Nikuradse, J. (1933). “Stromungsgesetze in rauhen Rohren”. *Forsch. Geb. Ingenieurwes.*, Ausg. Beill., vol. 4, p. 361. [Translation: “Laws of flows in rough pipes”, *NACA Technical Memorandum*, no. 1292, 62 pp., 1950].

Nezu, I., and Nakagawa, H. (1993). *Turbulence in Open-channel Flows*. IAHR, Monograph Series, Balkema, Rotterdam, The Netherlands.

Nezu, I., and Rodi, W. (1986). "Open-channel flow measurements with a laser Doppler anemometer." *Journal of Hydraulic Engineering*, ASCE, vol. 112, pp. 335–355.

Ozdemir, C.E., Hsu, T-J., and Balachandar, S. (2014). "Direct numerical simulations of transition and turbulence in smooth-walled Stokes boundary layer". *Physics of Fluids*, vol. 26, no. 045108.

Pedocchi, F. (2009). "Bed morphology and sediment transport under oscillatory flow". *Ph.D. thesis*. Ven Te Chow Hydrosystems Laboratory, University of Illinois at Urbana-Champaign, USA.

Pedocchi, F., Cantero, M.I., and Garcia, M.H. (2011). "Turbulent kinetic energy balance of an oscillatory boundary layer in the transition to the fully turbulent regime". *Journal of Turbulence*, vol. 12, no. 32, pp. 1-27.

Pedocchi, F., and Garcia, M.H. (2009). "Friction coefficient for oscillatory flow: the rough-smooth turbulent transition", *Journal of Hydraulic Research*, IAHR, vol. 47, no. 4, pp. 438-444.

Poisson, S.D. (1831). "Memoire sur les Equations Generales de l'Equilibre et du Mouvement des Corps Solides Elastiques et des Fluides". *J. Ec. Polytech.*, Paris, vol. 13, pp. 139-166.

Potter Industries Inc. (2008). <http://www.pottersbeads.com> (Sep. 22, 2008).

Prandtl, L. (1905). "Ueber Flussigkeitbewegung bei sehr kleiner Reibung". *Verh. Int. Math. Kongr., 3rd*. Heidelberg, 1904, pp. 484-491. [Translated: "Motion of fluids with very little viscosity", *NACA Technical Memorandum*, no. 452, 1928.]

Rayleigh, Lord (1911). "On the motion of solid bodies through viscous liquid". *Philosophical Magazine*, series 6, vol. 21, no. 126, pp. 697-711.

Saint-Venant, B. (1843). “Mémoire sur la dynamique des fluides”. *C. R. Acad. Sci.*, Paris, Série A 17, pp. 1240-1242.

Schlichting, H. (1979). *Boundary Layer Theory*, 7th Edition, McGraw-Hill, New York.

Sleath, J.F.A. (1987). “Turbulent oscillatory flow over rough beds”. *Journal of Fluid Mechanics*, vol. 182, pp. 369-409.

Spalart, P.R., and Baldwin, B.S. (1987). “Direct simulation of a turbulent oscillating boundary layer”. *NASA Technical Memorandum 89460*, Ames Research Centre Moffett Field, California (USA).

Stokes, G.G. (1845). “On the theories of the internal friction of fluids in motion, and of the equilibrium and motion of elastic solids”. *Trans. Cambridge Philos. Soc.*, vol. 8, pp. 287-305.

Sumer, B.M., Jensen, B.L., and Fredsoe, J. (1987). “Turbulence in Oscillatory Boundary Layers”. *Advances in Turbulence: Proceedings of the 1st European Turbulence Conference*, Lyon (France), July 1986. Ed.: G. Comte-Bellot and J. Mathieu, Springer Verlag, Berlin, pp. 556-567.

Tani, I. (1977). “History of boundary-layer theory”. *Annual Review of Fluid Mechanics*, vol. 9, pp. 87-111.

Thiesen, M., Scheel, K., Diesselhorst, H. (1900). “Untersuchungen über die thermische Ausdehnung von festen und tropfbar flüssigen Körpern—bestimmung der Ausdehnung des Wassers für die zwischen 0° und 40° liegenden temperaturen”. *Wiss. Abh. Phys. Tech. Reichsanst*, vol. 3, p. 1–70.

Tropea, C., Yarin, A., and Foss, J. (Editors) (2007). *Springer Handbook of Experimental Fluid Mechanics*. Springer, Berlin, 1557 pp.

TSI Inc. (2005). *LDV system manuals set*, Shoreview, Minnesota.

Ursell, F (1953). "The long-wave paradox in the theory of gravity waves". *Proceedings of the Cambridge Philosophical Society*, 49 (4): 685–694.

Vittori, G., and Verzicco, R. (1998). "Direct simulation of transition in an oscillatory boundary layer". *Journal of Fluid Mechanics*, vol. 371, pp. 207-232.

Zhou, J., Adrian, R.J., Balachandar, S., and Kendall, T.M. (1999). "Mechanisms for generating coherent packets of hairpin vortices in channel flow". *Journal of Fluid Mechanics*, vol. 387, pp. 353-396.

Other useful references

Clauser, F.H. (1956). "The turbulent boundary layer". In *Advances in Applied Mechanics*, edited by H.L. Driden and T. von Karman, vol. 4, pp. 2-51.

Eckelmann, H. (1974). "The structure of the viscous sublayer and the adjacent wall region in a turbulent channel flow". *Journal of Fluid Mechanics*, vol. 65, part 3, pp. 439-459.

Einstein, H.A., and Li, H. (1956). "The viscous sublayer along a smooth boundary". *Journal of the Engineering Mechanics Division*, Proc. ASCE, vol. 82, no. EM 2, paper 945.

Keulegan, G.H. (1938). "Laws of turbulent flow in open channels." *Journal of Research of the National Bureau of Standards,* Research Paper 1151, vol. 21, pp. 707–741, Washington, D.C.

Kline, S.J., Reynolds, W.C., Schraub, F.A., and Runstadler, P.W. (1967). "The structure of turbulent boundary layers". *Journal of Fluid Mechanics*, vol. 30, part 4, pp. 741-773.

Laufer, J. (1950). “Investigation of turbulent flow in a two-dimensional channel”. NACA technical note, TN 2123.

Moody, L.F. (1944). “Friction factors for pipe flow”. *Transactions of the American Society of Civil Engineers*, ASCE, vol. 66. Reston (Virginia), USA.

Popovich, A.T., and Hummel, R.L. (1967). “Experimental Study of the Viscous Sublayer in Turbulent Pipe Flow”. *AIChE Journal*, vol. 13, issue 5, pp. 854-860.

Reichardt, H. (1938). “Messungen turbulenter Schwankungen”. *Naturwissenschaften*, Heft 24/25, pp. 404-408.

Vanoni, V.A. (1975). *Sedimentation engineering*, ASCE, New York (New York), USA.

van Rijn, L.C. (1993). *Principles of sediment transport in rivers, estuaries and coastal seas*. Aqua Publications, Amsterdam, Netherlands. ISBN 90-800356-2-9. vi, 715 pp.

Yalin, M.S., and Karahan, E. (1979). “Inception of sediment transport”. *Journal of the Hydraulics Division*, Proc. ASCE, vol. 105, no. HY11, paper 14975, pp. 1433-1443.

Yalin, M.S. (1992). *River mechanics*, Pergamon Press, New York.

APPENDIX A

COMPUTER CODE FOR PROCESSING RAW LDV DATA (IN MATLAB©)

LDV_PROCESSING

Program to read raw data from LDV measurements and process it to obtain a clean set of data that can be later analyzed. Applied to data from LDV measurements from experiments in the HydroLab.

Contents

Main routine

```
% LDV_PROCESSING
% Program to read raw data from LDV measurements and process it
% to obtain a clean set of data that can be later analyzed.
% Applied to data from LDV measurements from experiments in the HydroLab.
%
% JM Mier
% Ven Te Chow Hydrosystems Laboratory, UIUC
% May 2009
% Revised March 2012
%

% Data files location
dir_read = uigetdir('C:\USERS\JM MIER\Research\Thesis exps','Select Path for RAW DATA:');
dir_save = uigetdir(dir_read,'Select Path for PROCESSING RESULTS:');
dir_clean = uigetdir(dir_save,'Select Path for CLEAN DATA:');
file_traverse = dir([dir_read,'\*.txt']);
file_list = dir([dir_read,'\*.csv']);

% Display experiment name
dir_read_dividers = find(dir_read=='\');
disp(' ');
disp(['Flume:           ',dir_read(dir_read_dividers(end-3)+1:dir_read_dividers(end-2)-1)]);
disp(['Project:         ',dir_read(dir_read_dividers(end-2)+1:dir_read_dividers(end-1)-1)]);
disp(['Experiment:      ',dir_read(dir_read_dividers(end-1)+1:dir_read_dividers(end-0)-1)]);
disp(['LDV_Processing:  ',dir_read(dir_read_dividers(end-0)+1:end)]);

% Case example to plot
disp(' ');
X_case = input('X_case = ');
Y_case = input('Y_case = ');
```

```
Z_case = input('Z_case = ');

% Read raw velocity data
[X Y Z] = ReadPositions_LDV(file_traverse, dir_read);         % m
[T1_raw T2_raw T3_raw V1_raw V2_raw V3_raw] = ReadData_LDV(file_list, dir_read, X, Y, Z, X_case, Y_case,
Z_case);         % m/s
% Note that velocities may or may not have been transformed yet into U,V,W cartesian reference

% Cycles detection and filtering of bad data points
[T1_sel T2_sel T3_sel V1_sel V2_sel V3_sel n_cycles_T1 n_cycles_T2 n_cycles_T3 zeroT_pos_T1 zeroT_pos_T2
zeroT_pos_T3 Ts] = SelectData_LDV(dir_save, X, Y, Z, T1_raw, T2_raw, T3_raw, V1_raw, V2_raw, V3_raw,
X_case, Y_case, Z_case, rep);

% Even-time sampling
[T_even V1_even V2_even V3_even n_cycles_T_even Fs a] = InterpData_LDV(X, Y, Z, T1_sel, T2_sel, T3_sel,
V1_sel, V2_sel, V3_sel, n_cycles_T1, n_cycles_T2, n_cycles_T3, zeroT_pos_T1, zeroT_pos_T2, zeroT_pos_T3,
Ts, X_case, Y_case, Z_case, dir_save);
% A lot of good data is discarded in this process
% and a lot of points will be empty because of lack of good interpolation conditions

% Transformation matrix
% Convert V1,V2,V3 from LDV reference system into U,V,W in cartesian axis reference.
% This can only be done after we have even-time sampled data
[U V W] = Transform_LDV(X, Y, Z, V1_even, V2_even, V3_even, T_even, X_case, Y_case, Z_case, dir_save);

% Conversion from cell structures to 5D matrices (easier to handle)
% Implies having the same number of elements in every cell
% Padding with NaNs to handle different number of cycles at each location
n_cycles = n_cycles_T_even;
n_cycles_max = max(max(max(n_cycles)));
x = X;
y = Y;
z = Z;
t = T_even;
u = NaN(numel(x),numel(y),numel(z),n_cycles_max, numel(t));
v = NaN(numel(x),numel(y),numel(z),n_cycles_max, numel(t));
w = NaN(numel(x),numel(y),numel(z),n_cycles_max, numel(t));
for i = 1 : numel(x)
    for j = 1 : numel(y)
        for k = 1 : numel(z)
            for nc = 1 : n_cycles(i,j,k)
                u(i,j,k,nc,:) = U{i,j,k}(nc,:);
                v(i,j,k,nc,:) = V{i,j,k}(nc,:);
                w(i,j,k,nc,:) = W{i,j,k}(nc,:);
            end
        end
    end
end

% For now on, only this variables: t,x,y,z,u,v,w
% After this point, all locations have same number of cycles

% Calculate real z=0 level (z0)
% Needs to be calculated for every (x,y) location
z0 = zeros(numel(x),numel(y));
for i = 1 : numel(x)
```

```
    for j = 1 : numel(y)
        z0(i,j) = ZeroLevel_LDV(z,t(1,:),u(i,j,:,:,:),Ts,dir_save);
    end
end

% Quality control of data
CheckData_LDV (x,y,z,t(1,:),u,v,w,n_cycles,X_case,Y_case,Z_case,dir_save,Ts,Fs,a);
% If data doesn't look good, repeat the processing, or check the original raw data.

% Save clean data
file_save = 'Run_clean-all.mat';
save([dir_clean '\' file_save],
'x','y','z','z0','t','u','v','w','n_cycles','Ts','Fs','a','X_case','Y_case','Z_case');

% Final statements
disp('LDV processing finished!');
% end of program
```

ReadPositions_LDV

```
function [X Y Z] = ReadPositions_LDV(file_traverse, dir_name)
% READPOSITIONS_LDV
% Function to read LDV measurement positions from
% text file (.txt) with columns (x,y,z).
% ONLY these 3 columns must be present in the file.
%
% file_traverse: file in the folder containing the traverse info
% dir_name: folder where the files are
%
% JM Mier
% May 2009
%

% Input parameters about the file
line1 = 1;                      % first line of data (to get rid of headers)
column1 = 1;                    % first column of data (to get rid of headers)

file = file_traverse.name;
file_fullpath = [dir_name '\' file];

% Reading the data
data = dlmread(file_fullpath,'\t',line1-1, column1-1);      % first line/column to read is 0, for this
x = data(:,1)/1000;              % position of data measurements in the horizontal-longitudinal
y = data(:,2)/1000;              % position of data measurements in the horizontal-transversal
z = data(:,3)/1000;              % position of data measurements in the vertical

% Obtaining the grid locations of the LDV measurements for each axis
X = unique(x);
Y = unique(y);
Z = unique(z);

% end of function
end
```

ReadData_LDV

```
function [T1 T2 T3 V1 V2 V3] = ReadData_LDV(file_list, dir_name, X, Y, Z, X_case, Y_case, Z_case)
% READDATA_LDV
% Function to read data from text files with columns
% and transform it into matrices.
% Data MUST have been taken first moving in Z,
% then in Y, and last in X.
%
% file_list: list of files in the folder
% dir_name: folder where the files are
% X, Y, Z: vectors containing the coordinates of the measurement locations in the flume
% X_case, Y_case, Z_case: coordinates of the location of the case to show in the plots
%
% JM Mier
% Ven Te Chow Hydrosystems Laboratory, UIUC
% May 2009
% Revised: March 2012
%

% Input parameters about the files
line1 = 3;                      % first line of data (to get rid of headers)
column1 = 1;                    % first column of data (to get rid of headers)

T1 = cell(numel(X),numel(Y),numel(Z));        % preallocation for speed
T2 = cell(numel(X),numel(Y),numel(Z));        % preallocation for speed
T3 = cell(numel(X),numel(Y),numel(Z));        % preallocation for speed
V1 = cell(numel(X),numel(Y),numel(Z));        % preallocation for speed
V2 = cell(numel(X),numel(Y),numel(Z));        % preallocation for speed
V3 = cell(numel(X),numel(Y),numel(Z));        % preallocation for speed

% Main loop (through space)
hw = waitbar(0,'Reading raw data files ...'); set(hw,'Name',[num2str(round(0*100)),' %']); drawnow;

for i = 1 : numel(X)    %numel(file_list)
    for j = 1 : numel(Y)
        for k = 1 : numel(Z)
            n_file = (i-1)*numel(Y)*numel(Z) + (j-1)*numel(Z) + k;
            file = file_list(n_file).name;
            file_fullpath = [dir_name '\' file];

% Reading the data
data = dlmread(file_fullpath,',',line1-1,column1-1);      % first line/column to read is 0, for this
dlmread function. (',' means delimited by commas)
t1 = data(:,4);              % time of data measurements in channel 1
v1 = data(:,5);              % velocity measurements in channel 1
t2 = data(:,6);              % time of data measurements in channel 2
v2 = data(:,7); %zeros(numel(u),1);      % velocity measurements in channel 2
t3 = data(:,8);              % time of data measurements in channel 3
v3 = data(:,9);              % velocity measurements in channel 3

% Assign default value for empty data
% dlmread gives a value of "0" for empty data, thus when both t=0 and v=0 means it was empty data
% Use "-1" because it will be removed later on during the processing
t1(t1==0 & v1==0) = -1;
t2(t2==0 & v2==0) = -1;
```

```
t3(t3==0 & v3==0) = -1;

% Data conversion into a matrix (X,Y,Z,S)
T1{i,j,k} = t1;               % store in a cell matrix because every file can have a different number of
samples, so this dimension is free in a cell structure
T2{i,j,k} = t2;               % store in a cell matrix because every file can have a different number of
samples, so this dimension is free in a cell structure
T3{i,j,k} = t3;               % store in a cell matrix because every file can have a different number of
samples, so this dimension is free in a cell structure
V1{i,j,k} = v1;               % store in a cell matrix because every file can have a different number of
samples, so this dimension is free in a cell structure
V2{i,j,k} = v2;               % store in a cell matrix because every file can have a different number of
samples, so this dimension is free in a cell structure
V3{i,j,k} = v3;               % store in a cell matrix because every file can have a different number of
samples, so this dimension is free in a cell structure

pw = n_file/numel(file_list);
waitbar(pw,hw); set(hw,'Name',[num2str(round(pw*100)),' %']); drawnow;

% end of main loop
        end
    end
end
close(hw);

% end of function
end
```

SelectData_LDV

```
function [T1 T2 T3 V1 V2 V3 n_cycles_T1 n_cycles_T2 n_cycles_T3 zeroT_pos_T1 zeroT_pos_T2 zeroT_pos_T3
Ts] = SelectData_LDV(dir_name, X, Y, Z, T1_raw, T2_raw, T3_raw, V1_raw, V2_raw, V3_raw, X_case, Y_case,
Z_case, rep)
% SELECTDATA_LDV
% Function to detect the cycles of the data and also to select and filter
% the good part of the raw data obtained with LDV.
%
% X, Y, Z: vectors containing the coordinates of the measurement locations in the flume
% T1_raw, T2_raw, T3_raw, V1_raw, V2_raw, V3_raw: raw measurements of velocities and time
% X_case, Y_case, Z_case: coordinates of the location of the case to show in the plots
% rep: control variable to indicate if this routine is run repeatedly, so that certain parts of the code
don't get executed
%
% ** It is ready to take differente number of data points in each channel
% ** However, results would be nicer with same number of data points in each channel (although they may
have different time stamps) **
% ** Use coincidence mode to achieve that (even with a rough coincidence interval) **
%
% JM Mier
% Ven Te Chow Hydrosystems Laboratory, UIUC
% May 2009
% Revised: March 2012
%
```

```
% Cycles detection
% Identify first row of data after sync pulse in each bin
% and compute number of cycles at each location
for i = 1 : numel(X)
    for j = 1 : numel(Y)
        for k = 1 : numel(Z)
            zeroT_pos_T1_raw{i,j,k} = find(T1_raw{i,j,k}>circshift(T1_raw{i,j,k},-1))+1;
            zeroT_pos_T2_raw{i,j,k} = find(T2_raw{i,j,k}>circshift(T2_raw{i,j,k},-1))+1;
            zeroT_pos_T3_raw{i,j,k} = find(T3_raw{i,j,k}>circshift(T3_raw{i,j,k},-1))+1;
            % when one channel has no data at some particular location (x,y,z)
            if numel(zeroT_pos_T1_raw{i,j,k}) == 0;  zeroT_pos_T1_raw{i,j,k} = [1]; end;
            if numel(zeroT_pos_T2_raw{i,j,k}) == 0;  zeroT_pos_T2_raw{i,j,k} = [1]; end;
            if numel(zeroT_pos_T3_raw{i,j,k}) == 0;  zeroT_pos_T3_raw{i,j,k} = [1]; end;
            % for the case of steady flows (only one cycle)
            if (numel(zeroT_pos_T1_raw{i,j,k}) == 1 && numel(zeroT_pos_T2_raw{i,j,k}) == 1 &&
numel(zeroT_pos_T3_raw{i,j,k}) == 1)
                zeroT_pos_T1_raw{i,j,k} = [1;zeroT_pos_T1_raw{i,j,k}];
                zeroT_pos_T2_raw{i,j,k} = [1;zeroT_pos_T2_raw{i,j,k}];
                zeroT_pos_T3_raw{i,j,k} = [1;zeroT_pos_T3_raw{i,j,k}];
            end
            n_cycles_T1_raw(i,j,k) = numel(zeroT_pos_T1_raw{i,j,k})-1;    % the -1 supresses one extra
count from circshift; the last cycle, which is usually incomplete is still considered, since the data
points are still good data; the first cycle, on the contrary, is always incomplete and wrong data, and
never makes it into the count.
            n_cycles_T2_raw(i,j,k) = numel(zeroT_pos_T2_raw{i,j,k})-1;    % the -1 supresses one extra
count from circshift; the last cycle, which is usually incomplete is still considered, since the data
points are still good data; the first cycle, on the contrary, is always incomplete and wrong data, and
never makes it into the count.
            n_cycles_T3_raw(i,j,k) = numel(zeroT_pos_T3_raw{i,j,k})-1;    % the -1 supresses one extra
count from circshift; the last cycle, which is usually incomplete is still considered, since the data
points are still good data; the first cycle, on the contrary, is always incomplete and wrong data, and
never makes it into the count.
        end
    end
end

% Discard first cycle
% Will only have effect in oscillatory flow, where the first cycle is incomplete and data has wrong time
stamp
for i = 1 : numel(X)
    for j = 1 : numel(Y)
        for k = 1 : numel(Z)

            % New vectors T1,T2,T3,V1,V2,V3
            if rep == 1         % Remove first cycle data only when this routine is executed for the
first time
                V1{i,j,k} = V1_raw{i,j,k}(zeroT_pos_T1_raw{i,j,k}(1):end);
                V2{i,j,k} = V2_raw{i,j,k}(zeroT_pos_T2_raw{i,j,k}(1):end);
                V3{i,j,k} = V3_raw{i,j,k}(zeroT_pos_T3_raw{i,j,k}(1):end);
                T1{i,j,k} = T1_raw{i,j,k}(zeroT_pos_T1_raw{i,j,k}(1):end);
                T2{i,j,k} = T2_raw{i,j,k}(zeroT_pos_T2_raw{i,j,k}(1):end);
                T3{i,j,k} = T3_raw{i,j,k}(zeroT_pos_T3_raw{i,j,k}(1):end);
            elseif rep == 2     % Don't remove first cycle data when this routine is executed subsequent
times
                V1{i,j,k} = V1_raw{i,j,k};
                V2{i,j,k} = V2_raw{i,j,k};
```

```
                V3{i,j,k} = V3_raw{i,j,k};
                T1{i,j,k} = T1_raw{i,j,k};
                T2{i,j,k} = T2_raw{i,j,k};
                T3{i,j,k} = T3_raw{i,j,k};
            end

            % New zeroT positons and number of cycles
            zeroT_pos_T1{i,j,k} = [1 ; find(T1{i,j,k}>circshift(T1{i,j,k},-1))+1];
            zeroT_pos_T2{i,j,k} = [1 ; find(T2{i,j,k}>circshift(T2{i,j,k},-1))+1];
            zeroT_pos_T3{i,j,k} = [1 ; find(T3{i,j,k}>circshift(T3{i,j,k},-1))+1];

            n_cycles_T1(i,j,k) = numel(zeroT_pos_T1{i,j,k})-1;
            n_cycles_T2(i,j,k) = numel(zeroT_pos_T2{i,j,k})-1;
            n_cycles_T3(i,j,k) = numel(zeroT_pos_T3{i,j,k})-1;
        end
    end
end

% We are not discarding the last cycle (usually not complete, but good data points),
% because it is still useful for the interpolation.

% Plot number of cycles
figure()
hb = bar3h([squeeze(n_cycles_T1(:,Y_case,:)) squeeze(n_cycles_T2(:,Y_case,:))
squeeze(n_cycles_T3(:,Y_case,:))],'grouped');
set(hb,'EdgeColor','none'); colormap([0 1 0;0 0 1;1 0 1]);
legend('V1','V2','V3');
title(['Number of cycles for points in section:  Y = ', num2str(Y(Y_case)*1000), ' mm']);
xlabel('X locations');
ylabel('# cycles');
zlabel('Z locations');
view([-65 22]);
set(gcf,'position',[228 228 560 420]);          % Back to original size in left window

% Portion of good data selection ("chopping")
% Period of the signal (for steady flow, input the total duration of the signal to analyze)
Ts = input('Period (unsteady) or Max duration (steady) of input signal (in seconds) = ');
for i = 1 : numel(X)
    for j = 1 : numel(Y)
        for k = 1 : numel(Z)
            % New vectors T1,T2,T3,V1,V2,V3
            V1{i,j,k} = V1{i,j,k}(find(T1{i,j,k} >= 0 & T1{i,j,k} < Ts));
            V2{i,j,k} = V2{i,j,k}(find(T2{i,j,k} >= 0 & T2{i,j,k} < Ts));
            V3{i,j,k} = V3{i,j,k}(find(T3{i,j,k} >= 0 & T3{i,j,k} < Ts));
            T1{i,j,k} = T1{i,j,k}(find(T1{i,j,k} >= 0 & T1{i,j,k} < Ts));
            T2{i,j,k} = T2{i,j,k}(find(T2{i,j,k} >= 0 & T2{i,j,k} < Ts));
            T3{i,j,k} = T3{i,j,k}(find(T3{i,j,k} >= 0 & T3{i,j,k} < Ts));

            % New zeroT positons and number of cycles
            zeroT_pos_T1{i,j,k} = [1 ; find(T1{i,j,k}>circshift(T1{i,j,k},-1))+1];
            zeroT_pos_T2{i,j,k} = [1 ; find(T2{i,j,k}>circshift(T2{i,j,k},-1))+1];
            zeroT_pos_T3{i,j,k} = [1 ; find(T3{i,j,k}>circshift(T3{i,j,k},-1))+1];

            n_cycles_T1(i,j,k) = numel(zeroT_pos_T1{i,j,k})-1;
            n_cycles_T2(i,j,k) = numel(zeroT_pos_T2{i,j,k})-1;
            n_cycles_T3(i,j,k) = numel(zeroT_pos_T3{i,j,k})-1;
```

```
        end
    end
end

% Number of cycles to analyze (for steady flow, usually input "1")
Nc = input('Max number of cycles to analyze (for steady flow =1) = ');
for i = 1 : numel(X)
    for j = 1 : numel(Y)
        for k = 1 : numel(Z)
            % New vectors T1,T2,T3,V1,V2,V3
            V1{i,j,k} =
V1{i,j,k}(zeroT_pos_T1{i,j,k}(1):zeroT_pos_T1{i,j,k}(min([n_cycles_T1(i,j,k),Nc])+1)-1);
            V2{i,j,k} =
V2{i,j,k}(zeroT_pos_T2{i,j,k}(1):zeroT_pos_T2{i,j,k}(min([n_cycles_T2(i,j,k),Nc])+1)-1);
            V3{i,j,k} =
V3{i,j,k}(zeroT_pos_T3{i,j,k}(1):zeroT_pos_T3{i,j,k}(min([n_cycles_T3(i,j,k),Nc])+1)-1);
            T1{i,j,k} =
T1{i,j,k}(zeroT_pos_T1{i,j,k}(1):zeroT_pos_T1{i,j,k}(min([n_cycles_T1(i,j,k),Nc])+1)-1);
            T2{i,j,k} =
T2{i,j,k}(zeroT_pos_T2{i,j,k}(1):zeroT_pos_T2{i,j,k}(min([n_cycles_T2(i,j,k),Nc])+1)-1);
            T3{i,j,k} =
T3{i,j,k}(zeroT_pos_T3{i,j,k}(1):zeroT_pos_T3{i,j,k}(min([n_cycles_T3(i,j,k),Nc])+1)-1);

            % New zeroT positons and number of cycles
            zeroT_pos_T1{i,j,k} = [1 ; find(T1{i,j,k}>circshift(T1{i,j,k},-1))+1];
            zeroT_pos_T2{i,j,k} = [1 ; find(T2{i,j,k}>circshift(T2{i,j,k},-1))+1];
            zeroT_pos_T3{i,j,k} = [1 ; find(T3{i,j,k}>circshift(T3{i,j,k},-1))+1];

            n_cycles_T1(i,j,k) = numel(zeroT_pos_T1{i,j,k})-1;
            n_cycles_T2(i,j,k) = numel(zeroT_pos_T2{i,j,k})-1;
            n_cycles_T3(i,j,k) = numel(zeroT_pos_T3{i,j,k})-1;
        end
    end
end

% Remove individual cycles
p = menu('Remove any individual cycles?','Yes','No');

if p == 1

    disp(' ');
    disp('Analysing individual cycles:');

    n_cycles_T = zeros(numel(X),numel(Y),numel(Z));

    figure();
    hp1 = subplot(3,1,1);
    hp2 = subplot(3,1,2);
    hp3 = subplot(3,1,3);
    linkaxes([hp1 hp2 hp3],'x');
    set(gcf,'position',[1 80 1024 600]);          % Maximizes figure window

    % Loop through all files and plot the data
    for i = 1 : numel(X)
        for j = 1 : numel(Y)
            for k = 1 : numel(Z)
```

```
                t1p = T1{i,j,k};
                v1p = V1{i,j,k};
                cla(hp1);
                plot(hp1,t1p,v1p,'.g','MarkerSize',1);
                legend(hp1,'V1','Location','Best');
                title(hp1,['Data selection at z = ', num2str(Z(k)*1000), ' mm']);
                ylabel(hp1,'Velocity (m/s)');

                t2p = T2{i,j,k};
                v2p = V2{i,j,k};
                cla(hp2);
                plot(hp2,t2p,v2p,'.b','MarkerSize',1);
                legend(hp2,'V2','Location','Best');
                ylabel(hp2,'Velocity (m/s)');

                t3p = T3{i,j,k};
                v3p = V3{i,j,k};
                cla(hp3);
                plot(hp3,t3p,v3p,'.m','MarkerSize',1);
                legend(hp3,'V3','Location','Best');
                xlabel(hp3,'Time (s)');
                xlim([0 Ts]);
                ylabel(hp3,'Velocity (m/s)');

                r = input(['Plotting: Z_case = ', num2str(k),'. Remove any cycles at this location?
(Yes=1 , No=0): ']);
                while r == 1
                    n_cycles_T(i,j,k) = max([n_cycles_T1(i,j,k) n_cycles_T2(i,j,k) n_cycles_T3(i,j,k)]);
% Max number of cycles in any of the 3 channels, for this location (X,Y,Z)
                    c = 0;      % Count of number of cycles deleted at this location

                    for nc = 1 : n_cycles_T(i,j,k)

                        nc_1 = min([nc , n_cycles_T1(i,j,k)+c]);              % In case the 3 channels
don't have the same number of cycles
                        t1p = T1{i,j,k}(zeroT_pos_T1{i,j,k}(1):zeroT_pos_T1{i,j,k}(nc_1+1-c)-1);
                        v1p = V1{i,j,k}(zeroT_pos_T1{i,j,k}(1):zeroT_pos_T1{i,j,k}(nc_1+1-c)-1);
                        cla(hp1);
                        plot(hp1,t1p,v1p,'.g','MarkerSize',1);
                        legend(hp1,'V1','Location','Best');
                        title(hp1,['Data selection at z = ', num2str(Z(k)*1000), ' mm']);
                        ylabel(hp1,'Velocity (m/s)');

                        nc_2 = min([nc , n_cycles_T2(i,j,k)+c]);              % In case the 3 channels
don't have the same number of cycles
                        t2p = T2{i,j,k}(zeroT_pos_T2{i,j,k}(1):zeroT_pos_T2{i,j,k}(nc_2+1-c)-1);
                        v2p = V2{i,j,k}(zeroT_pos_T2{i,j,k}(1):zeroT_pos_T2{i,j,k}(nc_2+1-c)-1);
                        cla(hp2);
                        plot(hp2,t2p,v2p,'.b','MarkerSize',1);
                        legend(hp2,'V2','Location','Best');
                        ylabel(hp2,'Velocity (m/s)');

                        nc_3 = min([nc , n_cycles_T3(i,j,k)+c]);              % In case the 3 channels
don't have the same number of cycles
                        t3p = T3{i,j,k}(zeroT_pos_T3{i,j,k}(1):zeroT_pos_T3{i,j,k}(nc_3+1-c)-1);
```

```
                        v3p = V3{i,j,k}(zeroT_pos_T3{i,j,k}(1):zeroT_pos_T3{i,j,k}(nc_3+1-c)-1);
                        cla(hp3);
                        plot(hp3,t3p,v3p,'.m','MarkerSize',1);
                        legend(hp3,'V3','Location','Best');
                        xlabel(hp3,'Time (s)');
                        xlim([0 Ts]);
                        ylabel(hp3,'Velocity (m/s)');

                        rr = input(['Plotting: Z_case = ', num2str(k), ', nc = ', num2str(nc),'. Remove
this cycle? (Yes=1 , No=0): ']);
                        if rr == 1
                            % New T1,T2,T3 and V1,V2,V3 vectors
                            if nc <= nc_1; V1{i,j,k}(zeroT_pos_T1{i,j,k}(nc-c):zeroT_pos_T1{i,j,k}(nc+1-
c)-1) = []; end;
                            if nc <= nc_2; V2{i,j,k}(zeroT_pos_T2{i,j,k}(nc-c):zeroT_pos_T2{i,j,k}(nc+1-
c)-1) = []; end;
                            if nc <= nc_3; V3{i,j,k}(zeroT_pos_T3{i,j,k}(nc-c):zeroT_pos_T3{i,j,k}(nc+1-
c)-1) = []; end;
                            if nc <= nc_1; T1{i,j,k}(zeroT_pos_T1{i,j,k}(nc-c):zeroT_pos_T1{i,j,k}(nc+1-
c)-1) = []; end;
                            if nc <= nc_2; T2{i,j,k}(zeroT_pos_T2{i,j,k}(nc-c):zeroT_pos_T2{i,j,k}(nc+1-
c)-1) = []; end;
                            if nc <= nc_3; T3{i,j,k}(zeroT_pos_T3{i,j,k}(nc-c):zeroT_pos_T3{i,j,k}(nc+1-
c)-1) = []; end;

                            % New zeroT positons and number of cycles
                            zeroT_pos_T1{i,j,k} = [1 ; find(T1{i,j,k}>circshift(T1{i,j,k},-1))+1];
                            zeroT_pos_T2{i,j,k} = [1 ; find(T2{i,j,k}>circshift(T2{i,j,k},-1))+1];
                            zeroT_pos_T3{i,j,k} = [1 ; find(T3{i,j,k}>circshift(T3{i,j,k},-1))+1];

                            n_cycles_T1(i,j,k) = numel(zeroT_pos_T1{i,j,k})-1;
                            n_cycles_T2(i,j,k) = numel(zeroT_pos_T2{i,j,k})-1;
                            n_cycles_T3(i,j,k) = numel(zeroT_pos_T3{i,j,k})-1;

                            c = c+1;      % Count of number of cycles deleted at this location
                        end

                    end     % Loop in number of cycles (nc)

                    rrr = 0;
                    while (isempty(rrr) || rrr==0)                          % Loop to allow repeat at
the same location
                        rrr = input('Continue to next location? (Yes=1 , Repeat=2): ');
                        if rrr == 1; r = 0; end;
                    end

                end         % end of if statement to remove any cycles at this location

            end
        end
    end         % Close loop through cases
    disp('End of analysing individual cycles');
    set(gcf,'position',[228 228 560 420]);          % Back to original size in left window
end

% Remove bad points from reflections/saturation manually
```

```
% (based on data rate)
p = menu('Remove bad points from reflections/saturation?','Yes','No');

if p == 1

    disp(' ');
    disp('Analysing bad points from reflections/saturation:');

    hf = figure();
    hp1 = subplot(3,1,1);
    hp2 = subplot(3,1,2);
    hp3 = subplot(3,1,3);
    linkaxes([hp1 hp2 hp3],'x');
    set(gcf,'position',[1 80 1024 600]);          % Maximizes figure window

    % Loop through all files and plot the data
    for i = 1 : numel(X)
        for j = 1 : numel(Y)
            for k = 1 : numel(Z)

                t1p = T1{i,j,k};
                v1p = V1{i,j,k};
                d1p = abs(1./(T1{i,j,k}-circshift(T1{i,j,k},1)));
                cla(hp1);
                plot(hp1,t1p,v1p,'.g','MarkerSize',1);
                legend(hp1,'V1','Location','Best');
                title(hp1,['Data selection at z = ', num2str(Z(k)*1000), ' mm']);
                ylabel(hp1,'Velocity (m/s)');

                t2p = T2{i,j,k};
                v2p = V2{i,j,k};
                d2p = abs(1./(T2{i,j,k}-circshift(T2{i,j,k},1)));
                cla(hp2);
                plot(hp2,t2p,v2p,'.b','MarkerSize',1);
                legend(hp2,'V2','Location','Best');
                ylabel(hp2,'Velocity (m/s)');

                t3p = T3{i,j,k};
                v3p = V3{i,j,k};
                d3p = abs(1./(T3{i,j,k}-circshift(T3{i,j,k},1)));
                cla(hp3);
                plot(hp3,t3p,v3p,'.m','MarkerSize',1);
                legend(hp3,'V3','Location','Best');
                xlabel(hp3,'Time (s)');
                xlim(hp3,[0 Ts]);
                ylabel(hp3,'Velocity (m/s)');

                r = input(['Plotting: Z_case = ', num2str(k),'. Remove any data points at this location?
(Yes=1 , No=0): ']);
                if r == 1

                    % New figure with velocities and data rates for one channel
                    hff = figure();
                    hpv = subplot(2,1,1);
                    hpd = subplot(2,1,2);
                    linkaxes([hpv hpd],'x');
```

```
                    set(gcf,'position',[1 80 1024 600]);          % Maximizes figure window

                    for ch = 1 : 3          % Loop in each channel, with new plot of velocities and data
rates together only for that channel

                        if ch==1; tp = t1p; vp = v1p; dp = d1p; sp = '.g'; end;    % tp =
eval(['t',num2str(ch),'p']);           % Time variable
                        if ch==2; tp = t2p; vp = v2p; dp = d2p; sp = '.b'; end;    % vp =
eval(['v',num2str(ch),'p']);           % Velocity variable
                        if ch==3; tp = t3p; vp = v3p; dp = d3p; sp = '.m'; end;    % dp =
eval(['d',num2str(ch),'p']);           % Data rate variable

                        cla(hpv); hold(hpv,'off');
                        plot(hpv,tp,vp,sp,'MarkerSize',1); hold(hpv,'on');
                        legend(hpv,['V',num2str(ch)],'Location','Best');
                        ylabel(hpv,'Velocity (m/s)');
                        title(hpv,['Data selection at z = ', num2str(Z(k)*1000), ' mm']);

                        cla(hpd); hold(hpd,'off');
                        plot(hpd,tp,dp,sp,'MarkerSize',10); hold(hpd,'on');
                        legend(hpd,['Ch. ',num2str(ch)],'Location','Best');
                        xlabel(hpd,'Time (s)');
                        xlim([0 Ts]);
                        ylabel(hpd,'Data rate (Hz)');

                        % Remove points using automatic detection algorithm
                        aa = input('Show auto detection of bad points? (Yes=1 , No=0): ');
                        while aa == 1
                            % Percentile plot
                            hfff = figure;
                            plot([0:1:100],prctile(dp,[0:1:100]));
                            xlabel('Percentile (%)');
                            ylabel('Data Rate (Hz)');

                            dr_pct = 50;            % Percentile of the data rate to apply auto
detection of bad points
                            n_sum = 10;              % Number of adjacent points that need to be over
the percentile to be identified as bad points
                            aaa = 0;
                            while aaa == 0           % Loop to be able to adjust parameters multiple
times for the same channel
                                % Thresholding criteria
                                dp_overpct = dp > prctile(dp,dr_pct);
                                dp_overpct_sum = zeros(size(dp));
                                for n = 1 : n_sum
                                    dp_overpct_sum = dp_overpct_sum + circshift(dp_overpct,-(n-1));
                                end
                                dp_badpoints = zeros(size(dp));
                                for n = 1 : n_sum
                                    dp_badpoints = dp_badpoints | circshift(dp_overpct_sum==n_sum,n-1);
                                end
                                dp_badpoints = dp_badpoints & dp_overpct;

                                % Show potential bad points in the graph
                                figure(hff);
                                hpva = plot(hpv,tp(dp_badpoints),vp(dp_badpoints),'sr','MarkerSize',2);
```

```
                                hpda = plot(hpd,tp(dp_badpoints),dp(dp_badpoints),'sr','MarkerSize',10);

                                ppp = input(['Percentile = ',num2str(dr_pct),'%, Adjacent points =
',num2str(n_sum),'. Try other parameters? (Yes=1 , No=0): ']);
                                if ppp == 1
                                    figure(hfff);
                                    dr_pct = input('Enter percentile of data rate (0-100): '); if
isempty(dr_pct); dr_pct = 50; end;
                                    n_sum = input('Enter minimum number of adjacent points: '); if
isempty(n_sum); n_sum = 10; end;
                                    delete([hpva,hpda]);        % Delete auto detection symbols in plot
                                else
                                    aaa = 1;
                                    apply_auto = input('Delete points from auto detection? (Yes=1 ,
No=0): ');
                                end
                            end

                            if apply_auto == 1
                                % Modify variables, writing NaN where 'brushedData' was not NaN
(selected point)
                                tp(dp_badpoints) = NaN;
                                vp(dp_badpoints) = NaN;
                                dp(dp_badpoints) = NaN;

                                % Update plot
                                cla(hpv);
                                plot(hpv,tp,vp,sp,'MarkerSize',1); hold(hpv,'on');
                                legend(hpv,['V',num2str(ch)],'Location','Best');
                                ylabel(hpv,'Velocity (m/s)');
                                title(hpv,['Data selection at z = ', num2str(Z(k)*1000), ' mm']);

                                cla(hpd);
                                plot(hpd,tp,dp,sp,'MarkerSize',10); hold(hpd,'on');
                                legend(hpd,['Ch. ',num2str(ch)],'Location','Best');
                                xlabel(hpd,'Time (s)');
                                %xlim(hpd,[0 Ts]);
                                ylabel(hpd,'Data rate (Hz)');
                            else
                                delete([hpva,hpda]);        % Delete auto detection symbols in plot
                            end
                            close(hfff);    % Close percentile plot
                            aa_input = input('Repeat auto detection of bad points? (Yes=1 , No=0): ');
                            if aa_input == 1; aa = 1; else aa = 0; end;  % Exit condition of while loop
                        end         % End of removing points automatically

                        % Remove points manually
                        rr = input('Remove points manually? (Yes=1 , No=0): ');
                        if rr == 1

                            hpva = [];
                            hpda = [];
                            rrr = 0;
                            while rrr == 0                                          % Loop to be able
to brush multiple times at the same location
                                % Select points manually from the graph
```

```
                                brush(hff,'on');
                                rrr_input = input('Manually SELECT data points to delete. When done
choose (Show=0, Delete=1, Continue=2): ');
                                if rrr_input==2; rrr = 1; end;      % Exit condition of while loop

                                % Get the data points selected
                                hBrushLineV = findall(hpv,'tag','Brushing');
                                hBrushLineD = findall(hpd,'tag','Brushing');

                                if isempty(hBrushLineV); brushedDataV = {NaN(1,numel(tp)) ,
NaN(1,numel(vp))};            % When no points are brushed for this channel 'brushedData' is all NaNs.
                                else                          brushedDataV = get(hBrushLineV,
{'Xdata','Ydata'}); end;
                                if isempty(hBrushLineD); brushedDataD = {NaN(1,numel(tp)) ,
NaN(1,numel(dp))};            % When no points are brushed for this channel 'brushedData' is all NaNs.
                                else                          brushedDataD = get(hBrushLineD,
{'Xdata','Ydata'}); end;

                                brush(hff,'off');

                                % Show selection in the other graph not brushed
                                delete([hpva,hpda]);        % Delete previous selection in the other
graph not brushed
                                hpva =
plot(hpv,tp(~isnan(brushedDataD{1})),vp(~isnan(brushedDataD{2})),'sr','MarkerSize',2);
                                hpda =
plot(hpd,tp(~isnan(brushedDataV{1})),dp(~isnan(brushedDataV{2})),'sr','MarkerSize',10);

                                if rrr_input == 1             % Delete selected points and redraw
                                    % Modify variables, writing NaN where 'brushedData' was not NaN
(selected point)
                                    tp(~isnan(brushedDataV{1})) = NaN; tp(~isnan(brushedDataD{1})) =
NaN;
                                    vp(~isnan(brushedDataV{2})) = NaN; vp(~isnan(brushedDataD{2})) =
NaN;
                                    dp(~isnan(brushedDataV{2})) = NaN; dp(~isnan(brushedDataD{2})) =
NaN;

                                    % Update plot
                                    cla(hpv);
                                    plot(hpv,tp,vp,sp,'MarkerSize',1); hold(hpv,'on');
                                    legend(hpv,['V',num2str(ch)],'Location','Best');
                                    ylabel(hpv,'Velocity (m/s)');
                                    title(hpv,['Data selection at z = ', num2str(Z(k)*1000), ' mm']);

                                    cla(hpd);
                                    plot(hpd,tp,dp,sp,'MarkerSize',10); hold(hpd,'on');
                                    legend(hpd,['Ch. ',num2str(ch)],'Location','Best');
                                    xlabel(hpd,'Time (s)');
                                    %xlim(hpd,[0 Ts]);
                                    ylabel(hpd,'Data rate (Hz)');

                                    hpda = [];
                                    hpva = [];
                                end
                            end
```

```
                        end         % End of removing points manually

                        % Assign results back to each channel
                        if ch==1; t1p = tp; v1p = vp; d1p = dp; end;
                        if ch==2; t2p = tp; v2p = vp; d2p = dp; end;
                        if ch==3; t3p = tp; v3p = vp; d3p = dp; end;

                    end          % End of loop for each channel

                    close(hff);     % Close detailed figure

                    % Count number of data points deleted
                    c1 = numel(find(isnan(v1p)));
                    c2 = numel(find(isnan(v2p)));
                    c3 = numel(find(isnan(v3p)));

                    % Delete all selected values (which became NaN) from the variables
                    % New T1,T2,T3 and V1,V2,V3 vectors
                    V1{i,j,k}(isnan(v1p)) = [];
                    V2{i,j,k}(isnan(v2p)) = [];
                    V3{i,j,k}(isnan(v3p)) = [];
                    T1{i,j,k}(isnan(t1p)) = [];
                    T2{i,j,k}(isnan(t2p)) = [];
                    T3{i,j,k}(isnan(t3p)) = [];

                    % New zeroT positons and number of cycles
                    zeroT_pos_T1{i,j,k} = [1 ; find(T1{i,j,k}>circshift(T1{i,j,k},-1))+1];
                    zeroT_pos_T2{i,j,k} = [1 ; find(T2{i,j,k}>circshift(T2{i,j,k},-1))+1];
                    zeroT_pos_T3{i,j,k} = [1 ; find(T3{i,j,k}>circshift(T3{i,j,k},-1))+1];

                    n_cycles_T1(i,j,k) = numel(zeroT_pos_T1{i,j,k})-1;
                    n_cycles_T2(i,j,k) = numel(zeroT_pos_T2{i,j,k})-1;
                    n_cycles_T3(i,j,k) = numel(zeroT_pos_T3{i,j,k})-1;

                    % Display confirmation message of data points deleted
                    disp(['Deleted data points: ch.1 = ',num2str(c1),', ch.2 = ',num2str(c2),', ch.3 =
',num2str(c3)]);

                end         % end of if statement to remove any data points at this location

            end
        end
    end         % Close loop through cases
    disp('End of analysing bad points from reflections/saturation');
    set(gcf,'position',[228 228 560 420]);          % Back to original size in left window
end

% Remove spikes and outliers manually
p = menu('Remove any individual data points (outliers)?','Yes','No');

if p == 1

    disp(' ');
    disp('Analysing individual data points:');

    hf = figure();
```

```
    hp1 = subplot(3,1,1);
    hp2 = subplot(3,1,2);
    hp3 = subplot(3,1,3);
    linkaxes([hp1 hp2 hp3],'x');
    set(gcf,'position',[1 80 1024 600]);          % Maximizes figure window

    % Loop through all files and plot the data
    for i = 1 : numel(X)
        for j = 1 : numel(Y)
            for k = 1 : numel(Z)

                t1p = T1{i,j,k};
                v1p = V1{i,j,k};
                cla(hp1);
                plot(hp1,t1p,v1p,'.g','MarkerSize',10);
                legend(hp1,'V1','Location','Best');
                title(hp1,['Data selection at z = ', num2str(Z(k)*1000), ' mm']);
                ylabel(hp1,'Velocity (m/s)');

                t2p = T2{i,j,k};
                v2p = V2{i,j,k};
                cla(hp2);
                plot(hp2,t2p,v2p,'.b','MarkerSize',10);
                legend(hp2,'V2','Location','Best');
                ylabel(hp2,'Velocity (m/s)');

                t3p = T3{i,j,k};
                v3p = V3{i,j,k};
                cla(hp3);
                plot(hp3,t3p,v3p,'.m','MarkerSize',10);
                legend(hp3,'V3','Location','Best');
                xlabel(hp3,'Time (s)');
                xlim([0 Ts]);
                ylabel(hp3,'Velocity (m/s)');

                r = input(['Plotting: Z_case = ', num2str(k),'. Remove any data points at this location?
(Yes=1 , No=0): ']);
                if r == 1

                    rr = 0;
                    while rr == 0                                          % Loop to be able to brush
multiple times at the same location
                        % Select points manually from the graph
                        brush on;
                        rr_input = input('Manually SELECT data points to be deleted from the graph. When
done choose (Redraw=0, Continue=1): ');
                        if rr_input == 1; rr = 1; end;

                        % Get the data points selected
                        hBrushLine1 = findall(hp1,'tag','Brushing');
                        hBrushLine2 = findall(hp2,'tag','Brushing');
                        hBrushLine3 = findall(hp3,'tag','Brushing');

                        if isempty(hBrushLine1); brushedData1 = {NaN(1,numel(t1p)) , NaN(1,numel(v1p))};
% When no points are brushed for this channel 'brushedData' is all NaNs.
```

```
                        else                        brushedData1 = get(hBrushLine1, {'Xdata','Ydata'});
end;
                        if isempty(hBrushLine2); brushedData2 = {NaN(1,numel(t2p)) , NaN(1,numel(v2p))};
% When no points are brushed for this channel 'brushedData' is all NaNs.
                        else                        brushedData2 = get(hBrushLine2, {'Xdata','Ydata'});
end;
                        if isempty(hBrushLine3); brushedData3 = {NaN(1,numel(t3p)) , NaN(1,numel(v3p))};
% When no points are brushed for this channel 'brushedData' is all NaNs.
                        else                        brushedData3 = get(hBrushLine3, {'Xdata','Ydata'});
end;

                        % Remove the data points from the plot
                        RemovePointsCallback = @datamanager.dataEdit;
                        RemovePointsCallback(hf,[],'replace',NaN);     % Writes NaNs to the brushed
points

                        brush off;

                        % Modify variables, writing NaN where 'brushedData' was not NaN (selected point)
                        t1p(~isnan(brushedData1{1})) = NaN;
                        t2p(~isnan(brushedData2{1})) = NaN;
                        t3p(~isnan(brushedData3{1})) = NaN;
                        v1p(~isnan(brushedData1{2})) = NaN;
                        v2p(~isnan(brushedData2{2})) = NaN;
                        v3p(~isnan(brushedData3{2})) = NaN;
                    end

                    % Count number of data points deleted
                    c1 = numel(find(isnan(v1p)));
                    c2 = numel(find(isnan(v2p)));
                    c3 = numel(find(isnan(v3p)));

                    % Delete all selected values (which became NaN) from the variables
                    % New T1,T2,T3 and V1,V2,V3 vectors
                    V1{i,j,k}(isnan(v1p)) = [];
                    V2{i,j,k}(isnan(v2p)) = [];
                    V3{i,j,k}(isnan(v3p)) = [];
                    T1{i,j,k}(isnan(t1p)) = [];
                    T2{i,j,k}(isnan(t2p)) = [];
                    T3{i,j,k}(isnan(t3p)) = [];

                    % New zeroT positons and number of cycles
                    zeroT_pos_T1{i,j,k} = [1 ; find(T1{i,j,k}>circshift(T1{i,j,k},-1))+1];
                    zeroT_pos_T2{i,j,k} = [1 ; find(T2{i,j,k}>circshift(T2{i,j,k},-1))+1];
                    zeroT_pos_T3{i,j,k} = [1 ; find(T3{i,j,k}>circshift(T3{i,j,k},-1))+1];

                    n_cycles_T1(i,j,k) = numel(zeroT_pos_T1{i,j,k})-1;
                    n_cycles_T2(i,j,k) = numel(zeroT_pos_T2{i,j,k})-1;
                    n_cycles_T3(i,j,k) = numel(zeroT_pos_T3{i,j,k})-1;

                    % Display confirmation message of data points deleted
                    disp(['Deleted data points: ch.1 = ',num2str(c1),', ch.2 = ',num2str(c2),', ch.3 =
',num2str(c3)]);

                end         % end of if statement to remove any data points at this location
```

```
            end
        end
    end         % Close loop through cases
    disp('End of analysing individual data points');
    set(gcf,'position',[228 228 560 420]);          % Back to original size in left window
end

% Shift data
% Make starting point of cycle to be at a particular phase (usually for oscillatory flow to correct
trigger sync pulse delay)
p = menu('Shift data to correct trigger sync pulse delay?','Yes','No');

if p == 1

    disp(' ');
    disp('Trigger sync pulse delay correction:');

    ch = 0;
    while ch==0
        ch = input('Select reference channel for sync pulse (1,2,3): ');
        if isempty(ch) || ischar(ch) || ((ch~=1) && (ch~=2) && (ch~=3))
            disp('Error: wrong channel input.');
            ch = 0;                             % The input channel statement keeps being displayed
until it gets a correct value (1,2,3)
        end
    end
    T_shift1 = eval(['T', num2str(ch)]);
    V_shift1 = eval(['V', num2str(ch)]);

    Zout = input('Set distance from wall to consider outer flow conditions (in meters): '); %0.1;  %
Distance from wall to consider outer flow conditions (m)

    for i = 1 : numel(X)
        for j = 1 : numel(Y)

            % Calculate phase shift for each Z location in the outer flow
            Zout_loc = find(Z>=Zout);
            hw = waitbar(0,'Calculating phase shift ...'); set(hw,'Name',[num2str(round(0*100)),' %']);
drawnow;
            for k = Zout_loc(1) : Zout_loc(end)
                [cfit1,gof1] = fit(T_shift1{i,j,k},V_shift1{i,j,k},'sin8');
                x_fit1 = [0:Ts/10000:Ts];
                y_fit1(k,:) = cfit1(x_fit1);
                [max_value,max_loc] = max(y_fit1(k,:));
                time_shift(k) = x_fit1(max_loc)-0;      % Taking t=0 as reference

                pw = (k-Zout_loc(1)+1)/(Zout_loc(end)-Zout_loc(1)+1);
                waitbar(pw,hw); set(hw,'Name',[num2str(round(pw*100)),' %']); drawnow;
            end
            close(hw);

            % Plot of original data before phase shift
            hf = figure();
            for k = Zout_loc(1) : Zout_loc(end)
                hp(k) = plot(T_shift1{i,j,k},V_shift1{i,j,k},'.c','MarkerSize',1); hold on;
            end
```

```
            hp_group = hggroup; set(hp(Zout_loc(1):Zout_loc(end)),'Parent',hp_group);
set(get(get(hp_group,'Annotation'),'LegendInformation'),'IconDisplayStyle','on');    % Groups all legend
entries into one

            y_fit_mean = mean(y_fit1(Zout_loc(1):Zout_loc(end),:),1);
            plot(x_fit1,y_fit_mean,'-b');
            title('Analysis of phase shift due to trigger sync pulse');
            xlabel('Time (s)');
            ylabel('Velocity (m/s)');
            xlim([0 Ts]);
            y_limits_auto = get(gca,'YLim');

            % Calculate mean phase shift
            if (max(time_shift(Zout_loc(1):Zout_loc(end)))-
min(time_shift(Zout_loc(1):Zout_loc(end))))>Ts/2     % To avoid errors with having values of time shift
at +360°
                time_shift(time_shift>Ts/2) = time_shift(time_shift>Ts/2) - Ts;
            end

            time_shift_mean = mean(time_shift(Zout_loc(1):Zout_loc(end)));
            phase_shift_mean = time_shift_mean/Ts*360;

            disp(['Trigger pulse phase shift = ', num2str(phase_shift_mean) , '°']);

            % Apply phase shift
            c = input('Apply trigger pulse phase shift correction? (No=0, Yes=1, Custom=2): ');

            if c==2
                phase_shift_mean = input('Enter custom value for phase shift (deg): ');
                time_shift_mean = phase_shift_mean*Ts/360;
            end

            if c==1 || c==2     % Correct for trigger phase shift
                for k = 1 : numel(Z)
                    % New vectors T1,T2,T3 (V1,V2,V3 don't change)
                    T1{i,j,k} = T1{i,j,k} - time_shift_mean;  T1{i,j,k}(T1{i,j,k}<0) =
T1{i,j,k}(T1{i,j,k}<0) + Ts;  T1{i,j,k}(T1{i,j,k}>=Ts) = T1{i,j,k}(T1{i,j,k}>=Ts) - Ts;
                    T2{i,j,k} = T2{i,j,k} - time_shift_mean;  T2{i,j,k}(T2{i,j,k}<0) =
T2{i,j,k}(T2{i,j,k}<0) + Ts;  T2{i,j,k}(T2{i,j,k}>=Ts) = T2{i,j,k}(T2{i,j,k}>=Ts) - Ts;
                    T3{i,j,k} = T3{i,j,k} - time_shift_mean;  T3{i,j,k}(T3{i,j,k}<0) =
T3{i,j,k}(T3{i,j,k}<0) + Ts;  T3{i,j,k}(T3{i,j,k}>=Ts) = T3{i,j,k}(T3{i,j,k}>=Ts) - Ts;

                    % New zeroT positons and number of cycles
                    zeroT_pos_T1{i,j,k} = [1 ; find(T1{i,j,k}>circshift(T1{i,j,k},-1))+1];
                    zeroT_pos_T2{i,j,k} = [1 ; find(T2{i,j,k}>circshift(T2{i,j,k},-1))+1];
                    zeroT_pos_T3{i,j,k} = [1 ; find(T3{i,j,k}>circshift(T3{i,j,k},-1))+1];

                    n_cycles_T1(i,j,k) = numel(zeroT_pos_T1{i,j,k})-1;
                    n_cycles_T2(i,j,k) = numel(zeroT_pos_T2{i,j,k})-1;
                    n_cycles_T3(i,j,k) = numel(zeroT_pos_T3{i,j,k})-1;

                end

                % Calculate new fit line after shift
                T_shift2 = eval(['T', num2str(ch)]);
                V_shift2 = eval(['V', num2str(ch)]);
```

```
                hw = waitbar(0,'Calculating new fit ...'); set(hw,'Name',[num2str(round(0*100)),' %']);
drawnow;
                for k = Zout_loc(1) : Zout_loc(end)
                    [cfit2,gof2] = fit(T_shift2{i,j,k},V_shift2{i,j,k},'sin8');
                    x_fit2 = [0:Ts/10000:Ts];
                    y_fit2(k,:) = cfit2(x_fit2);

                    pw = (k-Zout_loc(1)+1)/(Zout_loc(end)-Zout_loc(1)+1);
                    waitbar(pw,hw); set(hw,'Name',[num2str(round(pw*100)),' %']); drawnow;
                end
                close(hw);

                % Plot of data after phase shift
                figure(hf);
                y_fit_mean = mean(y_fit2(Zout_loc(1):Zout_loc(end),:),1);
                plot(x_fit2,y_fit_mean,'-r');
                legend('Data before shift','Fit before shift','Fit after shift');
                xlim([0 Ts]);
                ylim(y_limits_auto);
                file_save = [dir_name,'\Trigger pulse phase shift correction','.jpg'];
                saveas(gcf,file_save,'jpg');

            else
               disp('No trigger pulse phase shift correction was applied.');
            end

        end
    end
end

% Remove cycles with only one data point
% (to avoid problems in the interpolation routine later on)
for i = 1 : numel(X)
    for j = 1 : numel(Y)
        for k = 1 : numel(Z)
            % New vectors T1,T2,T3,V1,V2,V3
            V1{i,j,k}(zeroT_pos_T1{i,j,k}(find(zeroT_pos_T1{i,j,k}-circshift(zeroT_pos_T1{i,j,k},-1)==-
1))) = [];
            V2{i,j,k}(zeroT_pos_T2{i,j,k}(find(zeroT_pos_T2{i,j,k}-circshift(zeroT_pos_T2{i,j,k},-1)==-
1))) = [];
            V3{i,j,k}(zeroT_pos_T3{i,j,k}(find(zeroT_pos_T3{i,j,k}-circshift(zeroT_pos_T3{i,j,k},-1)==-
1))) = [];
            T1{i,j,k}(zeroT_pos_T1{i,j,k}(find(zeroT_pos_T1{i,j,k}-circshift(zeroT_pos_T1{i,j,k},-1)==-
1))) = [];
            T2{i,j,k}(zeroT_pos_T2{i,j,k}(find(zeroT_pos_T2{i,j,k}-circshift(zeroT_pos_T2{i,j,k},-1)==-
1))) = [];
            T3{i,j,k}(zeroT_pos_T3{i,j,k}(find(zeroT_pos_T3{i,j,k}-circshift(zeroT_pos_T3{i,j,k},-1)==-
1))) = [];

            % New zeroT positons and number of cycles
            zeroT_pos_T1{i,j,k} = [1 ; find(T1{i,j,k}>circshift(T1{i,j,k},-1))+1];
            zeroT_pos_T2{i,j,k} = [1 ; find(T2{i,j,k}>circshift(T2{i,j,k},-1))+1];
            zeroT_pos_T3{i,j,k} = [1 ; find(T3{i,j,k}>circshift(T3{i,j,k},-1))+1];

            n_cycles_T1(i,j,k) = numel(zeroT_pos_T1{i,j,k})-1;
```

```
            n_cycles_T2(i,j,k) = numel(zeroT_pos_T2{i,j,k})-1;
            n_cycles_T3(i,j,k) = numel(zeroT_pos_T3{i,j,k})-1;
        end
    end
end

% Check for empty variables
% If T1,T2,T3,V1,V2,V3 became empty at any locations after all the data
% selection process, we need to give some dummy value for 1 cycle to avoid
% problems in the following routines.
% It won't affect the results since the even-time interpolation routine
% will give NaNs at these locations.
for i = 1 : numel(X)
    for j = 1 : numel(Y)
        for k = 1 : numel(Z)
            % New vectors T1,T2,T3,V1,V2,V3
            if isempty(T1{i,j,k}); T1{i,j,k} = [0;Ts]; V1{i,j,k} = [0;0]; end;
            if isempty(T2{i,j,k}); T2{i,j,k} = [0;Ts]; V2{i,j,k} = [0;0]; end;
            if isempty(T3{i,j,k}); T3{i,j,k} = [0;Ts]; V3{i,j,k} = [0;0]; end;

            % New zeroT positons and number of cycles
            zeroT_pos_T1{i,j,k} = [1 ; find(T1{i,j,k}>circshift(T1{i,j,k},-1))+1];
            zeroT_pos_T2{i,j,k} = [1 ; find(T2{i,j,k}>circshift(T2{i,j,k},-1))+1];
            zeroT_pos_T3{i,j,k} = [1 ; find(T3{i,j,k}>circshift(T3{i,j,k},-1))+1];

            n_cycles_T1(i,j,k) = numel(zeroT_pos_T1{i,j,k})-1;
            n_cycles_T2(i,j,k) = numel(zeroT_pos_T2{i,j,k})-1;
            n_cycles_T3(i,j,k) = numel(zeroT_pos_T3{i,j,k})-1;
        end
    end
end

% end of function
end
```

InterpData_LDV

```
function [T_even V1_even V2_even V3_even n_cycles_T_even Fs a] = InterpData_LDV(X, Y, Z, T1_sel, T2_sel,
T3_sel, V1_sel, V2_sel, V3_sel, n_cycles_T1, n_cycles_T2, n_cycles_T3, zeroT_pos_T1, zeroT_pos_T2,
zeroT_pos_T3, Ts, X_case, Y_case, Z_case, dir_save)
% INTERPDATA_LDV
% Function to interpolate the raw data obtained
% with LDV and transform it into even-time sampled data.
%
% X, Y, Z: vectors containing the coordinates of the measurement locations in the flume
% T1_sel, T2_sel, T3_sel, V1_sel, V2_sel, V3_sel: velocities and time selected after chopping and
filtering
% n_cycles_T1, n_cycles_T2, n_cycles_T3: matrix with number of cycles measured at each location
% zeroT_pos_T1, zeroT_pos_T2, zeroT_pos_T3: matrix with the positions of the first time stamp in every
cycle in vector T, for each location
% Ts: sampling interval (unid) or period of cycle (osc)
% X_case, Y_case, Z_case: coordinates of the location of the case to show in the plots
% dir_save: path of the directory in which to save the figures
%
```

```
% JM Mier
% Ven Te Chow Hydrosystems Laboratory, UIUC
% May 2009
% Revised v2: March 2012. Incorporated independent analysis for each channel and no need to have data in
coincidence mode
% Revised v3: June 2012. Incorporated secondary interpolation pass based on "nearest" to avoid loosing
too many data points when data rates are small
%

% Preliminary analysis of data rates
disp(' ');
disp('Preliminary analysis of Data Rates:');

% Loop the 3 channels (since T1,T2,T3 are different a priori, although very similar, depending on the
goodness of the coincidence of the channels)
hf1 = figure();
hf2 = figure();
color_linea = {[0 1 0];[0 0 1];[1 0 1]};

for ch = 1 : 3

    if ch == 1;      T = T1_sel; n_cycles = n_cycles_T1; zeroT_pos = zeroT_pos_T1;
    elseif ch == 2;  T = T2_sel; n_cycles = n_cycles_T2; zeroT_pos = zeroT_pos_T2;
    elseif ch == 3;  T = T3_sel; n_cycles = n_cycles_T3; zeroT_pos = zeroT_pos_T3;
    end

    % Identify min, max, p10, p50, p90 data rates of the ensemble
    clear T_minus dT_dif dT_dif_min dT_dif_max dT_dif_10 dT_dif_50 dT_dif_90;
    for i = 1 : numel(X)
        for j = 1 : numel(Y)
            for k = 1 : numel(Z)
                T_minus1{i,j,k} = circshift(T{i,j,k},-1);
                dT_dif{i,j,k} = T_minus1{i,j,k}-T{i,j,k};
                dT_dif{i,j,k}(dT_dif{i,j,k}<0) = NaN;
                dT_dif_min(i,j,k) = min(dT_dif{i,j,k}(zeroT_pos{i,j,k}(1):zeroT_pos{i,j,k}(end)-1));
                dT_dif_max(i,j,k) = max(dT_dif{i,j,k}(zeroT_pos{i,j,k}(1):zeroT_pos{i,j,k}(end)-1));
                dT_dif_10(i,j,k) = prctile(dT_dif{i,j,k}(zeroT_pos{i,j,k}(1):zeroT_pos{i,j,k}(end)-
1),10);
                dT_dif_50(i,j,k) = prctile(dT_dif{i,j,k}(zeroT_pos{i,j,k}(1):zeroT_pos{i,j,k}(end)-
1),50);
                dT_dif_90(i,j,k) = prctile(dT_dif{i,j,k}(zeroT_pos{i,j,k}(1):zeroT_pos{i,j,k}(end)-
1),90);
            end
        end
    end

    % Plot dT_dif
    figure(hf1);
    hs1(ch) = subplot(3,1,ch,'Parent',hf1);
    plot(hs1(ch),dT_dif{X_case,Y_case,Z_case},'Color',color_linea{ch});
    legend(hs1(ch),{['V',num2str(ch)]});
    if ch == 1; title(hs1(ch),['Time stamp differences at point:  (', num2str(X(X_case)*1000), ', ',
num2str(Y(Y_case)*1000), ', ', num2str(Z(Z_case)*1000), ') mm']); end;
    if ch == 3; xlabel(hs1(ch),'Data count (#)'); end;
    ylabel(hs1(ch),'time (s)');
    ylim(hs1(ch),[-0.01 max(dT_dif{X_case,Y_case,Z_case})]);
```

```
    set(hf1,'position',[228 228 560 420]);          % Back to original size in left window


    DataRate_min = min(min(min(1./dT_dif_max)));
    DataRate_max = max(max(max(1./dT_dif_min)));
    DataRate_10 = mean(mean(mean(1./dT_dif_10)));
    DataRate_50 = mean(mean(mean(1./dT_dif_50)));
    DataRate_90 = mean(mean(mean(1./dT_dif_90)));
    %dT_dif_pos = find(min(dT_dif));
    disp(sprintf('Data Rates (Hz) for Channel %g: min = %g, max = %g, p10 = %g, p50 = %g, p90 =
%g',ch,DataRate_min,DataRate_max,DataRate_10,DataRate_50,DataRate_90));

    % Plot Data Rates
    %figure(hf2);
    if ch == 1; ha2 = axes('Parent',hf2); end;
    plot(ha2,1./squeeze(dT_dif_10(X_case,Y_case,:)),[1:numel(Z)],'-.','Color',color_linea{ch});
hold(ha2,'on');
    plot(ha2,1./squeeze(dT_dif_50(X_case,Y_case,:)),[1:numel(Z)],'-','Color',color_linea{ch});
    plot(ha2,1./squeeze(dT_dif_90(X_case,Y_case,:)),[1:numel(Z)],'--','Color',color_linea{ch});
    if ch == 1
        set(ha2,'XScale','log');
        legend(ha2,'10%','50%','90%');
        title(ha2,['Data Rates percentiles for V1,V2,V3 at profile:  X = ', num2str(X(X_case)*1000), '
mm ,  Y = ', num2str(Y(Y_case)*1000), ' mm']);
        xlabel(ha2,'Data Rate (Hz)');
        ylabel(ha2,'Z location');
        %view([90 -90]);
        set(hf2,'position',[228 228 560 420]);          % Back to original size in left window
    end
    hp22(ch) = plot(ha2,1./squeeze(dT_dif_50(X_case,Y_case,:)),[1:numel(Z)],'-
','Color',color_linea{ch});   % Need to repeat this line only for the second legend box.
end

% Even-time sampling - final interpolation (linear) of velocities
disp(' ');
disp('Final interpolation for even-time sampling:');

% based on a*dT criteria
a = input('alpha (for a*dT) = ');   % Optimization parameter defining interval a*dT
Fs = input('Fs (Hz) = ');
dT = 1/Fs;
T_even = roundn(0:dT:Ts-dT,-6);     % Round to 6 decimal positions, to avoid possible errors in the
interpolation

for ch = 1 : 3                       % For each of the 3 channels
    if ch == 1;      T = T1_sel; n_cycles = n_cycles_T1; zeroT_pos = zeroT_pos_T1; V = V1_sel;
    elseif ch == 2;  T = T2_sel; n_cycles = n_cycles_T2; zeroT_pos = zeroT_pos_T2; V = V2_sel;
    elseif ch == 3;  T = T3_sel; n_cycles = n_cycles_T3; zeroT_pos = zeroT_pos_T3; V = V3_sel;
    end

    n_nans = zeros(numel(X),numel(Y),numel(Z));
    n_points = zeros(numel(X),numel(Y),numel(Z));
    ratio_new_old = zeros(numel(X),numel(Y),numel(Z));
    V_mean = zeros(numel(X),numel(Y),numel(Z));
    V_rms = zeros(numel(X),numel(Y),numel(Z));
    V_even = cell(numel(X),numel(Y),numel(Z));
```

```
    n_cycles_T_even = zeros(numel(X),numel(Y),numel(Z));

    for i = 1 : numel(X)
        for j = 1 : numel(Y)
            for k = 1 : numel(Z)
                % Padding with NaNs in case all 3 channels don't have the same number of cycles
                % It is still ok if number of cycles is different at different locations (will be
corrected later)
                n_cycles_T_even(i,j,k) = max([n_cycles_T1(i,j,k) n_cycles_T2(i,j,k)
n_cycles_T3(i,j,k)]);        % Max number of cycles in any of the 3 channels, for this location (X,Y,Z)
                V_even{i,j,k} = NaN(n_cycles_T_even(i,j,k),numel(T_even));

                for nc = 1 : n_cycles(i,j,k)
                    clear T_pos T_pos_i T_pos_ip1 T_i T_ip1;
                    % Positions where T_even falls within vector T
                    T_pos = interp1(T{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-
1),zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1,T_even);
                        if isempty(find(abs(isnan(T_pos)-1))); continue; end;   % If there are no
elements to interpolate in this cycle (nc), then continue to next cycle (all values will be given NaNs
from the initialization of U_even). Otherwise the code could break due to empty matrices [].
                    T_pos_i = floor(T_pos);
                    T_pos_ip1 = ceil(T_pos);
                    % Left bounds of T for T_even
                    T_i(find(abs(isnan(T_pos_i)-1))) = T{i,j,k}(T_pos_i(find(abs(isnan(T_pos_i)-1))));
                    if nc == 1          % For the first cycle, there is NaN to the left of the first
point, if no data point exists.
                        T_i(1:find(find(isnan(T_pos_i))<numel(T_i),1,'last')) = NaN;
                    else                % For the rest, there is the previous cycle to the left of the
first point, so that we get continuity between cycles.
                        T_i(1:find(find(isnan(T_pos_i))<numel(T_i),1,'last')) =
T{i,j,k}(zeroT_pos{i,j,k}(nc)-1) - Ts;
                    end
                    T_i(numel(T_i)+1:numel(T_even)) = T_i(end);
                    % Right bounds of T for T_even
                    T_ip1(find(abs(isnan(T_pos_ip1)-1))) = T{i,j,k}(T_pos_ip1(find(abs(isnan(T_pos_ip1)-
1))));
                    T_ip1(1:find(find(isnan(T_pos_ip1))<numel(T_ip1),1,'last')) =
T_ip1(find(find(isnan(T_pos_ip1))<numel(T_ip1),1,'last')+1);
                    if nc == n_cycles(i,j,k)        % For the last cycle, there is NaN to the right of
the last point, if no data point exists.
                        T_ip1(numel(T_ip1)+1:numel(T_even)) = NaN;
                    else                            % For the rest, there is the next cycle to the right
of the last point, so that we get continuity between cycles.
                        T_ip1(numel(T_ip1)+1:numel(T_even)) = T{i,j,k}(zeroT_pos{i,j,k}(nc+1)) + Ts;
                    end

    % Do interpolation only if there are real points within a close range of +-a*dT (T_i and T_ip1) both
sides of the interpolated point (T_even).
    % Otherwise store NaN value. Values out of the range of the data will become NaN as well (first and
last cycles).
                    if n_cycles(i,j,k) == 1             % For the case with only one cycle, don't do
continuity either at begining or end of the cycle: [U(nc)_all]
                        V_even{i,j,k}(nc,find(T_even-T_i<=a*dT & T_ip1-T_even<=a*dT)) =
interp1([T{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1)] ,
[V{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1)] , T_even(find(T_even-T_i<=a*dT & T_ip1-
T_even<=a*dT)),'linear');
```

```
                    elseif nc == 1                      % For the first cycle, we can only give
continuity at the end of the cycle: [U(nc)_all;U(nc+1)_1]
                        V_even{i,j,k}(nc,find(T_even-T_i<=a*dT & T_ip1-T_even<=a*dT)) =
interp1([T{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1) ; T{i,j,k}(zeroT_pos{i,j,k}(nc+1))+Ts]
, [V{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1) ; V{i,j,k}(zeroT_pos{i,j,k}(nc+1))] ,
T_even(find(T_even-T_i<=a*dT & T_ip1-T_even<=a*dT)),'linear');
                    elseif nc == n_cycles(i,j,k)        % For the last cycle, we can only give
continuity at the beginning of the cycle: [U(nc-1)_last;U(nc)]
                        V_even{i,j,k}(nc,find(T_even-T_i<=a*dT & T_ip1-T_even<=a*dT)) =
interp1([T{i,j,k}(zeroT_pos{i,j,k}(nc)-1)-Ts ; T{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1)]
, [V{i,j,k}(zeroT_pos{i,j,k}(nc)-1) ; V{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1)] ,
T_even(find(T_even-T_i<=a*dT & T_ip1-T_even<=a*dT)),'linear');
                    else                                % For the rest cycles, we give continuity at the
beginning and end of the cycle: [U(nc-1)_last;U(nc)_all;U(nc+1)_1]
                        V_even{i,j,k}(nc,find(T_even-T_i<=a*dT & T_ip1-T_even<=a*dT)) =
interp1([T{i,j,k}(zeroT_pos{i,j,k}(nc)-1)-Ts ; T{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1) ;
T{i,j,k}(zeroT_pos{i,j,k}(nc+1))+Ts] , [V{i,j,k}(zeroT_pos{i,j,k}(nc)-1) ;
V{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1) ; V{i,j,k}(zeroT_pos{i,j,k}(nc+1))] ,
T_even(find(T_even-T_i<=a*dT & T_ip1-T_even<=a*dT)),'linear');
                    end

    % Use nearest real point when there is only one point close to the T_even point and so interpolation
would give NaN.
    % This method helps to not loose too many points at the locations close to the wall where data rates
are very small.
    % Need to be careful not to take points far away from the T_even point. This distance should be kept
very small relative to the cycle length (about 1/1000) so that the effect of doing this "nearest"
approximation doesn't distort the resulting statistics.
                    if n_cycles(i,j,k) == 1             % For the case with only one cycle, don't do
continuity either at begining or end of the cycle: [U(nc)_all]
                        V_even{i,j,k}(nc,find(T_even-T_i<=min(a*dT,Ts/1000) & T_ip1-T_even>a*dT)) =
interp1([T{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1)] ,
[V{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1)] , T_even(find(T_even-T_i<=min(a*dT,Ts/1000) &
T_ip1-T_even>a*dT)),'nearest');
                        V_even{i,j,k}(nc,find(T_even-T_i>a*dT & T_ip1-T_even<=min(a*dT,Ts/1000))) =
interp1([T{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1)] ,
[V{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1)] , T_even(find(T_even-T_i>a*dT & T_ip1-
T_even<=min(a*dT,Ts/1000))),'nearest');
                    elseif nc == 1                      % For the first cycle, we can only give
continuity at the end of the cycle: [U(nc)_all;U(nc+1)_1]
                        V_even{i,j,k}(nc,find(T_even-T_i<=min(a*dT,Ts/1000) & T_ip1-T_even>a*dT)) =
interp1([T{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1) ; T{i,j,k}(zeroT_pos{i,j,k}(nc+1))+Ts]
, [V{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1) ; V{i,j,k}(zeroT_pos{i,j,k}(nc+1))] ,
T_even(find(T_even-T_i<=min(a*dT,Ts/1000) & T_ip1-T_even>a*dT)),'nearest');
                        V_even{i,j,k}(nc,find(T_even-T_i>a*dT & T_ip1-T_even<=min(a*dT,Ts/1000))) =
interp1([T{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1) ; T{i,j,k}(zeroT_pos{i,j,k}(nc+1))+Ts]
, [V{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1) ; V{i,j,k}(zeroT_pos{i,j,k}(nc+1))] ,
T_even(find(T_even-T_i>a*dT & T_ip1-T_even<=min(a*dT,Ts/1000))),'nearest');
                    elseif nc == n_cycles(i,j,k)        % For the last cycle, we can only give
continuity at the beginning of the cycle: [U(nc-1)_last;U(nc)]
                        V_even{i,j,k}(nc,find(T_even-T_i<=min(a*dT,Ts/1000) & T_ip1-T_even>a*dT)) =
interp1([T{i,j,k}(zeroT_pos{i,j,k}(nc)-1)-Ts ; T{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1)]
, [V{i,j,k}(zeroT_pos{i,j,k}(nc)-1) ; V{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1)] ,
T_even(find(T_even-T_i<=min(a*dT,Ts/1000) & T_ip1-T_even>a*dT)),'nearest');
                        V_even{i,j,k}(nc,find(T_even-T_i>a*dT & T_ip1-T_even<=min(a*dT,Ts/1000))) =
interp1([T{i,j,k}(zeroT_pos{i,j,k}(nc)-1)-Ts ; T{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1)]
```

```
, [V{i,j,k}(zeroT_pos{i,j,k}(nc)-1) ; V{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1)] ,
T_even(find(T_even-T_i>a*dT & T_ip1-T_even<=min(a*dT,Ts/1000))),'nearest');
                    else                                % For the rest cycles, we give continuity at the
beginning and end of the cycle: [U(nc-1)_last;U(nc)_all;U(nc+1)_1]
                        V_even{i,j,k}(nc,find(T_even-T_i<=min(a*dT,Ts/1000) & T_ip1-T_even>a*dT)) =
interp1([T{i,j,k}(zeroT_pos{i,j,k}(nc)-1)-Ts ; T{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1) ;
T{i,j,k}(zeroT_pos{i,j,k}(nc+1))+Ts] , [V{i,j,k}(zeroT_pos{i,j,k}(nc)-1) ;
V{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1) ; V{i,j,k}(zeroT_pos{i,j,k}(nc+1))] ,
T_even(find(T_even-T_i<=min(a*dT,Ts/1000) & T_ip1-T_even>a*dT)),'nearest');
                        V_even{i,j,k}(nc,find(T_even-T_i>a*dT & T_ip1-T_even<=min(a*dT,Ts/1000))) =
interp1([T{i,j,k}(zeroT_pos{i,j,k}(nc)-1)-Ts ; T{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1) ;
T{i,j,k}(zeroT_pos{i,j,k}(nc+1))+Ts] , [V{i,j,k}(zeroT_pos{i,j,k}(nc)-1) ;
V{i,j,k}(zeroT_pos{i,j,k}(nc):zeroT_pos{i,j,k}(nc+1)-1) ; V{i,j,k}(zeroT_pos{i,j,k}(nc+1))] ,
T_even(find(T_even-T_i>a*dT & T_ip1-T_even<=min(a*dT,Ts/1000))),'nearest');
                    end

                end             % end for "nc" cycle
    % Check number of NaNs and good points ** It would be the same for all three colors if we were
sampling in coincidence mode, so they all have the same timing **
                n_nans(i,j,k) = sum(sum(isnan(V_even{i,j,k}),2))/numel(V_even{i,j,k});
                n_points(i,j,k) = (numel(V_even{i,j,k})-
sum(sum(isnan(V_even{i,j,k}),2)))/numel(V_even{i,j,k});
                ratio_new_old(i,j,k) = (numel(V_even{i,j,k})-
sum(sum(isnan(V_even{i,j,k}),2)))/numel(V{i,j,k});

    % Compute mean (in time) and standard deviation for each point
                V_mean(i,j,k) = nanmean(nanmean(V_even{i,j,k},2));
                V_rms(i,j,k) = nanmean(nanstd(V_even{i,j,k},0,2));
            end
        end
    end

    if ch == 1;      V1_even = V_even; n_nans_V1 = n_nans; n_points_V1 = n_points; ratio_new_old_V1 =
ratio_new_old; V1_mean = V_mean; V1_rms = V_rms;
    elseif ch == 2;  V2_even = V_even; n_nans_V2 = n_nans; n_points_V2 = n_points; ratio_new_old_V2 =
ratio_new_old; V2_mean = V_mean; V2_rms = V_rms;
    elseif ch == 3;  V3_even = V_even; n_nans_V3 = n_nans; n_points_V3 = n_points; ratio_new_old_V3 =
ratio_new_old; V3_mean = V_mean; V3_rms = V_rms;
    end

end             % end of loop for each channel

% end of function
end
```

Transform_LDV

```
function [U V W] = Transform_LDV(X, Y, Z, V1_even, V2_even, V3_even, T_even, X_case, Y_case, Z_case,
dir_save)
% TRANSFORM_LDV
% Function to transform LDV velocities V1,V2,V3 (in probe non-orthogonal reference system)
% into U,V,W (in orthogonal cartesian reference system), given the angles of the probe
% (pitch, rotation, yaw) and the half-angles of the beams (alpha (ch.1), beta (ch.2), gamma (ch.3))
%
```

```
% X, Y, Z: vectors containing the coordinates of the measurement locations in the flume
```

```
% V1_even,V2_even,V3_even: velocities for each channel in probe reference system
% T_even: time vector for even-time data of velocities
% X_case, Y_case, Z_case: coordinates of the location of the case to show in the plots
% dir_save: path of the directory in which to save the figures
% ** Must be already in even-time sampled form, so that the triplets of velocity measurements correspond
to each other in the same instant in time
% ** Also, must have same number of data points!!
%
% JM Mier
% Ven Te Chow Hydrosystems Laboratory, UIUC
% March 2012
%

% Input data
% probe angles in air
pitch = -1.0 %-4.9 %input('enter pitch angle = ');
rotation = 90 %input('enter rotation angle = ');
yaw = 0 %input('enter yaw angle = ');

% beam half-angles in air
alpha = 1.98 %3.95 %input('enter half-angle ch.1 (green) = ');
beta = 3.95 %input('enter half-angle ch.2 (blue) = ');
gamma = 2.35 %input('enter half-angle ch.3 (violet) = ');

% ** CAREFUL! Use UNDERWATER angles if measuring in water! **
w = menu({'Measuring underwater?' ;'(will correct half-angles)'},'Yes','No');
if w == 1
    n_water = 1.32;         % Refraction coefficient of water
    n_air = 1.00;           % Refraction coefficient of air

    % Apply Snell law (n1 * sin k1 = n2 * sin k2)
    pitch = asin(sin(pitch/180*pi())*n_air/n_water)*180/pi();
    rotation = asin(sin(rotation/180*pi())*n_air/n_water)*180/pi();
    yaw = asin(sin(yaw/180*pi())*n_air/n_water)*180/pi();
    alpha = asin(sin(alpha/180*pi())*n_air/n_water)*180/pi();
    beta = asin(sin(beta/180*pi())*n_air/n_water)*180/pi();
    gamma = asin(sin(gamma/180*pi())*n_air/n_water)*180/pi();
end

% Transformation matrix
% such that (V1,V2,V3) = M * (U,V,W)
% see TSI manual for more info
% ** CAREFUL! Transformation matrix depends on the arrangement of the LDV probes and beams! **

% For 3D measurements with
% side: 5-beam probe (green vertical + , blue horizontal -)
% top: 2-beam subm probe (violet spanwise +)

M = [0 sin((alpha-pitch)/180*pi()) cos((alpha-pitch)/180*pi()) ; ...
    -1 0 0; ...
    0 1 0];

disp(' ');
disp('Transformation matrix:');
disp(['      ','  U        ','  V        ','  W        ']);
disp([['Ch.1  ';'Ch.2  ';'Ch.3  '],num2str(roundn(M,-4))]);
```

```
% Inverse matrix
% such that (U,V,W) = MM * (V1,V2,V3)
MM = inv(M);

disp(' ');
disp('Inverse matrix:');
disp(['  ','   Ch.1     ','   Ch.2     ','   Ch.3     ']);
disp([['U  ';'V  ';'W  '],num2str(roundn(MM,-4))]);

% Transformed velocities
U = cell(numel(X),numel(Y),numel(Z));
V = cell(numel(X),numel(Y),numel(Z));
W = cell(numel(X),numel(Y),numel(Z));

for i = 1 : numel(X)
    for j = 1 : numel(Y)
        for k = 1 : numel(Z)
%            U{i,j,k} = MM(1,1) .* V1_even{i,j,k} + MM(1,2) .* V2_even{i,j,k} + MM(1,3) .*
V3_even{i,j,k};
%            V{i,j,k} = MM(2,1) .* V1_even{i,j,k} + MM(2,2) .* V2_even{i,j,k} + MM(2,3) .*
V3_even{i,j,k};
%            W{i,j,k} = MM(3,1) .* V1_even{i,j,k} + MM(3,2) .* V2_even{i,j,k} + MM(3,3) .*
V3_even{i,j,k};

            % Simplified for the 3D case in particular to avoid NaNs from one channel to give NaNs on
everything else
            V3_even_nonnan{i,j,k} = V3_even{i,j,k};
            V3_even_nonnan{i,j,k}(isnan(V3_even{i,j,k})) = 0;
            U{i,j,k} =                          + MM(1,2) .* V2_even{i,j,k}
;
            V{i,j,k} =                                                      + MM(2,3) .*
V3_even{i,j,k};
            W{i,j,k} = MM(3,1) .* V1_even{i,j,k}                            + MM(3,3) .*
V3_even_nonnan{i,j,k};
        end
    end
end

% end of function
end
```

ZeroLevel_LDV

```
function z0 = ZeroLevel_LDV(z,t,u,Ts,dir_save)
% ZEROLEVEL_LDV
% Function to calculate the real z=0 level of the bed from a LDV profile.
%
% z: vector containing the coordinates of the measurement locations in the flume
% t: processed time vector
% u: processed velocity in streamwise direction
% Ts: cycle duration
% dir_save: path of the directory in which to save the figures
%
```

```
% JM Mier
% Ven Te Chow Hydrosystems Laboratory, UIUC
% April 2013
%

% OSCILLATORY FLOW ONLY %%
% May 2013 - It can work for unidirectional flow as well, although be cautious

disp(' ');
disp('Calculation of z=0 level:');

%dt = t(2)-t(1);
%Ts = roundn((t(end)-t(1))+dt,-2);
phi = roundn(t/Ts*360,-2);

% Mean velocity profiles (U only)
% Average velocity cycle (ensemble average in cycles (nc) for every (x,y,z) location)
u_mean = nanmean(u,4);          % Results in u_mean(1,1,z,1,phi)
u_mean_smooth = SmoothPlot_LDV(u_mean,5,4,1);
u_mean_smooth = SmoothPlot_LDV2(u_mean_smooth,repmat(shiftdim(z,-2),[1,1,1,1,numel(phi)]),3,2,0);

% Plot
n_phi_plot = input(['Select number of profiles to plot (max=' , num2str(numel(phi)) , '): ']);      %
Number of lines to plot in profiles
n_phi_plot = min(n_phi_plot, numel(phi));
phi_plot = phi(1 : roundn(numel(phi)/n_phi_plot,0) : numel(phi));               % Phases to plot

figure1 = figure();
for h = 1 : numel(phi_plot)
    phi_pos(h) = find(phi==phi_plot(h));
    u_prof = squeeze(u_mean_smooth(:,:,:,:,phi_pos(h)));
    figure(figure1);
    hp = plot(u_prof,z,'o','MarkerSize',2); hc{h} = get(hp,'Color'); hold all;
    pause(0.1);
end

figure(figure1);
legend(num2str(phi_plot'),4);
title('Mean Velocity (smoothed) profiles (X\_av , NC\_av)  -  U_m_e_a_n (m/s)');
xlabel('U_m_e_a_n (m/s)');
ylabel('z (m)');
ylim([0 0.001]);
xlim([min([0,get(gca,'XLim')]) , max(get(gca,'XLim'))]);

% Linear fit to the bottom points in profile
% Such that z = m*u + b
Zloc1_fit = input('Select first Z location from the bottom to include in fit: ');
Zloc2_fit = input('Select last Z location from the bottom to include in fit: ');
cfit1 = cell(numel(phi),1);
gof1 = cell(numel(phi),1);
for h = 1 : numel(phi)
    [cfit1{h},gof1{h}] =
fit(squeeze(u_mean_smooth(:,:,Zloc1_fit:Zloc2_fit,:,h)),z(Zloc1_fit:Zloc2_fit),'poly1');
    m1(h) = cfit1{h}.p1;
    b1(h) = cfit1{h}.p2;
end
```

```
% Plot fit lines with data
for h = 1 : numel(phi_plot)
    phi_pos(h) = find(phi==phi_plot(h));
    z_lines_prof = [b1(phi_pos(h));z];
    u_lines_prof = (z_lines_prof-b1(phi_pos(h)))./m1(phi_pos(h));
    figure(figure1);
%    hp = plot(u_prof,z,'o','MarkerSize',2); hc = get(hp,'Color'); hold all;
    hpp = plot(u_lines_prof,z_lines_prof,'-','MarkerSize',2,'Color',hc{h});
set(get(get(hpp,'Annotation'),'LegendInformation'),'IconDisplayStyle','off'); % Exclude line from
legend);
    pause(0.1);
end
saveas(gcf,[dir_save '\Zero-level calculation_linear fit'],'jpg');

% Result of fit
% Method A: Calculation of z0 assuming all profiles meet at u=0

% Phases to use are only those with a developed profile (near u_max)
phi_loc_u_high = (abs(squeeze(u_mean_smooth(:,:,1,:,:)))>max(squeeze(u_mean_smooth(:,:,1,:,:)))/2);

% Average 'b', taking only those profiles with higher velocity to avoid
% bad results near zero velocity profiles
b1_mean = mean(b1(phi_loc_u_high));

% Plot 'b' coefficient
figure
plot(phi,b1); hold on;
plot(phi(phi_loc_u_high),b1(phi_loc_u_high),'or');
legend('all values','used for z0');
title('Coefficient of linear fit to find z=0 level');
xlabel('Phi (°)');
ylabel('b (m)');
xlim([0 360]);
ylim([b1_mean-0.0005 b1_mean+0.0005]);
saveas(gcf,[dir_save '\Zero-level calculation_coefficient'],'jpg');

% Show result (rounded to 1 micron increments)
z0_coef = b1_mean;
disp(['A) Coefficient method: z0 = ',num2str(roundn(z0_coef,-6)*1000),' mm']);

% Method B: Calculation of z0 from statistics of linear fit lines crossing points (doesn't assume u=0)
% Find the crossing point of each line with all the others
% line1: y1 = m1 x1 + b1
% line2: y2 = m2 x2 + b2
% Crossing point: y1 = y2 and x1 = x2, so:
% m1 x + b1 = m2 x + b2 ; x = (b2-b1)/(m1-m2) ; y = m1 x + b1
% If we call x = u, and y = z, then: z0 (z=0 level) would be the mean of all 'y', and u0 (bottom drift)
would be the mean of all 'x'

m1_matrix = repmat(m1,[numel(phi) 1]);
b1_matrix = repmat(b1,[numel(phi) 1]);
m2_matrix = repmat(m1',[1 numel(phi)]);
b2_matrix = repmat(b1',[1 numel(phi)]);

x_crossing_matrix = (b2_matrix-b1_matrix)./(m1_matrix-m2_matrix);
```

```
x_crossing_matrix(~(tril(ones(numel(phi)),-20) + triu(ones(numel(phi)),20))) = NaN;    % Exclude
intersections between neighboring lines (diagonal)
x_crossing_matrix(fliplr(~(tril(ones(numel(phi)),-20) + triu(ones(numel(phi)),20)))) = NaN;    % Exclude
intersections between lines symmetric respect to the velocity peak (antidiagonal)
x_crossing_matrix(~phi_loc_u_high,:) = NaN;       % Select only the phases with a developed profile
(near u_max)
x_crossing_matrix(:,~phi_loc_u_high) = NaN;       % Select only the phases with a developed profile
(near u_max)

y_crossing_matrix = m1_matrix.*x_crossing_matrix + b1_matrix;

% Transform matrices into a continuous vector of crossing points (x,y)
x_crossing_vector = x_crossing_matrix(~isnan(x_crossing_matrix));
y_crossing_vector = y_crossing_matrix(~isnan(x_crossing_matrix));

% Check in case there are no valid crossing points to use for calculations
if isempty(x_crossing_vector)
    z0_crossing = [];
    disp('B) Crossing points method: z0 = (unable to calculate)');

else
    % Plot lines crossing points
    figure
    plot(x_crossing_vector,y_crossing_vector,'.'); hold on;
    plot(nanmean(x_crossing_matrix,2),nanmean(y_crossing_matrix,2),'oc');
    legend('all points','average per phase');
    title('Crossing points between lines of linear fit to find z=0 level');
    xlabel('U (m/s)');
    ylabel('z (m)');

    % Statistics for x (u)
    x_bin_edge = [-max(max(u_mean_smooth)):0.001:max(max(u_mean_smooth))];
    x_count = histc(x_crossing_vector,x_bin_edge);
    x_threshold = x_bin_edge(x_count>0.1*max(x_count));
    x_mean = mean(x_crossing_vector(x_crossing_vector>x_threshold(1) &
x_crossing_vector<x_threshold(end)));
    x_mode = mode(roundn(x_crossing_vector,-3));

    % Statistics for y (z)
    y_bin_edge = [-0.001:0.000001:0.001];
    y_count = histc(y_crossing_vector,y_bin_edge);
    y_threshold = y_bin_edge(y_count>0.1*max(y_count));
    y_mean = mean(y_crossing_vector(y_crossing_vector>y_threshold(1) &
y_crossing_vector<y_threshold(end)));
    y_mode = mode(roundn(y_crossing_vector,-6));

    % Include results in plot
    plot(x_mean,y_mean,'sr');
    plot(nanmean(nanmean(x_crossing_matrix)),nanmean(nanmean(y_crossing_matrix)),'sm');
    plot(x_mode,y_mode,'^r');
    plot(mode(roundn(nanmean(x_crossing_matrix),-3)),mode(roundn(nanmean(y_crossing_matrix),-6)),'^m');
    legend('all points','average per phase','Mean (all points)','Mean (avg per phase)','Mode (all
points)','Mode (avg per phase)');
    xlim([x_mean-0.2 x_mean+0.2]);
    ylim([y_mean-0.0001 y_mean+0.0001]);

    saveas(gcf,[dir_save '\Zero-level calculation_crossing points'],'jpg');
```

```
    % Show result (rounded to 1 micron increments)
    z0_crossing = y_mean;
    disp(['B) Crossing points method: z0 = ',num2str(roundn(z0_crossing,-6)*1000),' mm']);
end

% Choose which method to take as final result
% The result from method B) should be more accurate.
method = 0;
while method==0         % Loop to keep asking for answer until a valid choice is selected
    method = input('Choose method for z0 (Method A)=1 , Method B)=2 , Custom=3): ');
    if     method==1; z0 = z0_coef;
    elseif method==2; z0 = z0_crossing;
    elseif method==3; z0 = input('Enter z0 value (in mm): ')/1000;
    else              method = 0;
    end
end

% Round z0 to 10 microns increments
z0 = roundn(z0,-5);
disp(['Final value (rounded to 10 microns increments): z0 = ',num2str(z0*1000),' mm']);

% end of function
end
```

CheckData_LDV

```
function [] = CheckData_LDV(x,y,z,t,u,v,w,n_cycles,X_case,Y_case,Z_case,dir_save,Ts,Fs,a)
% CHECKDATA_LDV
% Function to generate several figures to double-check the quality
% of the LDV data after processing.
%
% x, y, z: vectors containing the coordinates of the measurement locations in the flume
% t, u, v, w: processed time and velocities
% n_cycles: number of cycles measured at each location
% X_case, Y_case, Z_case: coordinates of the location of the case to show in the plots
% dir_save: path of the directory in which to save the figures
% Ts: cycle duration
% Fs: sampling frequency
% a: interval amplitude for even-time interpolation routine, such that Interval = +-a*dT
%
% JM Mier
% Ven Te Chow Hydrosystems Laboratory, UIUC
% May 2009
% Revised: March 2012 - 2D plots of running average velocities
% Revised: August 2012 - 2D plots of running RMS velocities
% Revised: May 2013 - 2D plots of percent error of running average and running RMS
%

% OSCILLATORY FLOW ONLY %%
% May 2013 - It can work for unidirectional flow as well, although be cautious

% Number of valid data points after processing
%Ts = roundn((t(end)-t(1))+1/Fs,-2);
```

```
phi = roundn(t/Ts*360,-2);
for i = 1 : numel(x)
    for j = 1 : numel(y)
        Plot_LDV_contours(phi,z,squeeze(sum(~isnan(u(i,j,:,:,:)),4)),'Valid data
points','U_n_\__p_o_i_n_t_s','#',dir_save,'Unp');
        Plot_LDV_contours(phi,z,squeeze(sum(~isnan(v(i,j,:,:,:)),4)),'Valid data
points','V_n_\__p_o_i_n_t_s','#',dir_save,'Vnp');
        Plot_LDV_contours(phi,z,squeeze(sum(~isnan(w(i,j,:,:,:)),4)),'Valid data
points','W_n_\__p_o_i_n_t_s','#',dir_save,'Wnp');
    end
end

% Number of cycles that need to be measured
% Limit the amount of data because the routine is very computer intensive
n_cycles_limit = 1024;
n_cycles =
min(cat(numel(size(n_cycles))+1,ones(size(n_cycles)).*n_cycles_limit,n_cycles),[],numel(size(n_cycles))+
1);
%n_cycles_min = min(min(min(n_cycles)));
n_cycles_max = max(max(max(n_cycles)));

% 1- Mean Velocity criteria

% 1a- Ensemble averaged velocities
% Average velocity cycle (ensemble average in cycles (nc) and in space (x) for every (y,z) locations)
u_mean = nanmean(nanmean(u,4),1);
v_mean = nanmean(nanmean(v,4),1);
w_mean = nanmean(nanmean(w,4),1);

% 1b- Running average velocities
% Average criterion (signal at every (y,z) location)
% running average over time (over cycles nc)
u_rmean = zeros(size(u(:,:,:,1:n_cycles_max,:)));
v_rmean = zeros(size(v(:,:,:,1:n_cycles_max,:)));
w_rmean = zeros(size(w(:,:,:,1:n_cycles_max,:)));
for nc = 1 : n_cycles_max
    u_rmean(:,:,:,nc,:) = nanmean(nanmean(u(:,:,:,1:nc,:),4),1);
    v_rmean(:,:,:,nc,:) = nanmean(nanmean(v(:,:,:,1:nc,:),4),1);
    w_rmean(:,:,:,nc,:) = nanmean(nanmean(w(:,:,:,1:nc,:),4),1);
end

% 1c- 2D plots of running average for several phases at one particular location (X,Y,Z), in the u-nc
plane
% Plot for all locations
p = menu('Plot of running average?','Yes','No');

if p == 1
    s = menu('Save pictures?','Yes','No');

    % X_case, Y_case, Z_case can be single values or vectors
    c = menu('Number of cases?','One','All');
    if c == 1
        X_cases = X_case;
        Y_cases = Y_case;
        Z_cases = Z_case;
    else
```

```
        X_cases = 1:numel(x);
        Y_cases = 1:numel(y);
        Z_cases = 1:numel(z);
    end

    n_lines = 5;                                    % Number of phases to plot
    n_lines = min(n_lines, numel(t));

    figure();
    hp1 = subplot(3,1,1); hold on;
    hp2 = subplot(3,1,2); hold on;
    hp3 = subplot(3,1,3); hold on;
    linkaxes([hp1 hp2 hp3],'x');
    set(gcf,'position',[1 80 1024 600]);            % Maximizes figure window

    % Loop through all files requested
    for i = 1 : numel(X_cases)
        for j = 1 : numel(Y_cases)
            for k = 1 : numel(Z_cases)

                leyenda1 = [];
%                leyenda2 = [];
%                leyenda3 = [];
                cla(hp1);
                cla(hp2);
                cla(hp3);
                tp = t;
                np = 1:n_cycles_max;

                for h = 1 : roundn(numel(tp)/n_lines,0) : numel(tp)     % loop in number of phases to
plot (n_lines)

                    up = squeeze(u_rmean(X_cases(i),Y_cases(j),Z_cases(k),np,h));
                    upp = squeeze(u(X_cases(i),Y_cases(j),Z_cases(k),np,h));
                    plot(hp1,np,up,'-','color',[h/numel(tp) 0 1-h/numel(tp)]); % hold on;
                    handlep = plot(hp1,np,upp,'.','color',[h/numel(tp) 0 1-
h/numel(tp)],'MarkerSize',10); hold on;
set(get(get(handlep,'Annotation'),'LegendInformation'),'IconDisplayStyle','off'); % Exclude line from
legend
                    leyenda1 = [leyenda1 ; ['Phi = ',num2str(roundn((h-
1)/numel(tp)*360,0),'%03.0f'),'°']];

                    vp = squeeze(v_rmean(X_cases(i),Y_cases(j),Z_cases(k),np,h));
                    vpp = squeeze(v(X_cases(i),Y_cases(j),Z_cases(k),np,h));
                    plot(hp2,np,vp,'-','color',[h/numel(tp) 0 1-h/numel(tp)]); % hold on;
                    handlep = plot(hp2,np,vpp,'.','color',[h/numel(tp) 0 1-
h/numel(tp)],'MarkerSize',10); hold on;
set(get(get(handlep,'Annotation'),'LegendInformation'),'IconDisplayStyle','off'); % Exclude line from
legend

                    wp = squeeze(w_rmean(X_cases(i),Y_cases(j),Z_cases(k),np,h));
                    wpp = squeeze(w(X_cases(i),Y_cases(j),Z_cases(k),np,h));
                    plot(hp3,np,wp,'-','color',[h/numel(tp) 0 1-h/numel(tp)]); % hold on;

                    handlep = plot(hp3,np,wpp,'.','color',[h/numel(tp) 0 1-
h/numel(tp)],'MarkerSize',10); hold on;
set(get(get(handlep,'Annotation'),'LegendInformation'),'IconDisplayStyle','off'); % Exclude line from
legend
```

```
                end                     % end of loop in number of phases

                xlim([np(1)-1 np(end)+1]);
                legend(hp1,leyenda1);
                title(hp1,['Running average along the cycles, for several phases, at z = ',
num2str(z(Z_cases(k))*1000), ' mm']);
                xlabel(hp3,'NC (-)');
                ylabel(hp1,'u (m/s)');
                ylabel(hp2,'v (m/s)');
                ylabel(hp3,'w (m/s)');

                if s == 1
                    file_save = [dir_save,'\Running average along
cycles_z=',num2str(z(Z_cases(k))*1000),'mm_Fs=',num2str(Fs),',a=',num2str(a),'.jpg'];
                    saveas(gcf,file_save,'jpg');
                end

                if c == 2
                    input(['Plotting: Z_case = ', num2str(Z_cases(k)),'. Press enter to continue.']);
                end
            end
        end
    end         % Close loop through cases
    if c == 2; disp('End of plots'); end;
    set(gcf,'position',[228 228 560 420]);          % Back to original size in left window
end

% 1d- Percentage error (pe) in the running average (with respect peak to peak (ptp) value to avoid DIV0)
pe_u_rmean = zeros(size(u(:,:,:,1:n_cycles_max,:)));
pe_v_rmean = zeros(size(v(:,:,:,1:n_cycles_max,:)));
pe_w_rmean = zeros(size(w(:,:,:,1:n_cycles_max,:)));
ptp_u_mean = abs(max(u_mean,[],5)); %abs(max(u_mean,[],5))+abs(min(u_mean,[],5));
ptp_v_mean = abs(max(u_mean,[],5)); %abs(max(v_mean,[],5))+abs(min(v_mean,[],5));
ptp_w_mean = abs(max(u_mean,[],5)); %abs(max(w_mean,[],5))+abs(min(w_mean,[],5));
% We use the ptp of u for the three components to be able to compare among
% them, avoiding very small values of ptp for the components with near zero average

for nc = 1 : n_cycles_max
    pe_u_rmean(:,:,:,nc,:) = abs((u_rmean(:,:,:,nc,:)-u_mean(:,:,:,1,:)))./repmat(ptp_u_mean,[1 1 1 1
numel(t)]);
    pe_v_rmean(:,:,:,nc,:) = abs((v_rmean(:,:,:,nc,:)-v_mean(:,:,:,1,:)))./repmat(ptp_v_mean,[1 1 1 1
numel(t)]);
    pe_w_rmean(:,:,:,nc,:) = abs((w_rmean(:,:,:,nc,:)-w_mean(:,:,:,1,:)))./repmat(ptp_w_mean,[1 1 1 1
numel(t)]);
end

% 1e- 2D plots of percent error for several phases at each particular location (X,Y,Z), in the pe-nc
plane
% Plot for all locations
p = menu('Plot of percent error in the running average?','Yes','No');

if p == 1

    s = menu('Save pictures?','Yes','No');
```

```
    % X_case, Y_case, Z_case can be single values or vectors
    c = menu('Number of cases?','One','All');
    if c == 1
        X_cases = X_case;
        Y_cases = Y_case;
        Z_cases = Z_case;
    else
        X_cases = 1:numel(x);
        Y_cases = 1:numel(y);
        Z_cases = 1:numel(z);
    end

    n_lines = 5;                                % Number of phases to plot
    n_lines = min(n_lines, numel(t));

    figure();
    hp1 = subplot(3,1,1); hold on;
    hp2 = subplot(3,1,2); hold on;
    hp3 = subplot(3,1,3); hold on;
    linkaxes([hp1 hp2 hp3],'x');
    set(gcf,'position',[1 80 1024 600]);            % Maximizes figure window

    % Loop through all files requested
    for i = 1 : numel(X_cases)
        for j = 1 : numel(Y_cases)
            for k = 1 : numel(Z_cases)

                leyenda1 = [];
                cla(hp1);
                cla(hp2);
                cla(hp3);
                tp = t;
                np = 1:n_cycles_max;

                for h = 1 : roundn(numel(tp)/n_lines,0) : numel(tp)     % loop in number of phases to
plot (n_lines)

                    up = squeeze(pe_u_rmean(X_cases(i),Y_cases(j),Z_cases(k),np,h)*100);
                    plot(hp1,np,up,'-','color',[0 h/numel(tp) 1-h/numel(tp)]); % hold on;
                    leyenda1 = [leyenda1 ; ['Phi = ',num2str(roundn((h-
1)/numel(tp)*360,0),'%03.0f'),'°']];

                    vp = squeeze(pe_v_rmean(X_cases(i),Y_cases(j),Z_cases(k),np,h)*100);
                    plot(hp2,np,vp,'-','color',[0 h/numel(tp) 1-h/numel(tp)]); % hold on;

                    wp = squeeze(pe_w_rmean(X_cases(i),Y_cases(j),Z_cases(k),np,h)*100);
                    plot(hp3,np,wp,'-','color',[0 h/numel(tp) 1-h/numel(tp)]); % hold on;
                end                     % end of loop in number of phases

                % Plot tolerance lines
                tolerance = 0.01;       % tolerance +/-, per unit.

                upp = ones(size(up))*tolerance*100;

                handlep = plot(hp1,np,upp,'-r');
set(get(get(handlep,'Annotation'),'LegendInformation'),'IconDisplayStyle','off'); % Exclude line from
legend
```

```
                vpp = ones(size(vp))*tolerance*100;
                handlep = plot(hp2,np,vpp,'-r');
set(get(get(handlep,'Annotation'),'LegendInformation'),'IconDisplayStyle','off'); % Exclude line from
legend
                wpp = ones(size(wp))*tolerance*100;
                handlep = plot(hp3,np,wpp,'-r');
set(get(get(handlep,'Annotation'),'LegendInformation'),'IconDisplayStyle','off'); % Exclude line from
legend

                % Axes decoration
                ylim(hp1,[0,20]);
                ylim(hp2,[0,20]);
                ylim(hp3,[0,20]);
                xlim([np(1)-1 np(end)+1]);
                legend(hp1,leyenda1);
                title(hp1,['Percent error for the running average along the cycles, at z = ',
num2str(z(Z_cases(k))*1000), ' mm']);
                xlabel(hp3,'NC (-)');
                ylabel(hp1,'Error in u_m_e_a_n (%)');
                ylabel(hp2,'Error in v_m_e_a_n (%)');
                ylabel(hp3,'Error in w_m_e_a_n (%)');

                if s == 1
                    file_save = [dir_save,'\Running average_percent
error_z=',num2str(z(Z_cases(k))*1000),'mm_Fs=',num2str(Fs),',a=',num2str(a),'.jpg'];
                    saveas(gcf,file_save,'jpg');
                end

                if c == 2
                    input(['Plotting: Z_case = ', num2str(Z_cases(k)),'. Press enter to continue.']);
                    %pause
                end
            end
        end
    end         % Close loop through cases
    if c == 2; disp('End of plots'); end;
    set(gcf,'position',[228 228 560 420]);          % Back to original size in left window
end


% 2- RMS Velocity criteria

% 2a- Fluctuating velocities
% Fluctuations with respect the average cycle (ensemble average in cycles (nc) and in space (x) for
every (y,z) locations)
uf = u - repmat(nanmean(u,4),[1 1 1 size(u,4) 1]);
vf = v - repmat(nanmean(v,4),[1 1 1 size(v,4) 1]);
wf = w - repmat(nanmean(w,4),[1 1 1 size(w,4) 1]);

% RMS velocities
u_rms = nanmean(nanstd(u,0,4),1);
v_rms = nanmean(nanstd(v,0,4),1);
w_rms = nanmean(nanstd(w,0,4),1);
```

```
% 2b- Running RMS velocities
% RMS criterion (signal at every (y,z) location)
```

```
% running RMS over time (over cycles nc)
u_rrms = zeros(size(u(:,:,:,1:n_cycles_max,:)));
v_rrms = zeros(size(v(:,:,:,1:n_cycles_max,:)));
w_rrms = zeros(size(w(:,:,:,1:n_cycles_max,:)));
for nc = 1 : n_cycles_max
    u_rrms(:,:,:,nc,:) = nanmean(nanstd(u(:,:,:,1:nc,:),0,4),1);
    v_rrms(:,:,:,nc,:) = nanmean(nanstd(v(:,:,:,1:nc,:),0,4),1);
    w_rrms(:,:,:,nc,:) = nanmean(nanstd(w(:,:,:,1:nc,:),0,4),1);
end

% 2c- 2D plots of running RMS for several phases at one particular location (X,Y,Z), in the u-nc plane
% Plot for all locations
p = menu('Plot of running RMS?','Yes','No');

if p == 1
    s = menu('Save pictures?','Yes','No');

    % X_case, Y_case, Z_case can be single values or vectors
    c = menu('Number of cases?','One','All');
    if c == 1
        X_cases = X_case;
        Y_cases = Y_case;
        Z_cases = Z_case;
    else
        X_cases = 1:numel(x);
        Y_cases = 1:numel(y);
        Z_cases = 1:numel(z);
    end

    n_lines = 5;                                % Number of phases to plot
    n_lines = min(n_lines, numel(t));

    figure();
    hp1 = subplot(3,1,1); hold on;
    hp2 = subplot(3,1,2); hold on;
    hp3 = subplot(3,1,3); hold on;
    linkaxes([hp1 hp2 hp3],'x');
    set(gcf,'position',[1 80 1024 600]);           % Maximizes figure window

    % Loop through all files requested
    for i = 1 : numel(X_cases)
        for j = 1 : numel(Y_cases)
            for k = 1 : numel(Z_cases)

                leyenda1 = [];
                cla(hp1);
                cla(hp2);
                cla(hp3);
                tp = t;
                np = 1:n_cycles_max;

                for h = 1 : roundn(numel(tp)/n_lines,0) : numel(tp)     % loop in number of phases to
plot (n_lines)
```

```
                    up = squeeze(u_rrms(X_cases(i),Y_cases(j),Z_cases(k),np,h));
                    upp = squeeze(abs(uf(X_cases(i),Y_cases(j),Z_cases(k),np,h)));
```

```
                    plot(hp1,np,up,'-','color',[h/numel(tp) 0 1-h/numel(tp)]); % hold on;
                    handlep = plot(hp1,np,upp,'.','color',[h/numel(tp) 0 1-
h/numel(tp)],'MarkerSize',10); hold on;
set(get(get(handlep,'Annotation'),'LegendInformation'),'IconDisplayStyle','off'); % Exclude line from
legend
                    leyenda1 = [leyenda1 ; ['Phi = ',num2str(roundn((h-
1)/numel(tp)*360,0),'%03.0f'),'°']];

                    vp = squeeze(v_rrms(X_cases(i),Y_cases(j),Z_cases(k),np,h));
                    vpp = squeeze(abs(vf(X_cases(i),Y_cases(j),Z_cases(k),np,h)));
                    plot(hp2,np,vp,'-','color',[h/numel(tp) 0 1-h/numel(tp)]); % hold on;
                    handlep = plot(hp2,np,vpp,'.','color',[h/numel(tp) 0 1-
h/numel(tp)],'MarkerSize',10); hold on;
set(get(get(handlep,'Annotation'),'LegendInformation'),'IconDisplayStyle','off'); % Exclude line from
legend

                    wp = squeeze(w_rrms(X_cases(i),Y_cases(j),Z_cases(k),np,h));
                    wpp = squeeze(abs(wf(X_cases(i),Y_cases(j),Z_cases(k),np,h)));
                    plot(hp3,np,wp,'-','color',[h/numel(tp) 0 1-h/numel(tp)]); % hold on;
                    handlep = plot(hp3,np,wpp,'.','color',[h/numel(tp) 0 1-
h/numel(tp)],'MarkerSize',10); hold on;
set(get(get(handlep,'Annotation'),'LegendInformation'),'IconDisplayStyle','off'); % Exclude line from
legend

                end                     % end of loop in number of phases

                xlim([np(1)-1 np(end)+1]);
                legend(hp1,leyenda1);
                title(hp1,['Running RMS along the cycles, for several phases, at z = ',
num2str(z(Z_cases(k))*1000), ' mm']);
                xlabel(hp3,'NC (-)');
                ylabel(hp1,'u'' , u_R_M_S (m/s)');
                ylabel(hp2,'v'' , v_R_M_S (m/s)');
                ylabel(hp3,'w'' , w_R_M_S (m/s)');

                if s == 1
                    file_save = [dir_save,'\Running RMS along
cycles_z=',num2str(z(Z_cases(k))*1000),'mm_Fs=',num2str(Fs),',a=',num2str(a),'.jpg'];
                    saveas(gcf,file_save,'jpg');
                end

                if c == 2
                    input(['Plotting: Z_case = ', num2str(Z_cases(k)),'. Press enter to continue.']);
                end
            end
        end
    end         % Close loop through cases
    if c == 2; disp('End of plots'); end;
    set(gcf,'position',[228 228 560 420]);          % Back to original size in left window
end

% 2d- Percentage error (pe) in the running RMS (with respect the final value of RMS calculated with all
cycles available)
```

```
pe_u_rrms = zeros(size(u(:,:,:,1:n_cycles_max,:)));
```

```
pe_v_rrms = zeros(size(v(:,:,:,1:n_cycles_max,:)));
pe_w_rrms = zeros(size(w(:,:,:,1:n_cycles_max,:)));
```

```
for nc = 1 : n_cycles_max
    pe_u_rrms(:,:,:,nc,:) = abs((u_rrms(:,:,:,nc,:)-u_rms(:,:,:,1,:)))./u_rms(:,:,:,1,:);
    pe_v_rrms(:,:,:,nc,:) = abs((v_rrms(:,:,:,nc,:)-v_rms(:,:,:,1,:)))./v_rms(:,:,:,1,:);
    pe_w_rrms(:,:,:,nc,:) = abs((w_rrms(:,:,:,nc,:)-w_rms(:,:,:,1,:)))./w_rms(:,:,:,1,:);
end

% 2e- 2D plots of percent error for several phases at each particular location (X,Y,Z), in the pe-nc
plane
% Plot for all locations
p = menu('Plot of percent error in the running RMS?','Yes','No');

if p == 1
    s = menu('Save pictures?','Yes','No');

    % X_case, Y_case, Z_case can be single values or vectors
    c = menu('Number of cases?','One','All');
    if c == 1
        X_cases = X_case;
        Y_cases = Y_case;
        Z_cases = Z_case;
    else
        X_cases = 1:numel(x);
        Y_cases = 1:numel(y);
        Z_cases = 1:numel(z);
    end

    n_lines = 5;                                % Number of phases to plot
    n_lines = min(n_lines, numel(t));

    figure();
    hp1 = subplot(3,1,1); hold on;
    hp2 = subplot(3,1,2); hold on;
    hp3 = subplot(3,1,3); hold on;
    linkaxes([hp1 hp2 hp3],'x');
    set(gcf,'position',[1 80 1024 600]);            % Maximizes figure window

    % Loop through all files requested
    for i = 1 : numel(X_cases)
        for j = 1 : numel(Y_cases)
            for k = 1 : numel(Z_cases)

                leyenda1 = [];
                cla(hp1);
                cla(hp2);
                cla(hp3);
                tp = t;
                np = 1:n_cycles_max;

                for h = 1 : roundn(numel(tp)/n_lines,0) : numel(tp)     % loop in number of phases to
plot (n_lines)

                    up = squeeze(pe_u_rrms(X_cases(i),Y_cases(j),Z_cases(k),np,h)*100);
                    plot(hp1,np,up,'-','color',[0 h/numel(tp) 1-h/numel(tp)]); % hold on;

                    leyenda1 = [leyenda1 ; ['Phi = ',num2str(roundn((h-
1)/numel(tp)*360,0),'%03.0f'),'°']];
```

```
                    vp = squeeze(pe_v_rrms(X_cases(i),Y_cases(j),Z_cases(k),np,h)*100);
                    plot(hp2,np,vp,'-','color',[0 h/numel(tp) 1-h/numel(tp)]); % hold on;

                    wp = squeeze(pe_w_rrms(X_cases(i),Y_cases(j),Z_cases(k),np,h)*100);
                    plot(hp3,np,wp,'-','color',[0 h/numel(tp) 1-h/numel(tp)]); % hold on;
                end                     % end of loop in number of phases

                % Plot tolerance lines
                tolerance = 0.01;       % tolerance +/-, per unit.
                upp = ones(size(up))*tolerance*100;
                handlep = plot(hp1,np,upp,'-r');
                vpp = ones(size(vp))*tolerance*100;
                handlep = plot(hp2,np,vpp,'-r');
                wpp = ones(size(wp))*tolerance*100;
                handlep = plot(hp3,np,wpp,'-r');

                % Axes decoration
                ylim(hp1,[0,20]);
                ylim(hp2,[0,20]);
                ylim(hp3,[0,20]);
                xlim([np(1)-1 np(end)+1]);
                legend(hp1,leyenda1);
                title(hp1,['Percent error for the running RMS along the cycles, at z = ',
num2str(z(Z_cases(k))*1000), ' mm']);
                xlabel(hp3,'NC (-)');
                ylabel(hp1,'Error in u_R_M_S (%)');
                ylabel(hp2,'Error in v_R_M_S (%)');
                ylabel(hp3,'Error in w_R_M_S (%)');
                if s == 1
                    file_save = [dir_save,'\Running RMS_percent
error_z=',num2str(z(Z_cases(k))*1000),'mm_Fs=',num2str(Fs),',a=',num2str(a),'.jpg'];
                    saveas(gcf,file_save,'jpg');
                end
                if c == 2
                    input(['Plotting: Z_case = ', num2str(Z_cases(k)),'. Press enter to continue.']);
                end
            end
        end
    end         % Close loop through cases
    if c == 2; disp('End of plots'); end;
    set(gcf,'position',[228 228 560 420]);          % Back to original size in left window
end
% end of function
```

Published with MATLAB® 7.14

APPENDIX B

COMPUTER CODE FOR ANALYSIS OF LDV DATA (IN MATLAB©)

LDV_ANALYSIS

Program to read, analyze and plot data from text files. Applied to data from LDV measurements from experiments in the HydroLab.

Contents

Main routine

```
% LDV_ANALYSIS
% Program to read, analyze and plot data from text files.
% Applied to data from LDV measurements from experiments in the HydroLab.
%
% JM Mier
% Ven Te Chow Hydrosystems Laboratory, UIUC
% May 2009
% Revised: March 2012
%

% Files location
% Directories to read data from and save results into
dir_read = uigetdir('C:\USERS\JM MIER\Research\Thesis exps','Select Path for DATA:');
dir_save = uigetdir(dir_read,'Select Path for RESULTS:');

% Generate list of data files
file_traverse = dir([dir_read,'\*.txt']);
file_list = dir([dir_read,'\*.csv']);
file_matlab = dir([dir_read,'\*.mat']);

% Experiment conditions
% Display experiment name
dir_read_dividers = find(dir_read=='\');
disp(' ');
disp(['Flume:           ',dir_read(dir_read_dividers(end-3)+1:dir_read_dividers(end-2)-1)]);
disp(['Project:         ',dir_read(dir_read_dividers(end-2)+1:dir_read_dividers(end-1)-1)]);
disp(['Experiment:      ',dir_read(dir_read_dividers(end-1)+1:dir_read_dividers(end-0)-1)]);
disp(['LDV_Analysis:    ',dir_read(dir_read_dividers(end-0)+1:end)]);

% Nominal case values
disp(' ');
disp('Nominal case values:');
```

```
T_nom = input(' Period of oscillation (s) = ');          % Period of oscillation (s)
A_nom = input(' Half-amplitude of oscillation (m) = ');  % Half-amplitude of oscillation (m)
ro_nom = 1000;                                           % Density of water (kg/m3)
nu_nom = 10^(-6);                                        % Kinematic viscosity of water (m2/s)
Umax_nom = 2*pi*A_nom/T_nom;                             % Max velocity of the oscillation (m/s)
Rew_nom = Umax_nom*A_nom/nu_nom;                         % Wave Reynolds number
Tau_b_max_laminar_nom = ro_nom*Umax_nom^2/Rew_nom^0.5;  % Max bed shear stress in laminar conditions
(N/m2)

disp([' Umax_nom = ',num2str(roundn(Umax_nom,-3)),' m/s']);
disp([' Rew_nom = ',num2str(roundn(Rew_nom,3)),' = ',num2str(roundn(Rew_nom/10^6,-2)),' x10^6']);
disp([' Tau_b_max_laminar_nom = ',num2str(roundn(Tau_b_max_laminar_nom,-2)),' N/m2']);

% Physical parameters
disp(' ');
disp('Physical parameters:');

g = 9.81; %input(' Gravitational acceleration (m/s2) = ');                      % Gravitational
acceleration (m/s2)
Temp = input(' Temperature of fluid (°C) = ');                                  % Input fluid
temperature (°C)
ro = (1 - ((Temp+288.9414)*(Temp-3.9863)^2)/(508929.2*(Temp+68.12963)))*1000;   % Density of water
(kg/m3)
nu = 1.79*10^(-6)/(1+0.03368*Temp+0.00021*Temp^2);                              % Kinematic viscosity of
water (m2/s)

disp([' Density of water (ro) = ',num2str(roundn(ro,0)),' kg/m3']);
disp([' Kinematic viscosity of water (nu) = ',num2str(roundn(nu*10^6,-2)),' x10^-6  m2/s']);

% Read clean data
disp(' ');
disp('Loading data from .mat file...');
load([dir_read '\' file_matlab(1).name]);

% Use corrected z=0 level
% When reading data from text files, the correction may have been already applied in the 'Test grid.txt'
file
z = z - z0;

% Get sampling frequency of the data set and cycle duration
disp([' Max number of cycles (Nmax) = ',num2str(roundn(max(max(max(n_cycles))),0)),' cycles']);
disp([' Cycle duration (Ts) = ',num2str(roundn(Ts,0)),' s']);
disp([' Sampling Frequency (Fs) = ',num2str(roundn(Fs,0)),' Hz']);

% Phase conversion
% Convert even time stamps into degrees of phase
phi_delta = 1/Fs*360/Ts;
disp([' Phase increment available: ',num2str(phi_delta),'°']);
phi_ini = input([' Initial phase at trigger (-180° to +180° , multiple of ',num2str(phi_delta),'°) =
']);            % Phase = 0° is the upward zero crossing of the pistons (max horizontal velocity to the
+X)
phi = roundn(interp1([0,Ts],[phi_ini,phi_ini+360],t(1,:)),-2);      % Round to avoid decimal errors
phi = phi + 360*(phi<0);
phi = phi - 360*(phi>=360);
disp([' Phases available: ',num2str(phi)]);
```

```
% Shift 5D matrices to get phi=0 at the first element of the "phi" dimension
% (avoids problems with contour plots)
phi_shift = find(phi<circshift(phi,[0 1]))-1;
phi = circshift(phi,[0 -phi_shift]);
u = circshift(u,[0 0 0 0 -phi_shift]);
v = circshift(v,[0 0 0 0 -phi_shift]);
w = circshift(w,[0 0 0 0 -phi_shift]);

% Data analysis parameters
% Select distance for outer flow
Zout = input(' Set distance from bottom to consider outer flow conditions (in meters): '); %0.1;  %
Distance from bottom to consider outer flow conditions (m)
Zout_loc = find(roundn(z,-3)>=Zout,1,'first');

% Select minimum number of points needed for averages and standard deviations
n_min_ave = 3;            % Minimum number of points in a series to compute mean values. Otherwise will
give NaN. (From observation, 1 is enough, 3 is ok, 10 is good).
n_min_std = 5;            % Minimum number of points in a series to compute rms values. Otherwise will
give NaN. (From observation, 2 is enough, 5 is ok, 10 is good).
n_min_skew = 7;           % Minimum number of points in a series to compute skewness values. Otherwise
will give NaN. (From observation, 3 is enough, 7 is ok, 10 is good).
% ** n_min_std NEEDS TO BE AT LEAST 2!! ** (Otherwise, if there is only 1 data point, std calculation
will give 0, which is not good)
% ** n_min_skew NEEDS TO BE AT LEAST 3!! ** (Otherwise, if there is only 2 data points, skew calculation
will give 0, which is not good)
```

MEAN FLOW ANALYSIS

```
% MEAN FLOW ANALYSIS
% Ensemble average in cycles (nc). Dimensions remaining: (x,y,z,1,phi)
% Spatial average in longitudinal (x). Dimensions remaining: (1,y,z,1,phi)

% Mean Velocities
u_mean = nanmean(nanmean(u,4),1); u_mean(sum(~isnan(u),4)< n_min_ave) = NaN;     % Second statement
included to avoid the calculation of MEAN in data sets with very few data points (which could bias all
subsequent calculations).
v_mean = nanmean(nanmean(v,4),1); v_mean(sum(~isnan(v),4)< n_min_ave) = NaN;     % Second statement
included to avoid the calculation of MEAN in data sets with very few data points (which could bias all
subsequent calculations).
w_mean = nanmean(nanmean(w,4),1); w_mean(sum(~isnan(w),4)< n_min_ave) = NaN;     % Second statement
included to avoid the calculation of MEAN in data sets with very few data points (which could bias all
subsequent calculations).

% Outer flow velocity
u_mean_out = nanmean(u_mean(:,:,Zout_loc+1:end,:,:),3);                % Outer flow velocity
u_mean_out_ref = Umax_nom*sind(phi);                                   % Reference velocity cycle from
sine wave

% Max velocity of the outer flow (m/s)
u_mean_out_max = (max(u_mean_out)+abs(min(u_mean_out)))/2;

% Phase of Umax in outer flow (gives 2 values, one for each half-cycle)
phi_loc_u_out_max = [find(u_mean_out==max(u_mean_out)) , find(u_mean_out==min(u_mean_out))];
```

TURBULENCE ANALYSIS

```
% TURBULENCE ANALYSIS
% Ensemble calculations in cycles (nc). Dimensions remaining: (x,y,z,1,phi)
% Spatial average in longitudinal (x). Dimensions remaining: (1,y,z,1,phi)

% Velocity fluctuations (u',v',w')
% uf = u - u_mean
% Calculation of fluctuating velocities (u',v',w') before any ensemble averaging
uf = u - repmat(u_mean,[size(u,1) 1 1 size(u,4) 1]);
vf = v - repmat(v_mean,[size(v,1) 1 1 size(v,4) 1]);
wf = w - repmat(w_mean,[size(w,1) 1 1 size(w,4) 1]);

% RMS Velocities
% u_rms = sqrt(mean(uf^2)) = std(u)
u_rms = nanmean(nanstd(u,0,4),1); u_rms(sum(~isnan(u),4)< n_min_std) = NaN;     % Second statement
included to avoid the calculation of STD in data sets with very few data points (when only 1 data point,
we were getting u_rms = 0, which biases all subsequent calculations).
v_rms = nanmean(nanstd(v,0,4),1); v_rms(sum(~isnan(v),4)< n_min_std) = NaN;     % Second statement
included to avoid the calculation of STD in data sets with very few data points (when only 1 data point,
we were getting u_rms = 0, which biases all subsequent calculations).
w_rms = nanmean(nanstd(w,0,4),1); w_rms(sum(~isnan(w),4)< n_min_std) = NaN;     % Second statement
included to avoid the calculation of STD in data sets with very few data points (when only 1 data point,
we were getting u_rms = 0, which biases all subsequent calculations).

% Turbulence Intensity
% uf2_mean = mean(uf x uf)
uf2_mean = nanmean(nanmean(uf.*uf,4),1); uf2_mean(sum(~isnan(uf.*uf),4)< n_min_ave) = NaN;     % Second
statement included to avoid the calculation of MEAN in data sets with very few data points (which could
bias all subsequent calculations).
vf2_mean = nanmean(nanmean(vf.*vf,4),1); vf2_mean(sum(~isnan(vf.*vf),4)< n_min_ave) = NaN;     % Second
statement included to avoid the calculation of MEAN in data sets with very few data points (which could
bias all subsequent calculations).
wf2_mean = nanmean(nanmean(wf.*wf,4),1); wf2_mean(sum(~isnan(wf.*wf),4)< n_min_ave) = NaN;     % Second
statement included to avoid the calculation of MEAN in data sets with very few data points (which could
bias all subsequent calculations).

% Turbulent Kinetic Energy
% tke = 1/2(uf2_mean + vf2_mean + wf2_mean)
% e = ro*tke = 1/2*ro*(uf2_mean + vf2_mean + wf2_mean)
% With the formula below, 'tke' will be calculated when at least one of the three components has a non-
Nan value, however, that could introduce some bias, since we don't know how much the NaN components
could contribute.
tke = 1/2*nansum(cat(numel(size(uf2_mean))+1, uf2_mean , vf2_mean , wf2_mean),numel(size(uf2_mean))+1);
% When using the 'nansum' formula, we need to make a provision for the case when all three components
are NaNs, so that the result won't be '0', but NaN.
tke(isnan(uf2_mean) & isnan(vf2_mean) & isnan(wf2_mean)) = NaN;

% Reynolds Shear Stress
% Rij = -ro x mean(ufi x ufj)
uf_vf_mean = nanmean(nanmean(uf.*vf,4),1); uf_vf_mean(sum(~isnan(uf.*vf),4)< n_min_ave) = NaN;     %
Second statement included to avoid the calculation of MEAN in data sets with very few data points (which
could bias all subsequent calculations).
uf_wf_mean = nanmean(nanmean(uf.*wf,4),1); uf_wf_mean(sum(~isnan(uf.*wf),4)< n_min_ave) = NaN;     %
Second statement included to avoid the calculation of MEAN in data sets with very few data points (which
could bias all subsequent calculations).
```

```
vf_wf_mean = nanmean(nanmean(vf.*wf,4),1); vf_wf_mean(sum(~isnan(vf.*wf),4)< n_min_ave) = NaN;     %
Second statement included to avoid the calculation of MEAN in data sets with very few data points (which
could bias all subsequent calculations).

Rxy = -ro * uf_vf_mean;
Rxz = -ro * uf_wf_mean;
Ryz = -ro * vf_wf_mean;

% Viscous Shear Stress
% tau_v = ro x nu x dui_mean/dxj
% Only XZ is considered (which produces shear stress in the direction of the main flow)
du_mean_dz = (circshift(u_mean,[0 0 -1 0 0])-circshift(u_mean,[0 0 1 0
0])).//(circshift(repmat(shiftdim(z,-2),[1 numel(y) 1 1 numel(phi)]),[0 0 -1 0 0])-
circshift(repmat(shiftdim(z,-2),[1 numel(y) 1 1 numel(phi)]),[0 0 1 0 0]));     % Central scheme: dudz =
(u_zp1-u_zm1)/(zp1-zm1). Need to make a correction for z=1 and z=end.
du_mean_dz(:,:,1,:,:) = (u_mean(:,:,2,:,:)-u_mean(:,:,1,:,:))./(z(2)-z(1));
% Correction for z=1.
du_mean_dz(:,:,end,:,:) = (u_mean(:,:,end,:,:)-u_mean(:,:,end-1,:,:))./(z(end)-z(end-1));
% Correction for z=end.
tau_v_xz = ro*nu*du_mean_dz;

% Total Shear Stress (viscous + turbulent components)
% tau = tau_v + Rij
% Only XZ is considered (which produces shear stress in the direction of the main flow)
% With the formula below, 'tau' will be calculated when at least one of the two components has a non-Nan
value, however, that could introduce some bias, since we don't know how much the NaN components could
contribute.
tau_xz_comp = nansum(cat(numel(size(tau_v_xz))+1, tau_v_xz , Rxz),numel(size(tau_v_xz))+1);
% When using the 'nansum' formula, we need to make a provision for the case when all two components are
NaNs, so that the result won't be '0', but NaN.
tau_xz_comp(isnan(tau_v_xz) & isnan(Rxz)) = NaN;

% Turbulence Production
% Pij = - mean(ufi x ufj) x dui_mean/dxj
% Only XZ is considered (which produces turbulence in the direction of the main flow)
P_xz = - uf_wf_mean .* du_mean_dz;

% Eddy viscosity
% nu_t_ij = - mean(ufi x ufj) / (dui_mean/dxj + duj_mean/dxi)
% Only XZ is considered (which produces turbulence in the direction of the main flow)
nu_t_xz = - uf_wf_mean ./ du_mean_dz;

% Remove nu_t from places with very small gradient to avoid unrealistically large values
nu_t_xz(abs(du_mean_dz)<max(max(abs(du_mean_dz)))*0.0001) = NaN;
nu_t_xz(:,:,z<0.0001,:,:) = NaN;

% Remove nu_t for z locations outside boundary layer
% Find trace of du/dz=0 lines
du_dz_zero_pos_matrix = squeeze(sign(du_mean_dz).*sign(circshift(du_mean_dz,[0 0 1 0 0])));
delta_max_pos = find(z<=delta_max,1,'last');
for h = 1 : numel(phi)
    du_dz_zero_pos = find(du_dz_zero_pos_matrix(2:end,h)==-1)+1;
    du_dz_zero_pos_first(1,h) = min(du_dz_zero_pos(1),delta_max_pos+3);
    du_dz_zero_pos_second(1,h) =
min([du_dz_zero_pos(du_dz_zero_pos>du_dz_zero_pos_first(1,h)+9);delta_max_pos+3]);
end
du_dz_zero_plot = z(max([du_dz_zero_pos_first;du_dz_zero_pos_second],[],1)-1);
```

```
% Remove nu_t for z locations outside boundary layer
for h = 1 : numel(phi)
    nu_t_xz(:,:,z>du_dz_zero_plot(h),:,h) = NaN;
end

% Skewness of velocity fluctuations
% Using the standard definition of skewness (it gives the same result whether using u or uf)
u_skew = nanmean(skewness(u,1,4),1); u_skew(sum(~isnan(u),4)< n_min_skew) = NaN;     % Second statement
included to avoid the calculation of STD in data sets with very few data points (when only 1 data point,
we were getting u_rms = 0, which biases all subsequent calculations).
v_skew = nanmean(skewness(v,1,4),1); v_skew(sum(~isnan(v),4)< n_min_skew) = NaN;     % Second statement
included to avoid the calculation of STD in data sets with very few data points (when only 1 data point,
we were getting u_rms = 0, which biases all subsequent calculations).
w_skew = nanmean(skewness(w,1,4),1); w_skew(sum(~isnan(w),4)< n_min_skew) = NaN;     % Second statement
included to avoid the calculation of STD in data sets with very few data points (when only 1 data point,
we were getting u_rms = 0, which biases all subsequent calculations).
```

BED SHEAR STRESS CALCULATION

```
% BED SHEAR STRESS
% tau_b
% Only XZ is considered (which produces shear stress in the direction of the main flow)

% Aux variables
% Phases near u_max
phi_loc_u_high = (abs(squeeze(u_mean_out(:,1,:,:,:)))>max(squeeze(u_mean_out(:,1,:,:,:)))/2);

% A) From extrapolation of tau down to the bottom
%-- A4) Evaluate tau at a particular location near the bottom
Zb_loc = input(' Select Z location from the bottom to evaluate Tau_b from Tau_xz: ');             % Z
location where to evaluate tau_b
Zb = mean(z(Zb_loc));         % Elevation of bottom where to evaluate tau_b (m)
tau_b_xz_comp = nanmean(tau_xz_comp(:,:,Zb_loc,:,:),3);

% C) From fit of u* to linear profiles, then tau_b = ro x u*2
tau_b_xz_linearfit = zeros(1,numel(y),1,1,numel(phi));
for j = 1 : numel(y)

    % Plot bottom points of u_mean profile
    figure1 = figure();
    for h = 1 : numel(phi_plot)
        phi_pos = find(phi==phi_plot(h));
        u_prof = squeeze(u_mean(:,j,:,:,phi_pos));
        figure(figure1);
        hp = plot(u_prof,z,'o','MarkerSize',2); hc{h} = get(hp,'Color'); hold all;
        pause(0.1);
    end
    figure(figure1);
    legend(num2str(phi_plot'),4);
    title('Mean Velocity profiles (X\_av , NC\_av)  -  U_m_e_a_n (m/s)');
    xlabel('U_m_e_a_n (m/s)');
    ylabel('z (m)');
    ylim([0 0.001]);
    xlim([min([0,get(gca,'XLim')]) , max([0,get(gca,'XLim')])]);
```

```
    % Linear fit to the bottom points in profile
    % Such that u = m*z + b (in theory, b=0 and m=u*2/nu)
    Zloc1_fit = input(' Select first Z location from the bottom to include for linear fit: ');
    Zloc2_fit = input(' Select last Z location from the bottom to include for linear fit: ');

    cfit1 = cell(numel(phi),1);
    gof1 = cell(numel(phi),1);
    m1 = zeros(1,numel(phi));
    b1 = zeros(1,numel(phi));
    for h = 1 : numel(phi)
        zfit = z(Zloc1_fit:Zloc2_fit);
        ufit = squeeze(u_mean(:,j,Zloc1_fit:Zloc2_fit,:,h));
        if sum(~isnan(ufit)) < 2    % To avoid error when doing fit with less than 2 data points
            m1(h) = NaN;
            b1(h) = NaN;
        else
            [cfit1{h},gof1{h}] = fit(zfit(~isnan(ufit)),ufit(~isnan(ufit)),'poly1');
            m1(h) = cfit1{h}.p1;
            b1(h) = cfit1{h}.p2;
        end
        tau_b_xz_linearfit(:,j,:,:,h) = ro*nu*m1(h);
    end

    % Plot fit lines with data
    for h = 1 : numel(phi_plot)
        phi_pos = find(phi==phi_plot(h));
        z_lines_prof = [0;z];
        u_lines_prof = (m1(phi_pos).*z_lines_prof + b1(phi_pos));
        figure(figure1);
        hpp = plot(u_lines_prof,z_lines_prof,'-','MarkerSize',2,'Color',hc{h});
set(get(get(hpp,'Annotation'),'LegendInformation'),'IconDisplayStyle','off'); % Exclude line from
legend);
        pause(0.1);
    end
    my_saveas(gcf,[dir_save '\Tau_b_linearfit calculation'],'jpg');
end

% D) From fit of u* to logarithmic profiles, then tau_b = ro x u*2
% In a U_mean vs logZ plot, the logarithmic region should show as a
% straight line.
% In oscillatory flow, the straight portion of the velocity profile
% happens at different Z for each phase, so we need to find a way to find
% that Z automatically.
% We do that by looking for the Z location where the slope (using logZ)
% becomes constant (i.e. the curvature is zero).

tau_b_xz_logfit = zeros(1,numel(y),1,1,numel(phi));
for j = 1 : numel(y)

    % Plot points of u_mean profile (log scale)
    figure1 = figure();
    for h = 1 : numel(phi_plot)
        phi_pos = find(phi==phi_plot(h));
        u_prof = squeeze(u_mean(:,j,:,:,phi_pos));
        figure(figure1);
        hp = plot(u_prof,z,'o','MarkerSize',2); hc{h} = get(hp,'Color'); hold all;
```

```
        pause(0.1);
    end
    figure(figure1);
    legend(num2str(phi_plot'),4);
    title('Mean Velocity profiles (X\_av , NC\_av)  -  U_m_e_a_n (m/s)');
    xlabel('U_m_e_a_n (m/s)');
    ylabel('z (m)');
    set(gca,'YScale','log');
    ylim([0.0001 roundn(max(z),-2)]);
    xlim([min([0,get(gca,'XLim')]) , max([0,get(gca,'XLim')])]);
    drawnow;

    % Slope of mean velocity profiles in logZ scale
    du_mean_dzlog = (circshift(u_mean,[0 0 -1 0 0])-circshift(u_mean,[0 0 1 0
0])).//(circshift(repmat(shiftdim(log(z),-2),[1 numel(y) 1 1 numel(phi)]),[0 0 -1 0 0])-
circshift(repmat(shiftdim(log(z),-2),[1 numel(y) 1 1 numel(phi)]),[0 0 1 0 0]));     % Central scheme:
dudz = (u_zp1-u_zm1)/(zp1-zm1). Need to make a correction for z=1 and z=end.
    du_mean_dzlog(:,:,1,:,:) = (u_mean(:,:,2,:,:)-u_mean(:,:,1,:,:))./(log(z(2))-log(z(1)));
% Correction for z=1.
    du_mean_dzlog(:,:,end,:,:) = (u_mean(:,:,end,:,:)-u_mean(:,:,end-1,:,:))./(log(z(end))-log(z(end-
1)));
% Correction for z=end.

    % Curvature of mean velocity profiles in logZ scale
    du_mean_dz2log = (circshift(du_mean_dzlog,[0 0 -1 0 0])-circshift(du_mean_dzlog,[0 0 1 0
0])).//(circshift(repmat(shiftdim(log(z),-2),[1 numel(y) 1 1 numel(phi)]),[0 0 -1 0 0])-
circshift(repmat(shiftdim(log(z),-2),[1 numel(y) 1 1 numel(phi)]),[0 0 1 0 0]));     % Central scheme:
dudz = (u_zp1-u_zm1)/(zp1-zm1). Need to make a correction for z=1 and z=end.
    du_mean_dz2log(:,:,1,:,:) = (du_mean_dzlog(:,:,2,:,:)-du_mean_dzlog(:,:,1,:,:))./(log(z(2))-
log(z(1)));
% Correction for z=1.
    du_mean_dz2log(:,:,end,:,:) = (du_mean_dzlog(:,:,end,:,:)-du_mean_dzlog(:,:,end-
1,:,:))./(log(z(end))-log(z(end-1)));
% Correction for z=end.

    % Logarithmic fit to the points in profile
    % Such that u = m*ln(z) + b (in theory: u/u* = 1/K ln(z/k0) + Bs, so b = -u*/K ln(k0) + u*Bs  and  m
= u*/K , with K=0.41 von Karman)
    cfit1 = cell(numel(phi),1);
    gof1 = cell(numel(phi),1);
    m1 = zeros(1,numel(phi));
    b1 = zeros(1,numel(phi));
    for h = 1 : numel(phi)
        % Method 2 for Zloc_fit
        % Zloc_fit are different for every phase and are obtained from the points between curvature
peaks
        curvature_profile = squeeze(du_mean_dz2log(:,j,:,:,h))*sign(squeeze(u_mean_out(:,j,:,:,h)));
        curvature_peaks = (curvature_profile(1:Zout_loc-1)<circshift(curvature_profile(1:Zout_loc-1),-1)
& curvature_profile(1:Zout_loc-1)<circshift(curvature_profile(1:Zout_loc-1),1));
        Zloc_fit = 1 + find(curvature_peaks(2:end-1));      % Removes possible false peak detected at
bottom or top of profile
        if isempty(Zloc_fit); Zloc_fit = 1; end;            % Just to avoid errors when it doesn't find
any curvature peaks
        Zloc1_fit = Zloc_fit(min(1,numel(Zloc_fit))); Zloc1_fit = max(Zloc1_fit,1);
        Zloc2_fit = Zloc_fit(min(2,numel(Zloc_fit))); Zloc2_fit = min(Zloc2_fit,numel(z));

        zfit = log(z(Zloc1_fit:Zloc2_fit));
```

```
        ufit = squeeze(u_mean(:,j,Zloc1_fit:Zloc2_fit,:,h));
        if sum(~isnan(ufit)) < 2    % To avoid error when doing fit with less than 2 data points
            m1(h) = NaN;
            b1(h) = NaN;
        else
            [cfit1{h},gof1{h}] = fit(zfit(~isnan(ufit)),ufit(~isnan(ufit)),'poly1');
            m1(h) = cfit1{h}.p1;
            b1(h) = cfit1{h}.p2;
        end
        tau_b_xz_logfit(:,j,:,:,h) = sign(m1(h))*ro*(0.41*m1(h))^2;
    end

    % Plot fit lines with data
    for h = 1 : numel(phi_plot)
        phi_pos = find(phi==phi_plot(h));
        z_lines_prof = [0;z];
        u_lines_prof = (m1(phi_pos).*log(z_lines_prof) + b1(phi_pos));
        figure(figure1);
        hpp = plot(u_lines_prof,z_lines_prof,'-','MarkerSize',2,'Color',hc{h});
set(get(get(hpp,'Annotation'),'LegendInformation'),'IconDisplayStyle','off'); % Exclude line from
legend);
        pause(0.1);
    end
    my_saveas(gcf,[dir_save '\Tau_b_logfit calculation'],'jpg');

end

% Use this fit only for phases near the velocity maximum (it doesn't work near the flow reversal)
tau_b_xz_logfit(:,:,:,:,~phi_loc_u_high) = NaN;

% E) Average with all methods
tau_b_xz_avg = nanmean(cat(6,tau_b_xz_comp,tau_b_xz_linearfit),6);

% Phase of Tau_b_max (gives 2 values, one for each half-cycle)
phi_loc_tau_b_max = [find(tau_b_xz_avg==max(tau_b_xz_avg)) , find(tau_b_xz_avg==min(tau_b_xz_avg))];
```

BOUNDARY LAYER PARAMETERS

```
% BOUNDARY LAYER PARAMETERS
% Boundary layer thickness (delta)
% Defined as the height where the velocity peak is in a vertical profile
velocity_peak_right = zeros(1,numel(phi));
velocity_peak_left = zeros(1,numel(phi));
Zloc_peak_right = zeros(1,numel(phi));
Zloc_peak_left = zeros(1,numel(phi));
delta_right = zeros(1,numel(phi));
delta_left = zeros(1,numel(phi));
for h = 1 : numel(phi)
    velocity_profile = squeeze(u_mean(:,j,:,:,h)); %*sign(squeeze(u_mean_out(:,1,:,:,h)));
    [velocity_peak_right(h),Zloc_peak_right(h)] = max(velocity_profile(1:Zout_loc-1));
    [velocity_peak_left(h),Zloc_peak_left(h)] = min(velocity_profile(1:Zout_loc-1));
    delta_right(h) = z(Zloc_peak_right(h));
    delta_left(h) = z(Zloc_peak_left(h));
end
```

```
% Take delta only when velocity peak is positive (for delta right) or negative (for delta left)
delta_right(velocity_peak_right<0) = NaN;
delta_left(velocity_peak_left>0) = NaN;

% Delta can not get thinner after reaching all the way out
delta_right(find(Zloc_peak_right==Zout_loc-1,1,'last')+1:find(isnan(delta_right),1,'first')) = NaN;
delta_left([1:find(isnan(delta_left),1,'first') , find(Zloc_peak_left==Zout_loc-1,1,'last')+1:end]) =
NaN;

% Reference delta is computed at the phase of max u_out:
delta_u_out_max = (delta_right(u_mean_out==max(u_mean_out)) +
delta_left(u_mean_out==min(u_mean_out)))/2;
delta_max = (max(delta_right) + max(delta_left))/2;

% Vertical scaling in boundary layer
% z_scale_out = delta
z_scale_out = zeros(size(u_mean_out));
z_scale_out(:,1,:,:,1:numel(phi)) = max([delta_right;delta_left],[],1);

% Shear Velocity
% Classic tau_b = rho x u*^2, so u* = sqrt(tau_b/rho)

% A) From tau_b (visc + turb) components
u_star_comp = sign(tau_b_xz_comp).*(abs(tau_b_xz_comp)/ro).^0.5;
u_star_comp_max = (max(u_star_comp)+abs(min(u_star_comp)))/2;

% C) From fit to linear profiles
u_star_linearfit = sign(tau_b_xz_linearfit).*(abs(tau_b_xz_linearfit)/ro).^0.5;
u_star_linearfit_max = (max(u_star_linearfit)+abs(min(u_star_linearfit)))/2;

% D) From fit to logarithmic profiles
u_star_logfit = sign(tau_b_xz_logfit).*(abs(tau_b_xz_logfit)/ro).^0.5;
u_star_logfit_max = (max(u_star_logfit)+abs(min(u_star_logfit)))/2;

% E) Average with all methods
% Due to the nature of the oscillatory flow, the values of u* from the
% logarithmic fit will have a lag with respect to the values obtained near
% the bed.
% If the logarithmic fit is included to compute the average, the phase
% should be shifted according to the phase difference between the peaks.
u_star_avg = sign(tau_b_xz_avg).*(abs(tau_b_xz_avg)/ro).^0.5;
u_star_avg_max = (max(u_star_avg)+abs(min(u_star_avg)))/2;

% Phases near u_star_max
phi_loc_u_star_high = (abs(squeeze(u_star_avg(:,1,:,:,:)))>max(squeeze(u_star_avg(:,1,:,:,:)))/2);

% Wave friction factor
% Typically Tau_b = ro u*^2, but also Tau_b = ro Cf U^2, where Cf is friction coefficient.
% In wave boundary layers fw = 2Cf, so fw = 2 (u*/U)^2
% It changes on every phase!!
fw_avg = 2.*(u_star_avg./u_mean_out).^2;

% Discard points near u=0 to avoid division by '0'
fw_avg(:,:,:,:,~phi_loc_u_high) = NaN;

% Reference fw is computed at the phase of max u_star:
```

```
fw_avg_u_star_max = (fw_avg(u_star_avg==max(u_star_avg)) + fw_avg(u_star_avg==min(u_star_avg)))/2;
fw_avg_u_out_max = (fw_avg(u_mean_out==max(u_mean_out)) + fw_avg(u_mean_out==min(u_mean_out)))/2;

% Vertical scaling in viscous sublayer
% z_scale_in = nu/u*
% It changes on every phase!!
z_scale_in_avg = nu./abs(u_star_avg);

% Discard points near u_star=0 to avoid division by '0'
z_scale_in_avg(:,:,:,:,~phi_loc_u_star_high) = NaN;

% Reference Zv is computed at the phase of max u_star:
z_scale_in_avg_u_star_max = (z_scale_in_avg(u_star_avg==max(u_star_avg)) +
z_scale_in_avg(u_star_avg==min(u_star_avg)))/2;

% Viscous sublayer thickness (delta_v)

%-- Method 1: Using the vicous length scale
% delta_v = 11.6 z_scale_in = 11.6 nu/u*    % Divide between viscous and log-law zones in steady flows
% delta_v = 5 z_scale_in = 5 nu/u*          % Purely viscous zone limit

% This method proved not to give good results for oscillatory flows,
% because the flow is unsteady and the viscous sublayer develops
% differently than in steady flows, which is what this method is based on.

delta_v_116 = 11.6.*squeeze(z_scale_in_avg(:,1,:,:,:))';
delta_v_5 = 5.*squeeze(z_scale_in_avg(:,1,:,:,:))';

%-- Method 2: from linear velocity profiles near bottom
% Instead of calculating it using the viscous length scale (nu/u*), which
% may not apply for oscillatory flows??
% So we find the distance from the bottom where velocity profiles are linear
% for each phase, which is also where d(tau)/dz = 0

% This method didn't work well, since the velocity profiles may not be
% linear near the bottom even though we might be inside the viscous
% sublayer.

%-- Method 3: From oscillatory flow theory (laminar regime only)
% From Hino et al. (1983), it is mentioned that the thickness of viscous
% sublayer can be predicted in the laminar regime from Rayleigh theory,
% being proportional to (nu*t)^0.5

% In this formula the time 't' needs to be adjusted to the starting point
% of the cycle for the near bed layer. In the case of the pure laminar
% regime goes 45° in advance of the outer flow, for other Rew, an
% appropriate value needs to be used.

% Find the zero crossing of u_star, which indicates more or less the starting point of the cycle near
bed
% Assumes u_star positive for first half-cycle and negative for second half-cycle
phi_pos = find(squeeze(u_star_avg(:,1,:,:,:))'<=0 & phi>=90 & phi<=270 &
~isnan(squeeze(u_star_avg(:,1,:,:,:))'),1,'first');
phi_pos_m1 = find(squeeze(u_star_avg(:,1,:,:,:))'>=0 & phi>=90 & phi<=270 &
~isnan(squeeze(u_star_avg(:,1,:,:,:))'),1,'last');
```

```
phi_dif_Rayleigh =
interp1q(squeeze(u_star_avg(:,1,:,:,[phi_pos,phi_pos_m1])),phi([phi_pos,phi_pos_m1])',0) - 180;
% Calculate delta_v
delta_v_Rayleigh(1,:) = (nu*(phi-phi_dif_Rayleigh-0)/360*T).^0.5;
delta_v_Rayleigh(2,:) = (nu*(phi-phi_dif_Rayleigh-180)/360*T).^0.5;
delta_v_Rayleigh(3,:) = (nu*(phi-phi_dif_Rayleigh-360)/360*T).^0.5;
delta_v_Rayleigh(imag(delta_v_Rayleigh)~=0) = NaN;
delta_v_Rayleigh(:,find(isnan(delta_v_Rayleigh(1,:)),1,'last')) = NaN;
delta_v_Rayleigh(:,find(isnan(delta_v_Rayleigh(2,:)),1,'last')) = NaN;
delta_v_Rayleigh(:,find(isnan(delta_v_Rayleigh(3,:)),1,'last')) = NaN;
delta_v_Rayleigh_all = min(delta_v_Rayleigh,[],1);

%-- Method 4: Using the definition of viscous sublayer
% Go back to the basic definition of viscous sublayer, where viscous forces
% are important compared to turbulent forces.
% Use a ratio of viscous force (from tau_v) to turbulent force (from tke),
% and establish an appropriate threshold that determines where the viscous
% zone limit can be set.

ratio_VT = tau_v_xz./(0.5*ro*uf2_mean);    % Using only the tke from U component

% The value of the ratio_VT that defines the viscous zone limit can be
% calculated from an analogy with the criteria used in steady flows:
% From steady flows (Nezu & Nakagawa, 1993), for z+=5 (pure viscous zone limit), we have that
u_rms/u_star = 1.8
% Also, near the bed we can say that tau_v ~= ro*u_star^2
% So, ratio_VT = tau_v / tke ~= ro*u_star^2 / (1/2*ro*(u_rms^2+v_rms^2+w_rms^2)). For full tke, with 3
components.
% So, ratio_VT = tau_v / tke_u ~= ro*u_star^2 / (1/2*ro*(u_rms^2)). A simplification when only u_rms is
known near bed.
% This results in ratio_VT_threshold = 2*(u_star/u_rms)^2 = 0.62
ratio_VT_threshold = 0.62;

%Find line of ratio_VT_threshold, starting from the bed
ratio_VT_line = zeros(2,numel(phi));
for line = 1 : 2    % One line for each half-cycle
ratio_VT = -ratio_VT;   % Change sign of ratio, to detect both lines independently, when flow is going
right and going left. This will allow the lines to overlap for certain phases.
for h = 1 : numel(phi)
    line_pos = find(squeeze(ratio_VT(:,1,:,:,h))<=ratio_VT_threshold &
[1;squeeze(ratio_VT(:,1,2:end,:,h))-squeeze(ratio_VT(:,1,1:end-1,:,h))<=0] &
squeeze(ratio_VT(:,1,:,:,h))>=0,1,'first');
    line_pos_m1 = line_pos - 1;
    if line_pos_m1==0; line_pos_m1 = 1;
    elseif all(isnan(ratio_VT(:,1,1:line_pos_m1,:,h))); line_pos_m1 = NaN;
    else line_pos_m1 = find(~isnan(ratio_VT(:,1,1:line_pos_m1,:,h)),1,'last');
    end
    %ratio_VT_line(h) = z(line_pos);
    if isnan(line_pos_m1); ratio_VT_line(line,h) = NaN;
    else ratio_VT_line(line,h) = interp1q(ratio_VT(:,1,[line_pos,line_pos_m1],:,h),
z([line_pos,line_pos_m1]),ratio_VT_threshold);
    end
end
end
delta_v_ratio_VT = max(ratio_VT_line,[],1);
```

```
% Reference delta_v is computed at the phase of max u_star:
delta_v_ratio_VT_u_star_max = (delta_v_ratio_VT(u_star_avg==max(u_star_avg)) +
delta_v_ratio_VT(u_star_avg==min(u_star_avg)))/2;

% Phase difference between Tau_b_max and U_max
% Phase lead of Tau_b_max with respect to Umax (positive means Tau_b_max happens before Umax)
phi_lead_right_avg = phi(phi_loc_u_out_max(1))-phi(phi_loc_tau_b_max(1));
phi_lead_left_avg = phi(phi_loc_u_out_max(2))-phi(phi_loc_tau_b_max(2));
phi_lead_avg = (phi_lead_right_avg + phi_lead_left_avg)/2;
```

QUADRANT ANALYSIS

```
% QUADRANT ANALYSIS
%Analysis of u'w' based on 4 quadrants:
%Q1: u'>0 & w'>0 (outward interaction)
%Q2: u'<0 & w'>0 (ejection)
%Q3: u'<0 & w'<0 (inward interaction)
%Q4: u'>0 & w'<0 (sweep)
%In the case of oscillatory flow, there is a catch: quadrant analysis is
%based on the interactions of u' with respect to the mean flow u_mean. So
%during the phases of returning flow, a negative fluctuation (u'<0) will
%indeed go with the mean flow (u_mean<0), and so will be a Q1 or Q4, not Q2 or Q3.
%A threshold (H) is defined to consider relevant interactions, such that: |u'w'| > H(u'w')

% Calculate quadrant number and magnitude for each data point at each location
Qn = zeros(size(uf));       % Quadrant number
Qn(or(uf>0 & repmat(u_mean,[1,1,1,size(uf,4),1])>0 , uf<0 & repmat(u_mean,[1,1,1,size(uf,4),1])<0) &
wf>0) = 1;
Qn(or(uf<0 & repmat(u_mean,[1,1,1,size(uf,4),1])>0 , uf>0 & repmat(u_mean,[1,1,1,size(uf,4),1])<0) &
wf>0) = 2;
Qn(or(uf<0 & repmat(u_mean,[1,1,1,size(uf,4),1])>0 , uf>0 & repmat(u_mean,[1,1,1,size(uf,4),1])<0) &
wf<0) = 3;
Qn(or(uf>0 & repmat(u_mean,[1,1,1,size(uf,4),1])>0 , uf<0 & repmat(u_mean,[1,1,1,size(uf,4),1])<0) &
wf<0) = 4;

Qmag = abs(uf.*wf);       % Quadrant magnitude

% Calculate quadrant threshold
Hcoef = 2;
%Analyzed the effect of the Hcoef, seems 2 is a good number.
Hmag = Hcoef * nanmean(nanmean(Qmag,4),1); Hmag(sum(~isnan(Qmag),4)< n_min_ave) = NaN;     % Second
statement included to avoid the calculation of MEAN in data sets with very few data points (which could
bias all subsequent calculations).
%Threshold defined using the mean of Qmag along all cycles for each phase at each location
Hmag = repmat(Hmag,[size(Qmag,1),1,1,size(Qmag,4),1]);      % To keep the same size as Qmag

% Calculate which points are above the threshold
Qsup = zeros(size(Qn));
Qsup = Qmag>Hmag;

% Calculate dominant quadrant for each phase at each location
%There might be several points above the threshold in all 4 different
%quadrants, so we need to decide which quadrant is dominant, if any.
```

```
%Criteria 3: Compare the magnitude of the sum(uf.wf) of all points above the threshold in each quadrant
Qmag_sum = zeros(size(mean(Qmag,4)));
Qmag_sum(:,:,:,1,:) = nansum((Qmag-Hmag)./Hmag.*(Qsup & Qn==1),4);
Qmag_sum(:,:,:,2,:) = nansum((Qmag-Hmag)./Hmag.*(Qsup & Qn==2),4);
Qmag_sum(:,:,:,3,:) = nansum((Qmag-Hmag)./Hmag.*(Qsup & Qn==3),4);
Qmag_sum(:,:,:,4,:) = nansum((Qmag-Hmag)./Hmag.*(Qsup & Qn==4),4);

[~,Qn_dom] = max(Qmag_sum,[],4);
Qn_dom(sum(Qsup,4)==0) = NaN;
```

UNCERTAINTY ANALYSIS

```
% UNCERTAINTY ANALYSIS
% Standard error of X_mean = standard deviation of X / sqrt(N)      % Has units of X
% Percent error = Standard error / X_mean * 100                     % In percentage
% See Benedict and Gould (1996) for reference on how to calculate standard deviation for common
turbulence statistics

% Mean velocity (u_mean)
u_std = nanmean(nanstd(u,0,4),1); u_std(sum(~isnan(u),4)< n_min_ave) = NaN;     % Standard deviation.
Second statement included to avoid the calculation of STD in data sets with very few data points.
u_N = sum(~isnan(u),4);                                                        % Number of valid data
points
u_ste = u_std./sqrt(u_N);                                                      % Standard error
u_ste_pct = u_ste./max(cat(6,abs(u_mean),u_rms),[],6) *100;                    % Percent error, using the
largest of u_mean and u_rms, to be more realistic, but avoiding divisions by zero as well.

% Turbulence Intensities (uf2_mean , vf2_mean , wf2_mean)
uf2 = uf.*uf;
vf2 = vf.*vf;
wf2 = wf.*wf;

uf2_std = nanmean(nanstd(uf2,0,4),1); uf2_std(sum(~isnan(uf2),4)< n_min_ave) = NaN;     % Standard
deviation. Second statement included to avoid the calculation of STD in data sets with very few data
points.
vf2_std = nanmean(nanstd(vf2,0,4),1); vf2_std(sum(~isnan(vf2),4)< n_min_ave) = NaN;     % Standard
deviation. Second statement included to avoid the calculation of STD in data sets with very few data
points.
wf2_std = nanmean(nanstd(wf2,0,4),1); wf2_std(sum(~isnan(wf2),4)< n_min_ave) = NaN;     % Standard
deviation. Second statement included to avoid the calculation of STD in data sets with very few data
points.

uf2_N = sum(~isnan(uf2),4);                                                      % Number of valid data
points
vf2_N = sum(~isnan(vf2),4);                                                      % Number of valid data
points
wf2_N = sum(~isnan(wf2),4);                                                      % Number of valid data
points

uf2_ste = uf2_std./sqrt(uf2_N);                                                    % Standard error
vf2_ste = vf2_std./sqrt(vf2_N);                                                    % Standard error
wf2_ste = wf2_std./sqrt(wf2_N);                                                    % Standard error

uf2_ste_pct = uf2_ste./uf2_mean *100;                                          % Percent error
```

```
vf2_ste_pct = vf2_ste./vf2_mean *100;                                          % Percent error
wf2_ste_pct = wf2_ste./wf2_mean *100;                                          % Percent error

uf2_ste_pct(uf2_ste_pct>100) = 100;                                              % To avoid spikes that
could distort the scale
vf2_ste_pct(vf2_ste_pct>100) = 100;                                              % To avoid spikes that
could distort the scale
wf2_ste_pct(wf2_ste_pct>100) = 100;                                              % To avoid spikes that
could distort the scale

% RMS Velocities (u_rms , v_rms , w_rms)
% From statistics, percent error of Urms = 1/2 of percent error of uf2_mean
u_rms_ste_pct = uf2_ste_pct/2;                                                 % Percent error
v_rms_ste_pct = vf2_ste_pct/2;                                                 % Percent error
w_rms_ste_pct = wf2_ste_pct/2;                                                 % Percent error

% Reynolds Stresses (uf_wf_mean)
uf_wf = uf.*wf;
uf_wf_std = nanmean(nanstd(uf_wf,0,4),1); uf_wf_std(sum(~isnan(uf_wf),4)< n_min_ave) = NaN;     %
Standard deviation. Second statement included to avoid the calculation of STD in data sets with very few
data points.
uf_wf_N = sum(~isnan(uf_wf),4);                                                % Number of valid data
points
uf_wf_ste = uf_wf_std./sqrt(uf_wf_N);                                          % Standard error
uf_wf_ste_pct = uf_wf_ste./(u_rms.*w_rms) *100;                                % Percent error, using Urms
and Wrms since uf_wf_mean has value close to zero, and can cause division by very small numbers, giving
unrealistic results

% Skewness of velocity fluctuations (uf3_mean , vf3_mean , wf3_mean)
uf3 = uf.*uf.*uf;
vf3 = vf.*vf.*vf;
wf3 = wf.*wf.*wf;

uf3_std = nanmean(nanstd(uf3,0,4),1); uf3_std(sum(~isnan(uf3),4)< n_min_skew) = NaN;     % Standard
deviation. Second statement included to avoid the calculation of STD in data sets with very few data
points.
vf3_std = nanmean(nanstd(vf3,0,4),1); vf3_std(sum(~isnan(vf3),4)< n_min_skew) = NaN;     % Standard
deviation. Second statement included to avoid the calculation of STD in data sets with very few data
points.
wf3_std = nanmean(nanstd(wf3,0,4),1); wf3_std(sum(~isnan(wf3),4)< n_min_skew) = NaN;     % Standard
deviation. Second statement included to avoid the calculation of STD in data sets with very few data
points.

uf3_N = sum(~isnan(uf3),4);                                                      % Number of valid data
points
vf3_N = sum(~isnan(vf3),4);                                                      % Number of valid data
points
wf3_N = sum(~isnan(wf3),4);                                                      % Number of valid data
points

uf3_ste = uf3_std./sqrt(uf3_N);                                                    % Standard error
vf3_ste = vf3_std./sqrt(vf3_N);                                                    % Standard error
wf3_ste = wf3_std./sqrt(wf3_N);                                                    % Standard error

uf3_ste_pct = uf3_ste./(u_rms.^3) *100;                                        % Percent error
vf3_ste_pct = vf3_ste./(v_rms.^3) *100;                                        % Percent error
```

```
wf3_ste_pct = wf3_ste./(w_rms.^3) *100;                                          % Percent error

uf3_ste_pct(uf3_ste_pct>100) = 100;                                               % To avoid spikes that
could distort the scale
vf3_ste_pct(vf3_ste_pct>100) = 100;                                               % To avoid spikes that
could distort the scale
wf3_ste_pct(wf3_ste_pct>100) = 100;                                               % To avoid spikes that
could distort the scale

% Turbulent kinetic energy (tke)
% We use the property of the sum of errors, such that: standard error of (X+Y) = Standard error of X +
Standard error of Y
tke_ste = 1/2 * nansum(cat(6,uf2_ste,vf2_ste,wf2_ste),6);                              % Standard error
tke_ste_pct = tke_ste./tke *100;                                                % Percent error

% Save all variables in MatLab format
file_save = 'LDV_analysis-all.mat';
save([dir_save '\' file_save]);
```

Published with MATLAB® 7.14

APPENDIX C

EXPERIMENTAL DATA

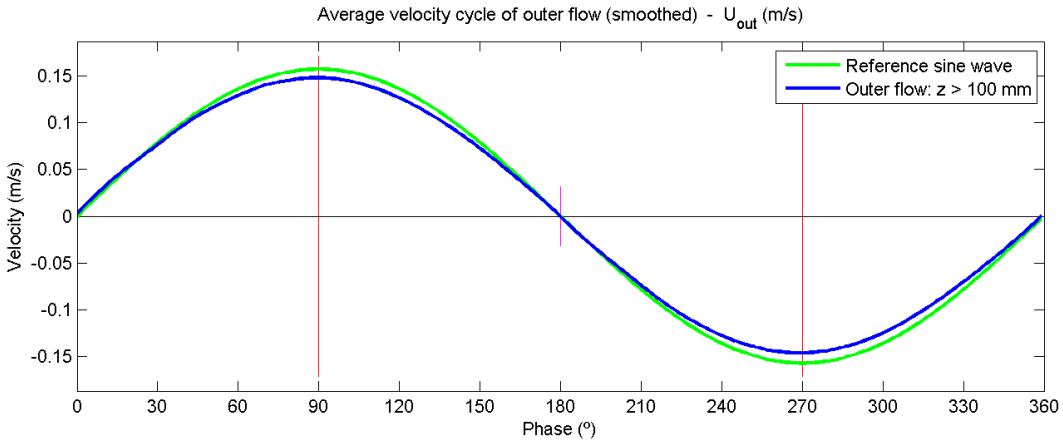
C.1 Experiment no. 1

C.1.1 Main parameters

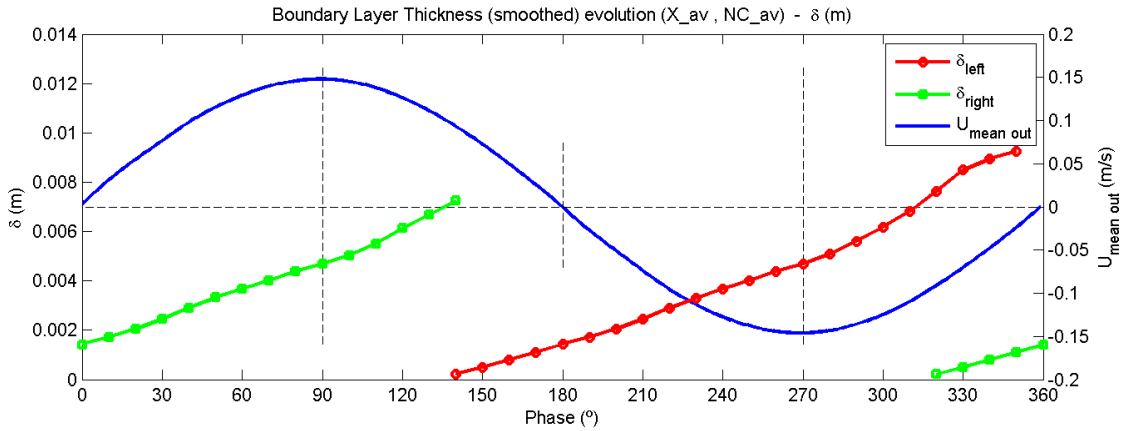
Exp no.	*Temp* (ºC)	ρ (kg/m^3)	ν (m^2/s)	*T* (s)	*2a* (m)	U_{out_max} (m/s)	Re_w (-)	N_{max} (cycles)
1	18.0	998.62	1.07E-06	10	0.468	0.147	3.2E+04	130

Exp no.	U_{*max} (m/s)	τ_{b_max} (N/m^2)	$\Delta\phi$ (deg)	f_{w_ref} (-)	Z_{v_min} (mm)	δ_{v_90} (mm)	δ_{90} (mm)	δ_{top} (mm)
1	0.011	0.11	39.5	0.0102	0.102	3.672	4.7	9

C.1.2 Outer flow velocity

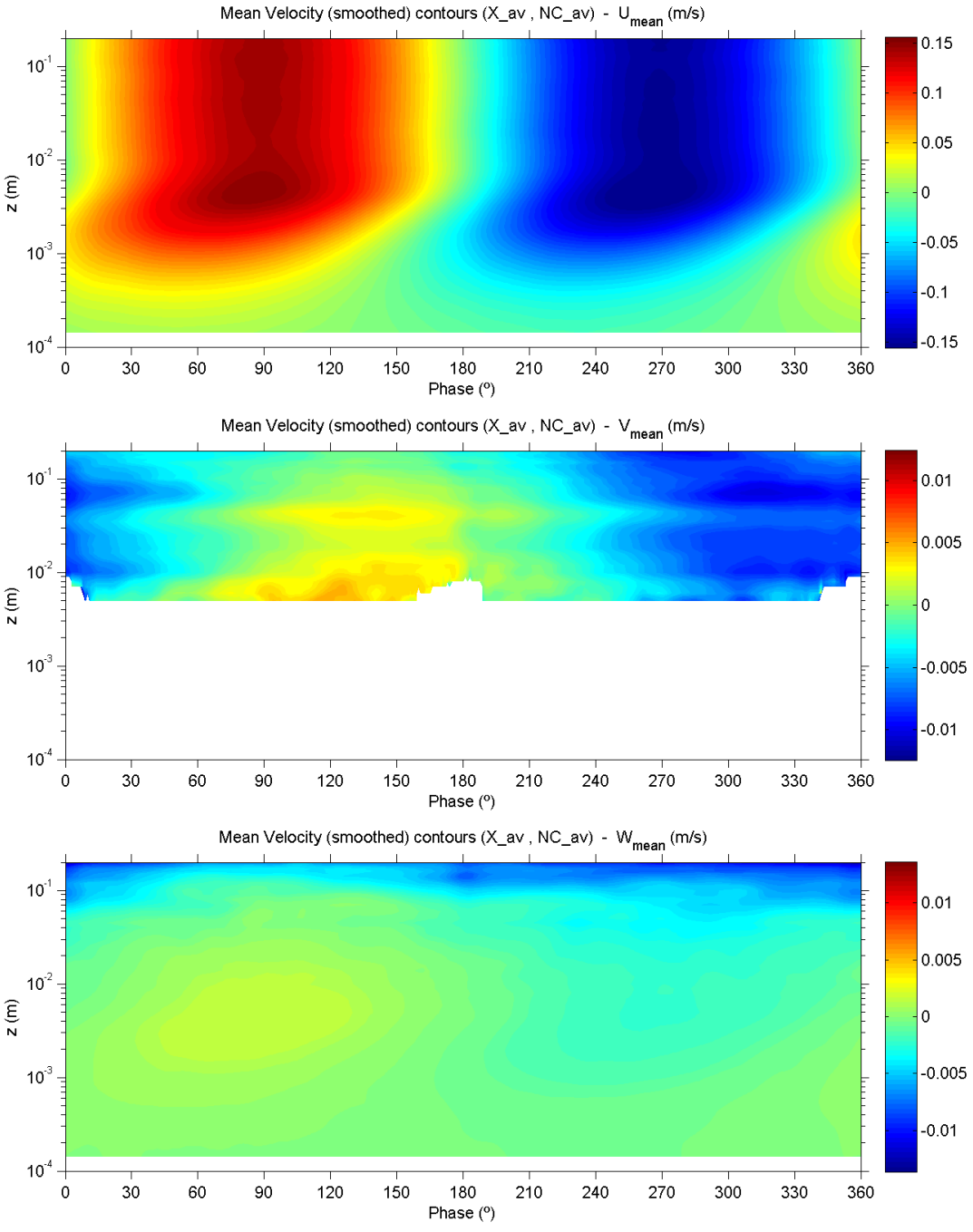


C.1.3 Boundary layer thickness

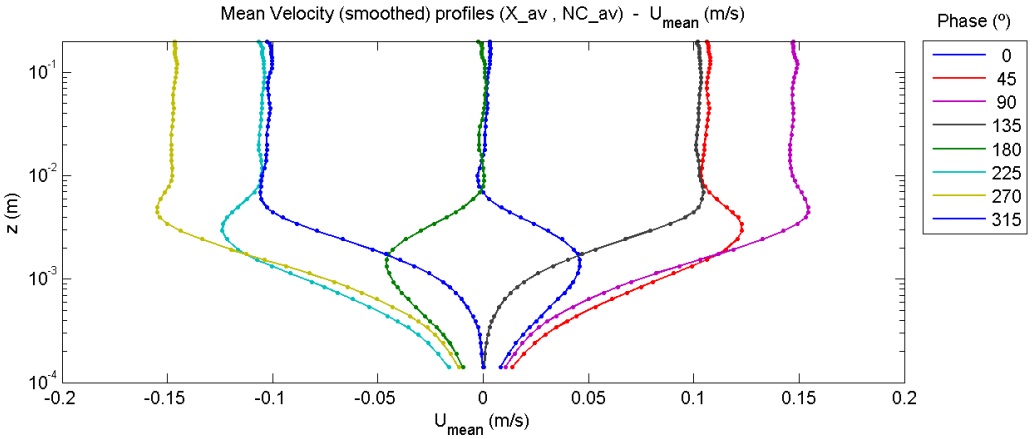


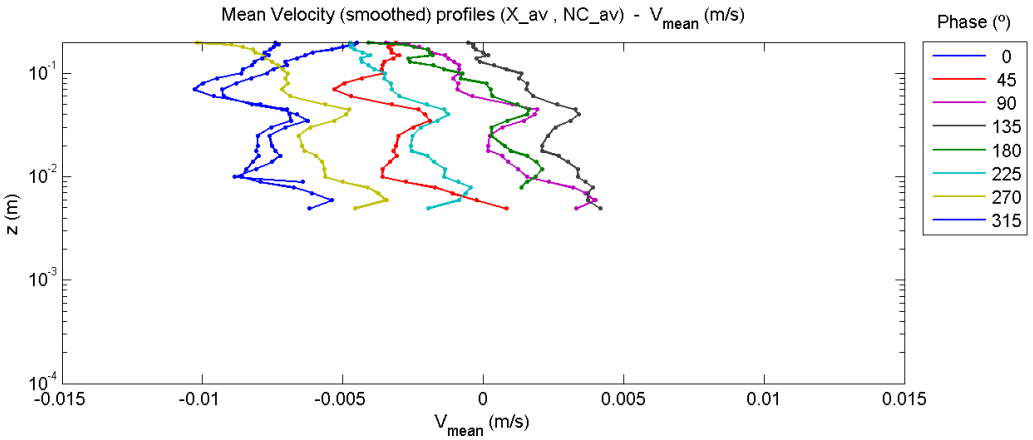
C.1.4 Mean velocities

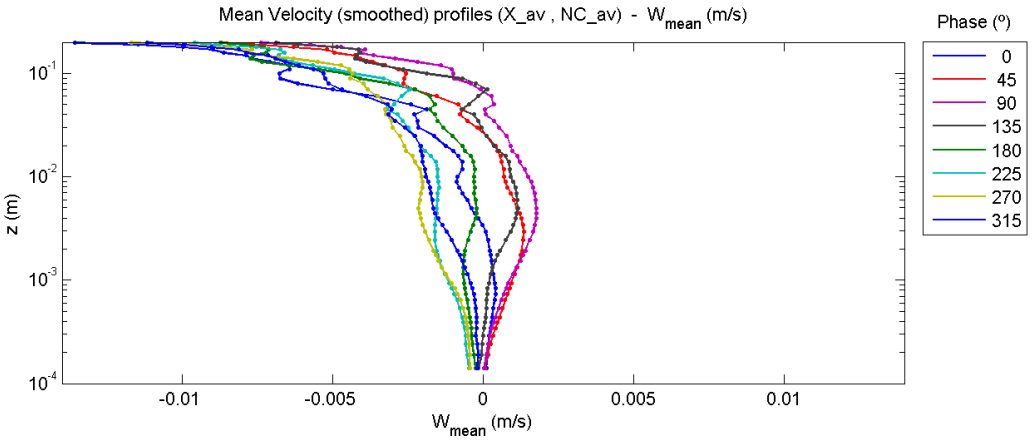
C.1.4.1 Contour plots



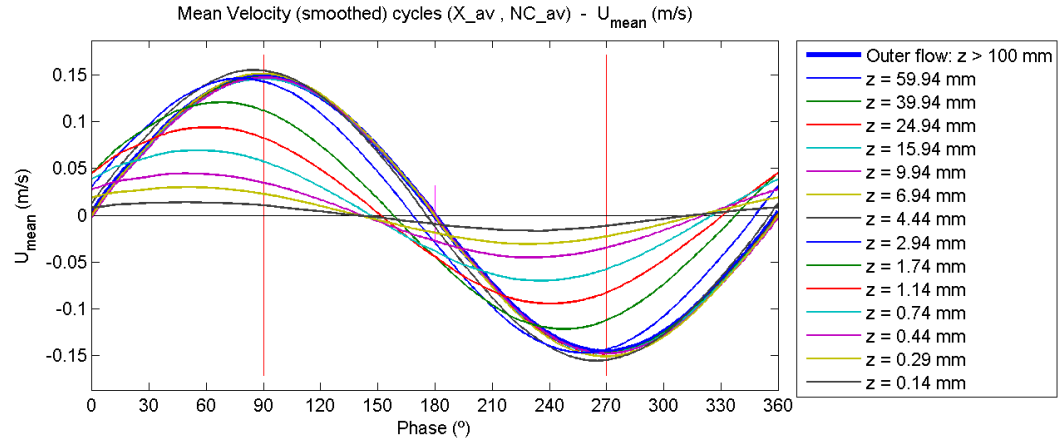
C.1.4.2 Profile plots

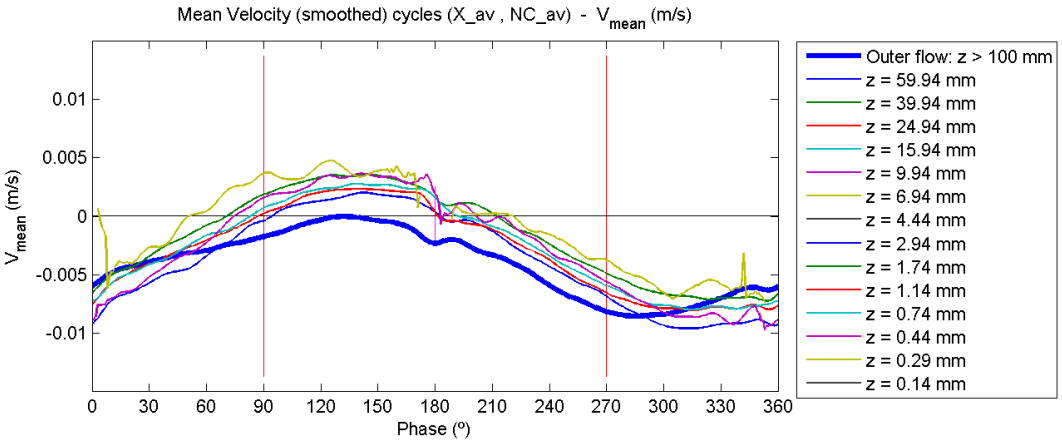


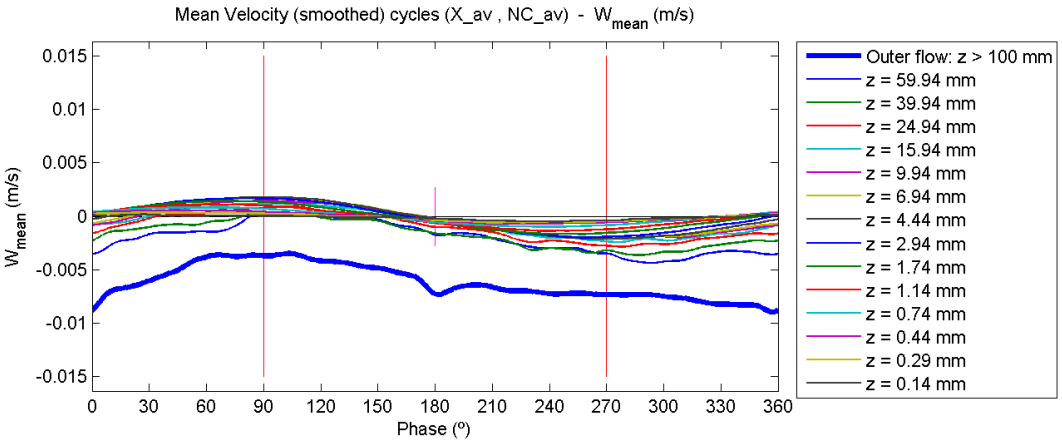




C.1.4.3 Evolution plots

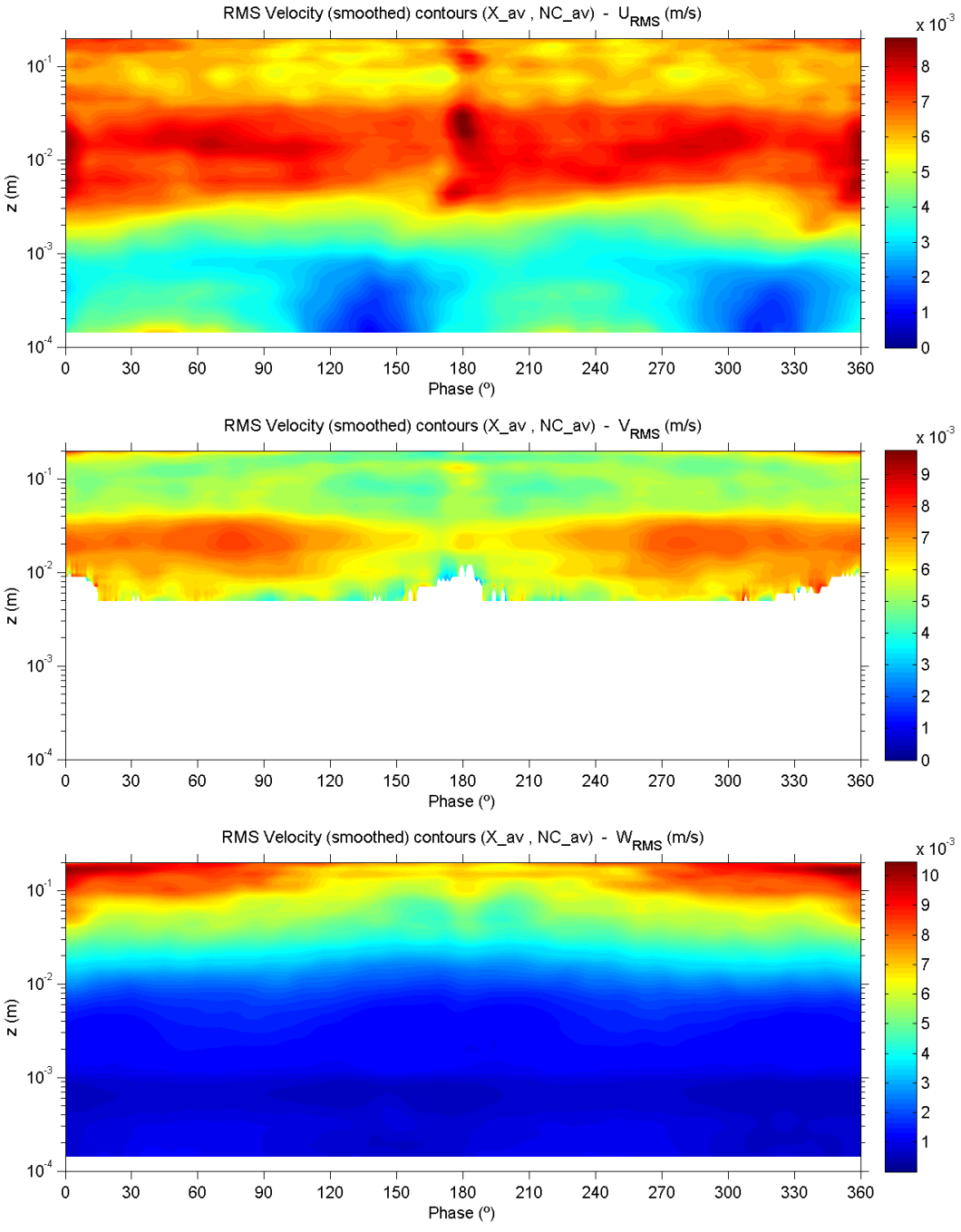




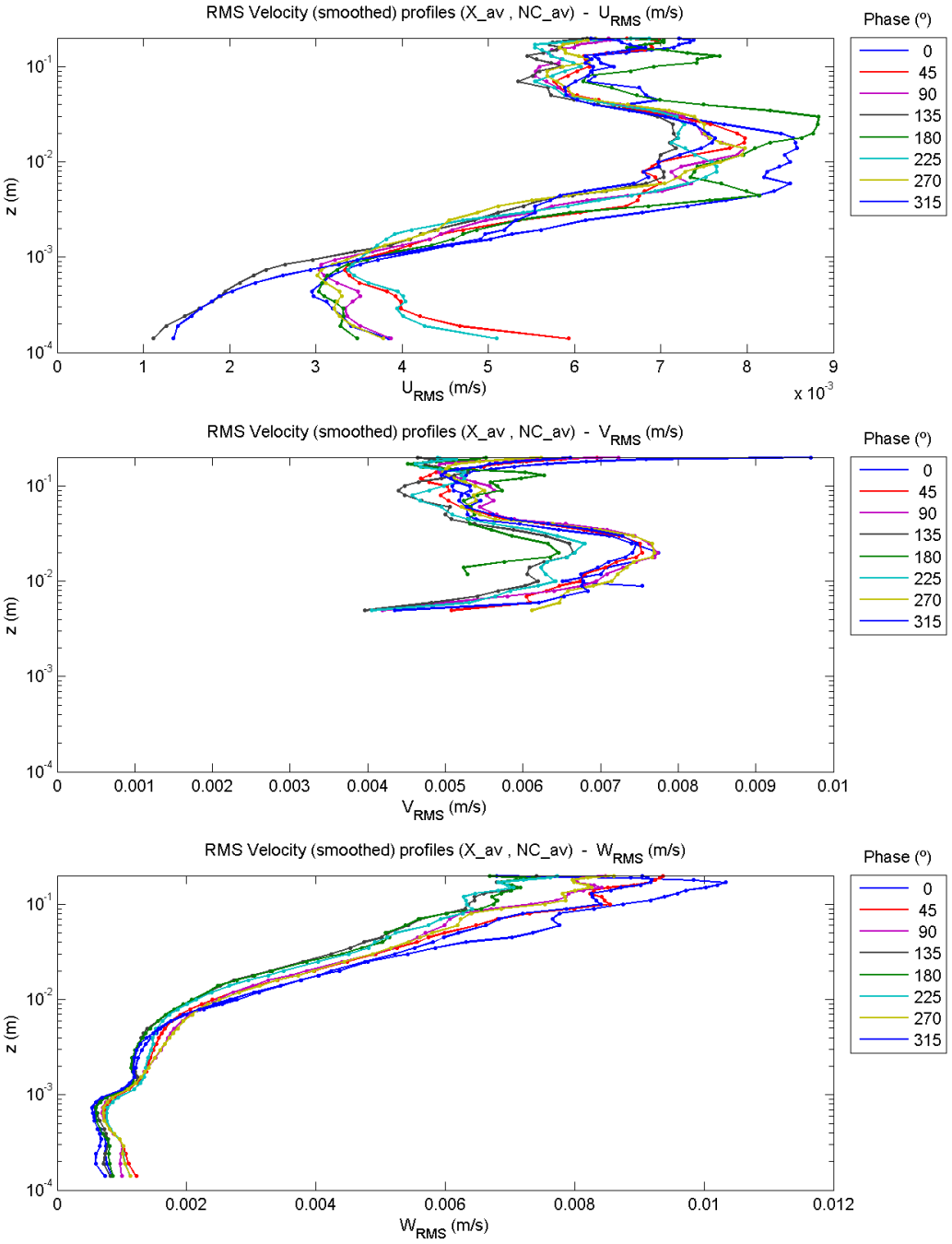


C.1.5 RMS velocity fluctuations

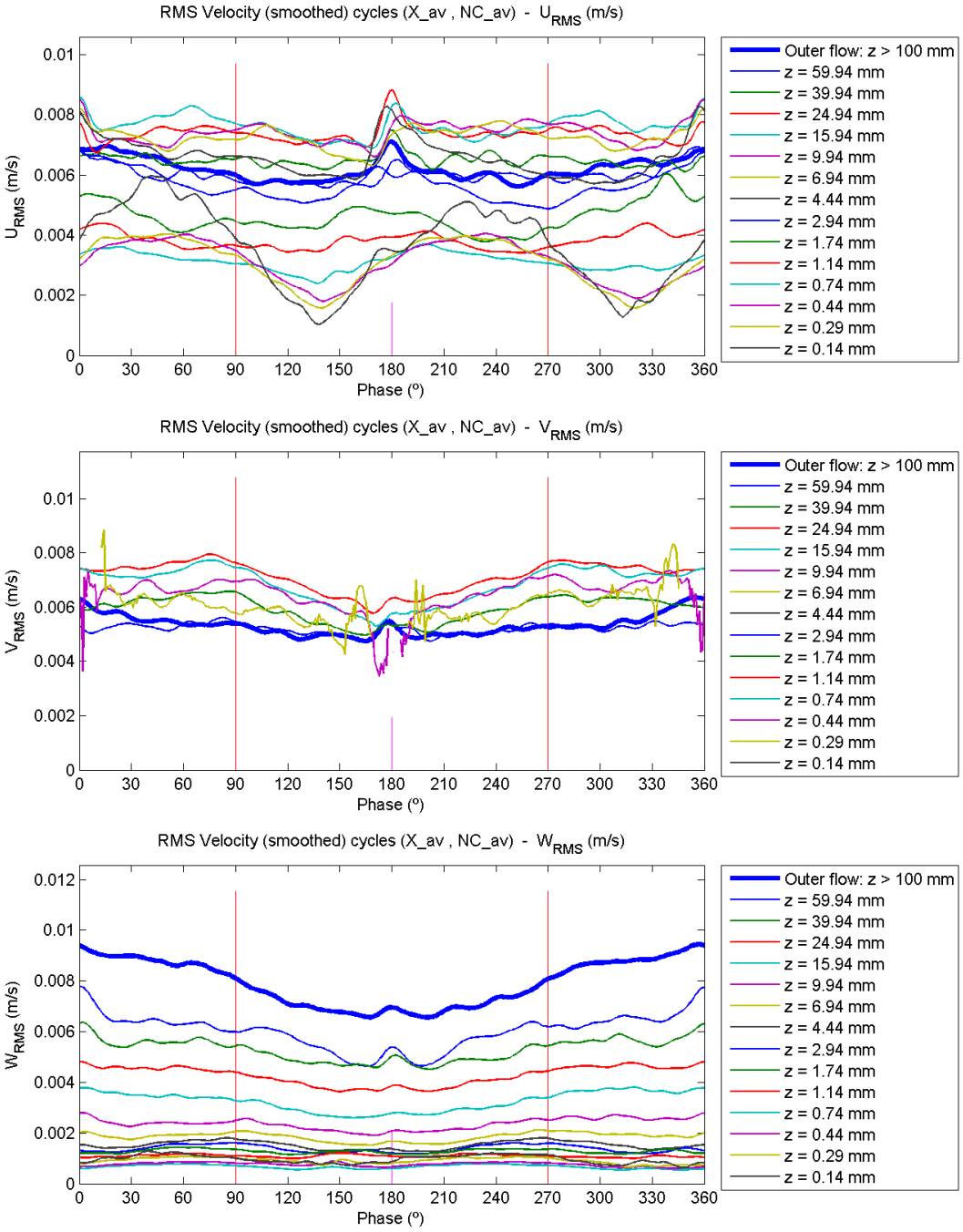
C.1.5.1 Contour plots



C.1.5.2 Profile plots

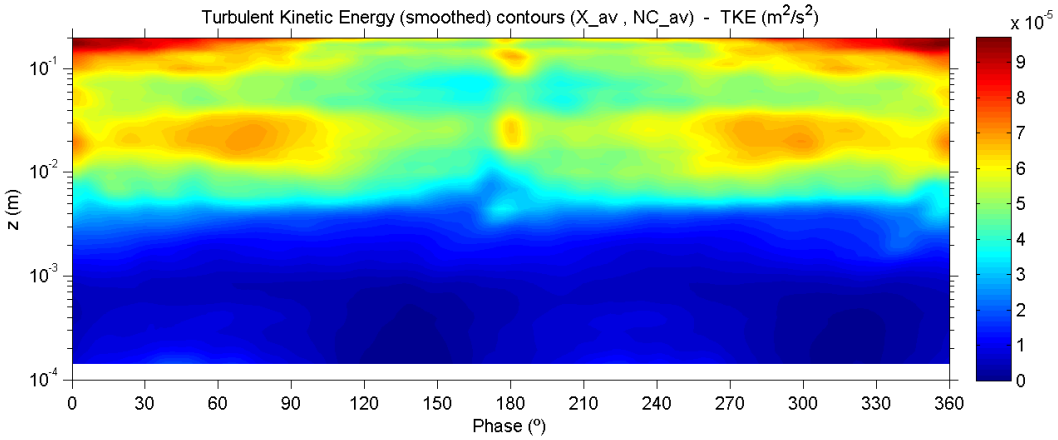


C.1.5.3 Evolution plots

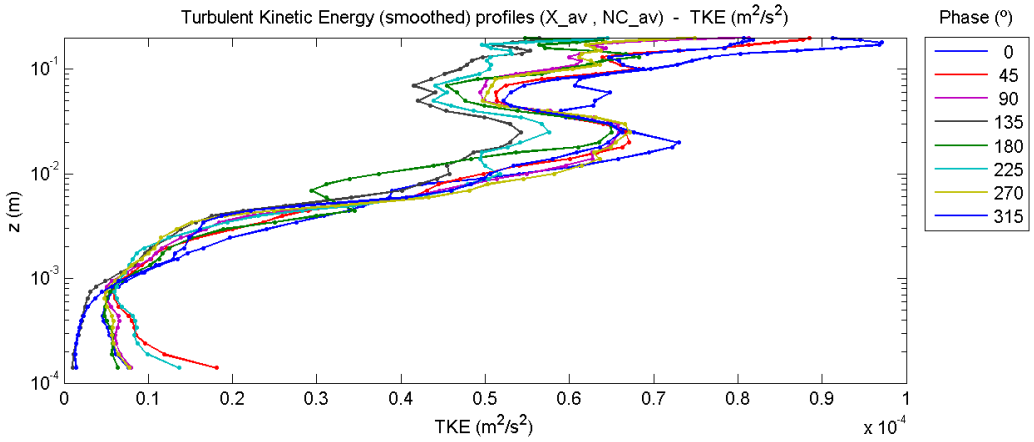


C.1.6 Turbulent kinetic energy

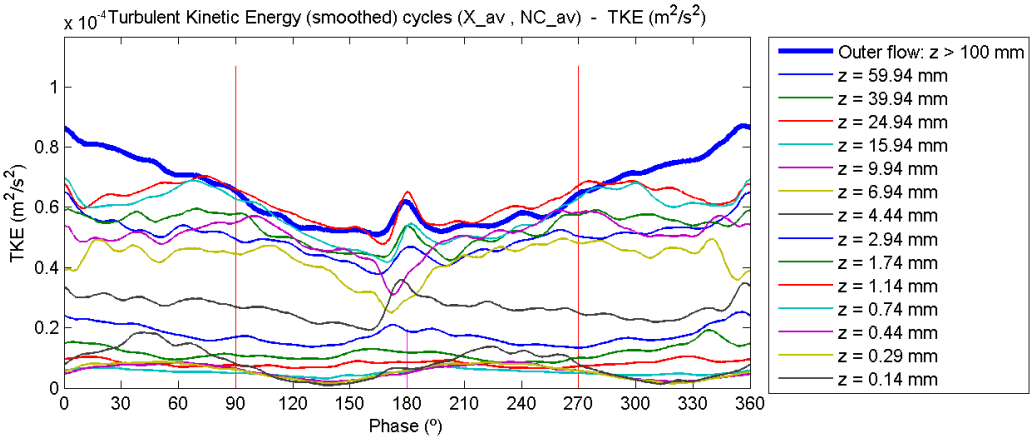
C.1.6.1 Contour plot



C.1.6.2 Profile plot

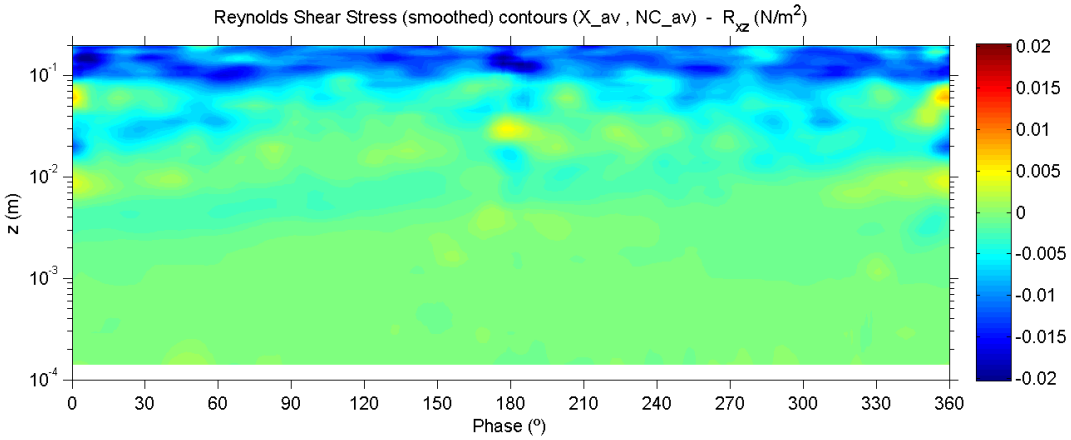


C.1.6.3 Evolution plot

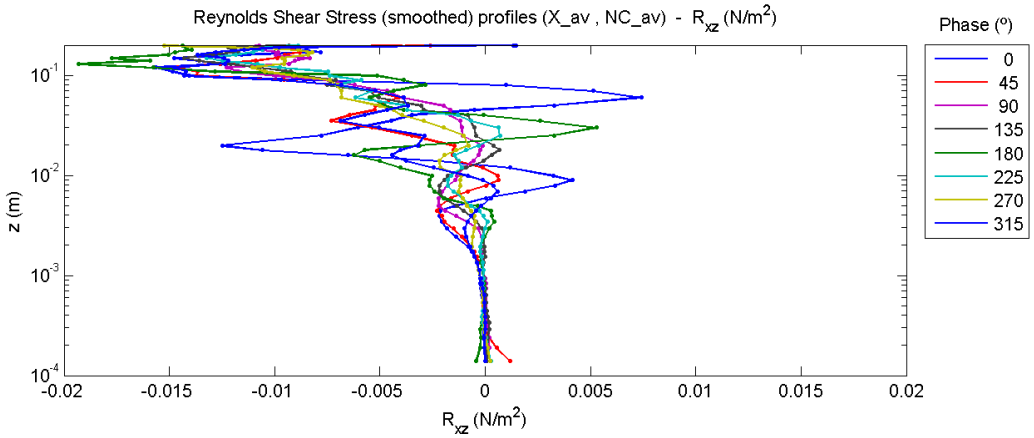


C.1.7 Turbulent shear stress (Reynolds shear stress)

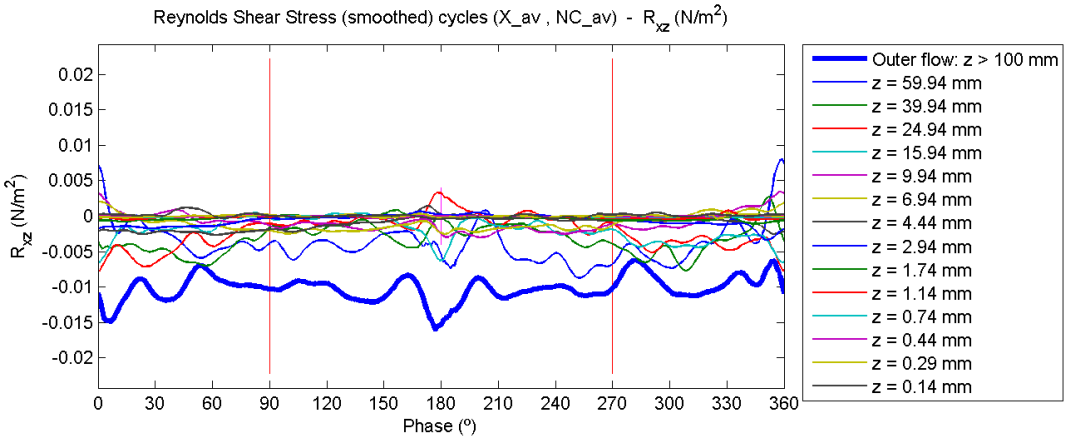
C.1.7.1 Contour plot



C.1.7.2 Profile plot

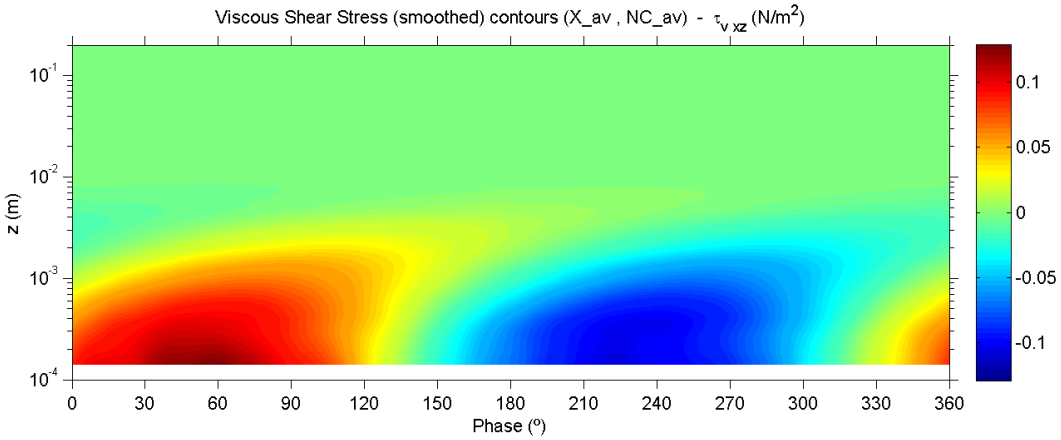


C.1.7.3 Evolution plot

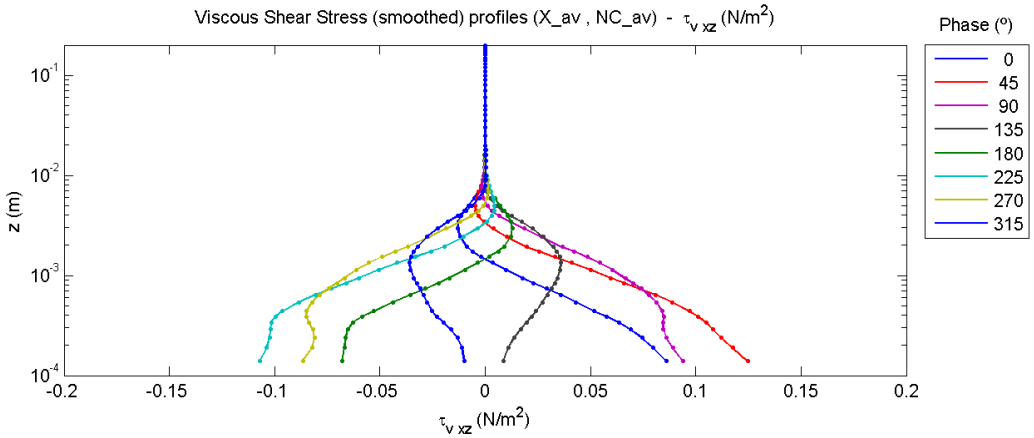


C.1.8 Viscous shear stress

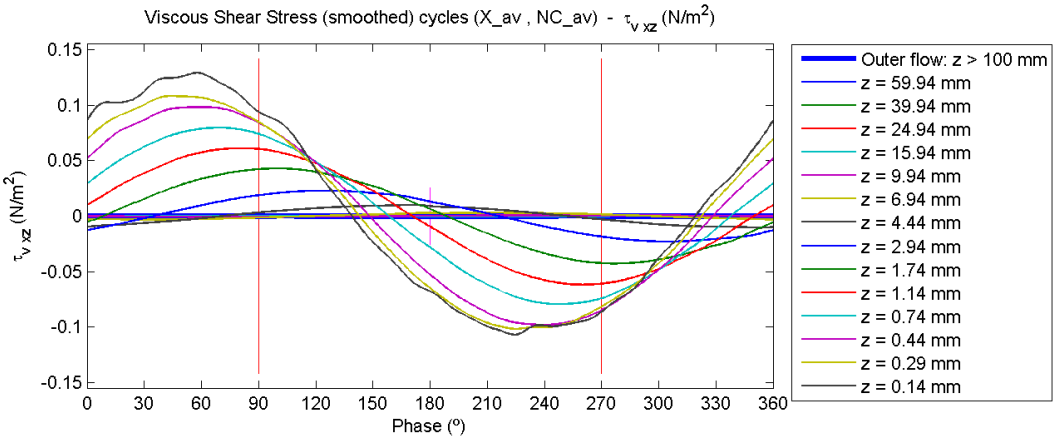
C.1.8.1 Contour plot



C.1.8.2 Profile plot

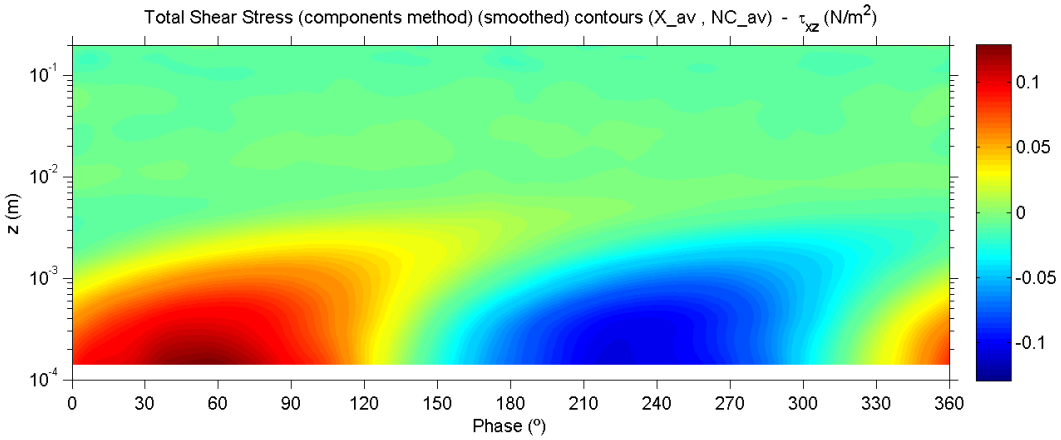


C.1.8.3 Evolution plot

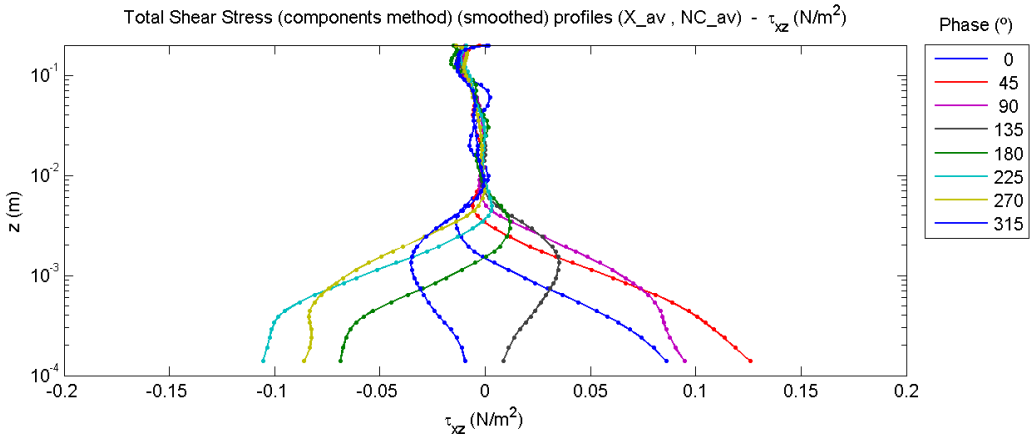


C.1.9 Total shear stress

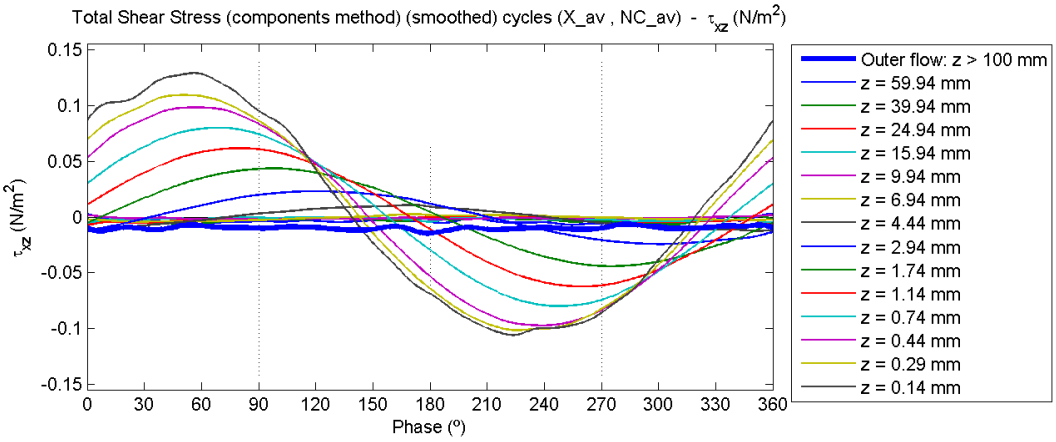
C.1.9.1 Contour plot



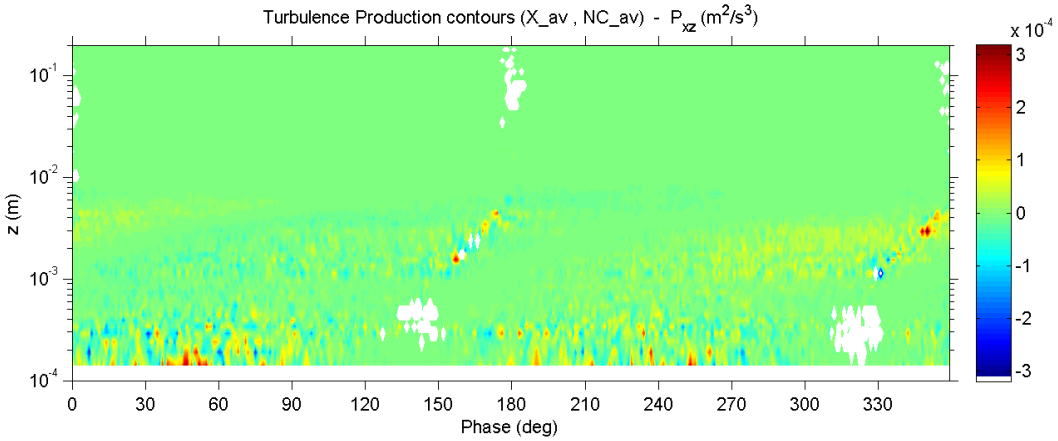
C.1.9.2 Profile plot



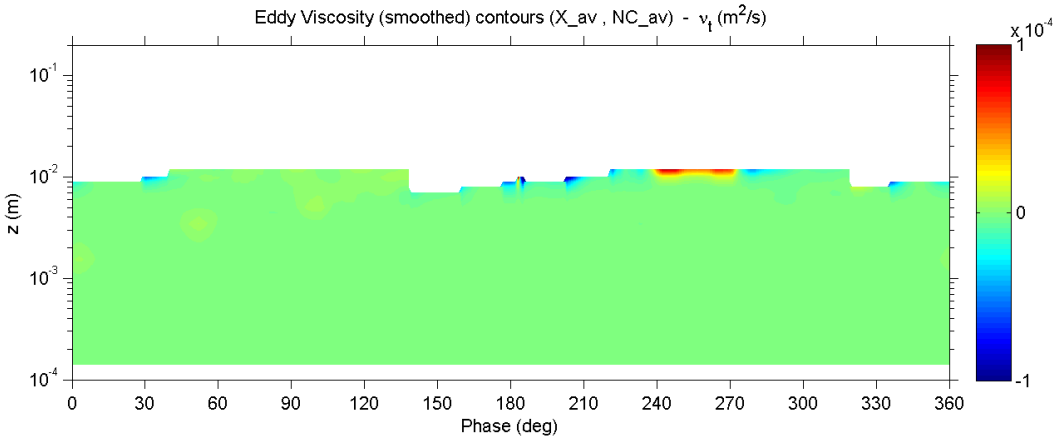
C.1.9.3 Evolution plot



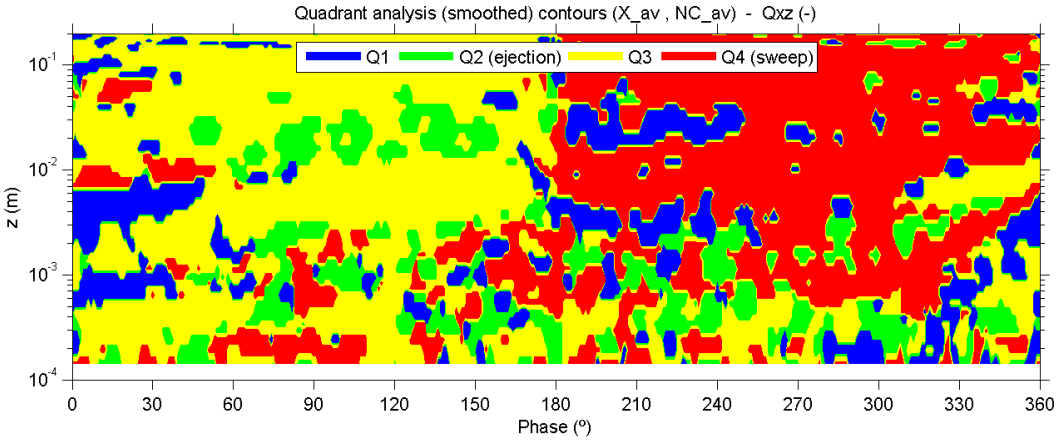
C.1.10 Turbulence production



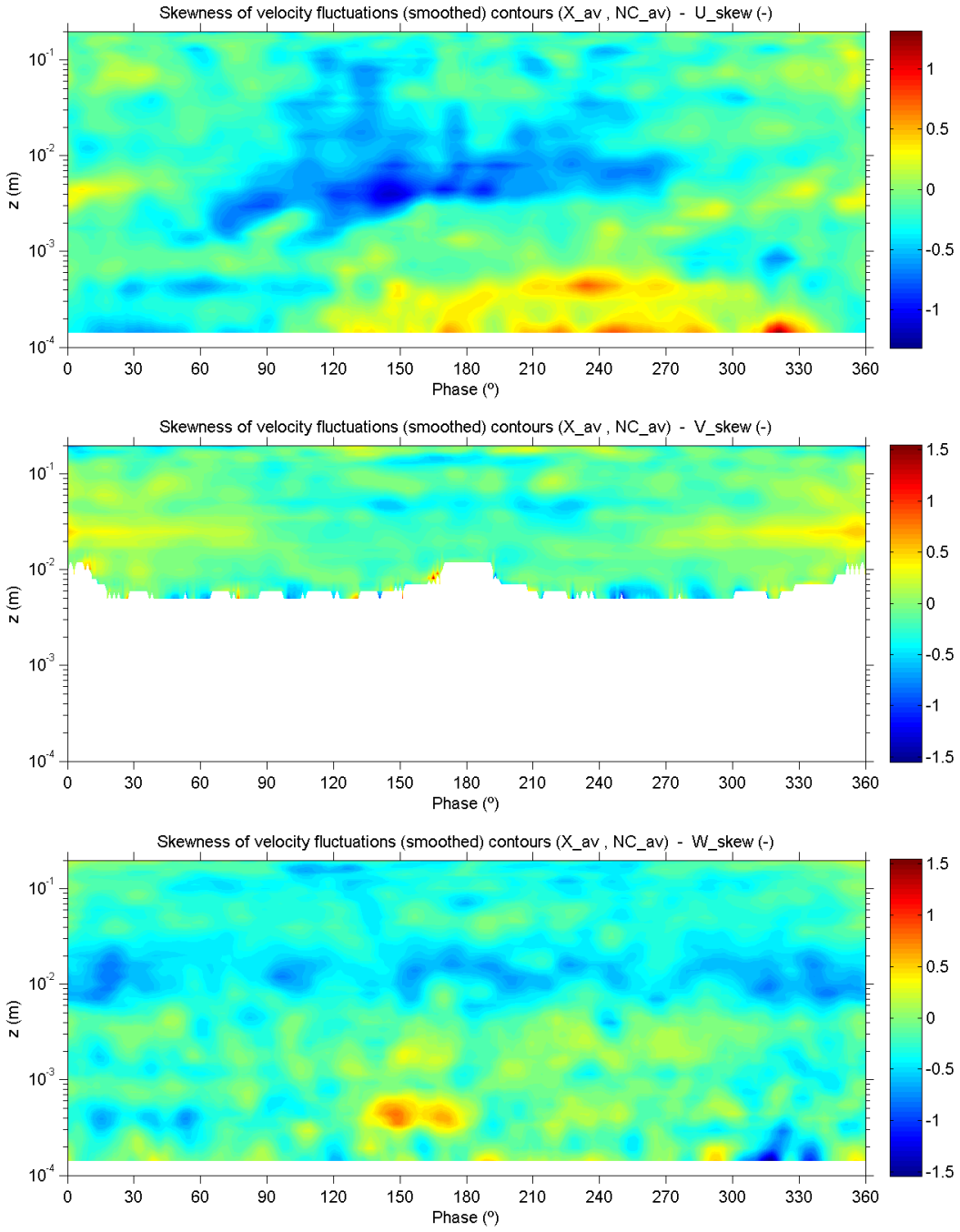
C.1.11 Turbulent viscosity (Eddy viscosity)



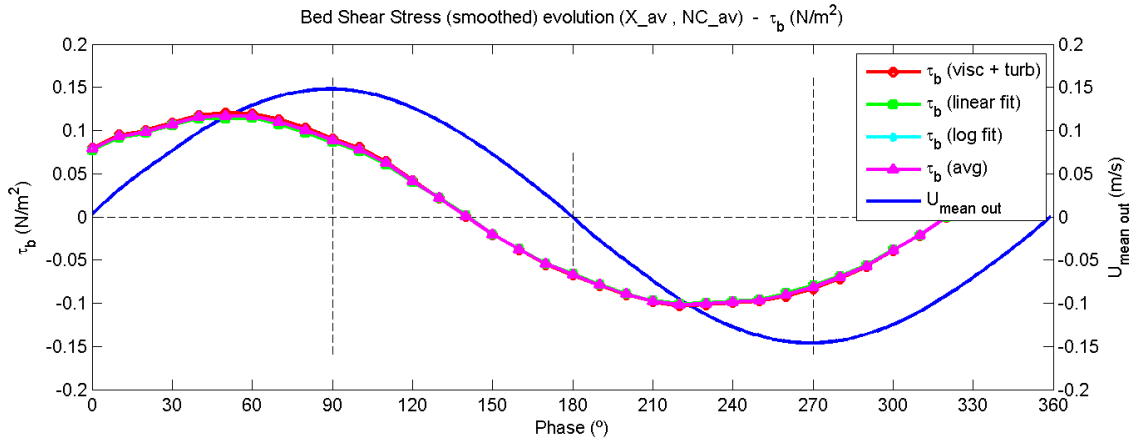
C.1.12 Quadrant analysis



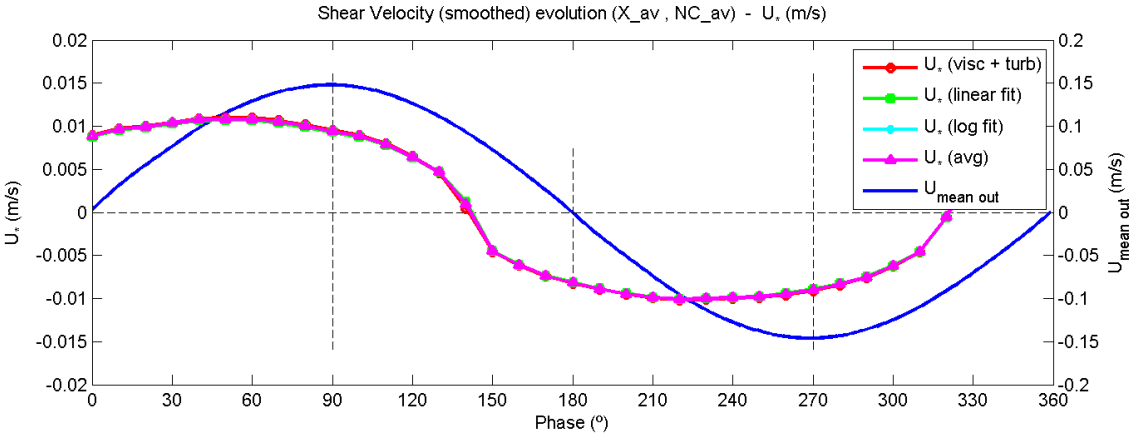
C.1.13 Skewness of velocity fluctuations



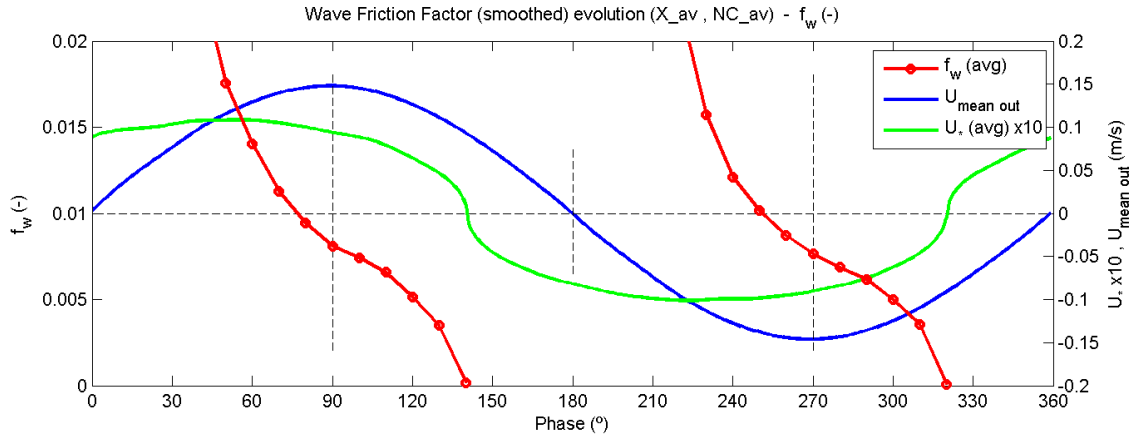
C.1.14 Bed shear stress



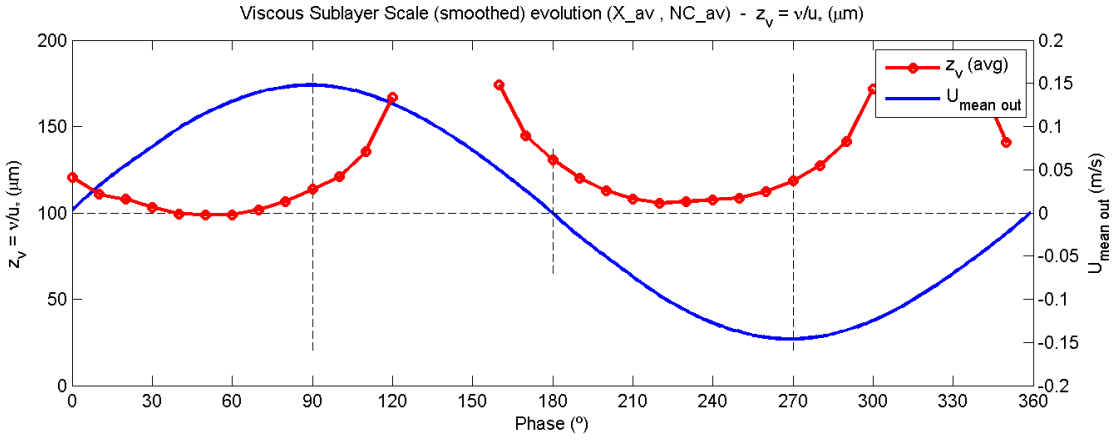
C.1.15 Shear velocity



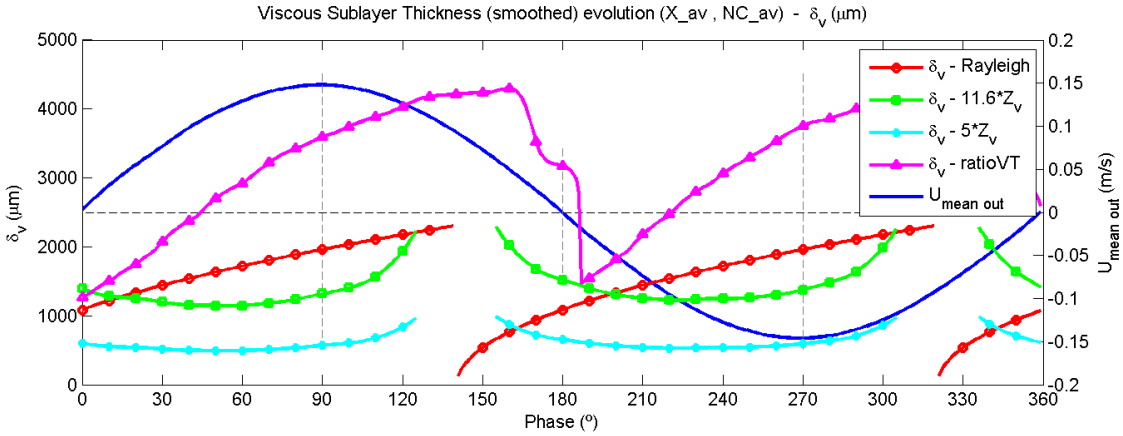
C.1.16 Wave friction factor



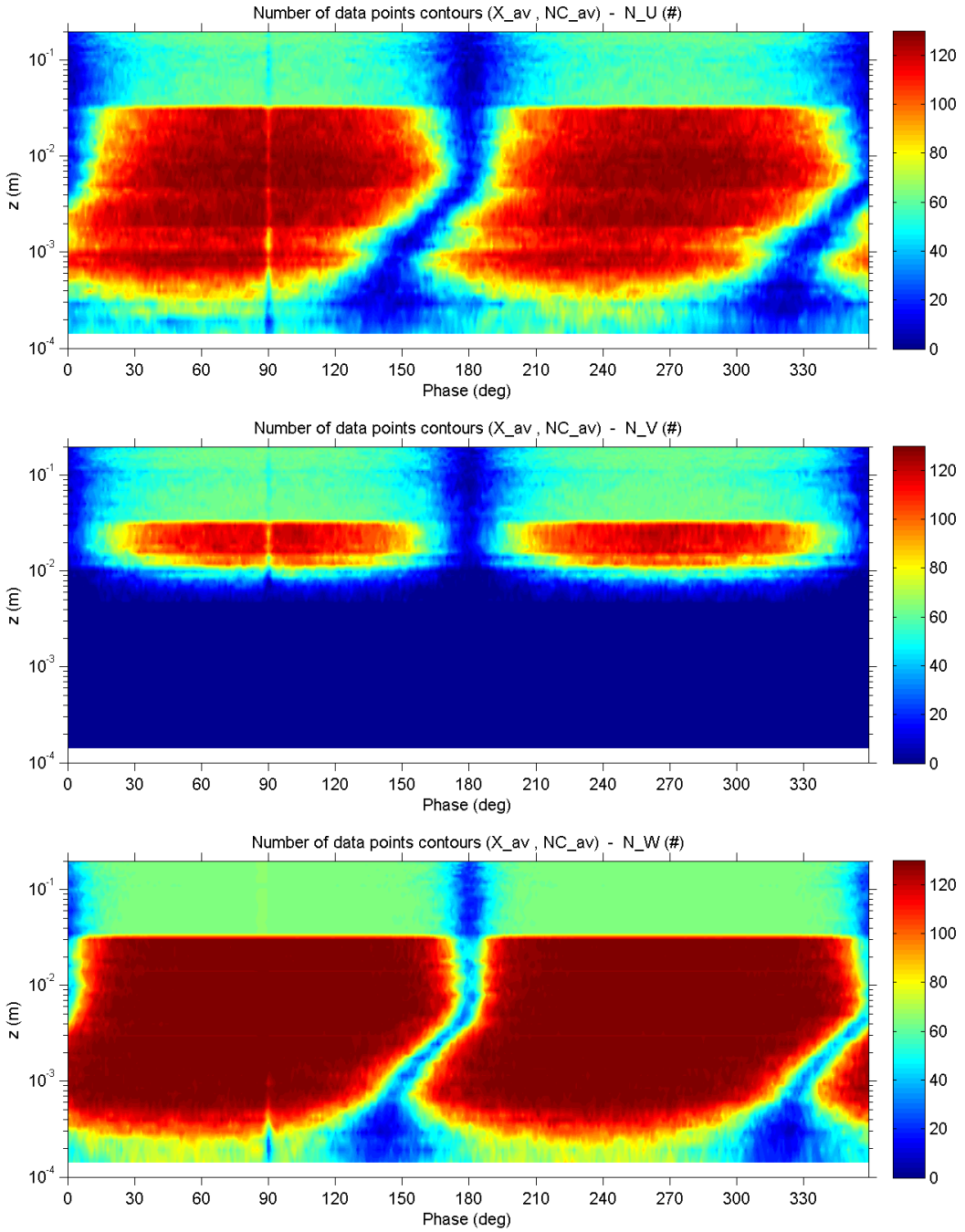
C.1.17 Viscous length scale



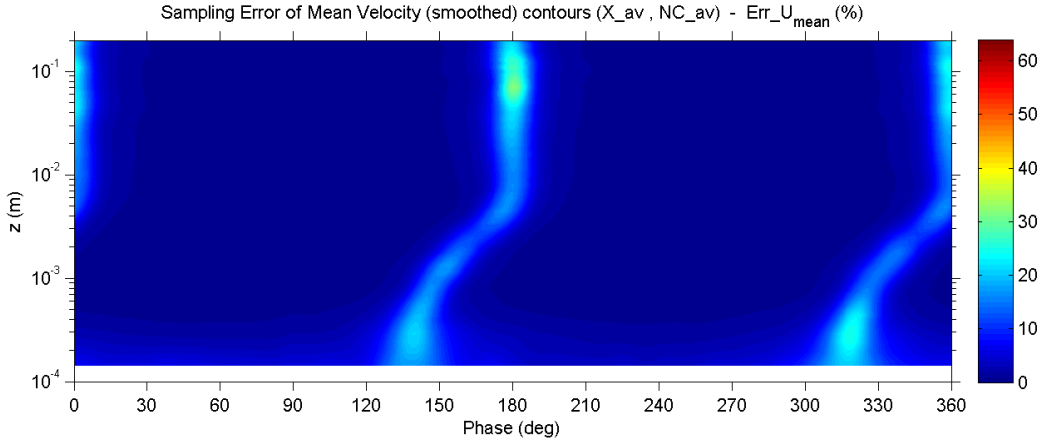
C.1.18 Viscous sublayer thickness



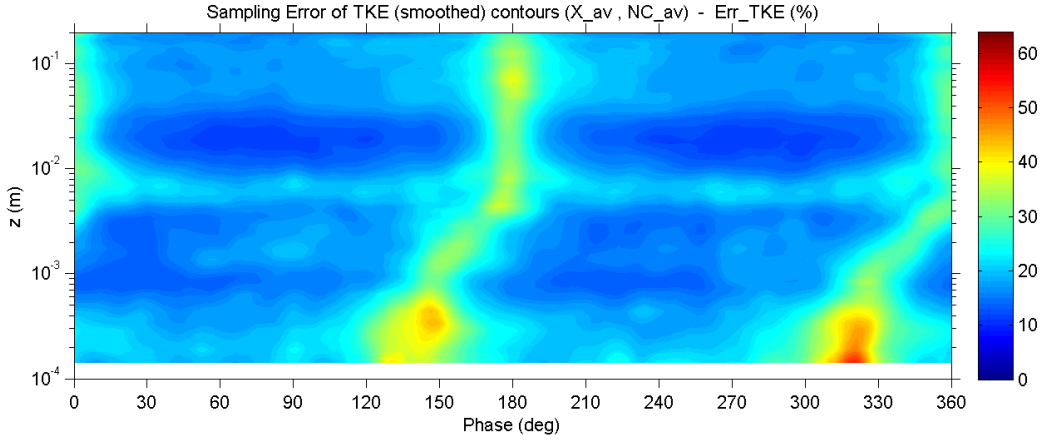
C.1.19 Number of valid data points



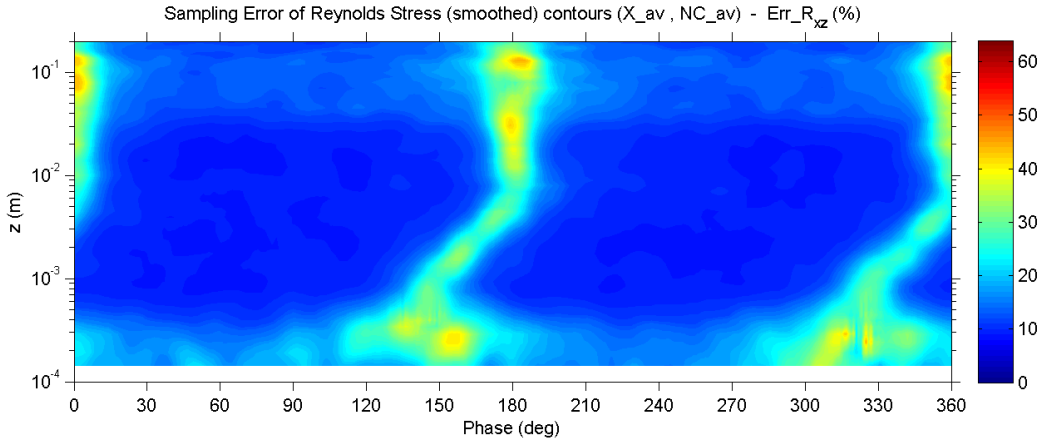
C.1.20 Percent error of mean velocity



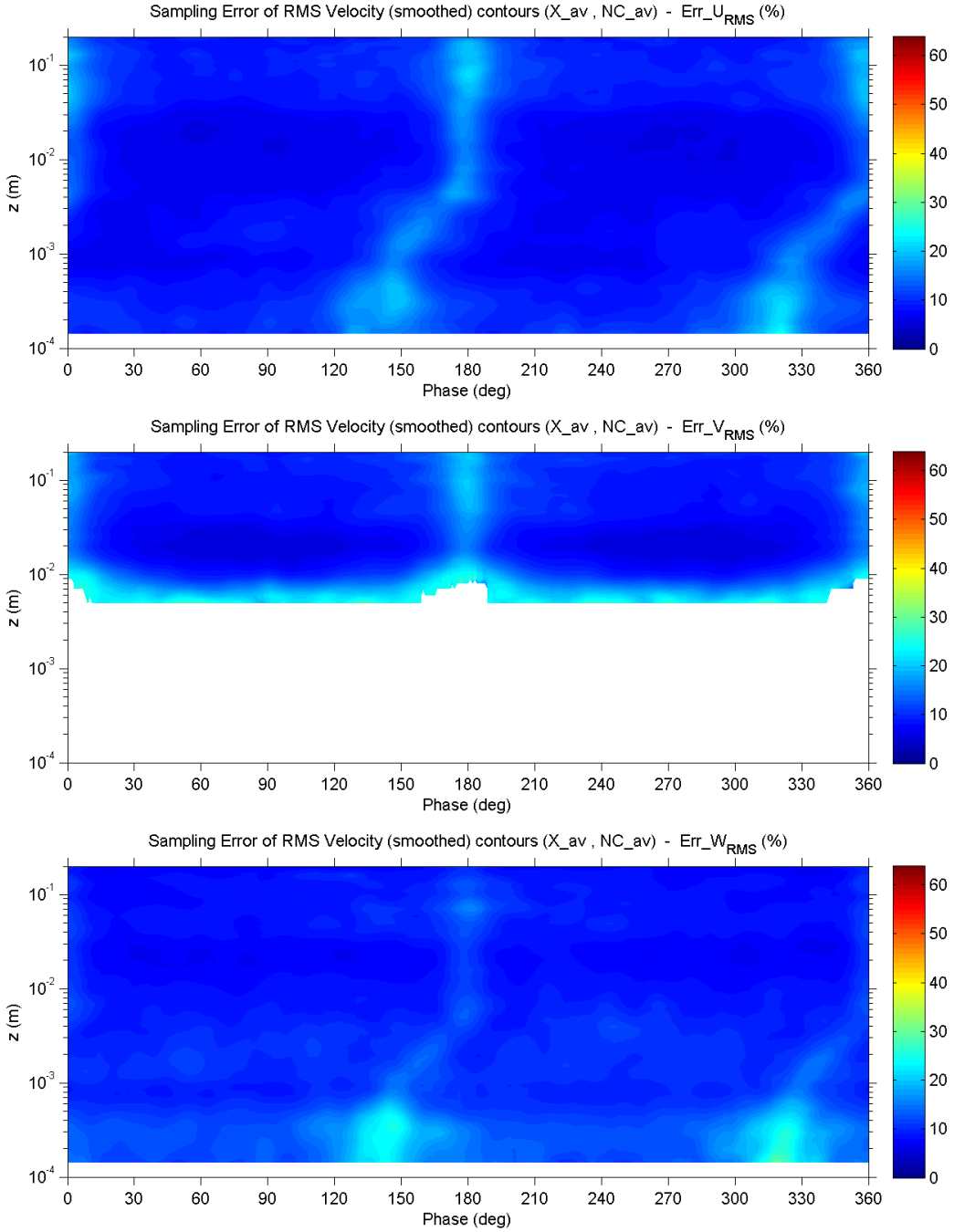
C.1.21 Percent error of turbulent kinetic energy



C.1.22 Percent error of Reynolds shear stress



C.1.23 Percent error of RMS velocity fluctuations



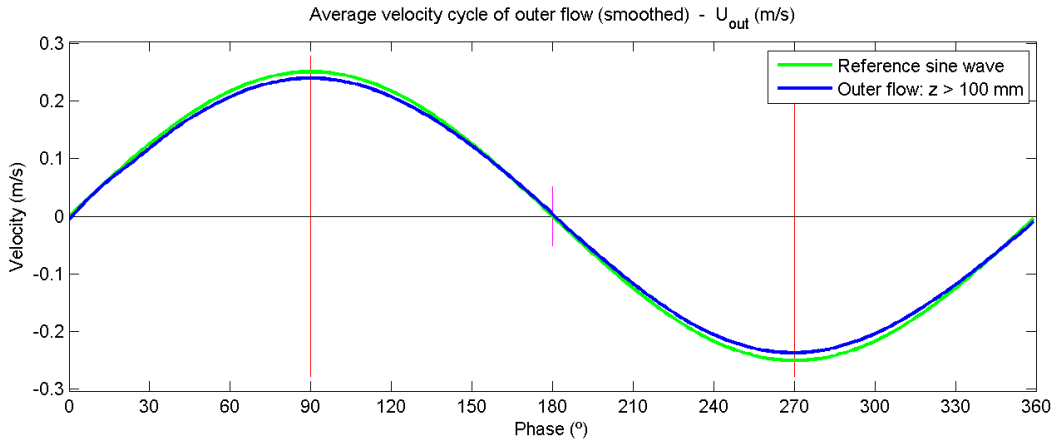
C.2 Experiment no. 2

C.2.1 Main parameters

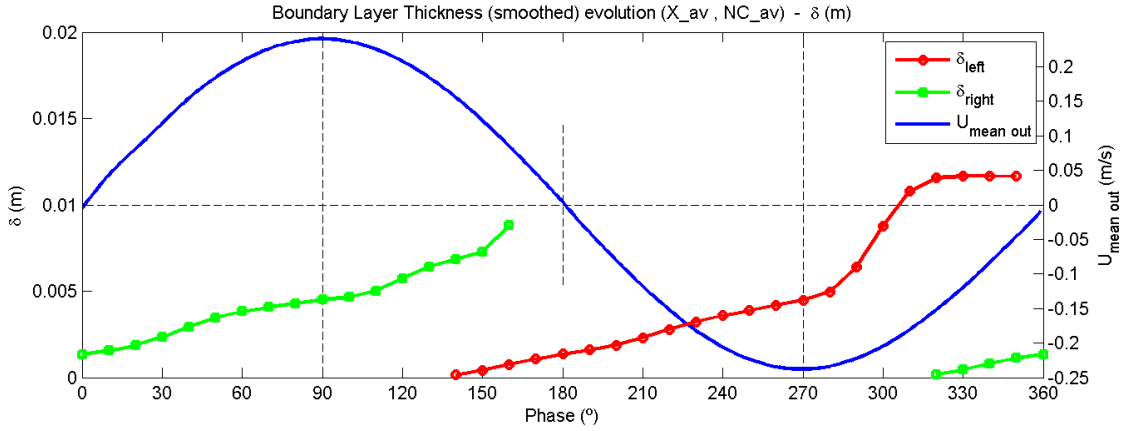
Exp no.	Temp (ºC)	ρ (kg/m^3)	ν (m^2/s)	T (s)	$2a$ (m)	U_{out_max} (m/s)	Re_w (-)	N_{max} (cycles)
2	16.6	998.87	1.11E-06	10	0.761	0.239	8.2E+04	130

Exp no.	U_{*max} (m/s)	τ_{b_max} (N/m^2)	$\Delta\phi$ (deg)	f_{w_ref} (-)	Z_{v_min} (mm)	δ_{v_90} (mm)	δ_{90} (mm)	δ_{top} (mm)
2	0.014	0.21	39.0	0.0073	0.077	2.655	4.5	12

C.2.2 Outer flow velocity

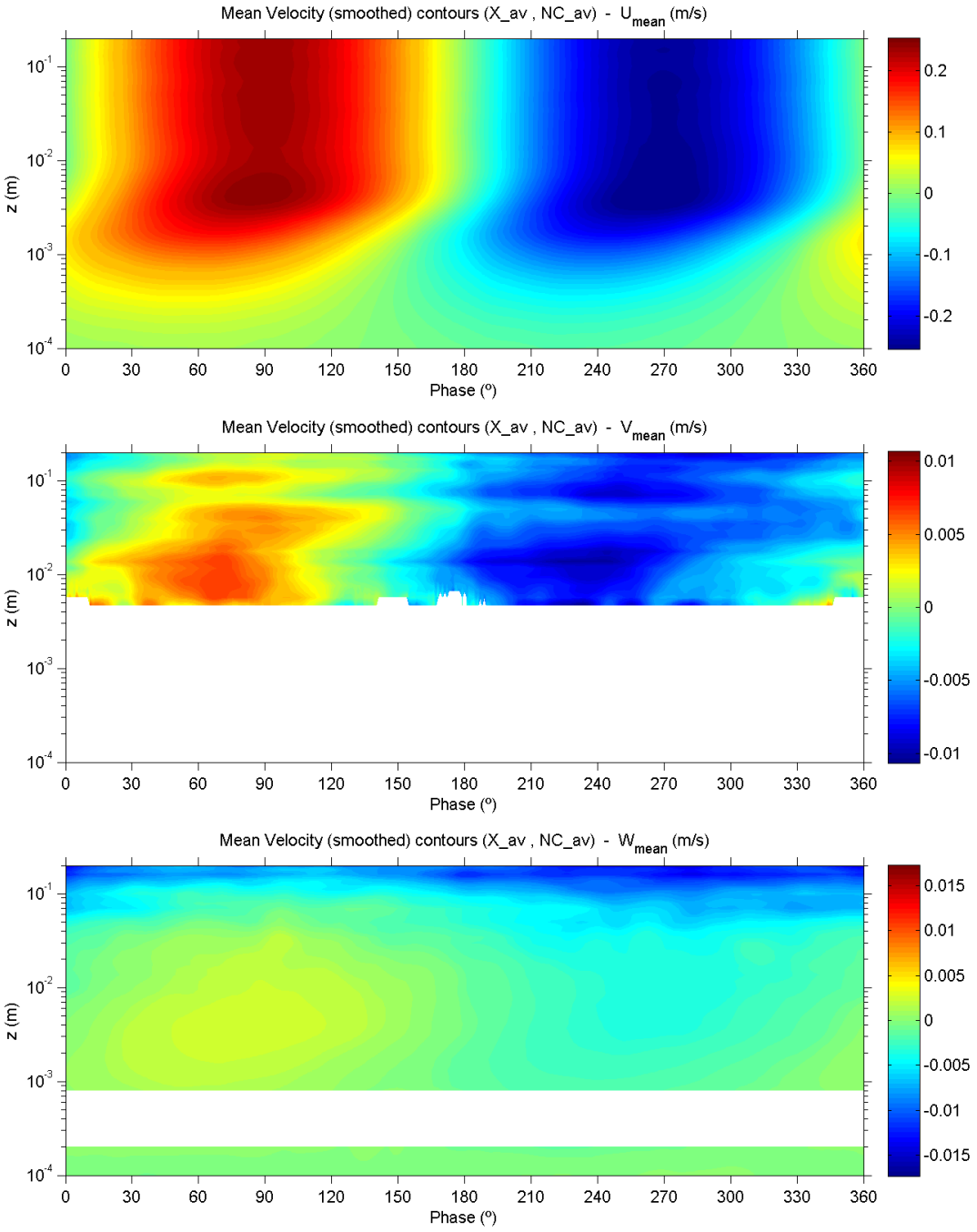


C.2.3 Boundary layer thickness

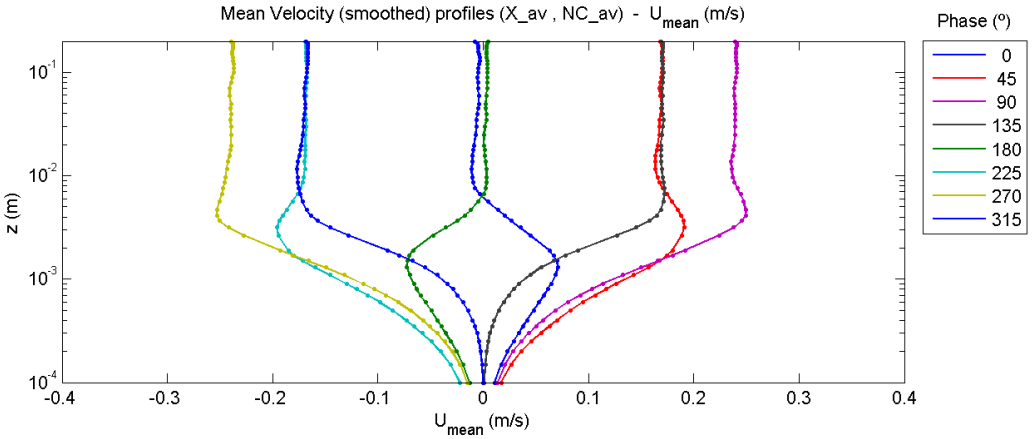


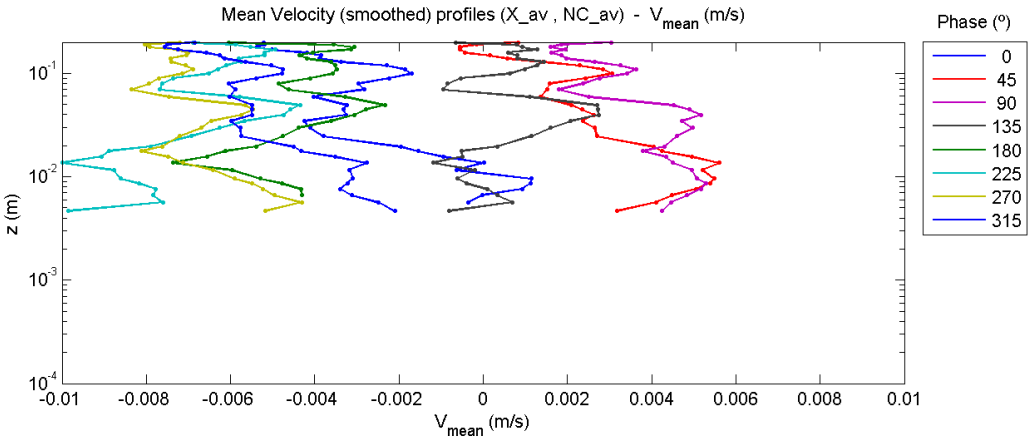
C.2.4 Mean velocities

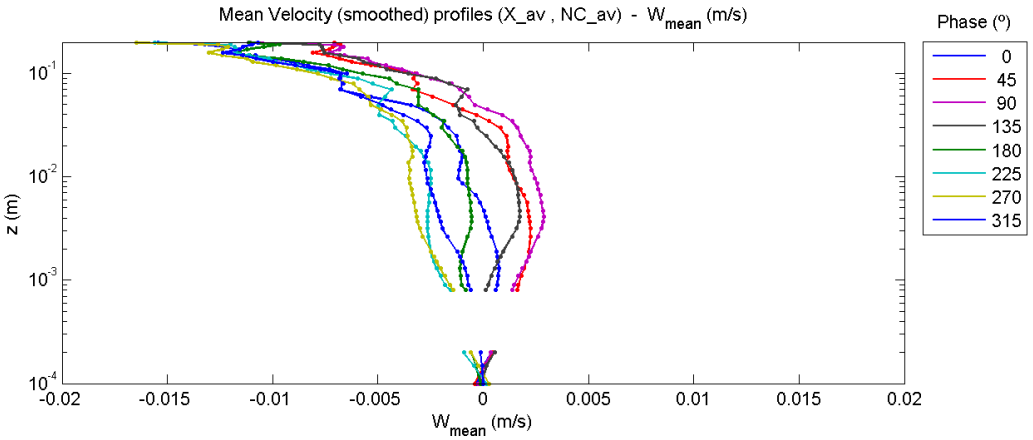
C.2.4.1 Contour plots



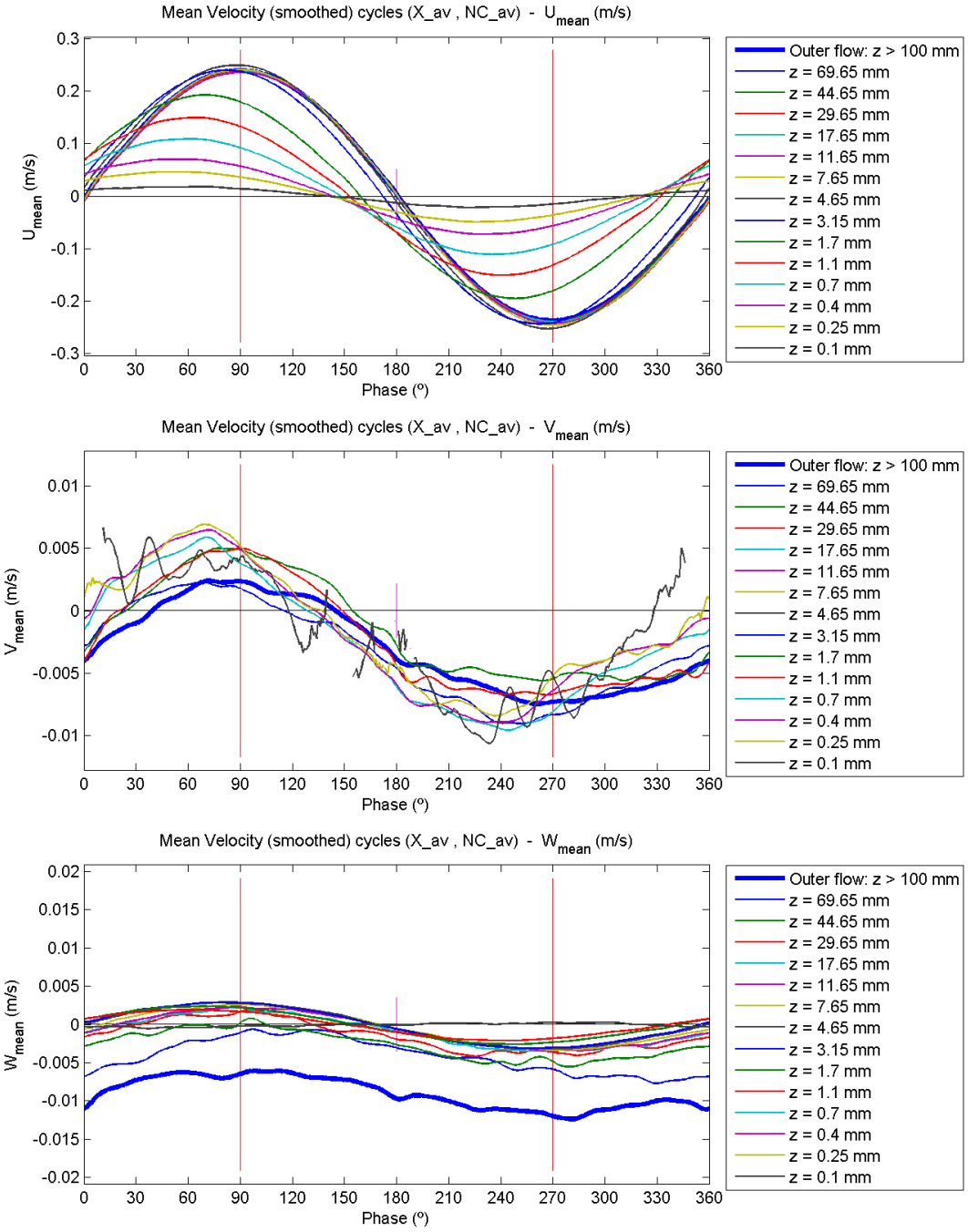
C.2.4.2 Profile plots





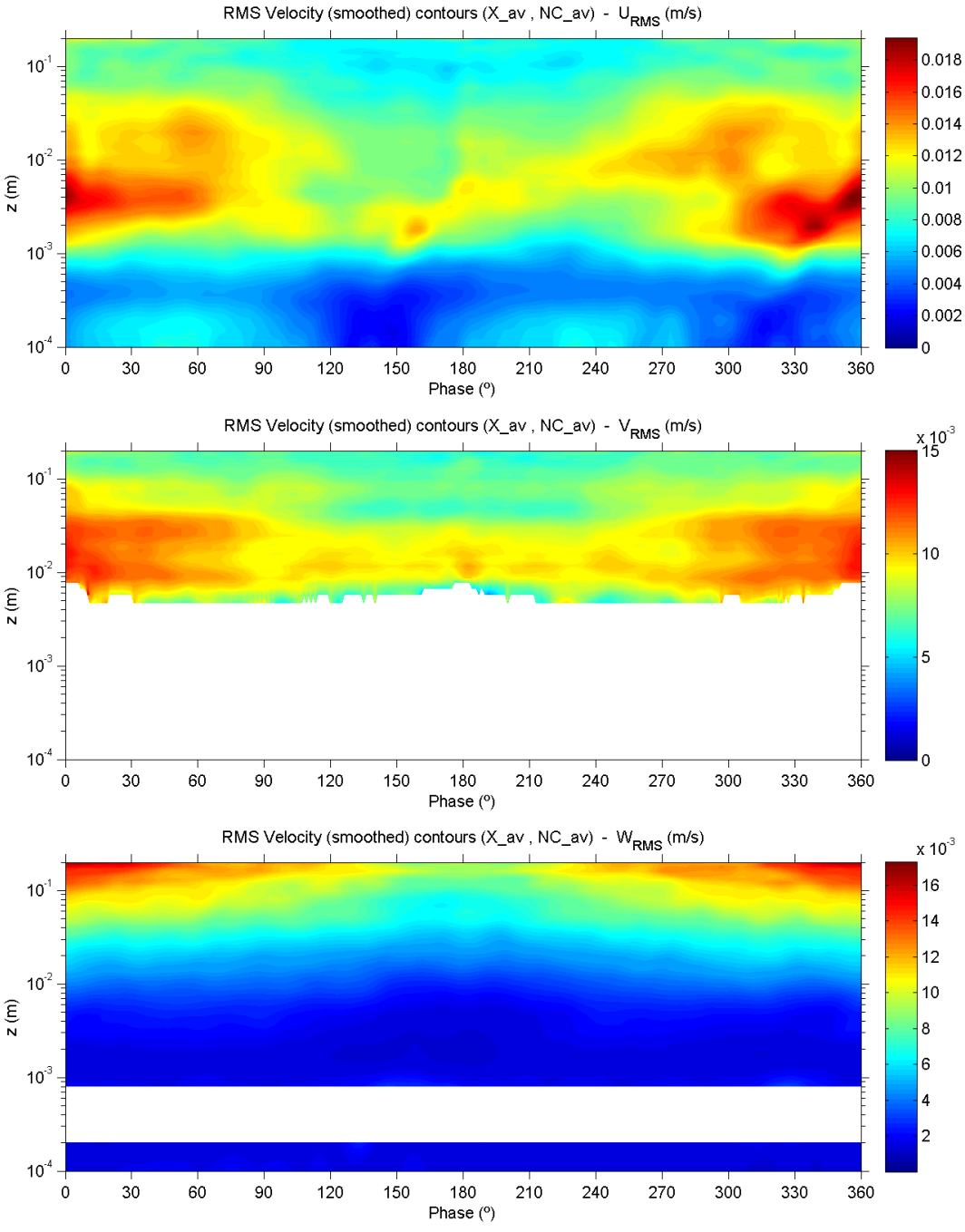


C.2.4.3 Evolution plots

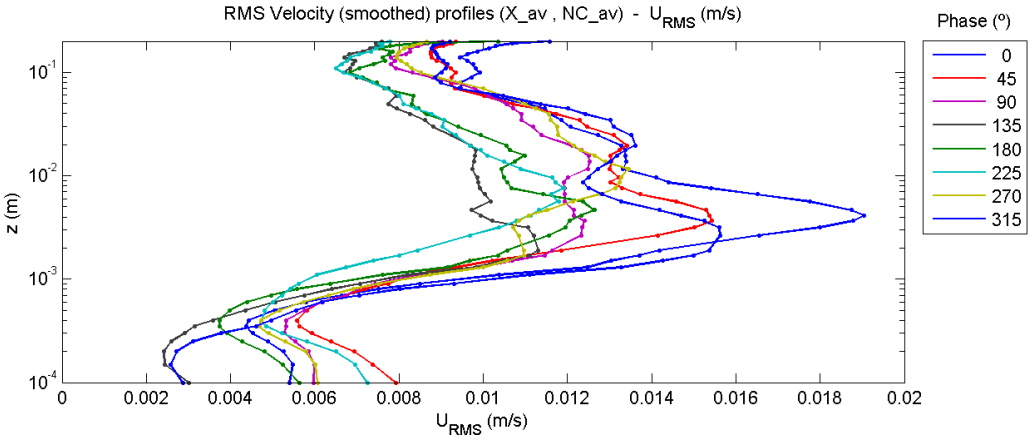


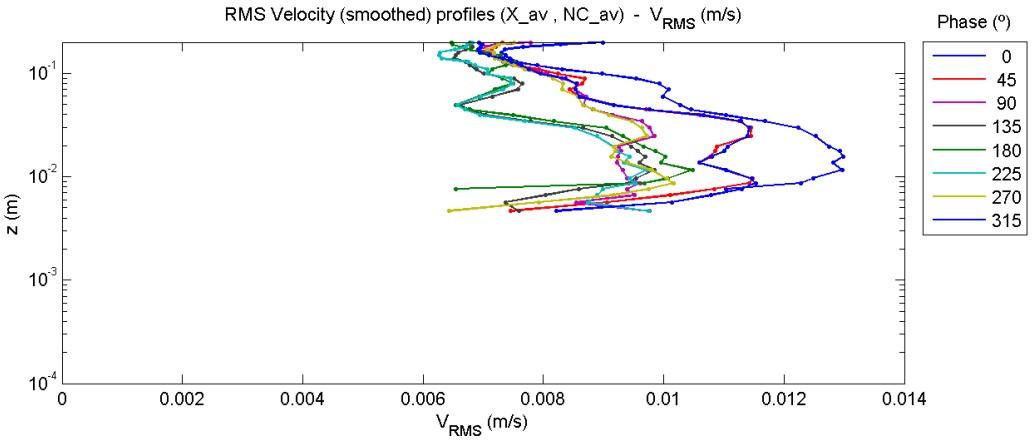
C.2.5 RMS velocity fluctuations

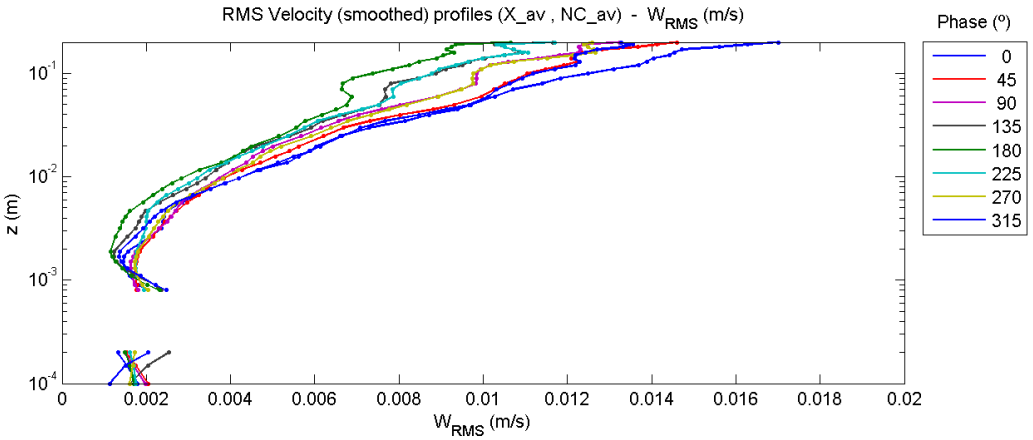
C.2.5.1 Contour plots



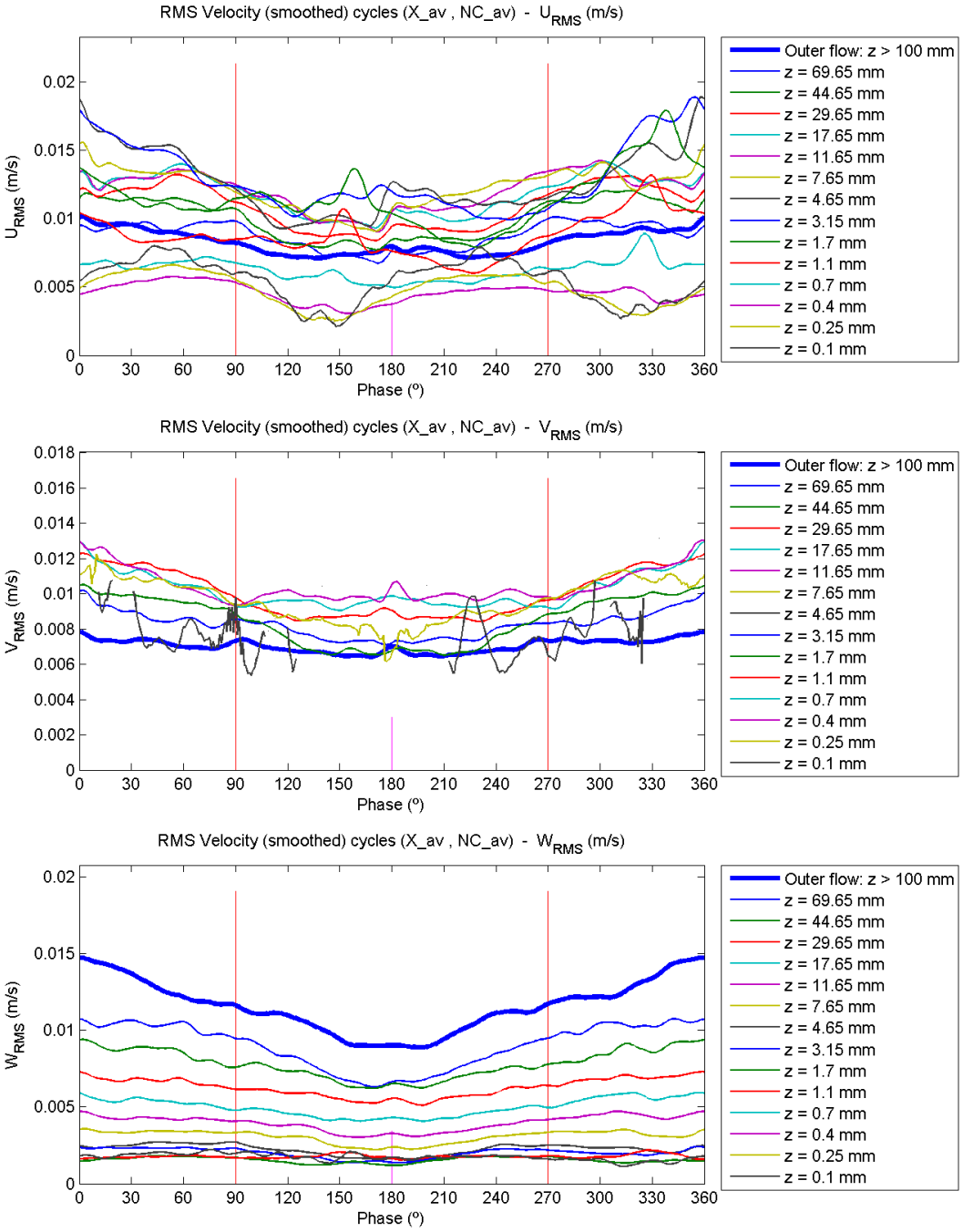
C.2.5.2 Profile plots





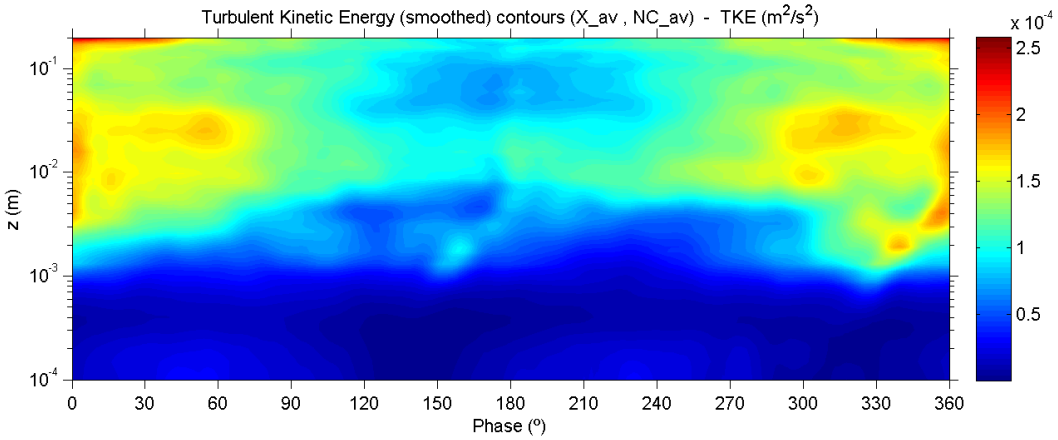


C.2.5.3 Evolution plots

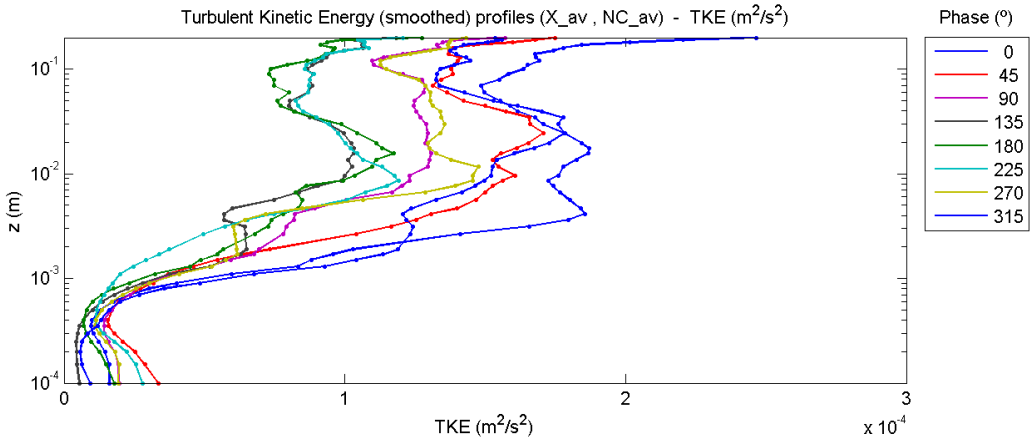


C.2.6 Turbulent kinetic energy

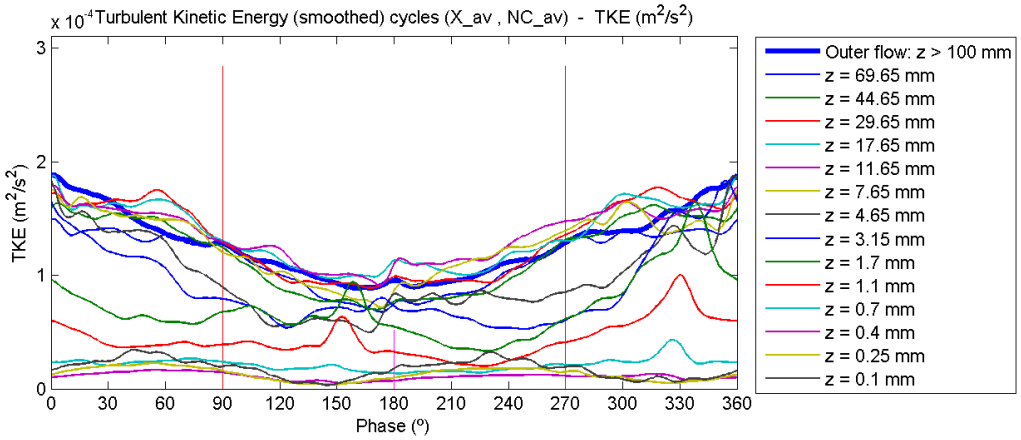
C.2.6.1 Contour plot



C.2.6.2 Profile plot

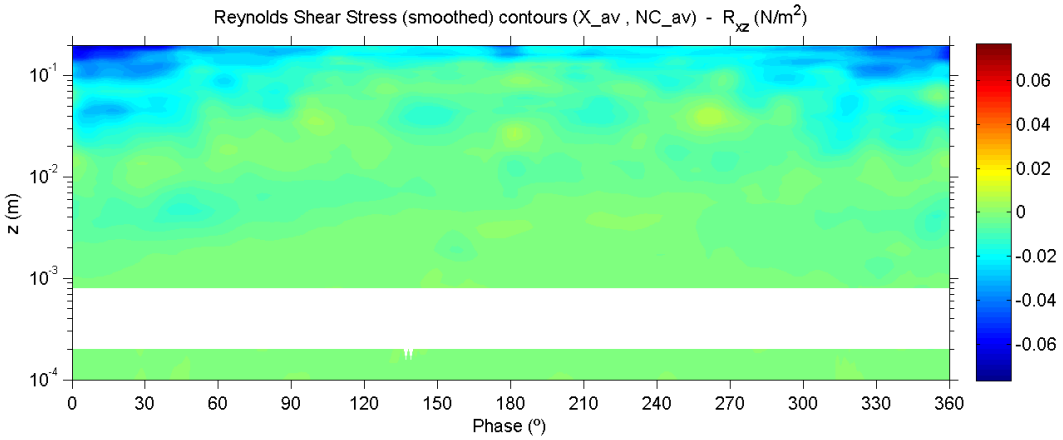


C.2.6.3 Evolution plot

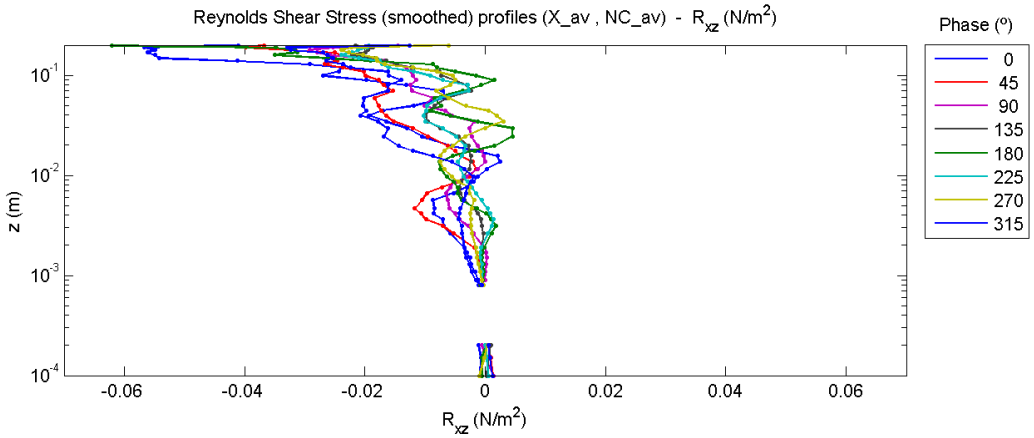


C.2.7 Turbulent shear stress (Reynolds shear stress)

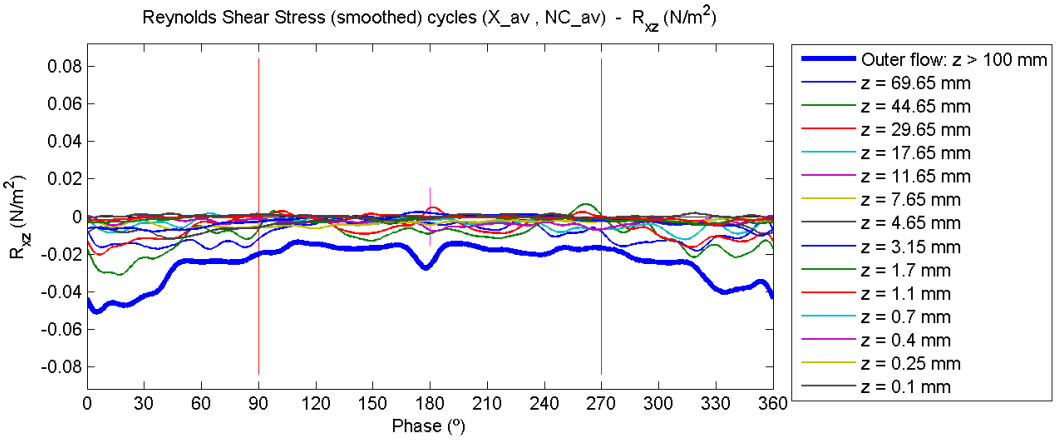
C.2.7.1 Contour plot



C.2.7.2 Profile plot

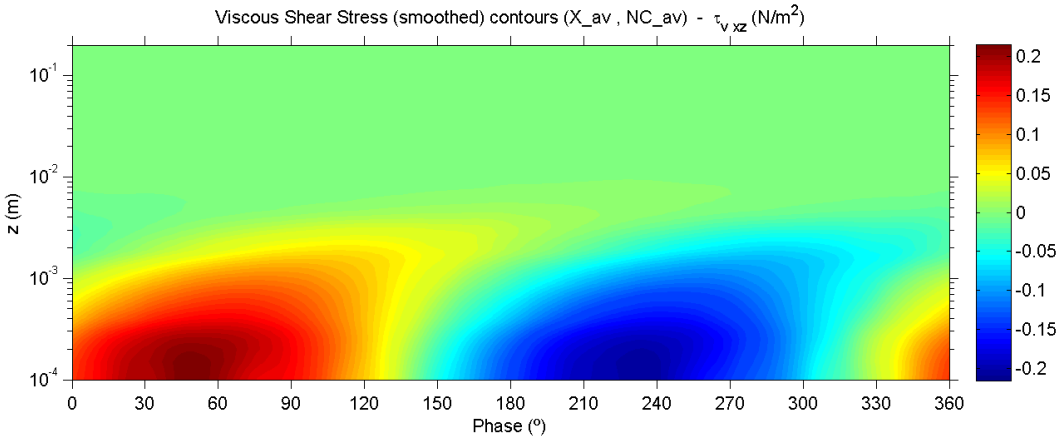


C.2.7.3 Evolution plot

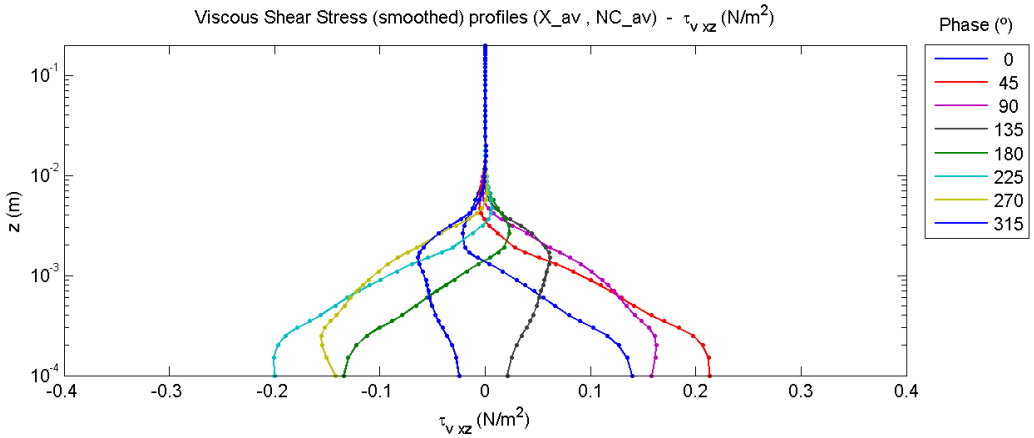


C.2.8 Viscous shear stress

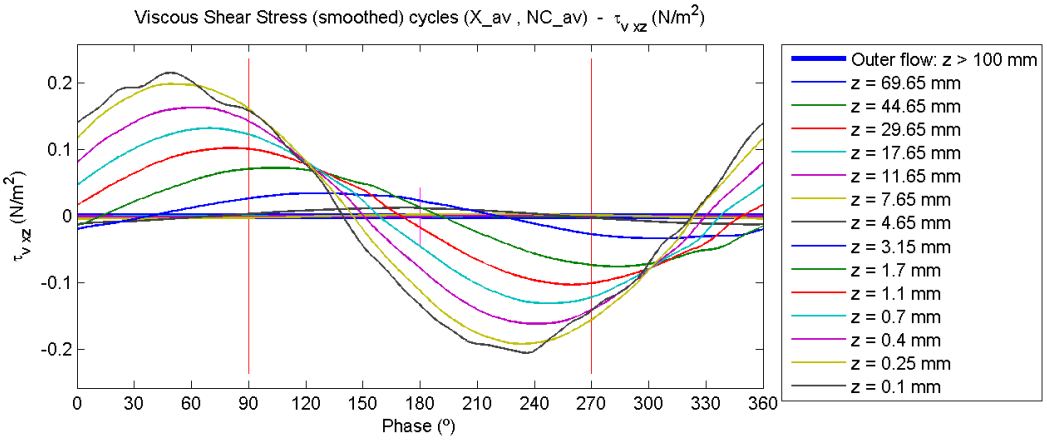
C.2.8.1 Contour plot



C.2.8.2 Profile plot

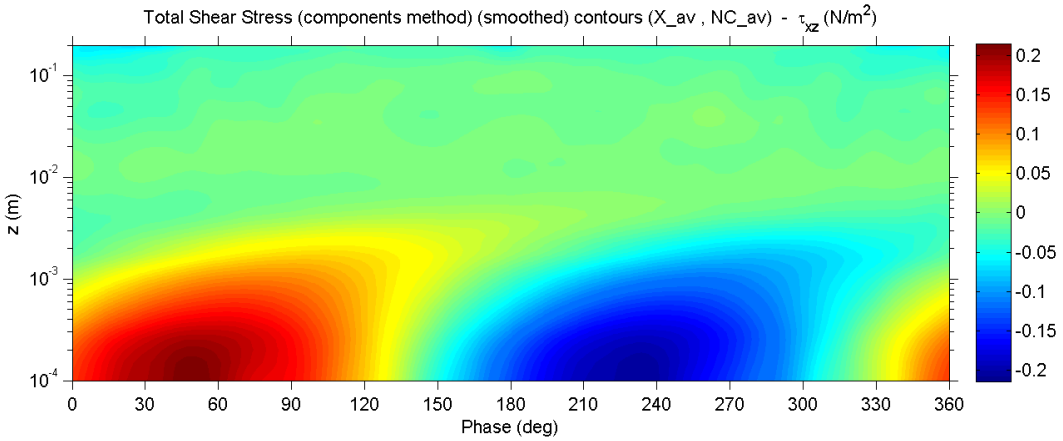


C.2.8.3 Evolution plot

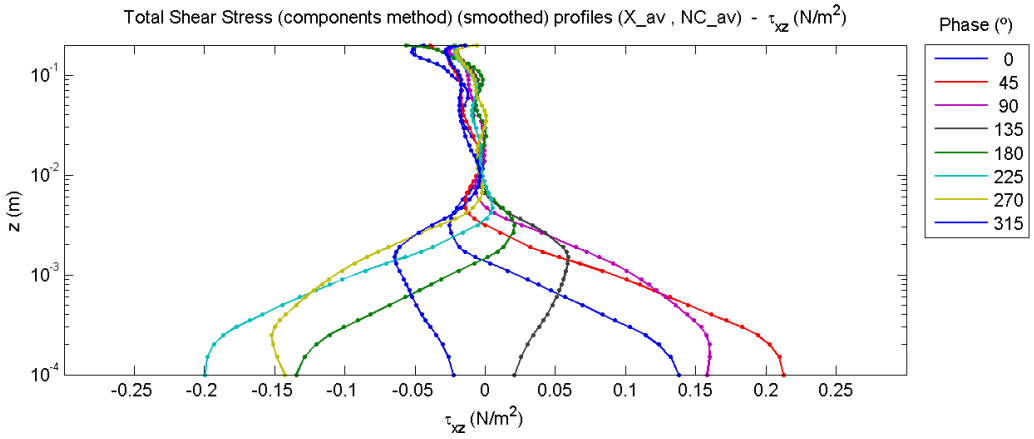


C.2.9 Total shear stress

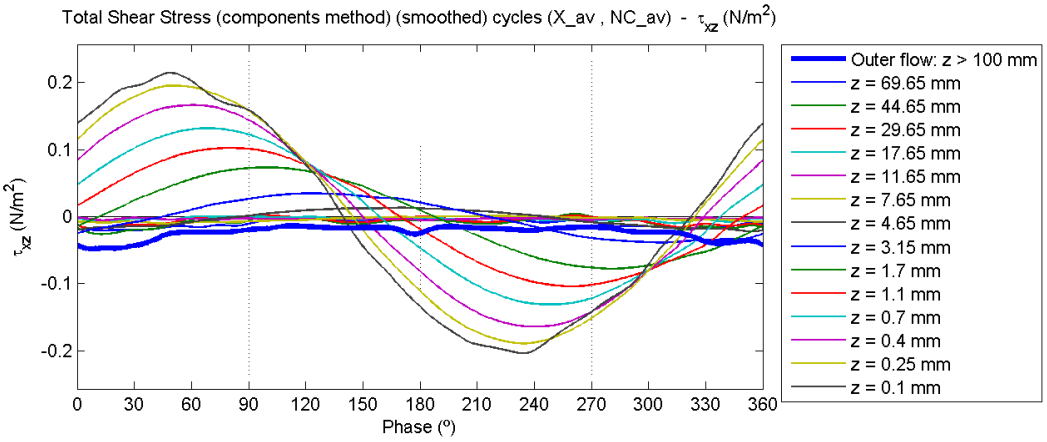
C.2.9.1 Contour plot



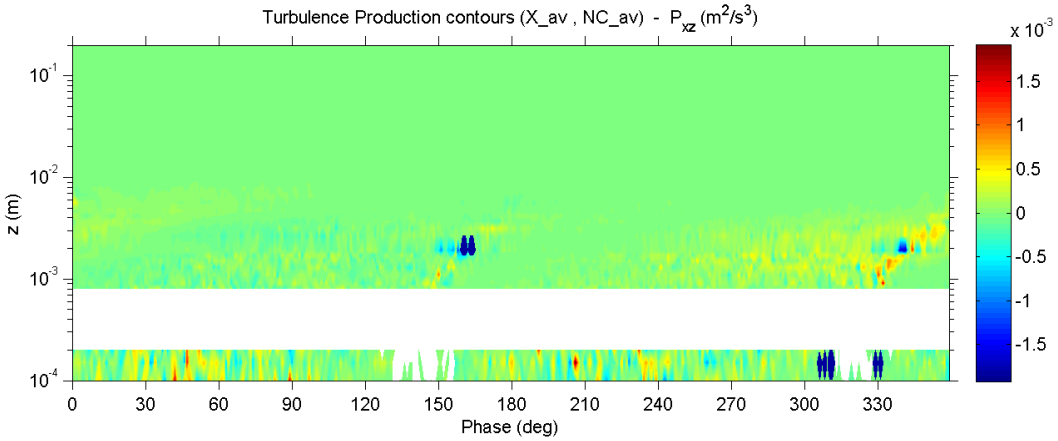
C.2.9.2 Profile plot



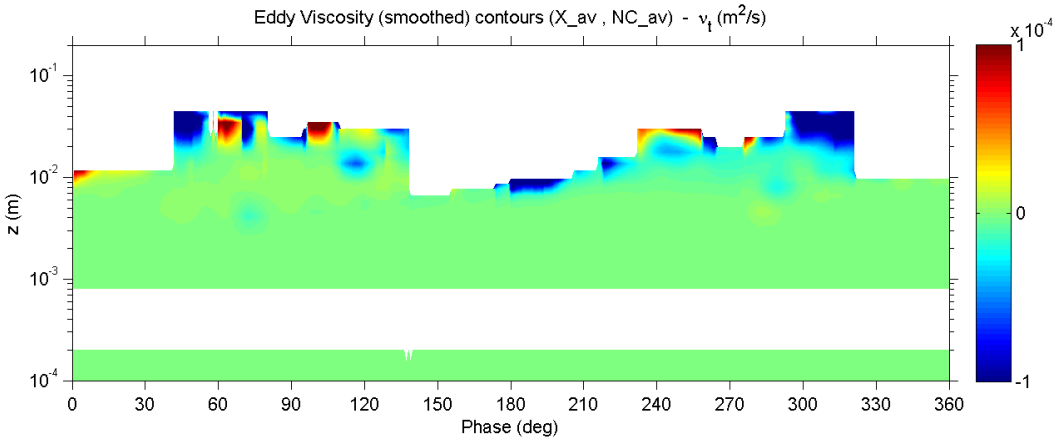
C.2.9.3 Evolution plot



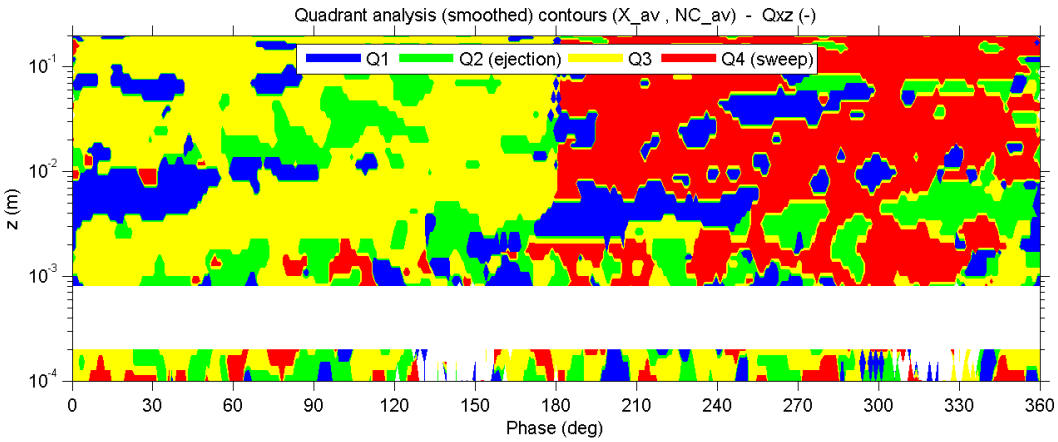
C.2.10 Turbulence production



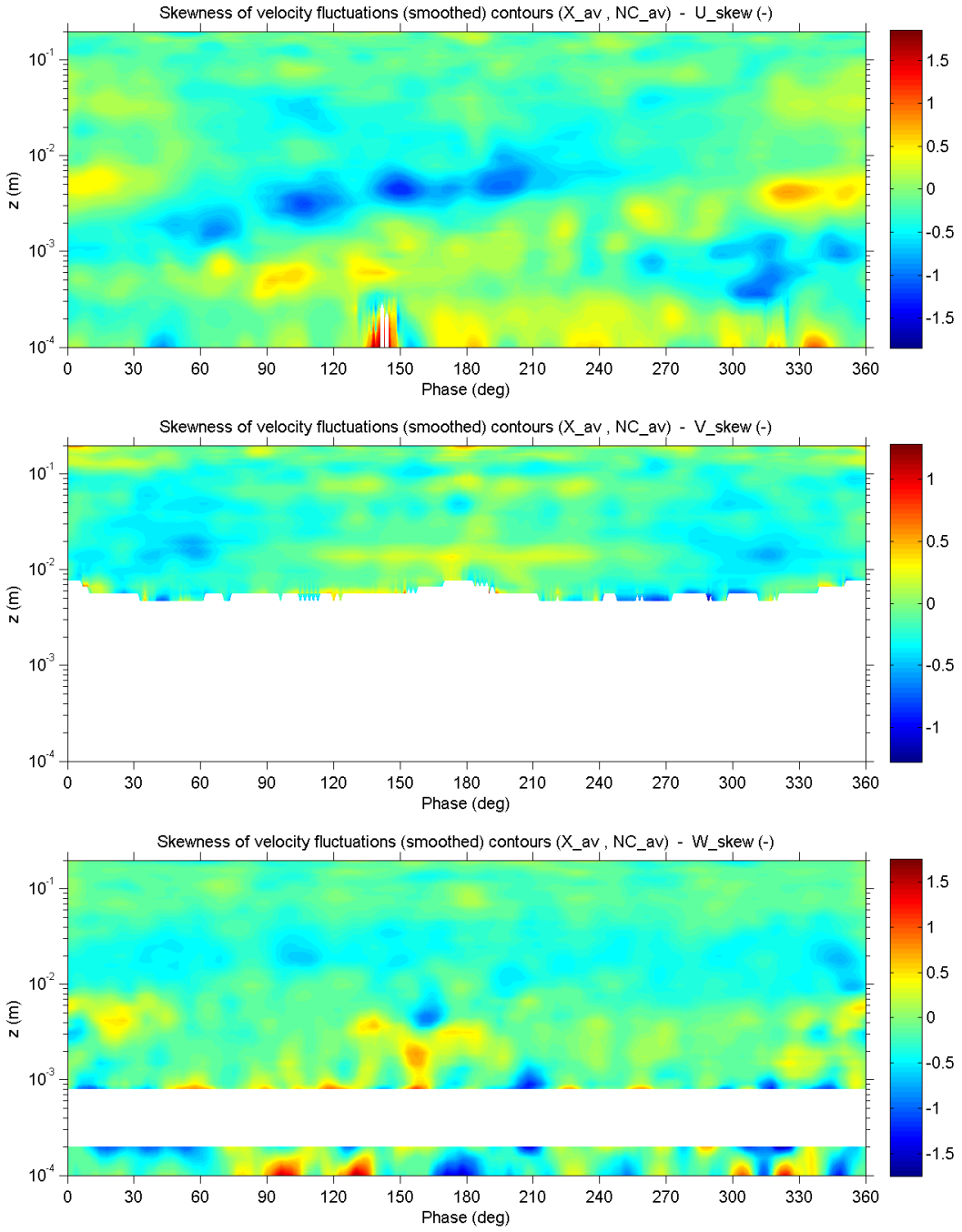
C.2.11 Turbulent viscosity (Eddy viscosity)



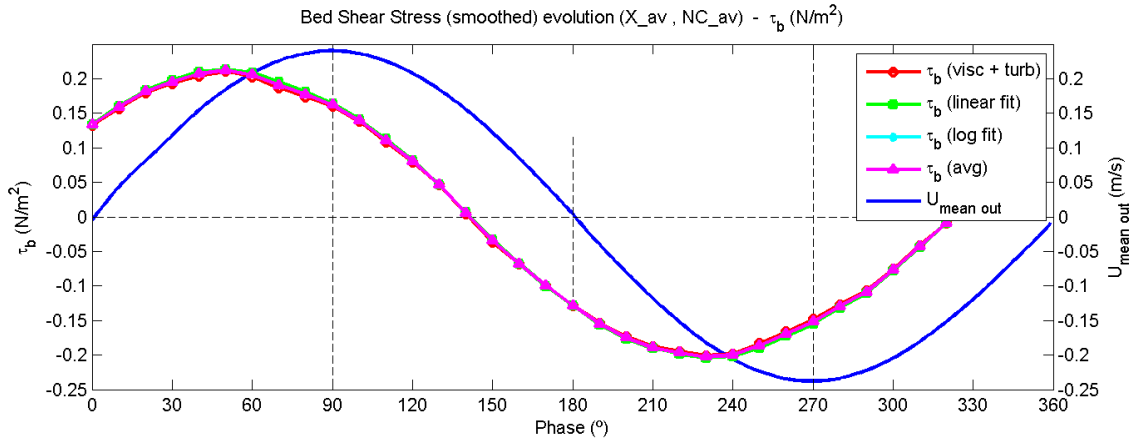
C.2.12 Quadrant analysis



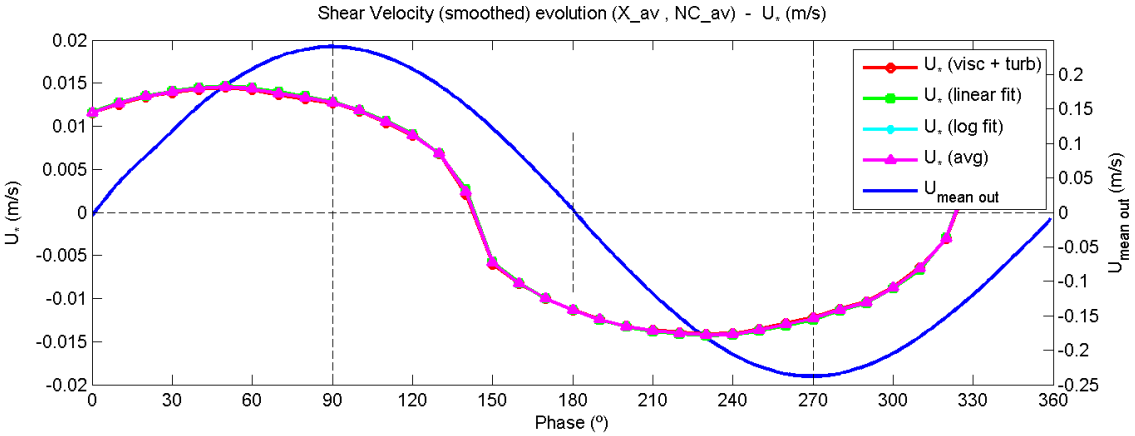
C.2.13 Skewness of velocity fluctuations



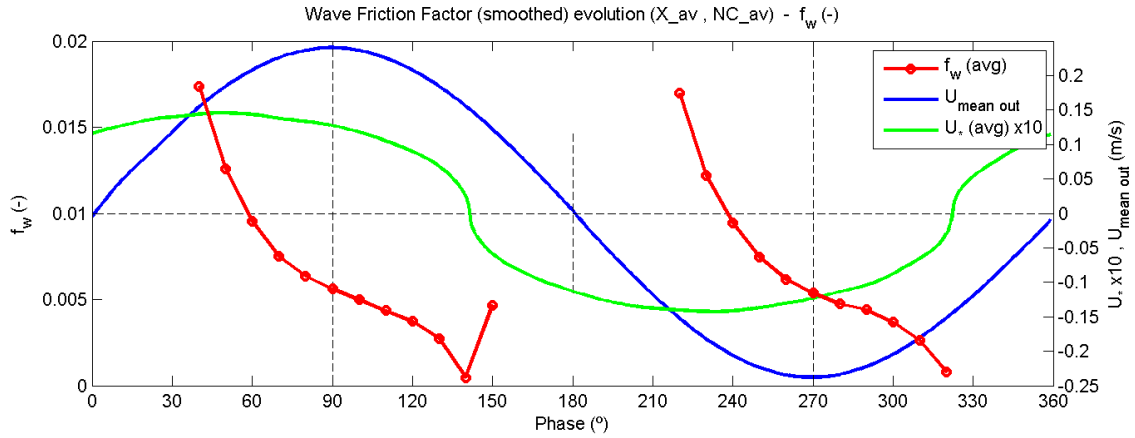
C.2.14 Bed shear stress



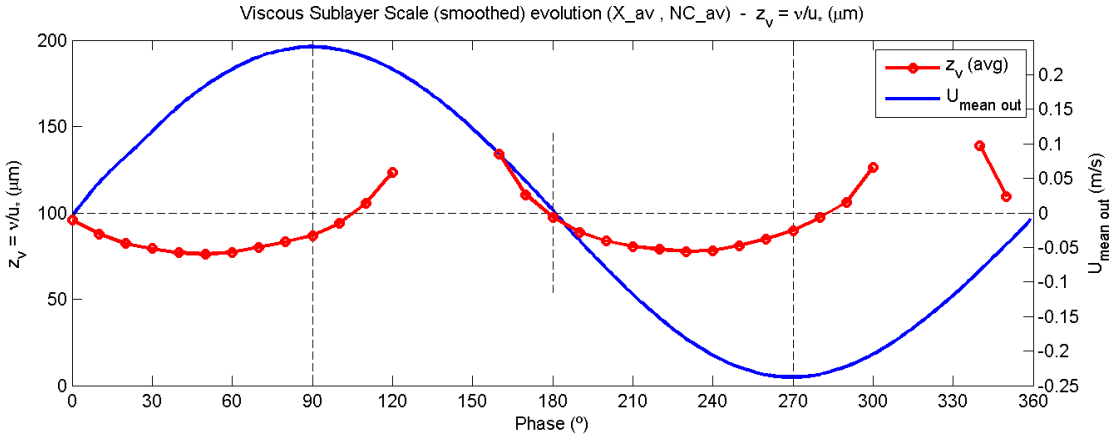
C.2.15 Shear velocity



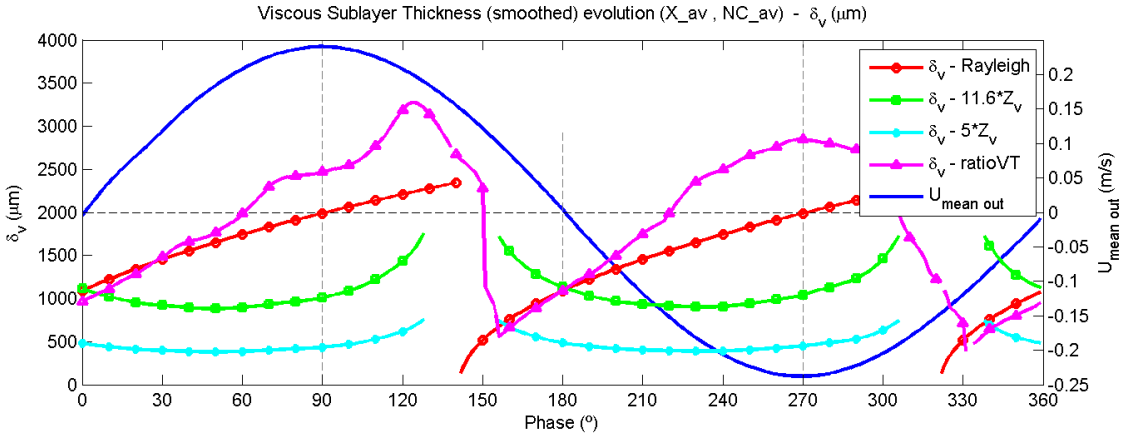
C.2.16 Wave friction factor



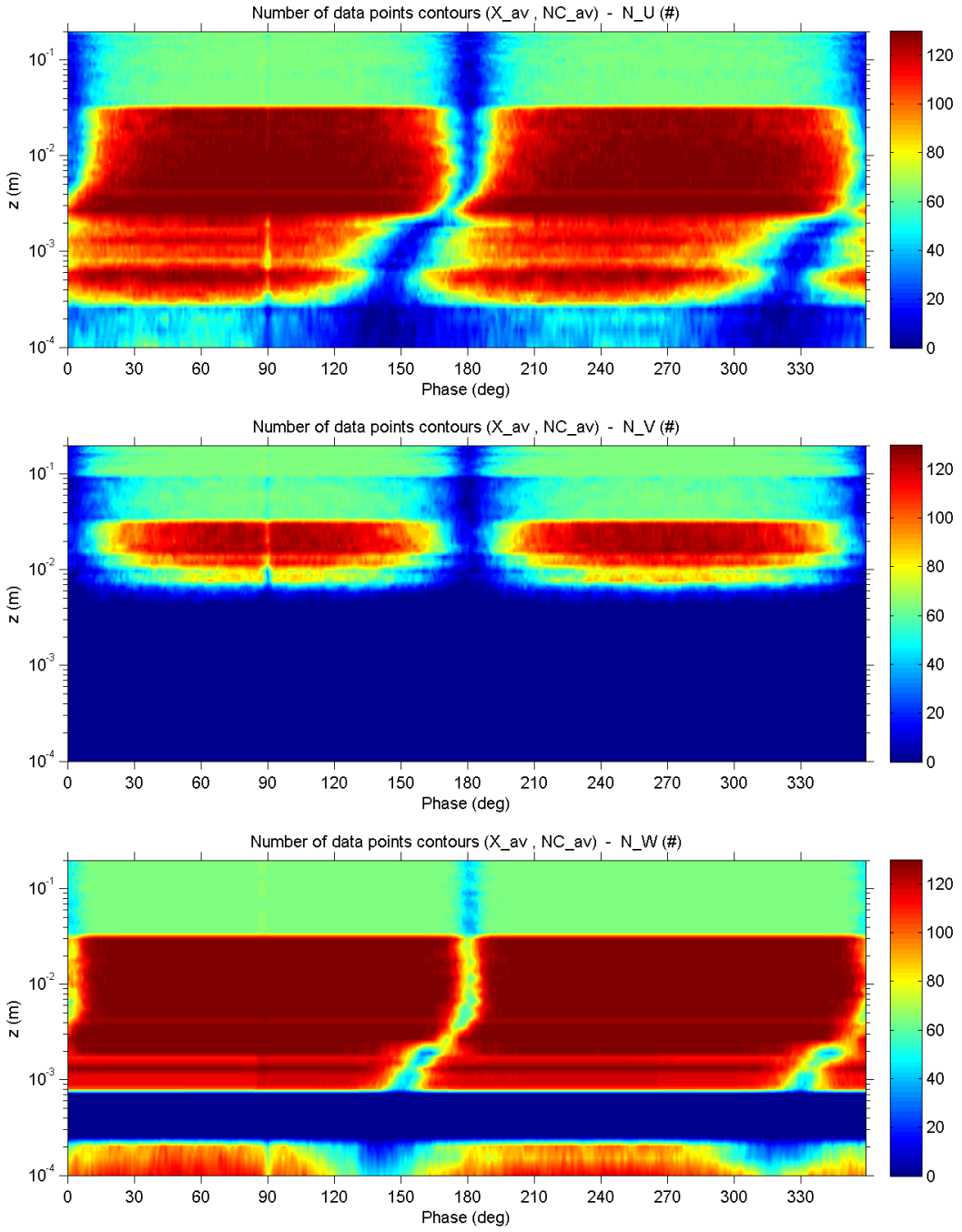
C.2.17 Viscous length scale



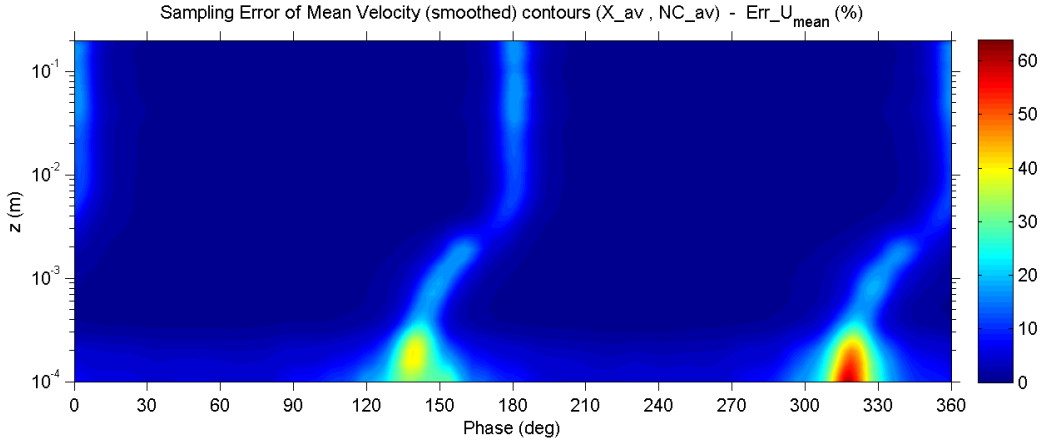
C.2.18 Viscous sublayer thickness



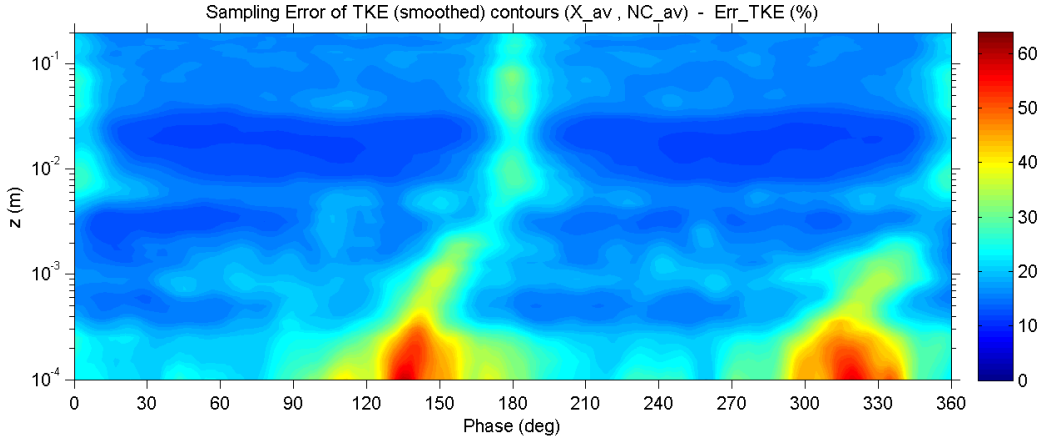
C.2.19 Number of valid data points



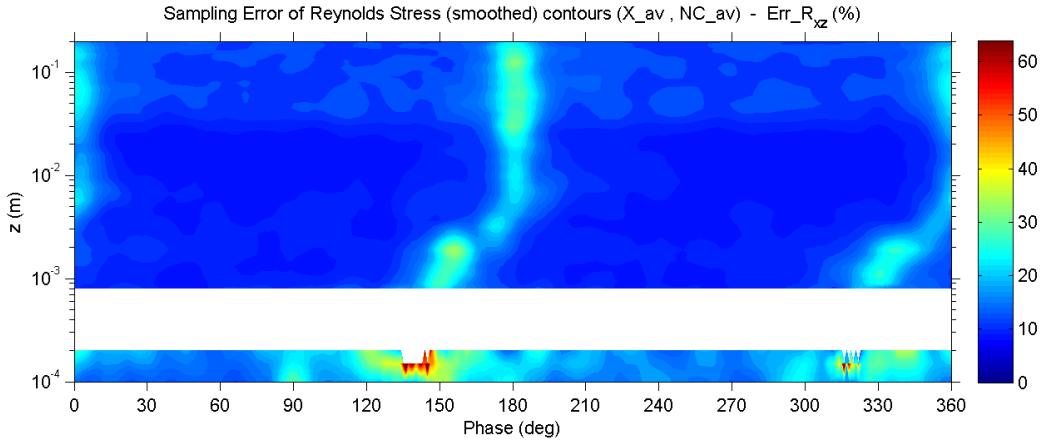
C.2.20 Percent error of mean velocity



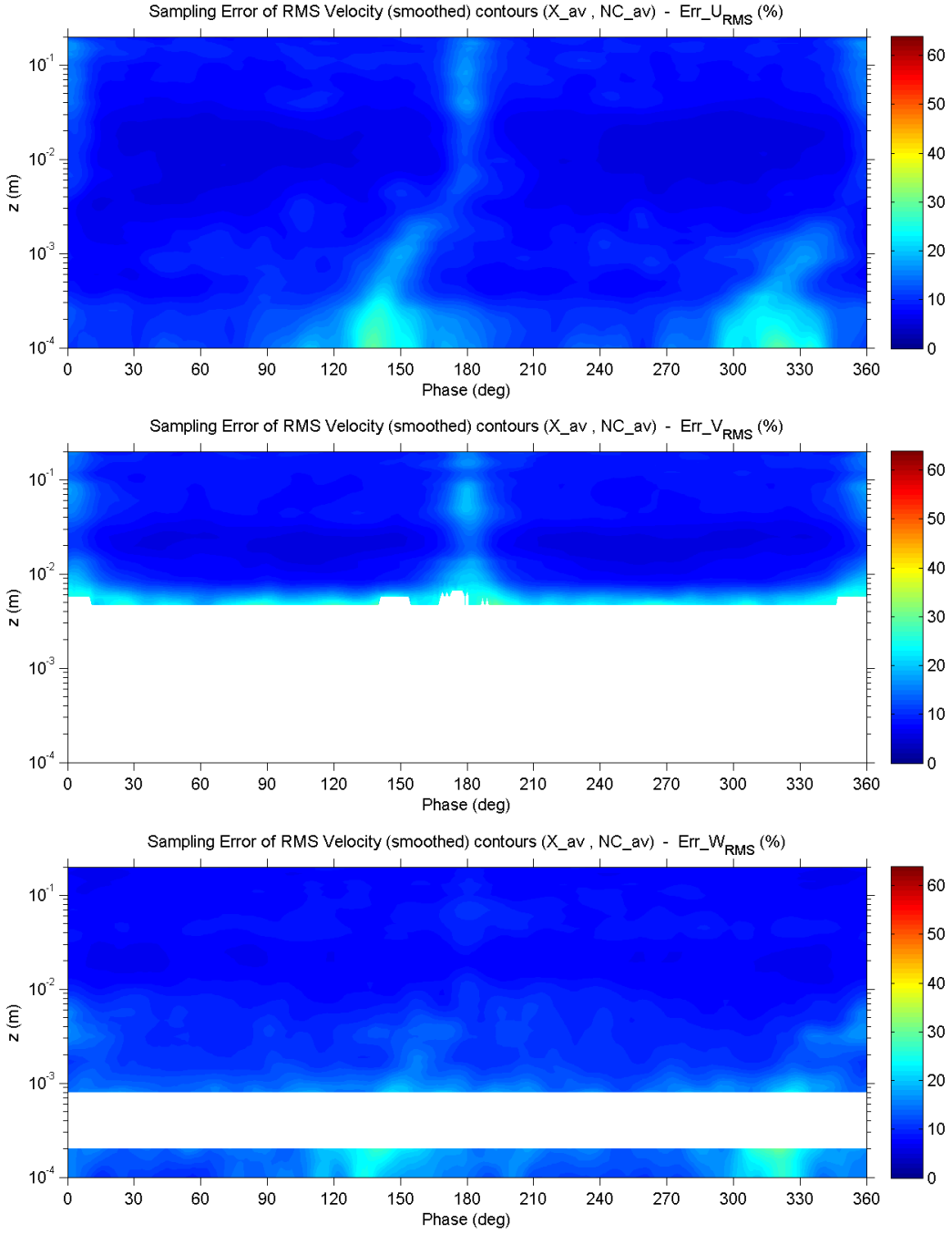
C.2.21 Percent error of turbulent kinetic energy



C.2.22 Percent error of Reynolds shear stress



C.2.23 Percent error of RMS velocity fluctuations



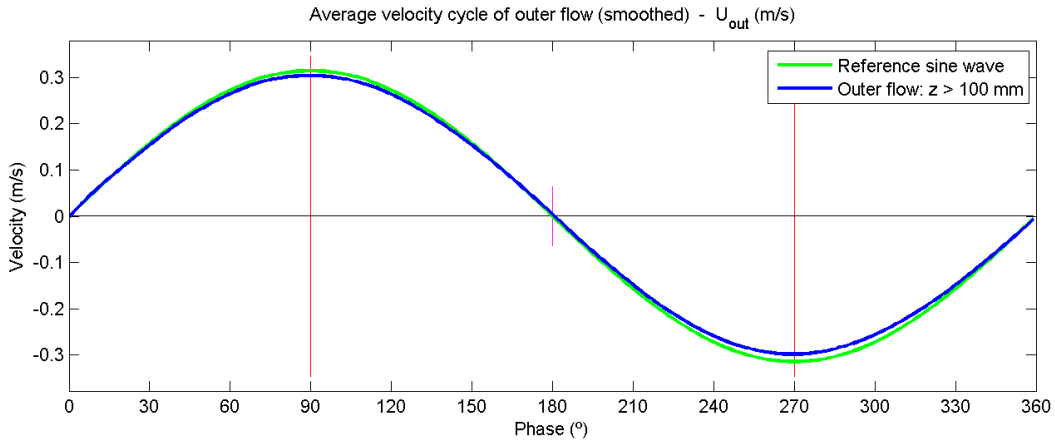
C.3 Experiment no. 3

C.3.1 Main parameters

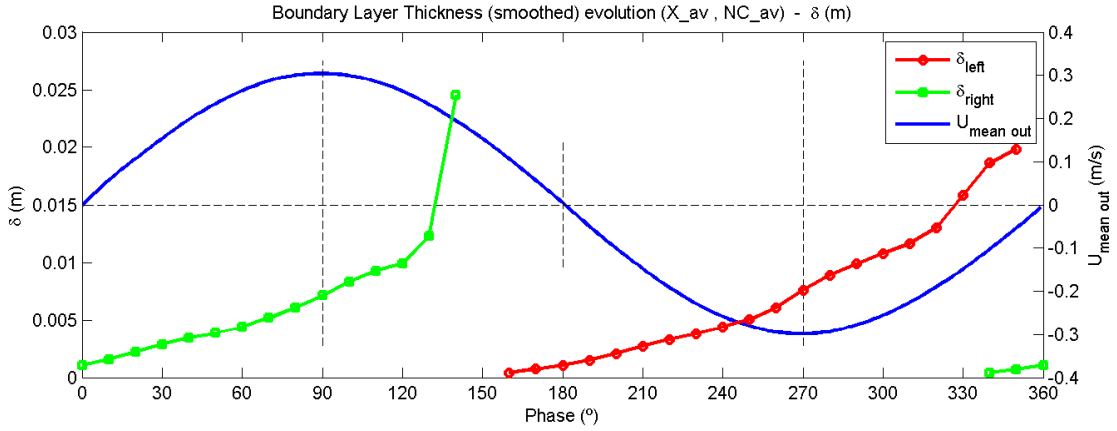
Exp no.	Temp (ºC)	ρ (kg/m^3)	ν (m^2/s)	T (s)	$2a$ (m)	U_{out_max} (m/s)	Re_w (-)	N_{max} (cycles)
3	23.2	997.52	9.45E-07	10	0.958	0.301	1.5E+05	130

Exp no.	U_{*max} (m/s)	τ_{b_max} (N/m^2)	$\Delta\phi$ (deg)	f_{w_ref} (-)	Z_{v_min} (mm)	δ_{v_90} (mm)	δ_{90} (mm)	δ_{top} (mm)
3	0.016	0.26	37.5	0.0057	0.059	0.693	7.3	19

C.3.2 Outer flow velocity

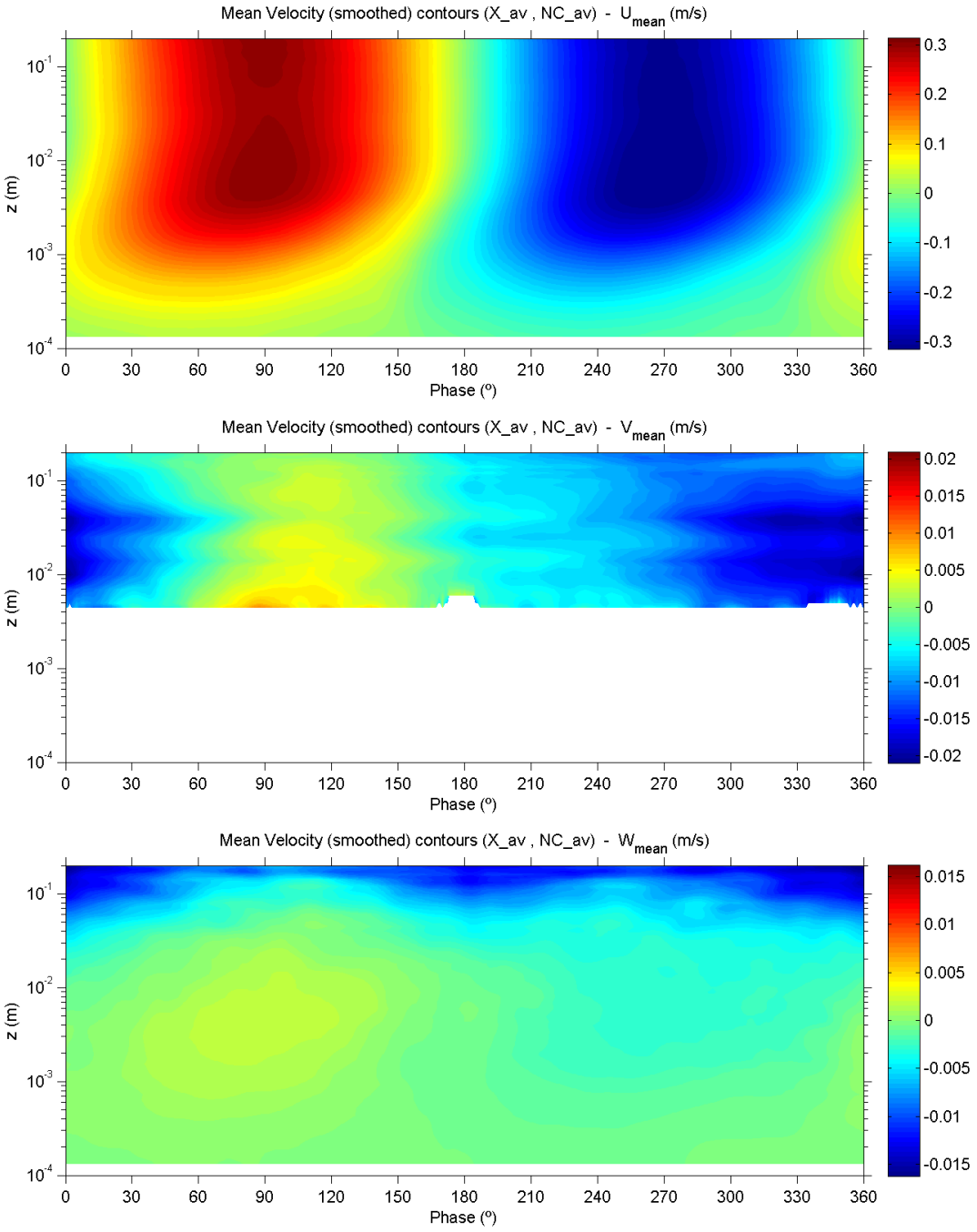


C.3.3 Boundary layer thickness

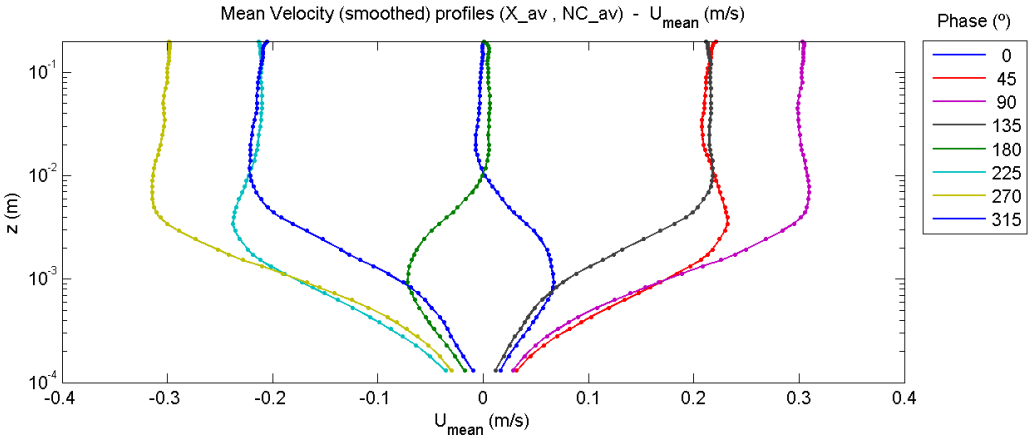


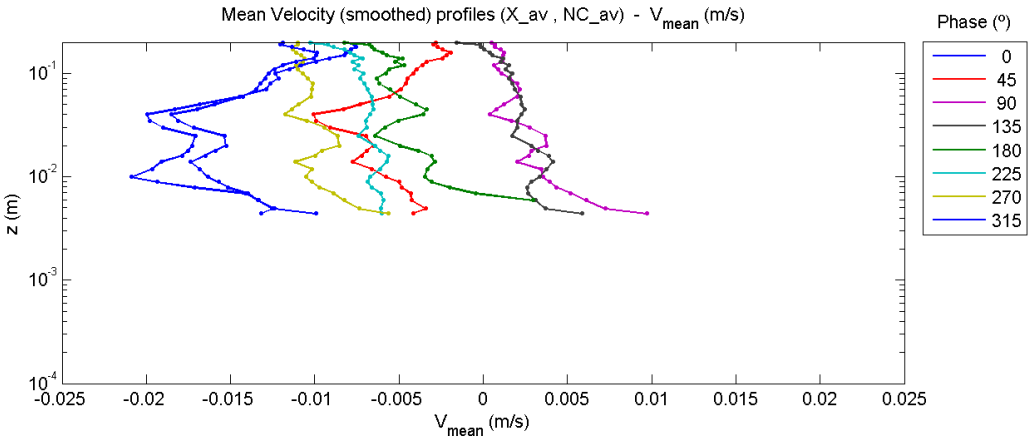
C.3.4 Mean velocities

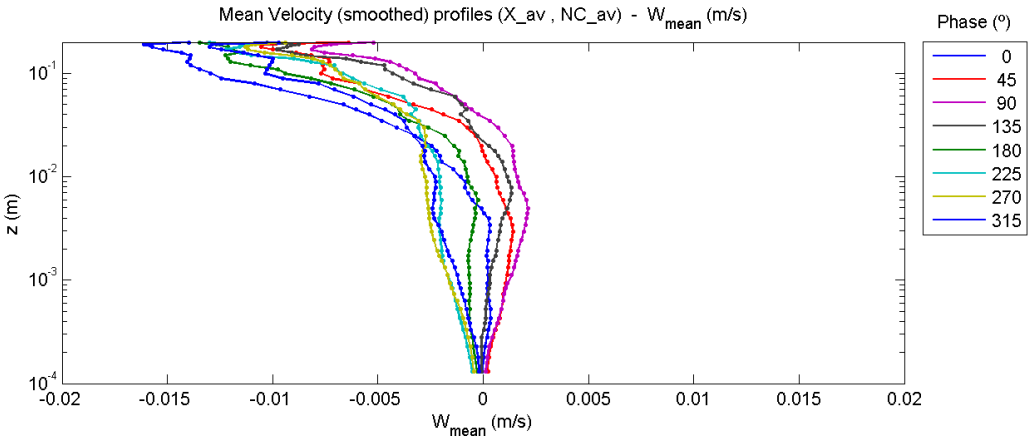
C.3.4.1 Contour plots



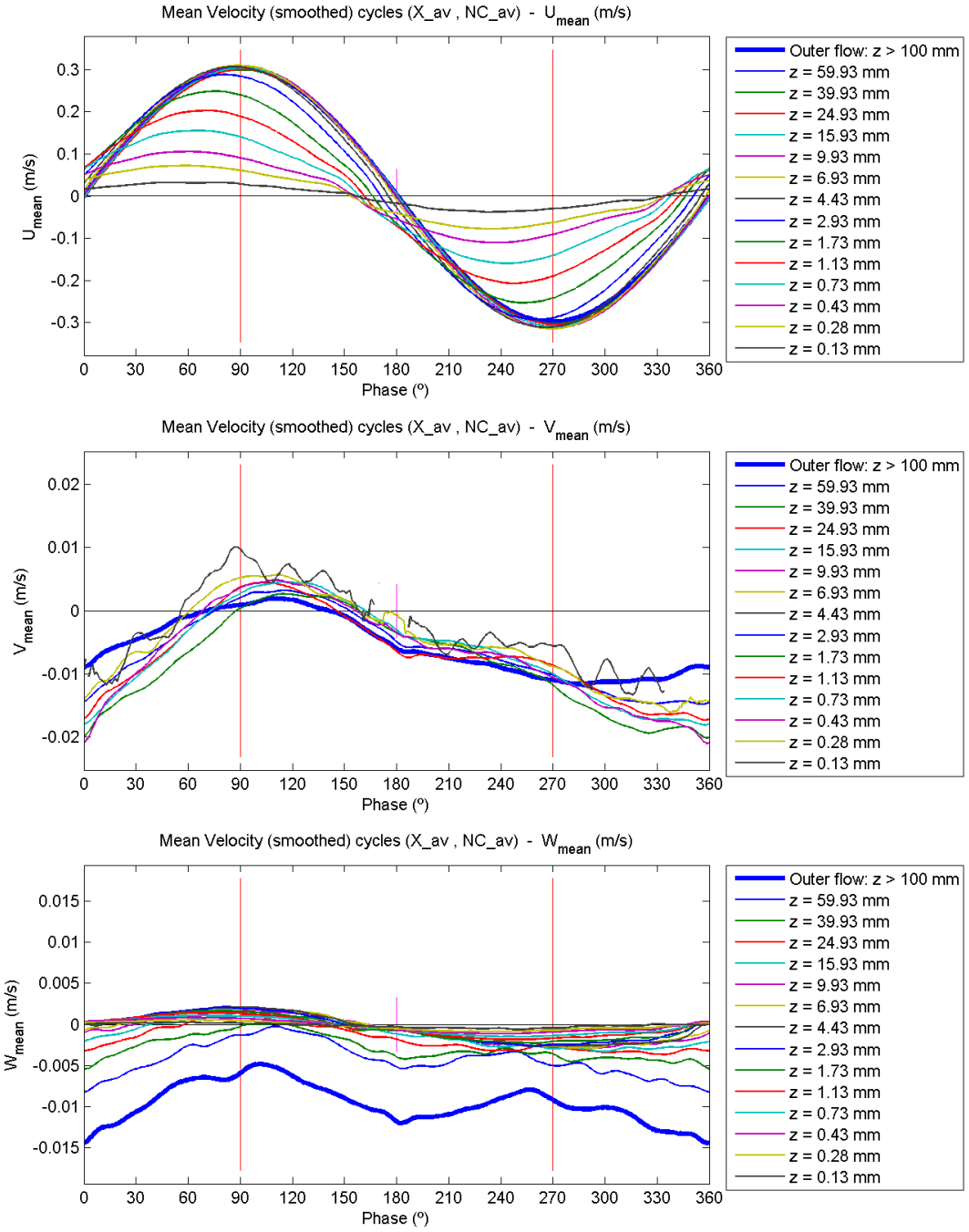
C.3.4.2 Profile plots





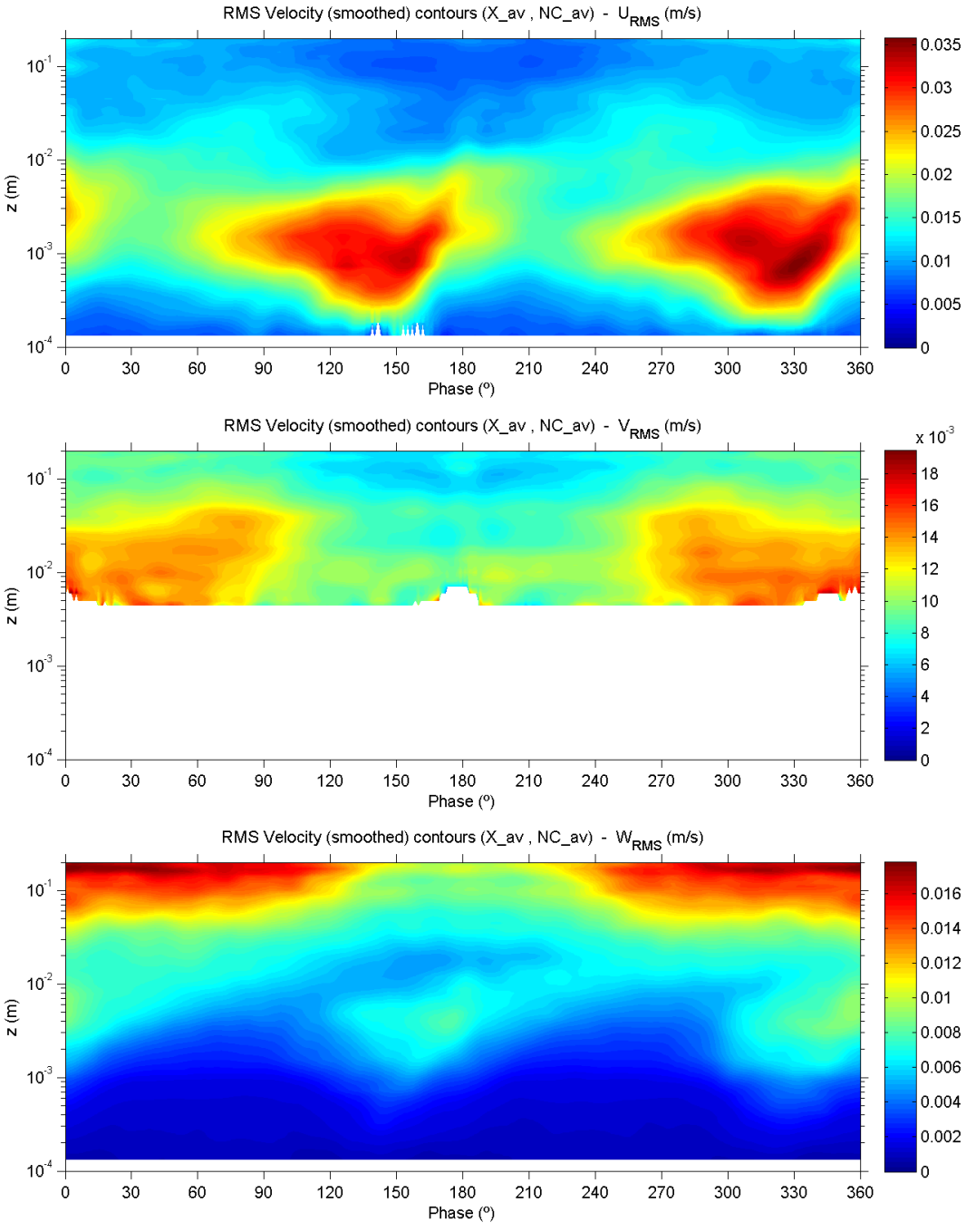


C.3.4.3 Evolution plots

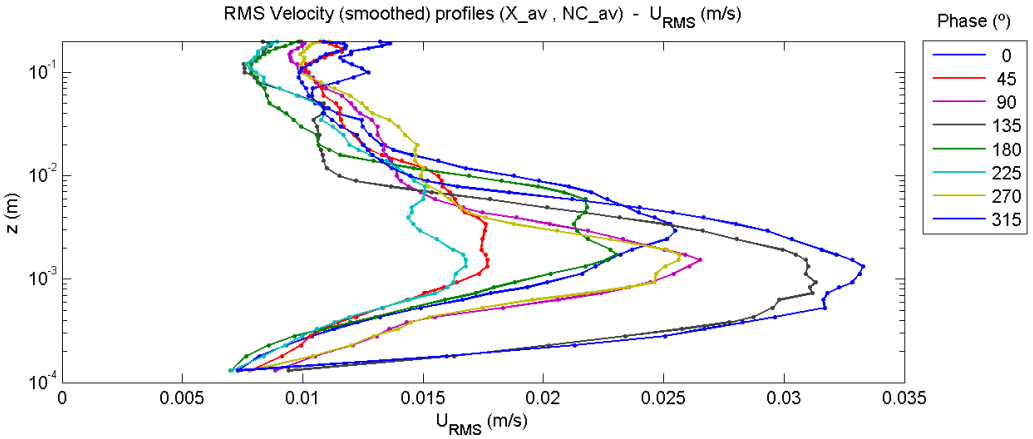


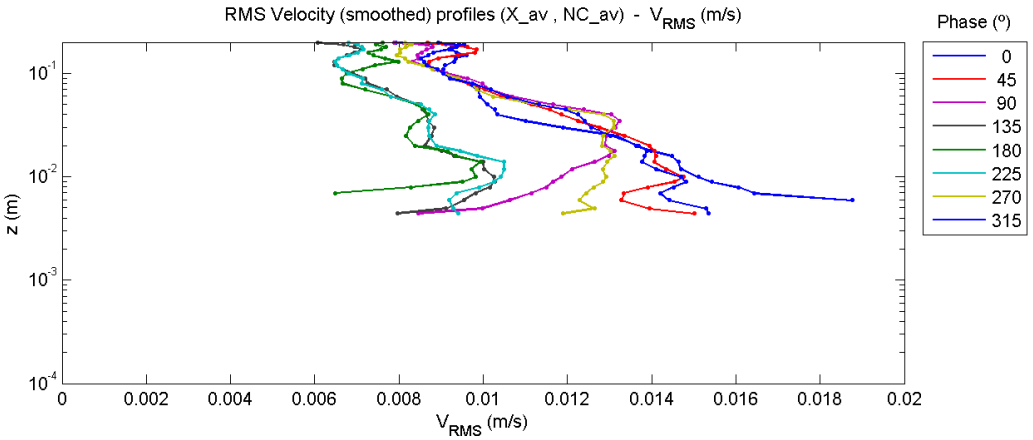
C.3.5 RMS velocity fluctuations

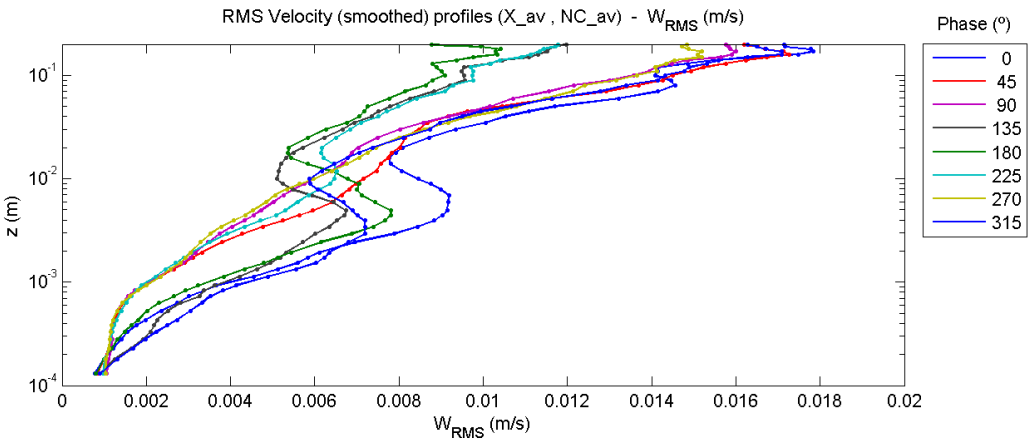
C.3.5.1 Contour plots



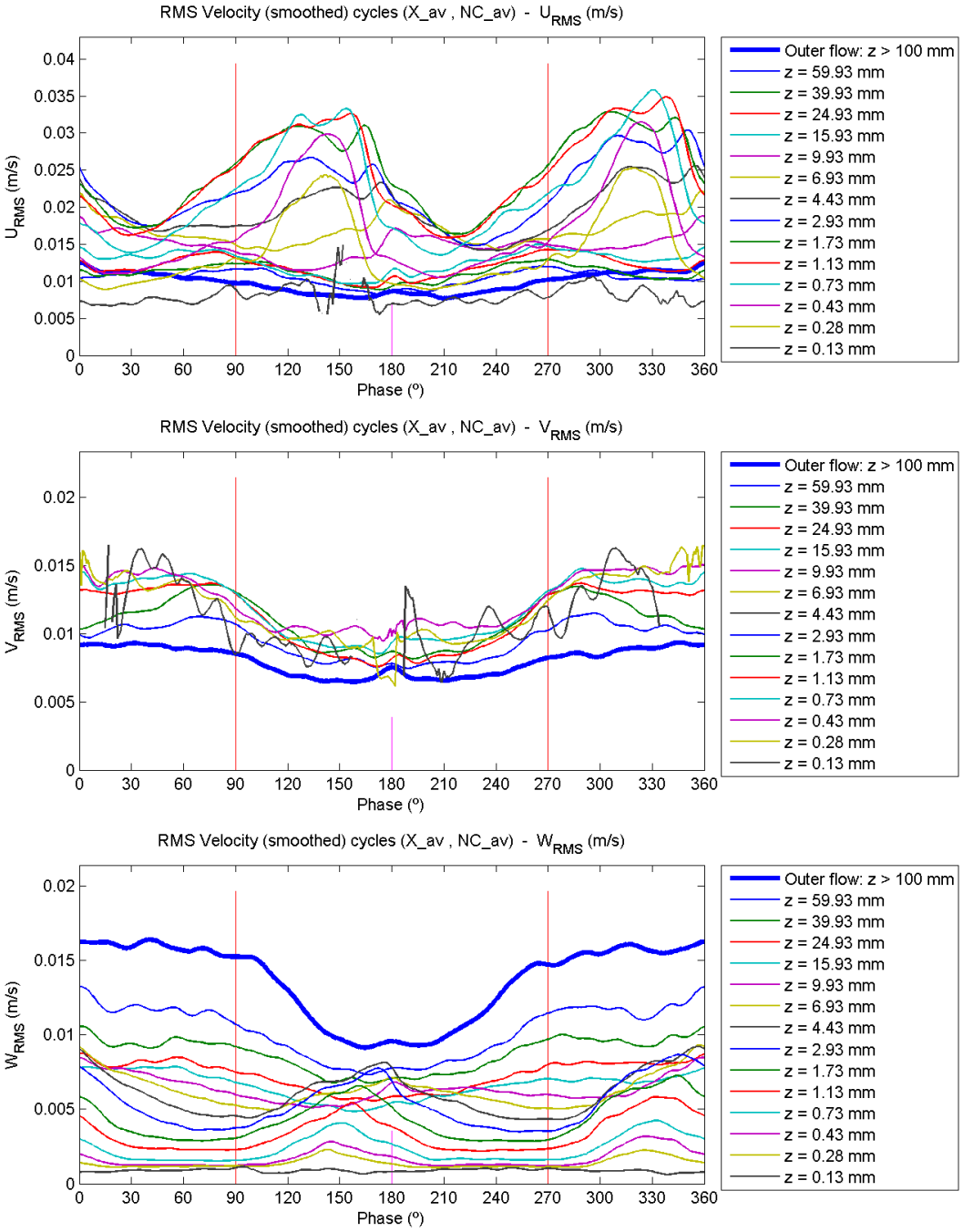
C.3.5.2 Profile plots





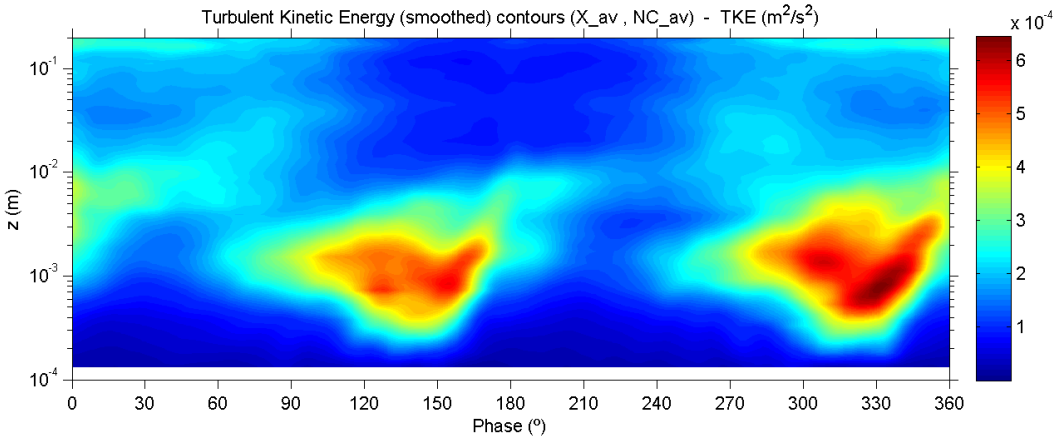


C.3.5.3 Evolution plots

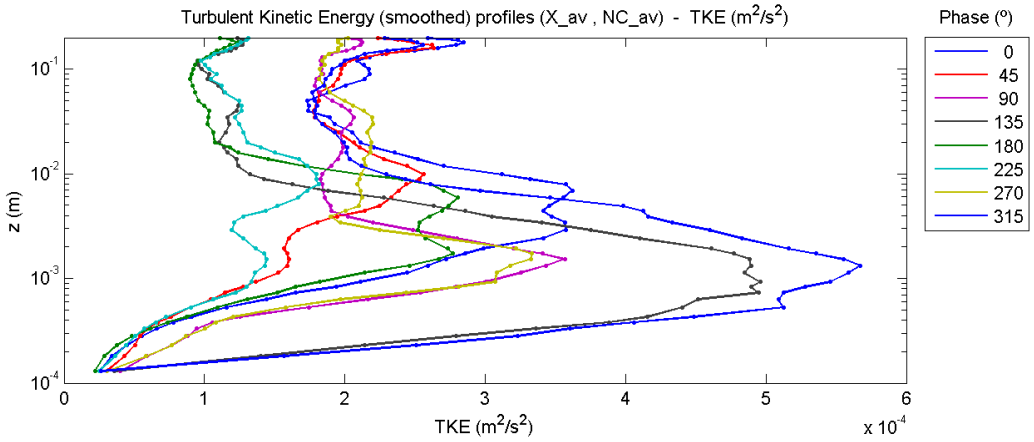


C.3.6 Turbulent kinetic energy

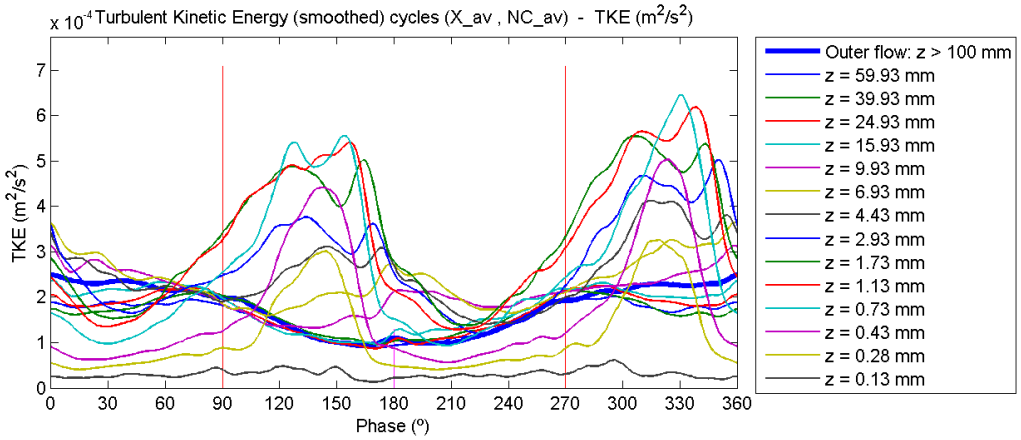
C.3.6.1 Contour plot



C.3.6.2 Profile plot

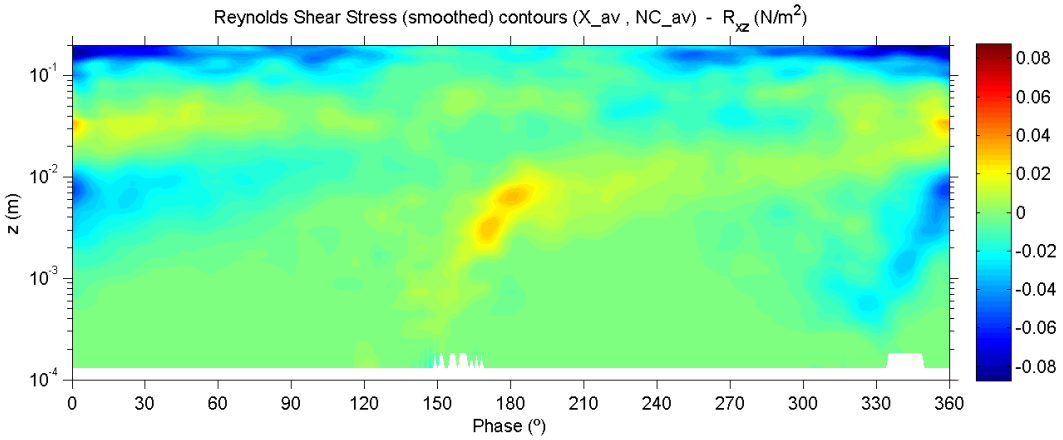


C.3.6.3 Evolution plot

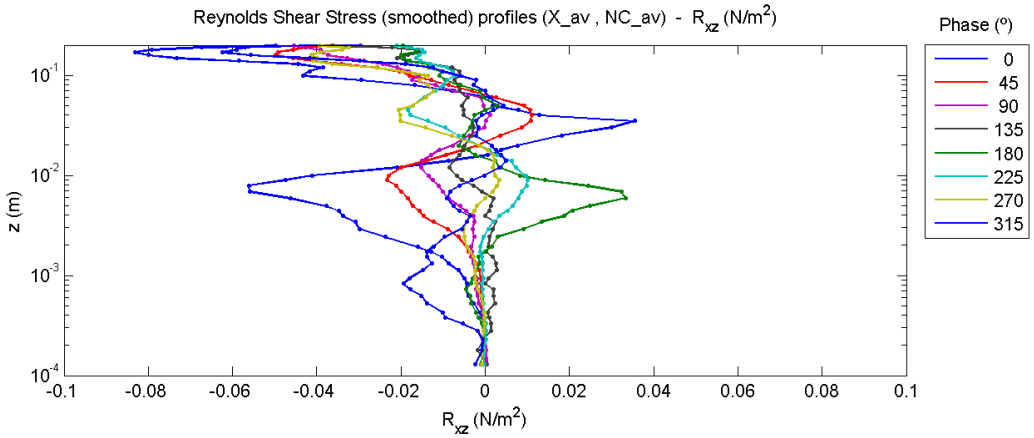


C.3.7 Turbulent shear stress (Reynolds shear stress)

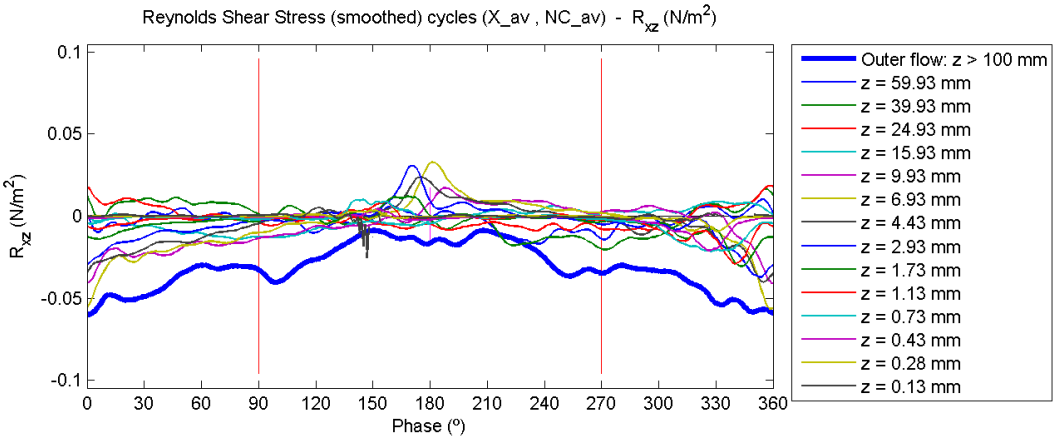
C.3.7.1 Contour plot



C.3.7.2 Profile plot

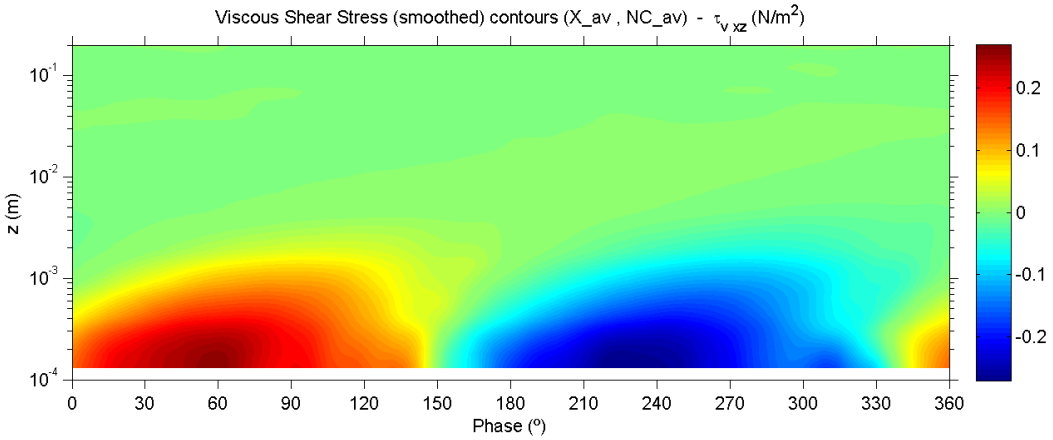


C.3.7.3 Evolution plot

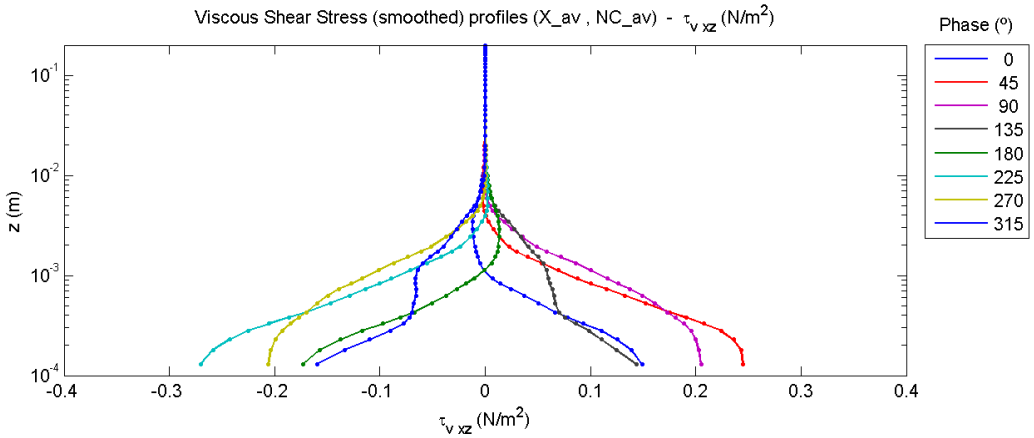


C.3.8 Viscous shear stress

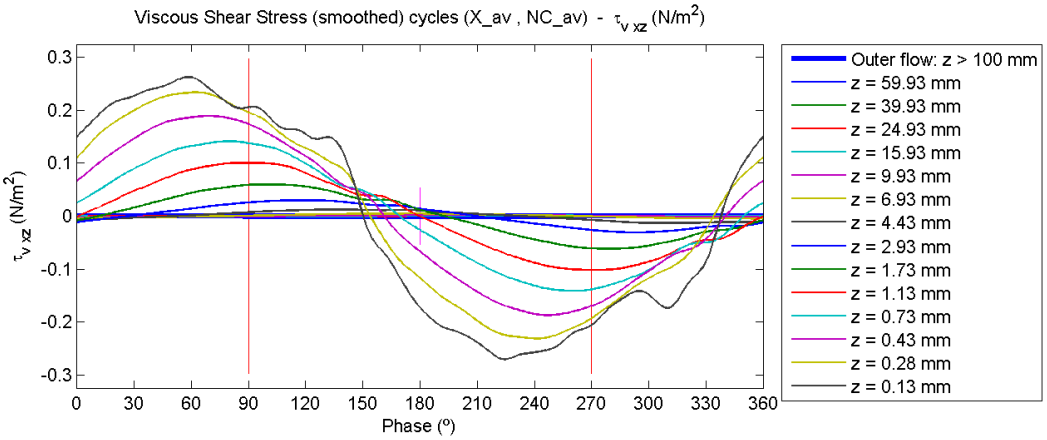
C.3.8.1 Contour plot



C.3.8.2 Profile plot

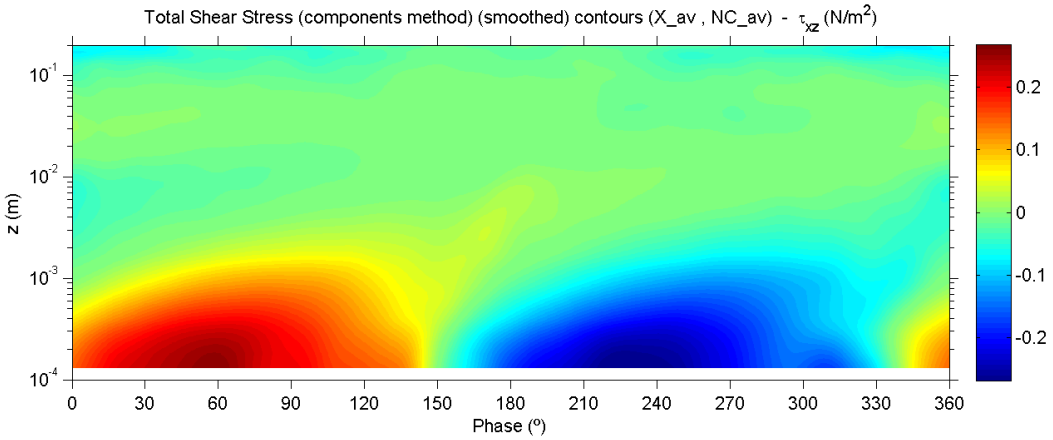


C.3.8.3 Evolution plot

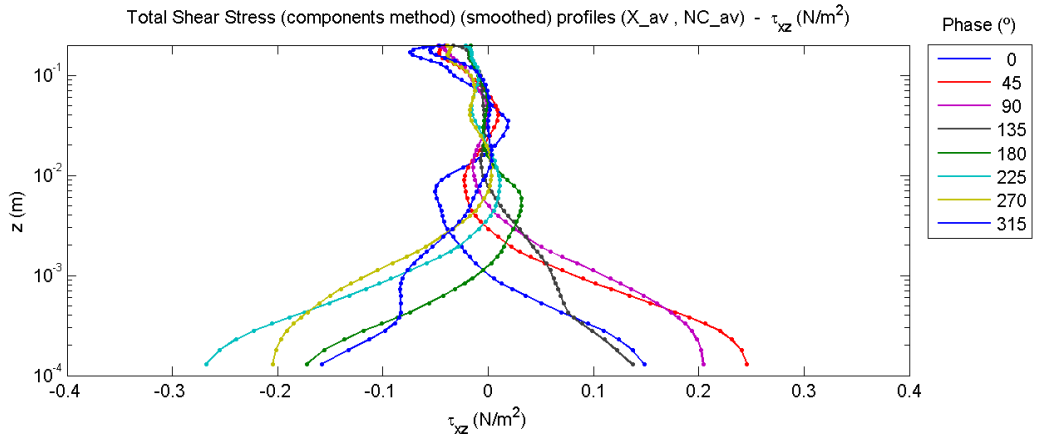


C.3.9 Total shear stress

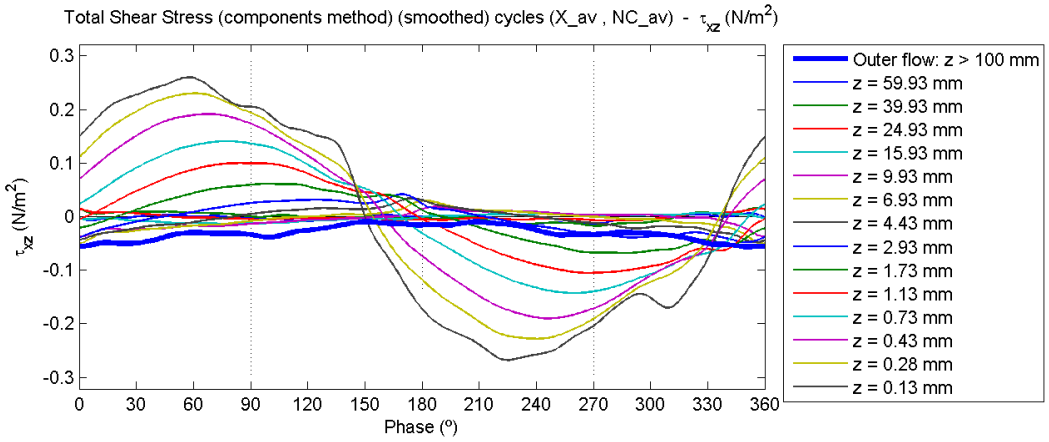
C.3.9.1 Contour plot



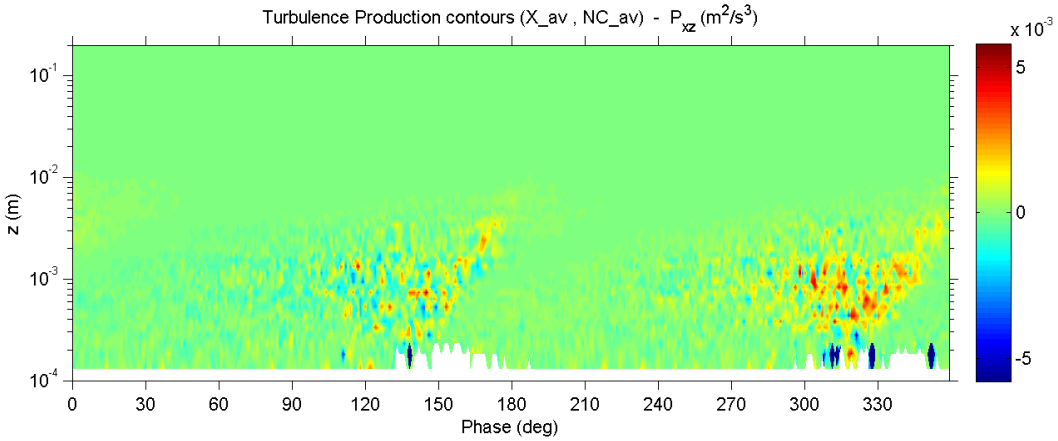
C.3.9.2 Profile plot



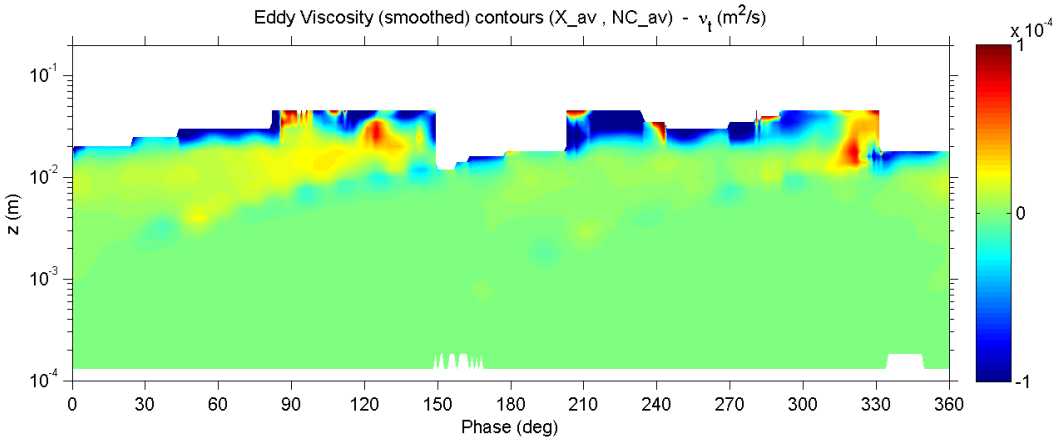
C.3.9.3 Evolution plot



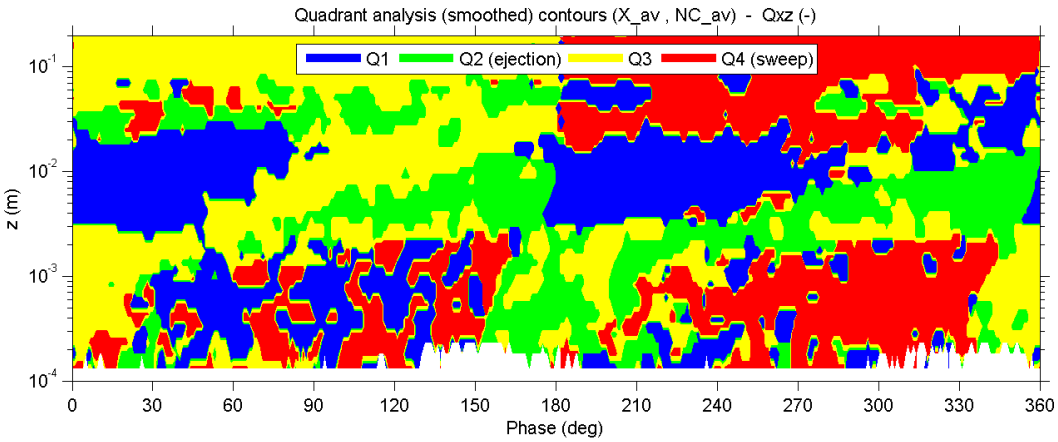
C.3.10 Turbulence production



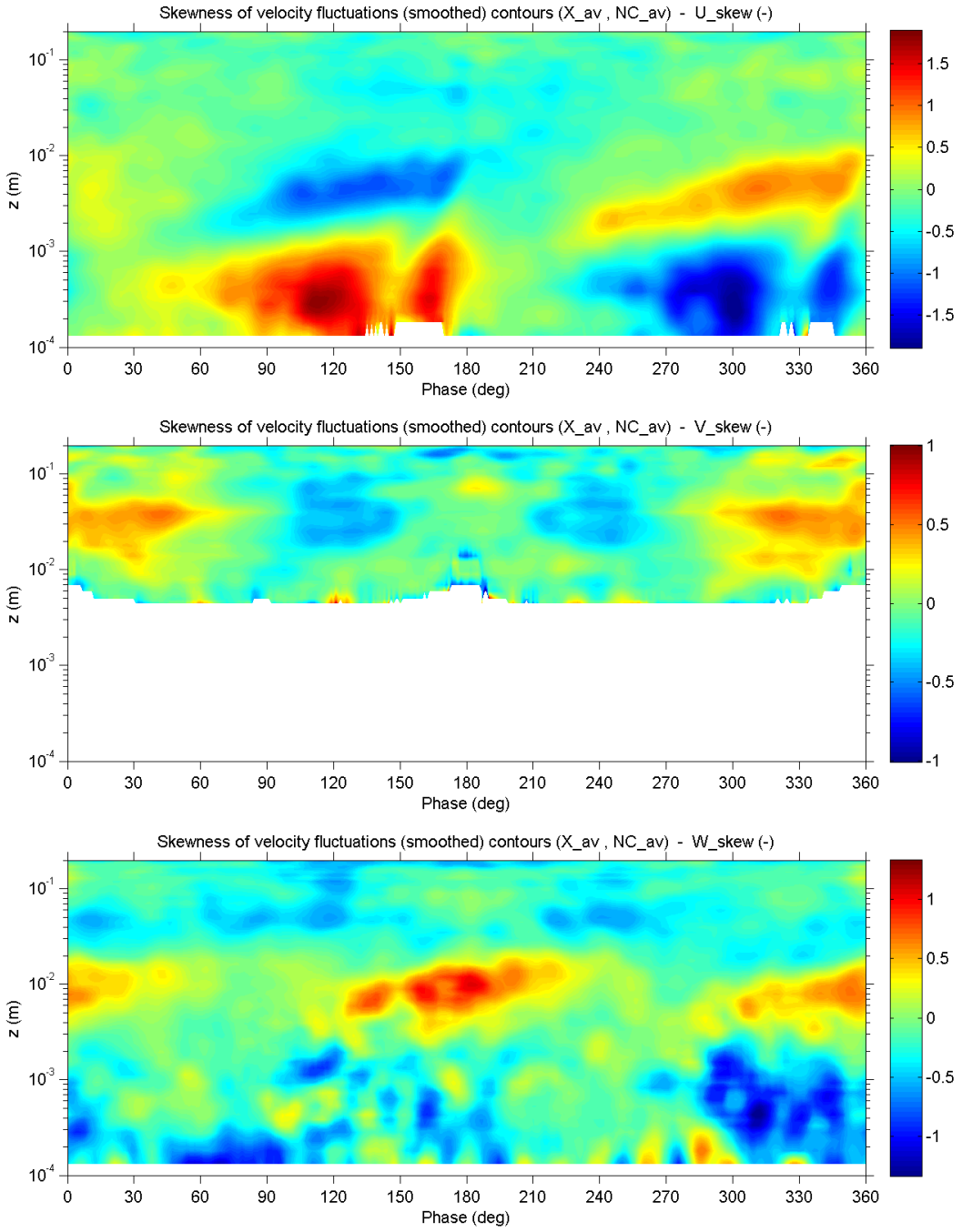
C.3.11 Turbulent viscosity (Eddy viscosity)



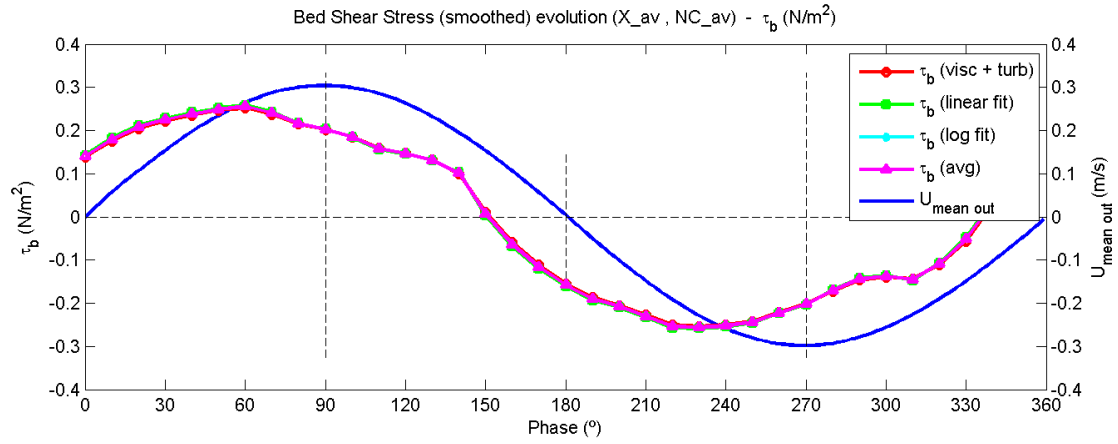
C.3.12 Quadrant analysis



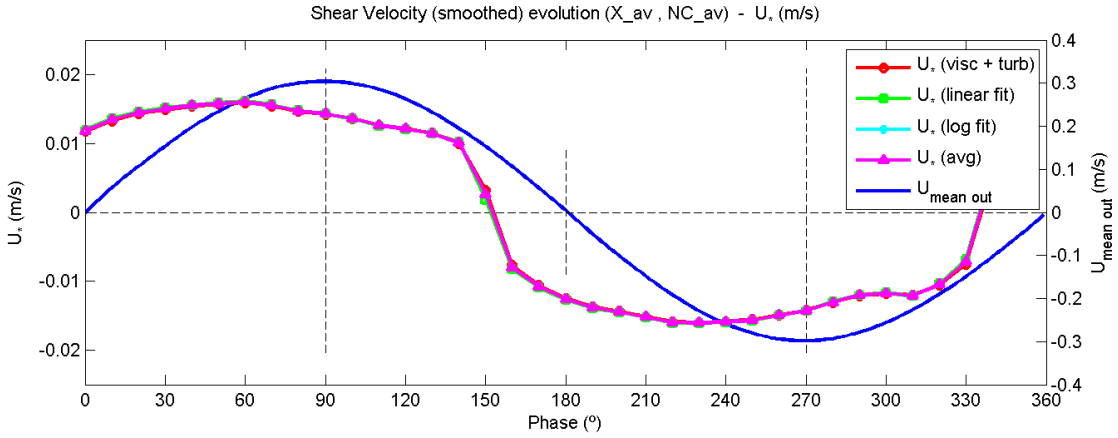
C.3.13 Skewness of velocity fluctuations



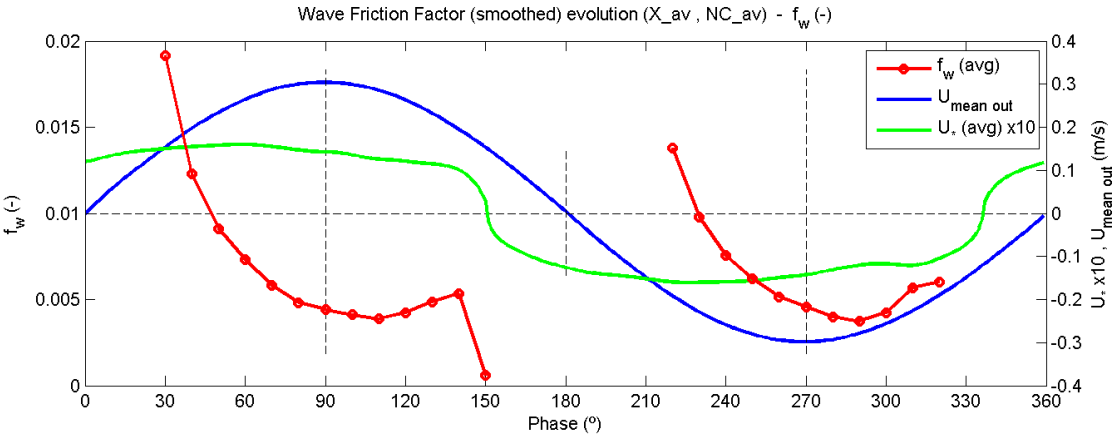
C.3.14 Bed shear stress



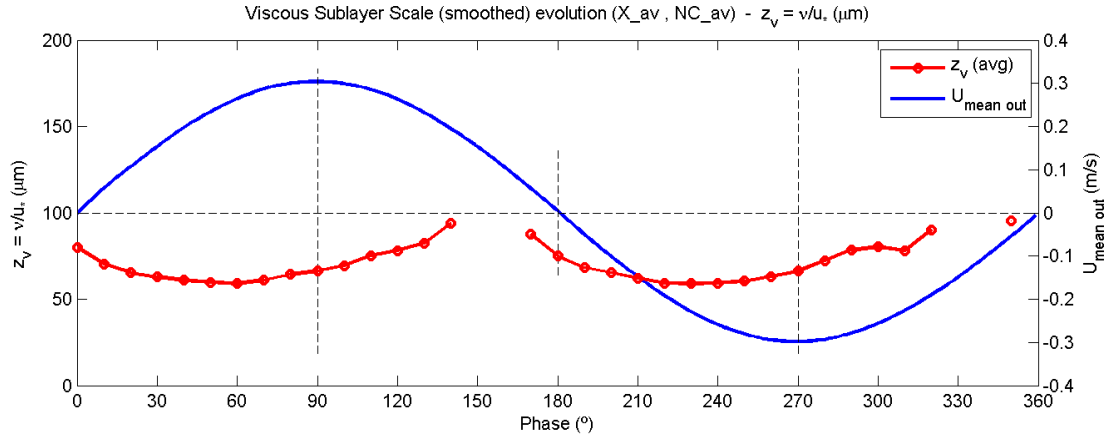
C.3.15 Shear velocity



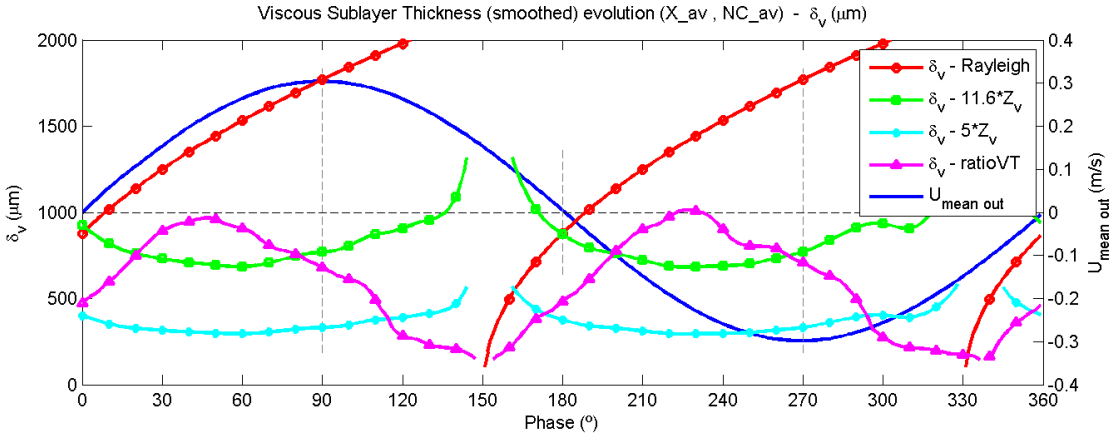
C.3.16 Wave friction factor



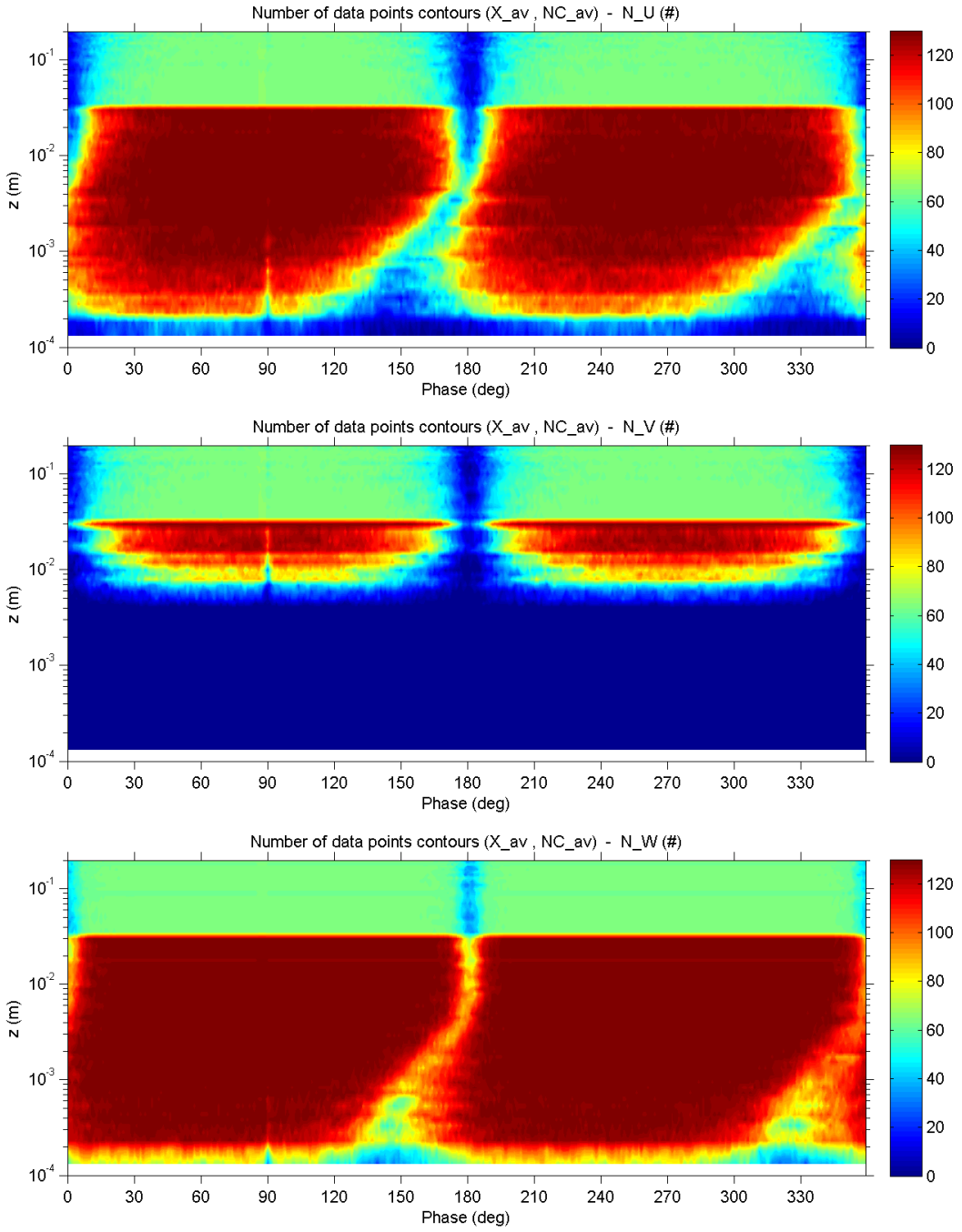
C.3.17 Viscous length scale



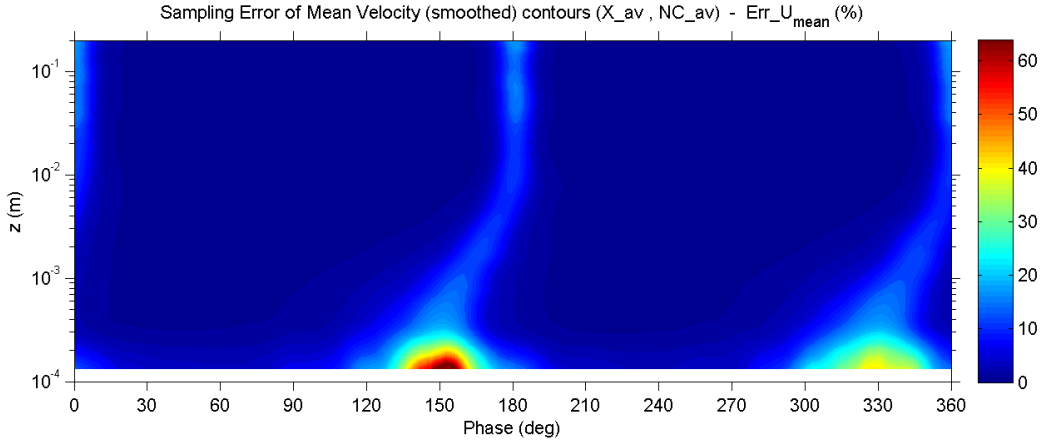
C.3.18 Viscous sublayer thickness



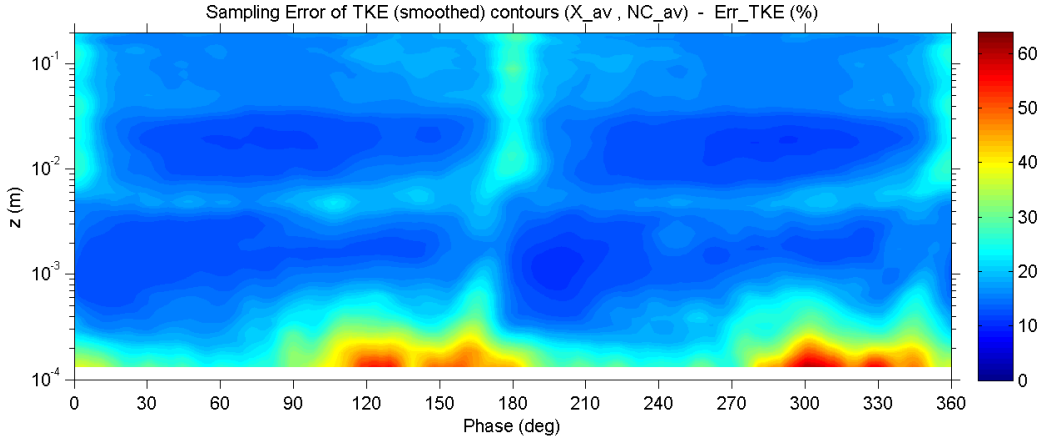
C.3.19 Number of valid data points



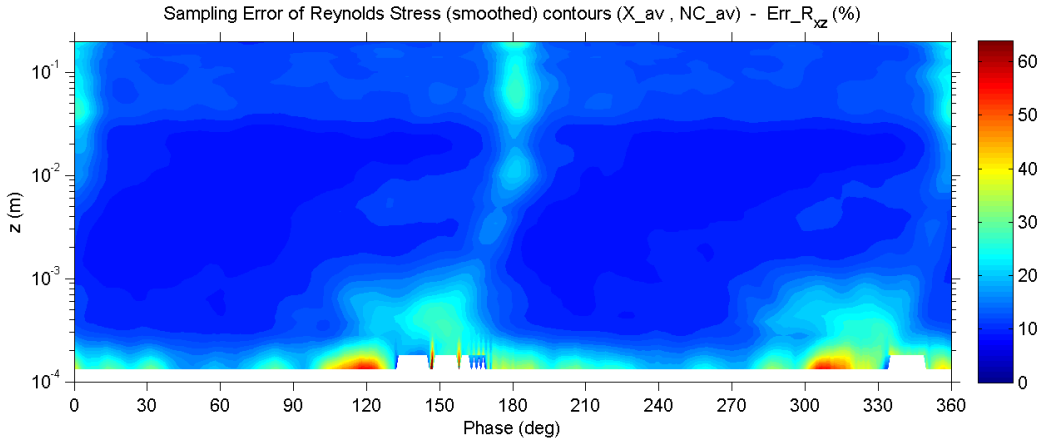
C.3.20 Percent error of mean velocity



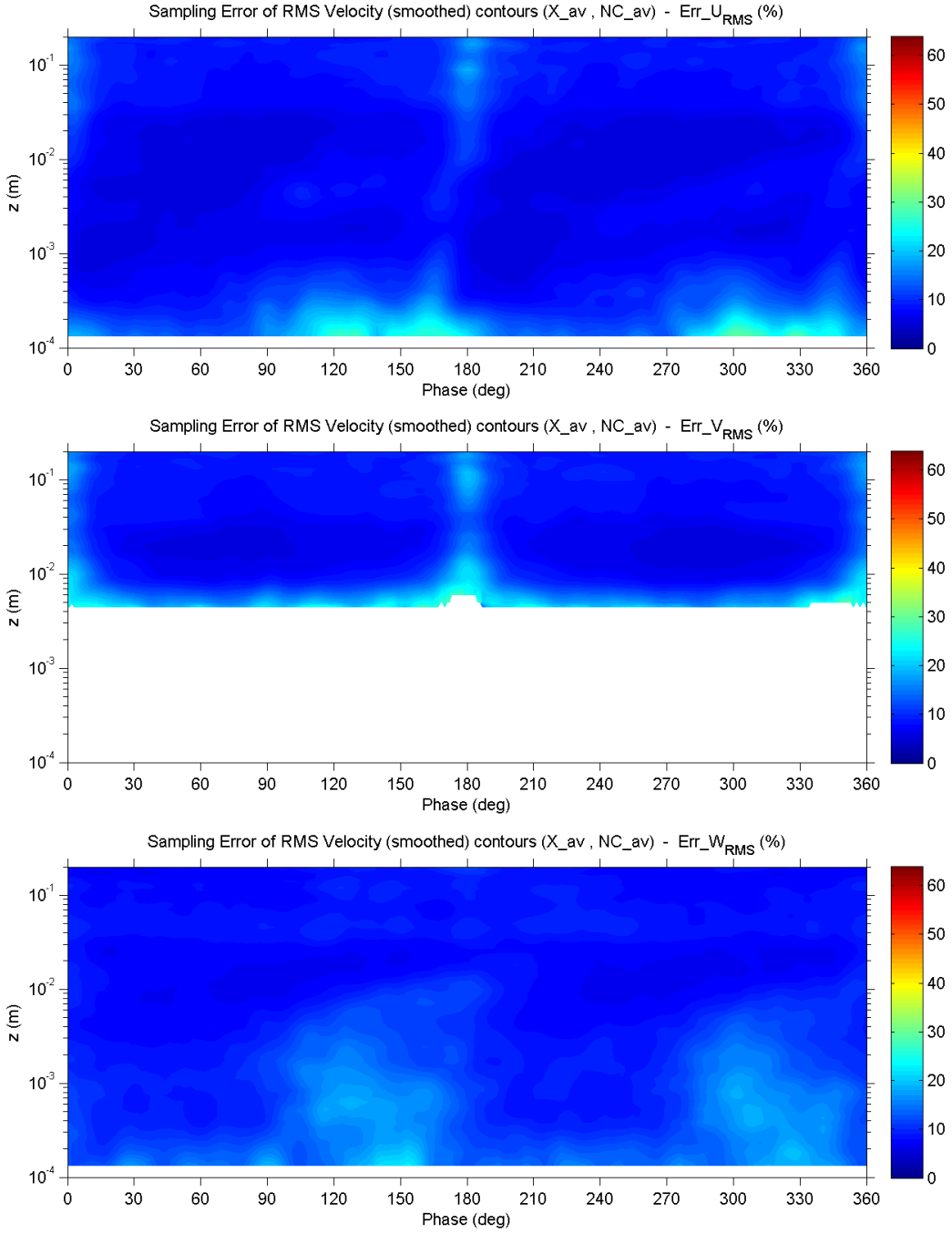
C.3.21 Percent error of turbulent kinetic energy



C.3.22 Percent error of Reynolds shear stress



C.3.23 Percent error of RMS velocity fluctuations



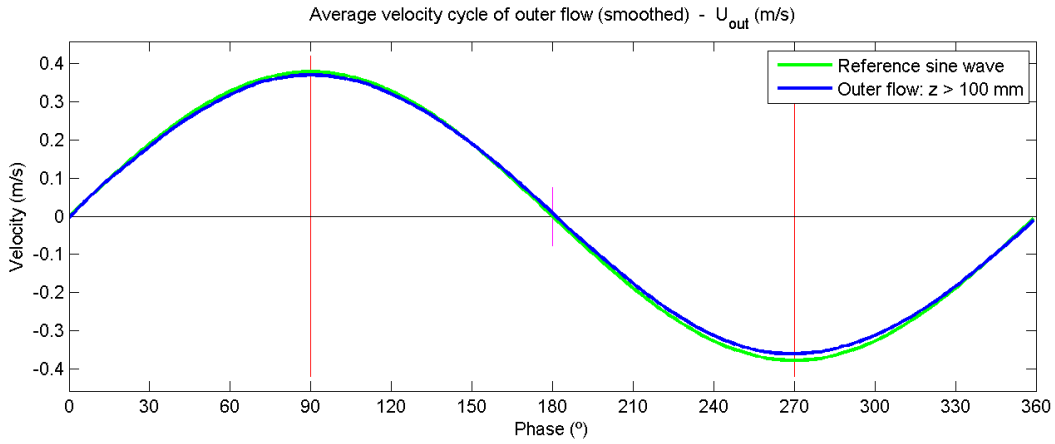
C.4 Experiment no. 4

C.4.1 Main parameters

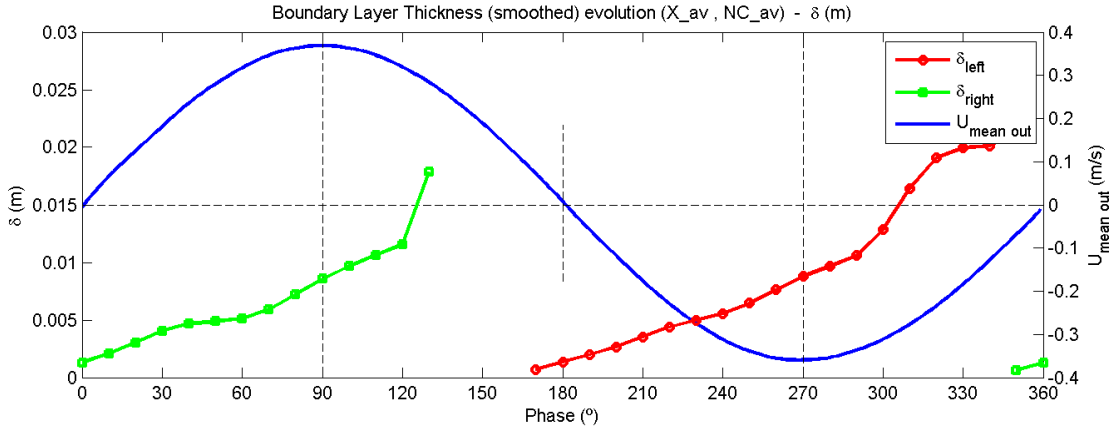
Exp no.	Temp (ºC)	ρ (kg/m^3)	ν (m^2/s)	T (s)	$2a$ (m)	U_{out_max} (m/s)	Re_w (-)	N_{max} (cycles)
4	23.6	997.42	9.36E-07	10	1.159	0.364	2.3E+05	130

Exp no.	U_{*max} (m/s)	τ_{b_max} (N/m^2)	$\Delta\phi$ (deg)	f_{w_ref} (-)	Z_{v_min} (mm)	δ_{v_90} (mm)	δ_{90} (mm)	δ_{top} (mm)
4	0.017	0.28	36.5	0.0043	0.056	0.326	8.6	22

C.4.2 Outer flow velocity

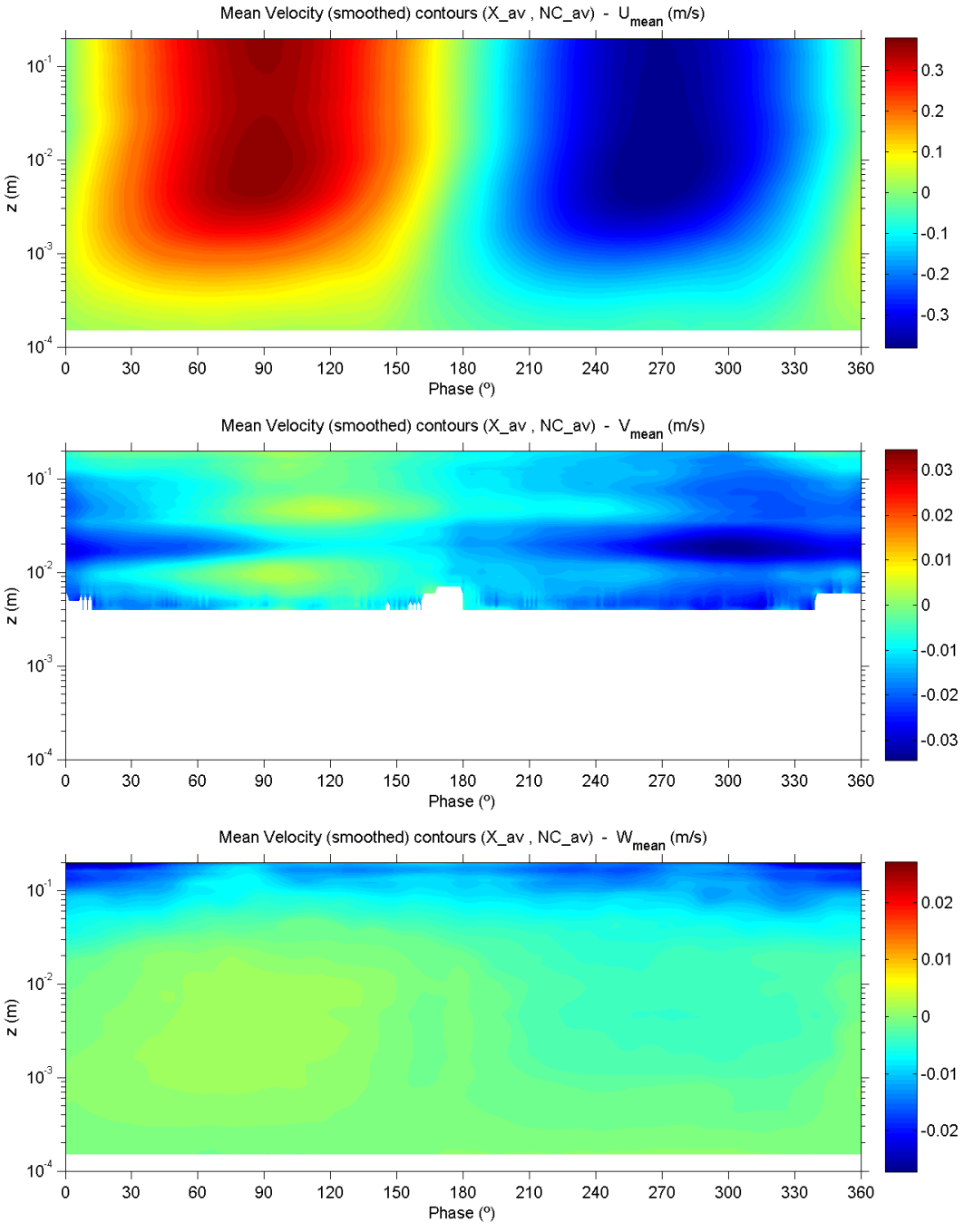


C.4.3 Boundary layer thickness

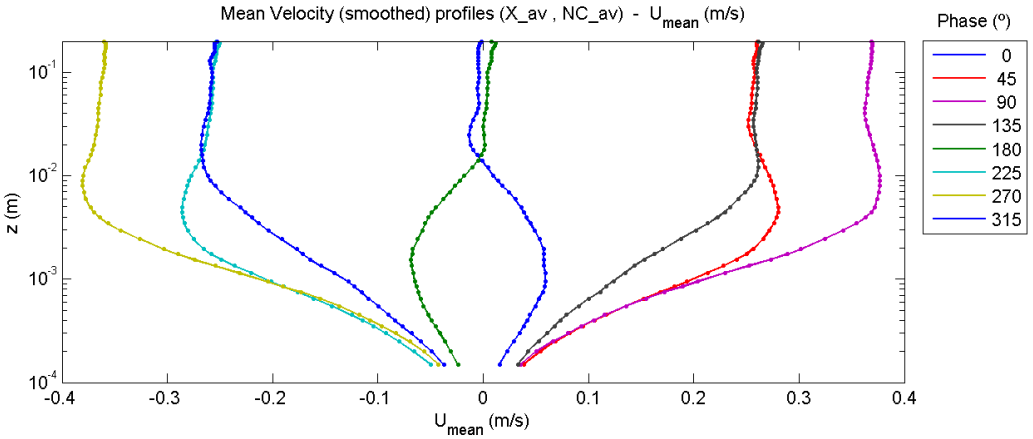


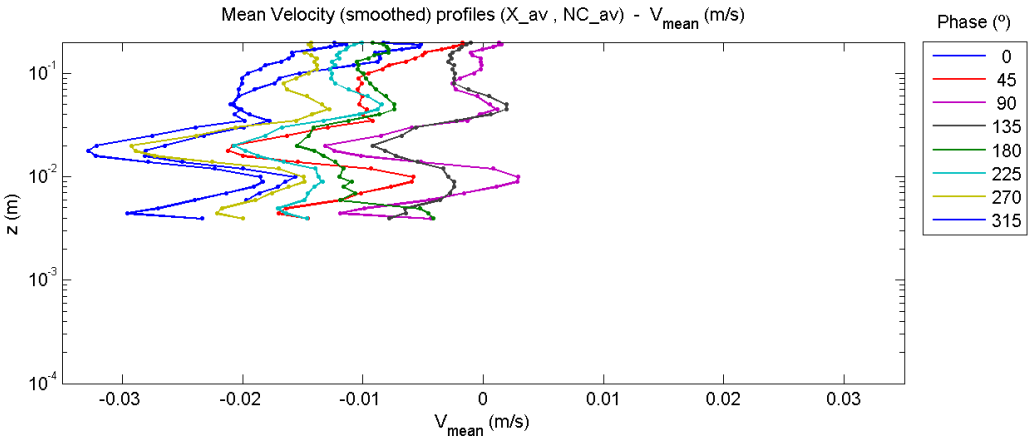
C.4.4 Mean velocities

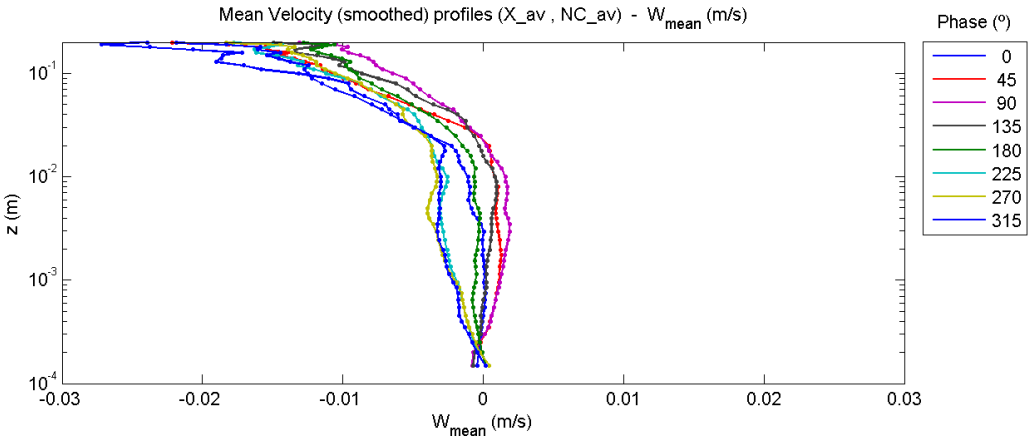
C.4.4.1 Contour plots



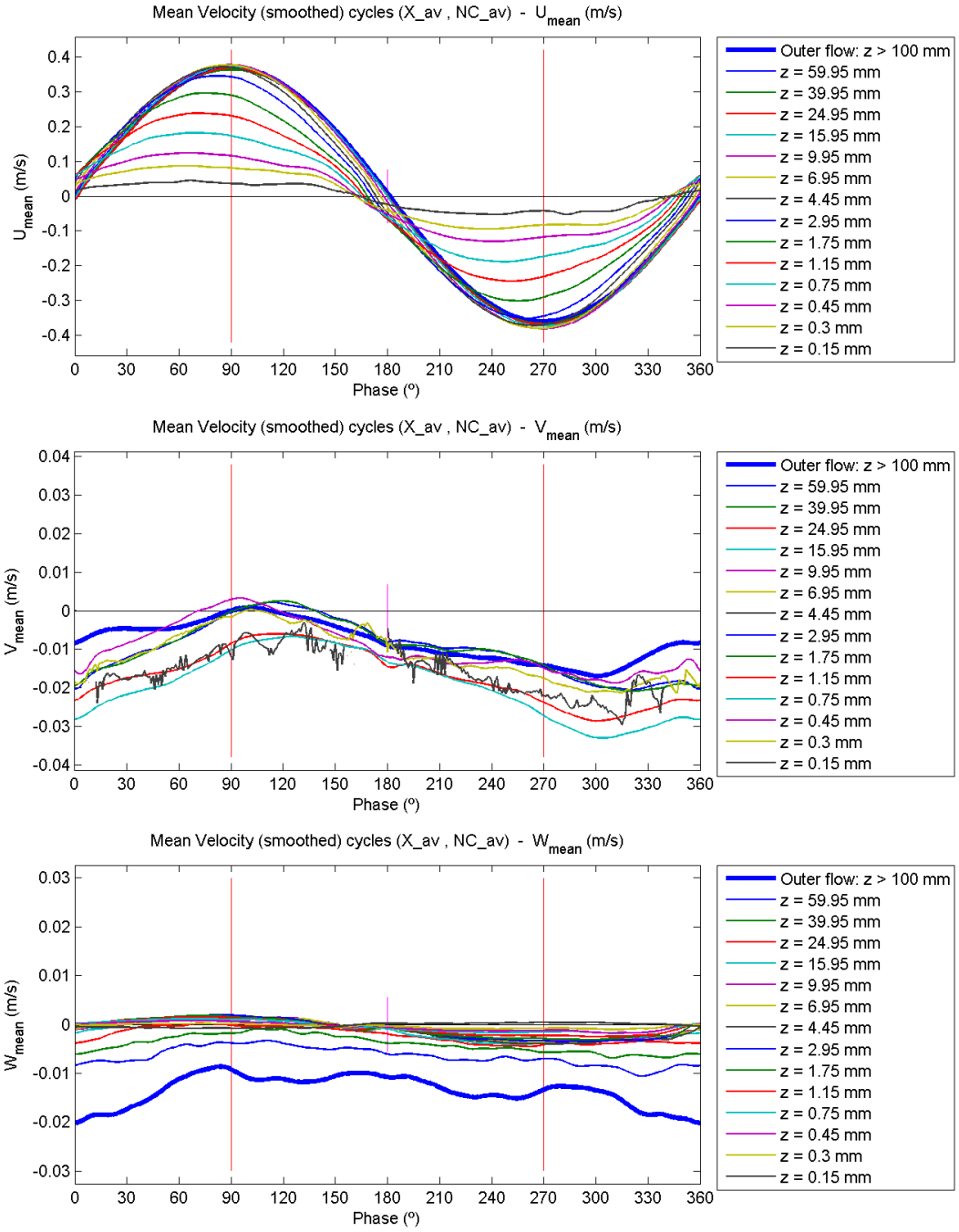
C.4.4.2 Profile plots





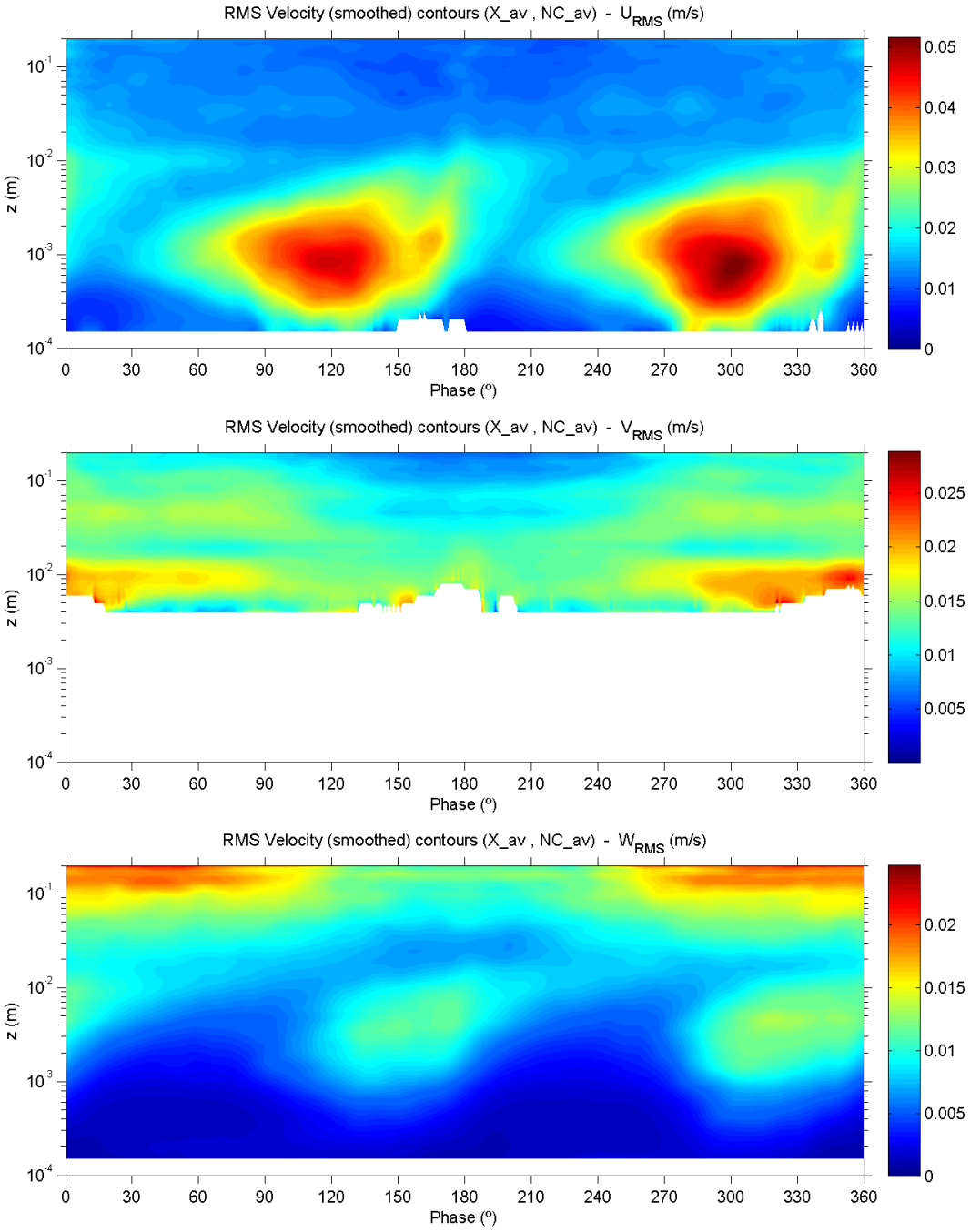


C.4.4.3 Evolution plots

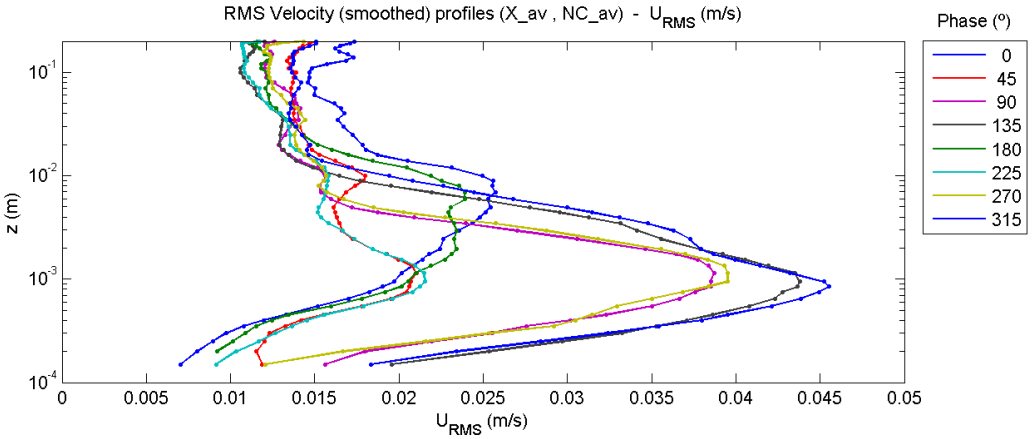


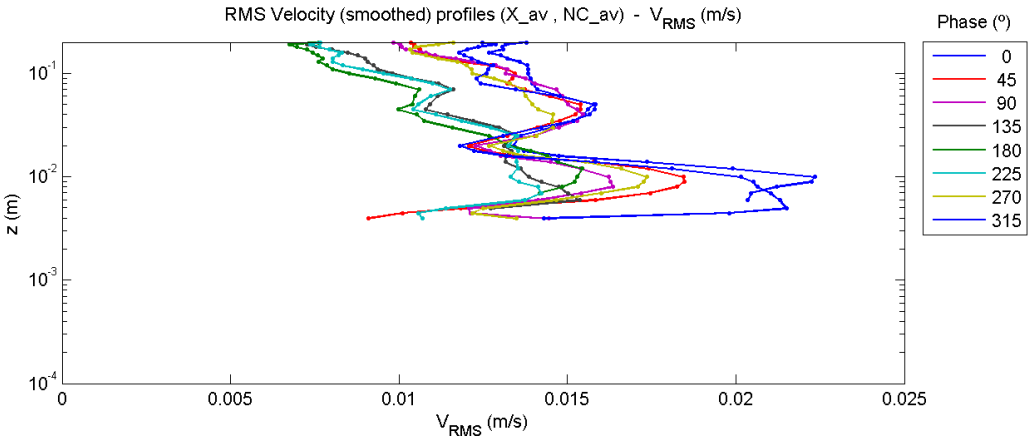
C.4.5 RMS velocity fluctuations

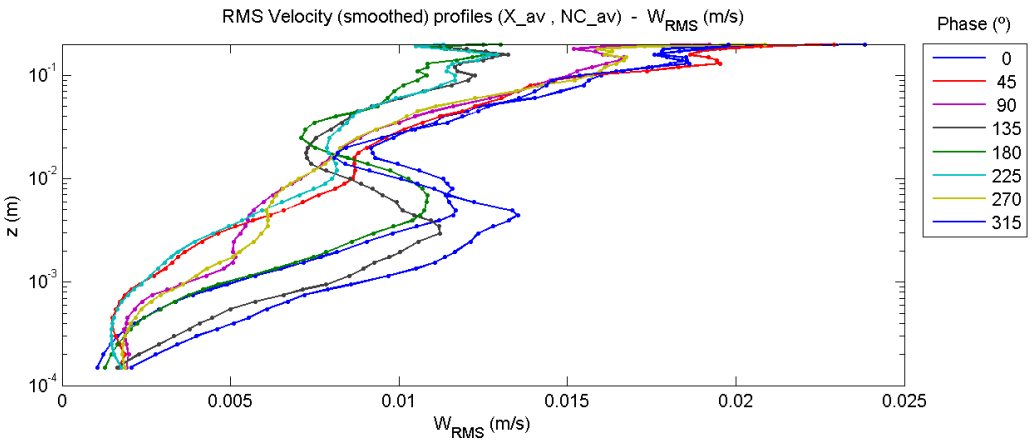
C.4.5.1 Contour plots



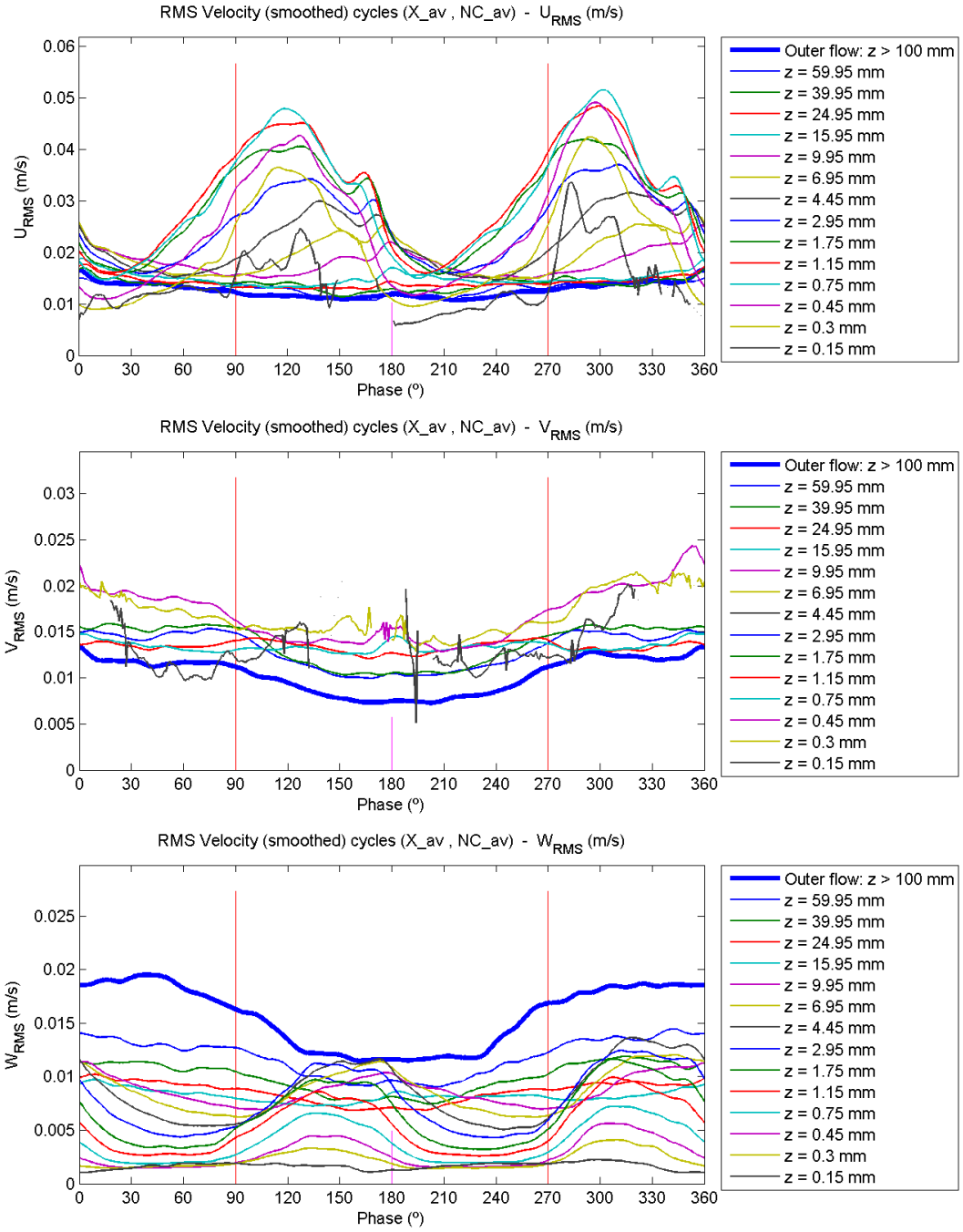
C.4.5.2 Profile plots





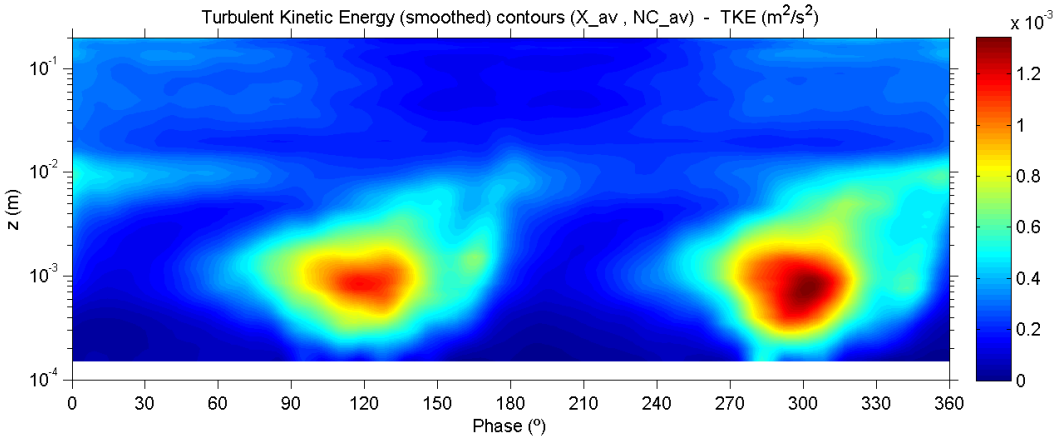


C.4.5.3 Evolution plots

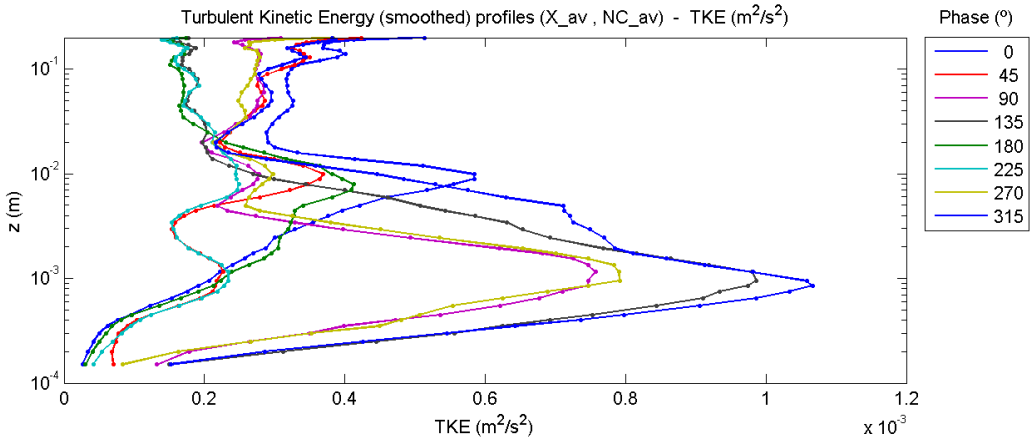


C.4.6 Turbulent kinetic energy

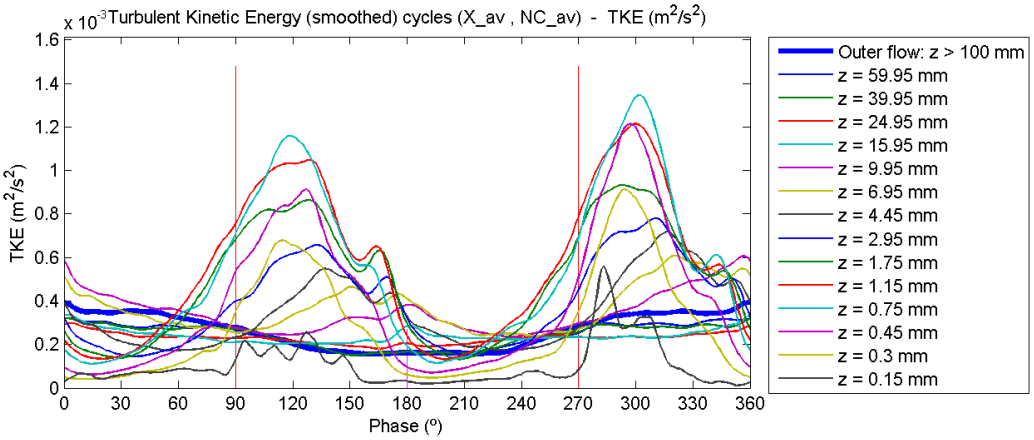
C.4.6.1 Contour plot



C.4.6.2 Profile plot

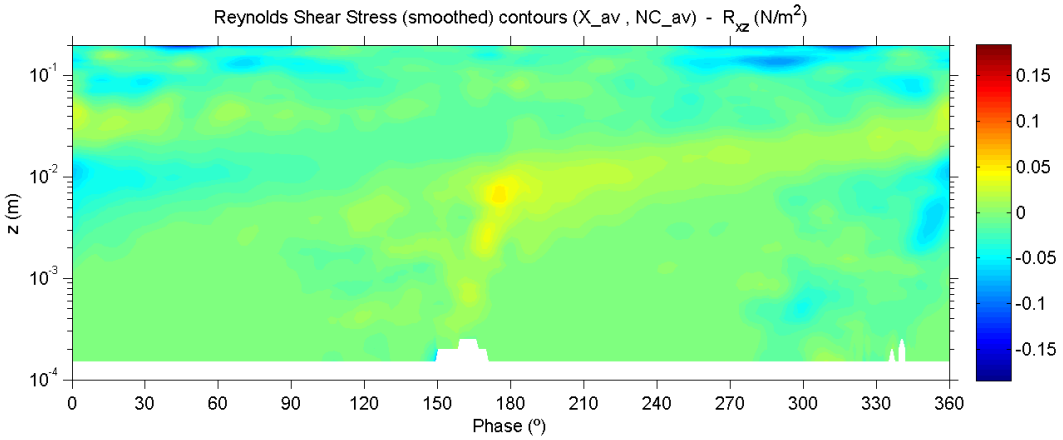


C.4.6.3 Evolution plot

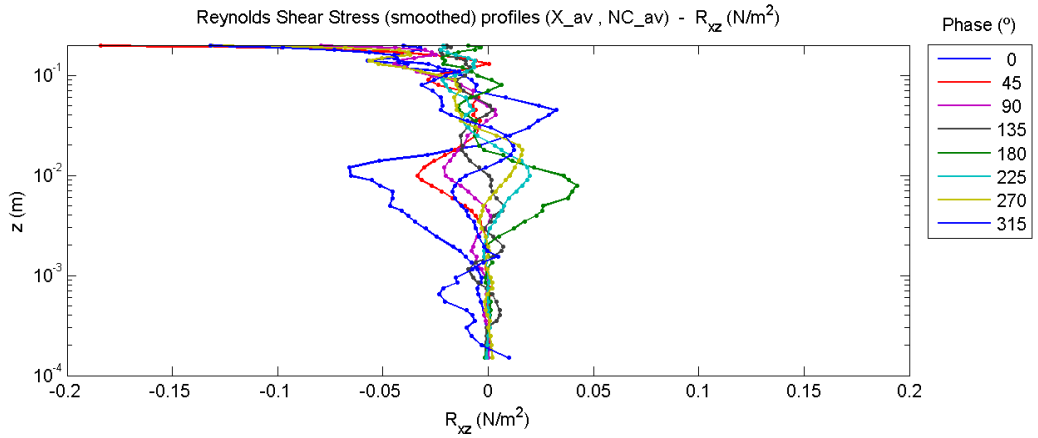


C.4.7 Turbulent shear stress (Reynolds shear stress)

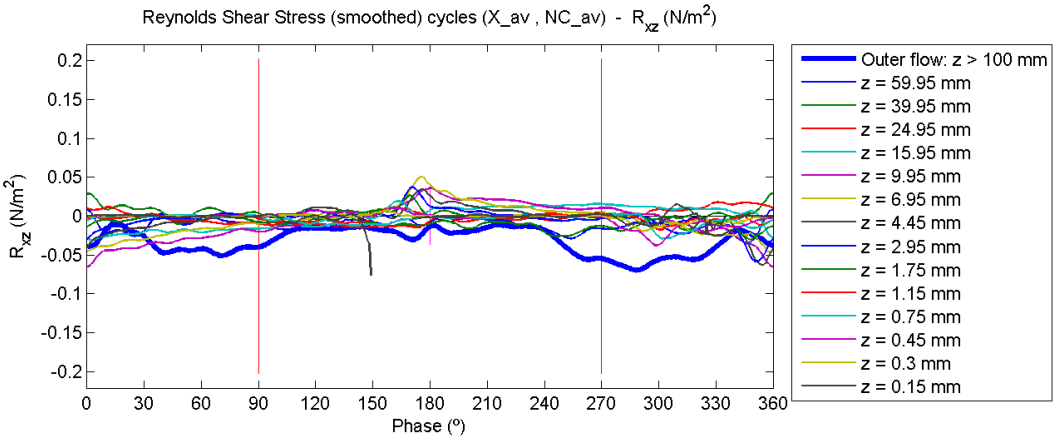
C.4.7.1 Contour plot



C.4.7.2 Profile plot

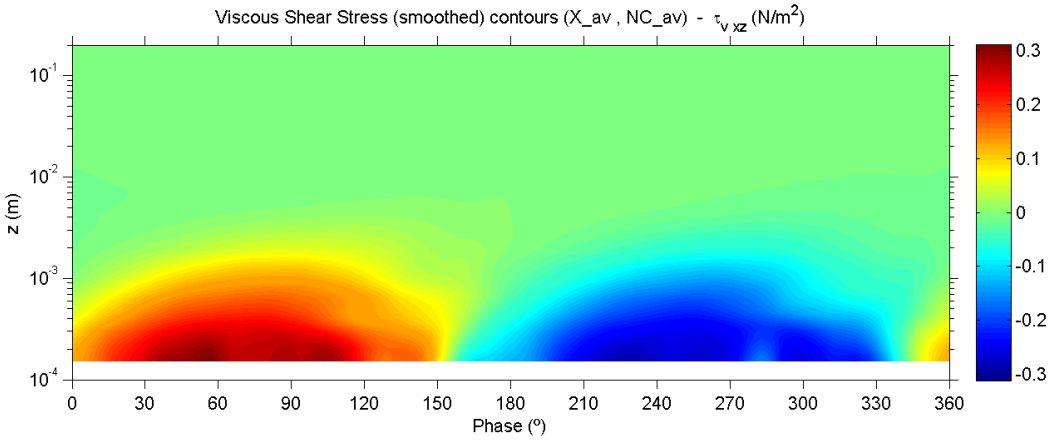


C.4.7.3 Evolution plot

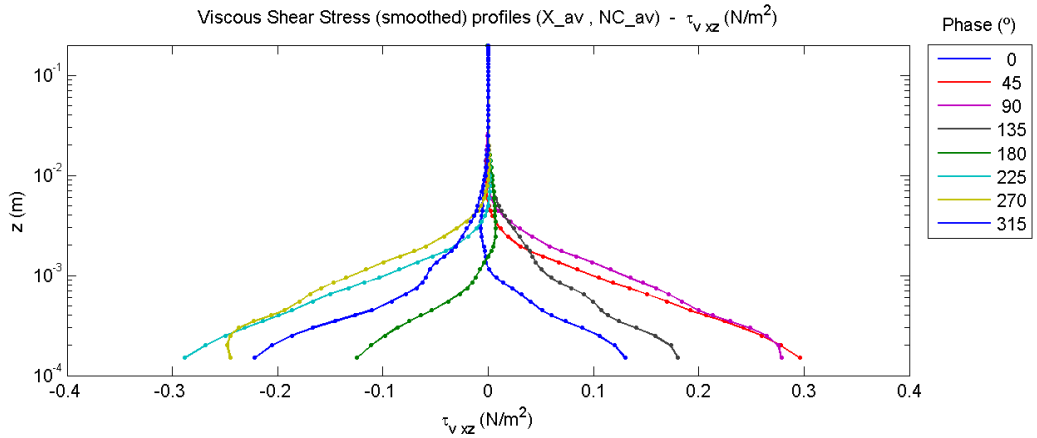


C.4.8 Viscous shear stress

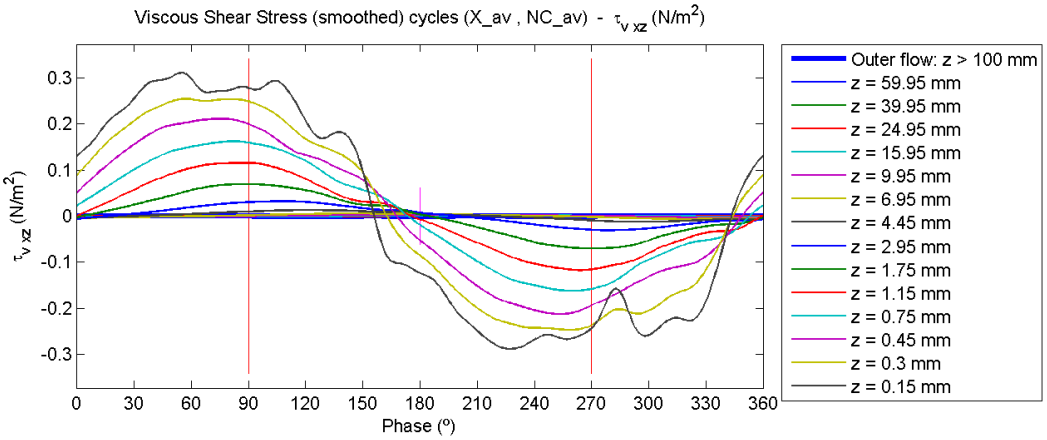
C.4.8.1 Contour plot



C.4.8.2 Profile plot

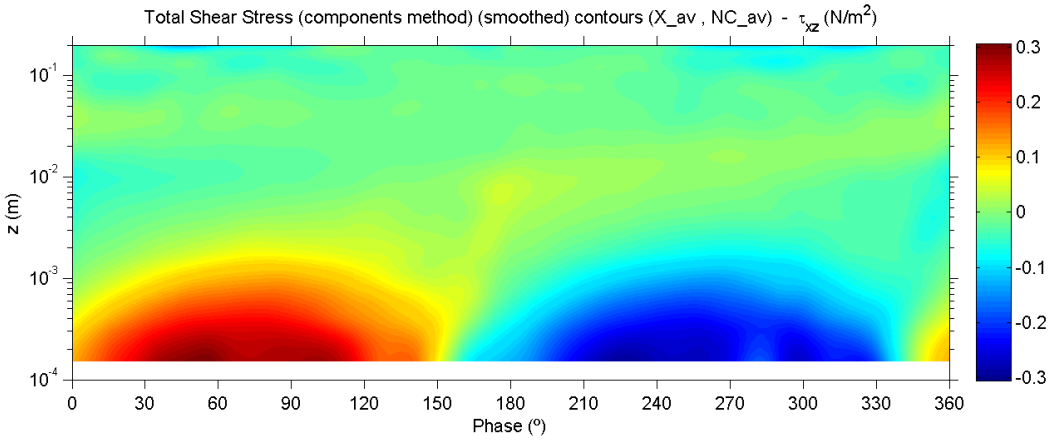


C.4.8.3 Evolution plot

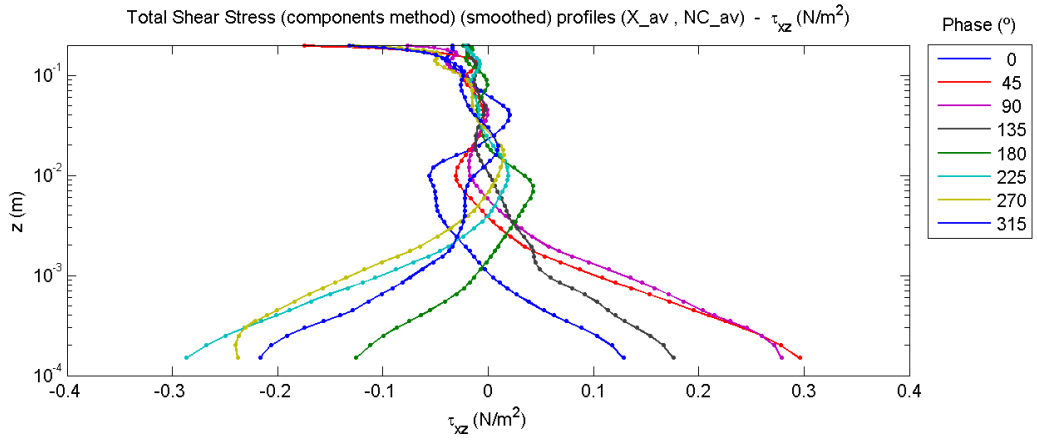


C.4.9 Total shear stress

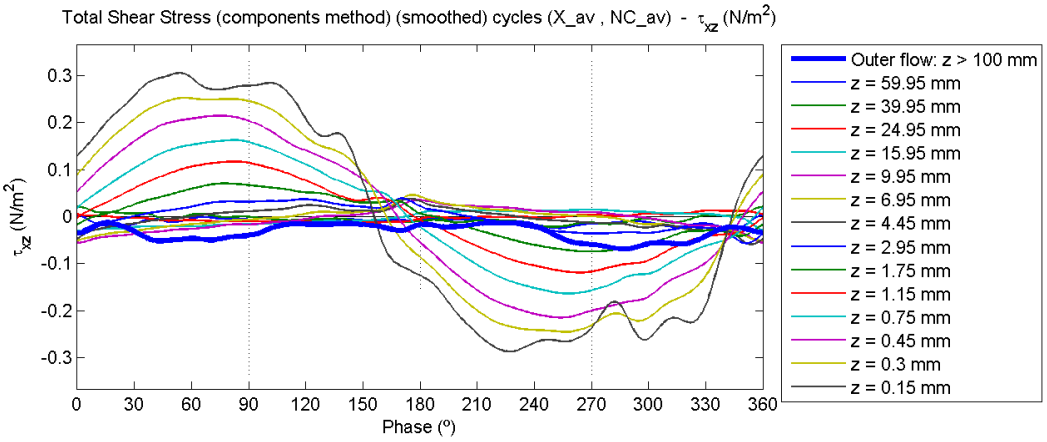
C.4.9.1 Contour plot



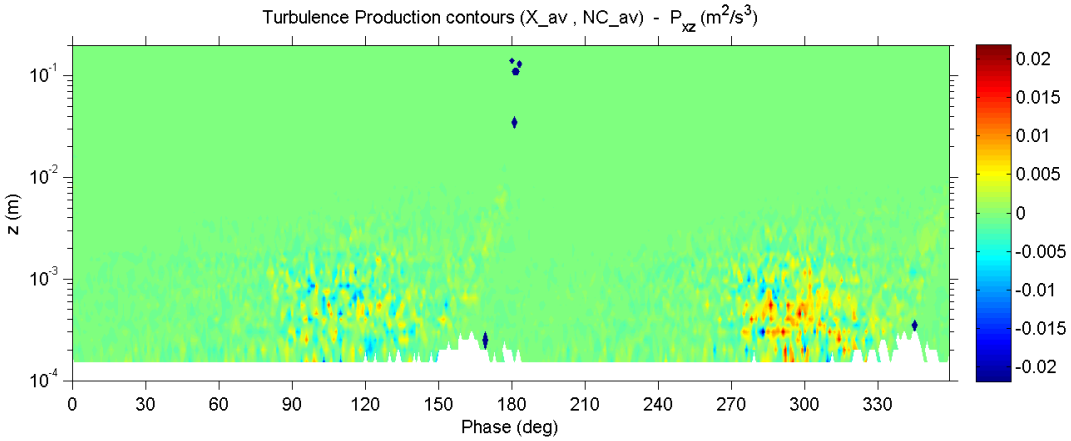
C.4.9.2 Profile plot



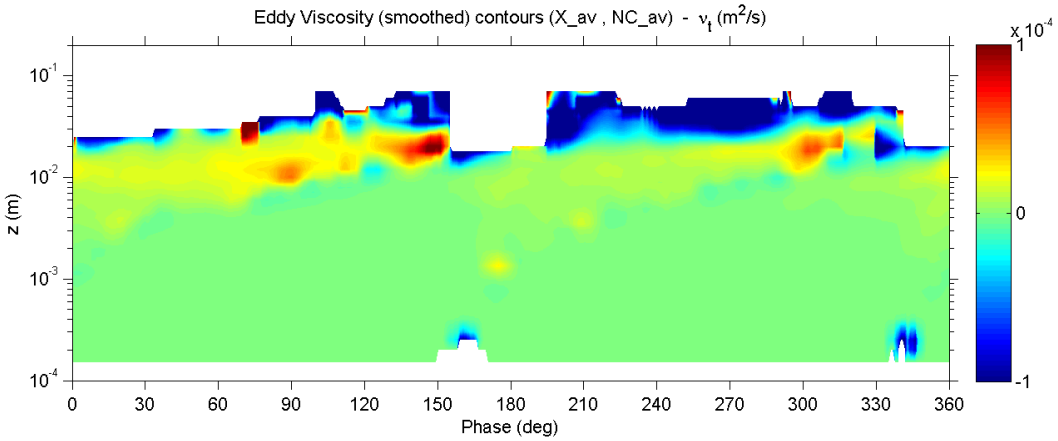
C.4.9.3 Evolution plot



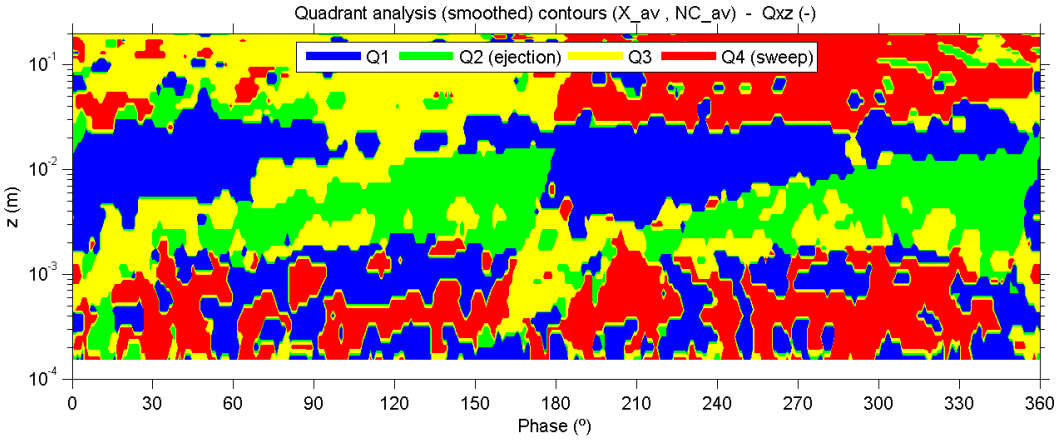
C.4.10 Turbulence production



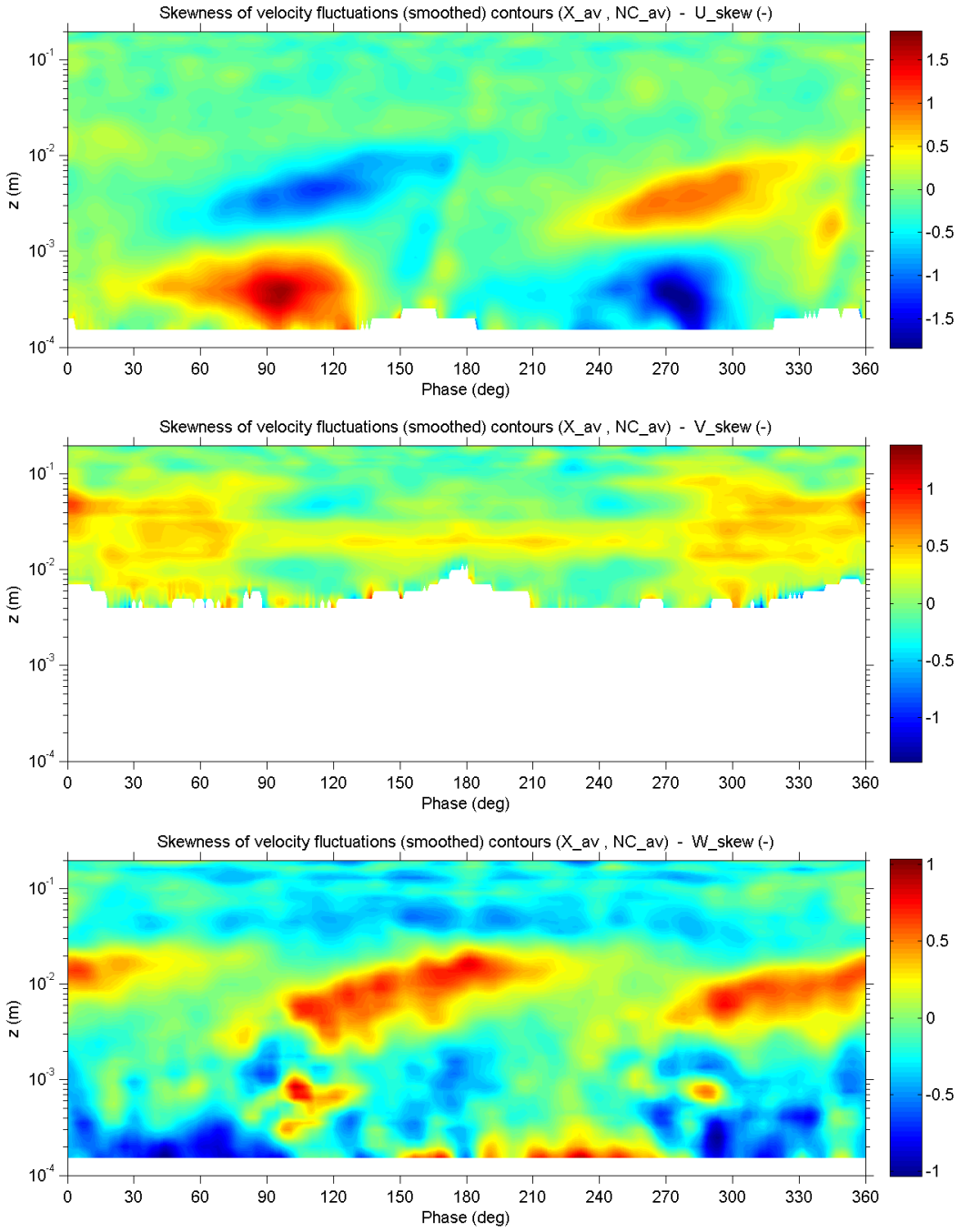
C.4.11 Turbulent viscosity (Eddy viscosity)



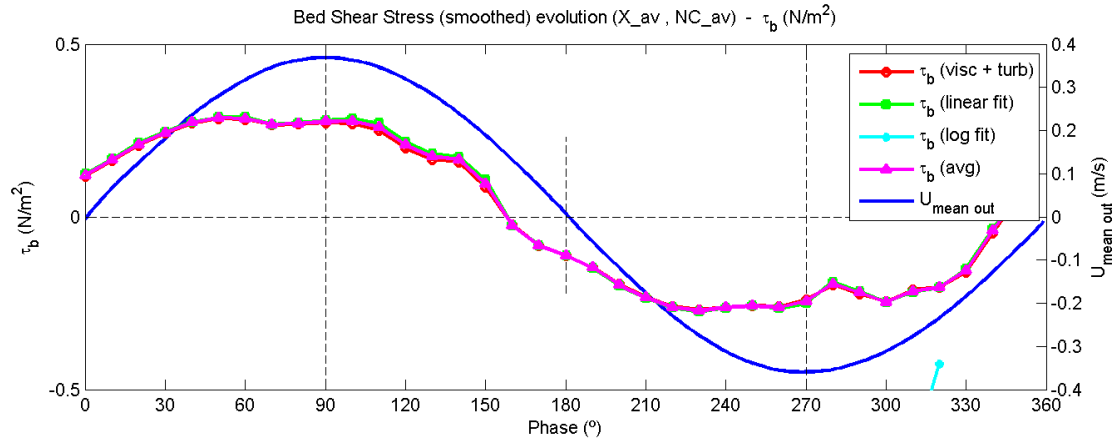
C.4.12 Quadrant analysis



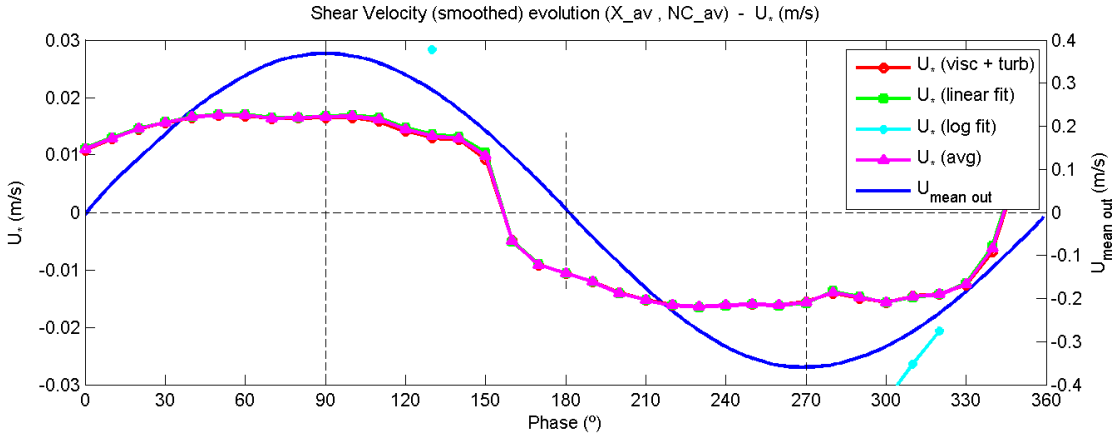
C.4.13 Skewness of velocity fluctuations



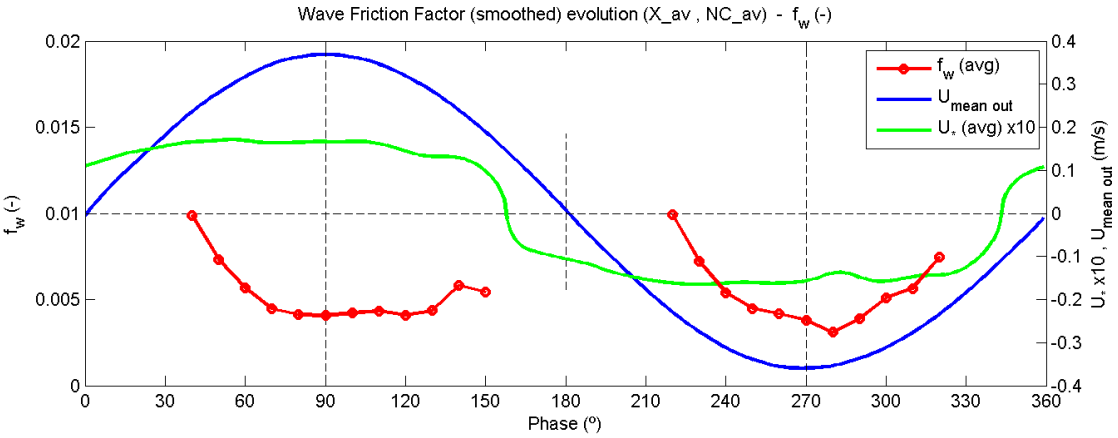
C.4.14 Bed shear stress



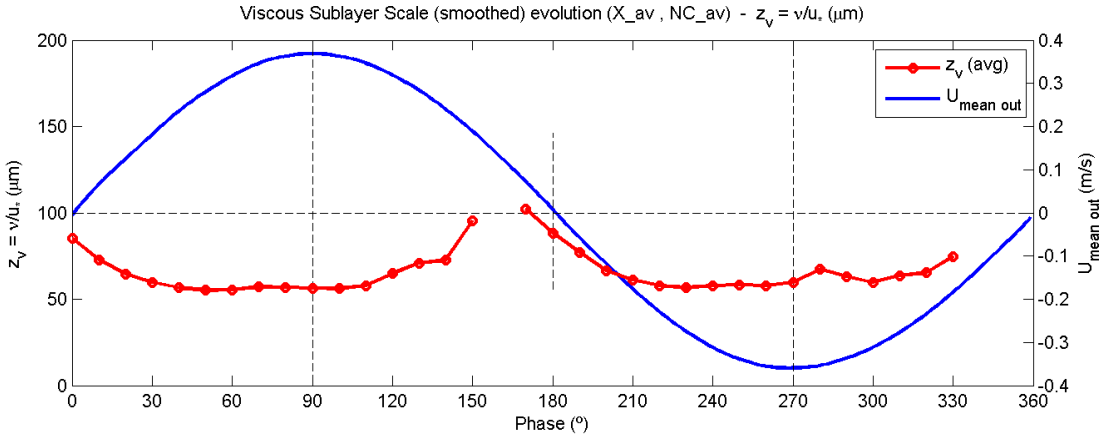
C.4.15 Shear velocity



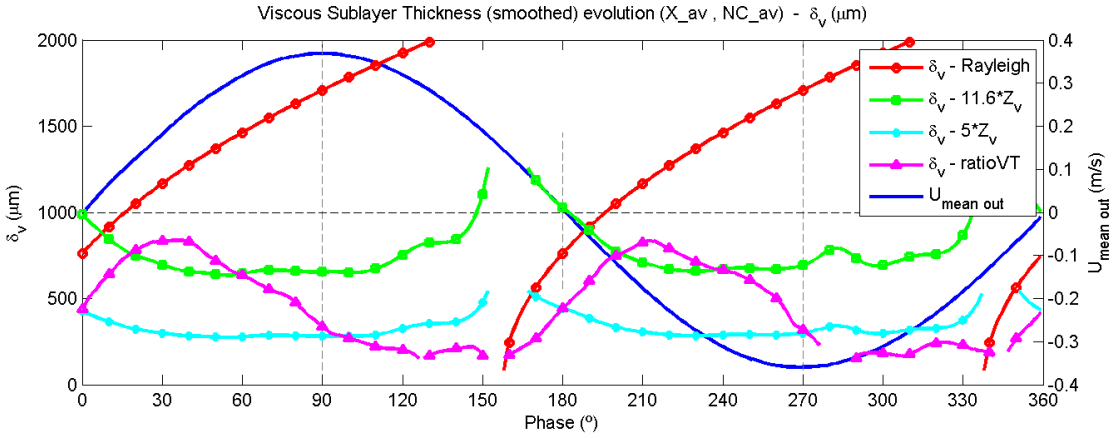
C.4.16 Wave friction factor



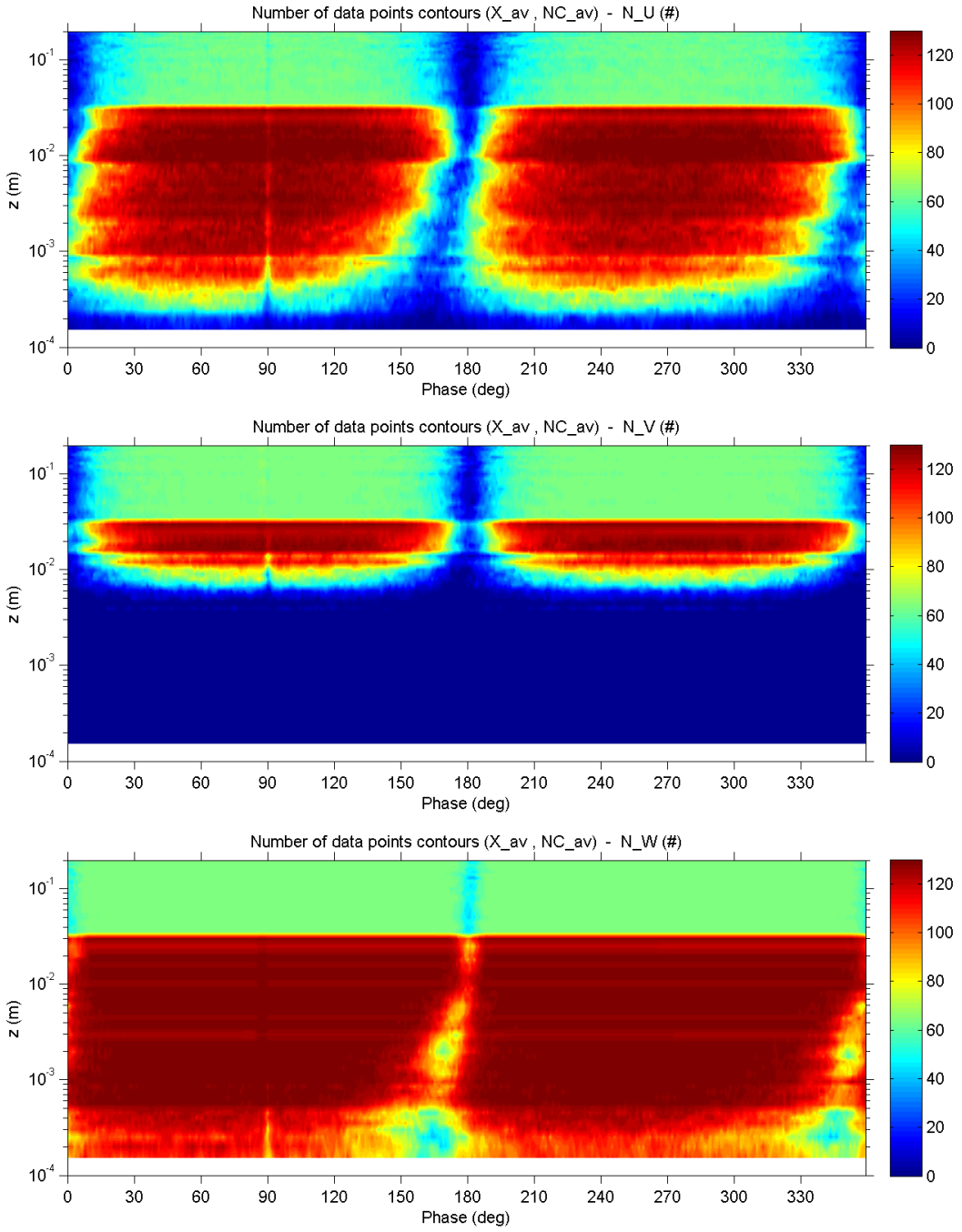
C.4.17 Viscous length scale



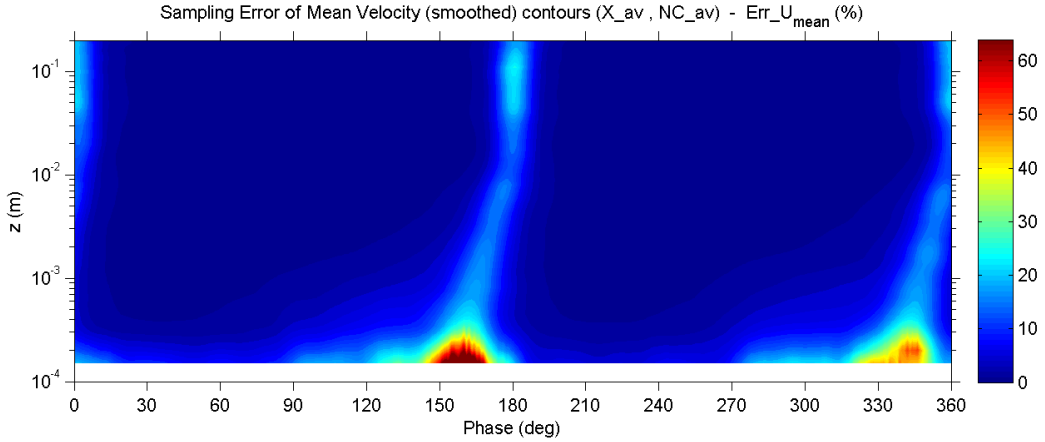
C.4.18 Viscous sublayer thickness



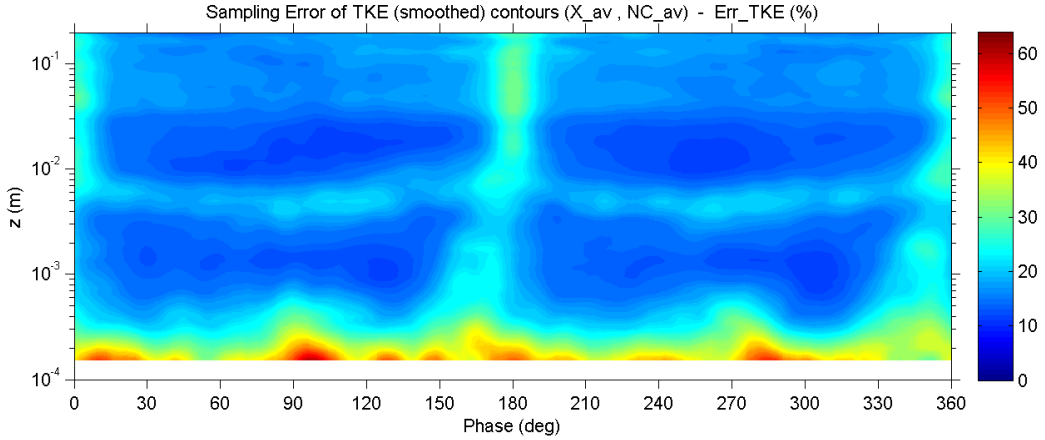
C.4.19 Number of valid data points



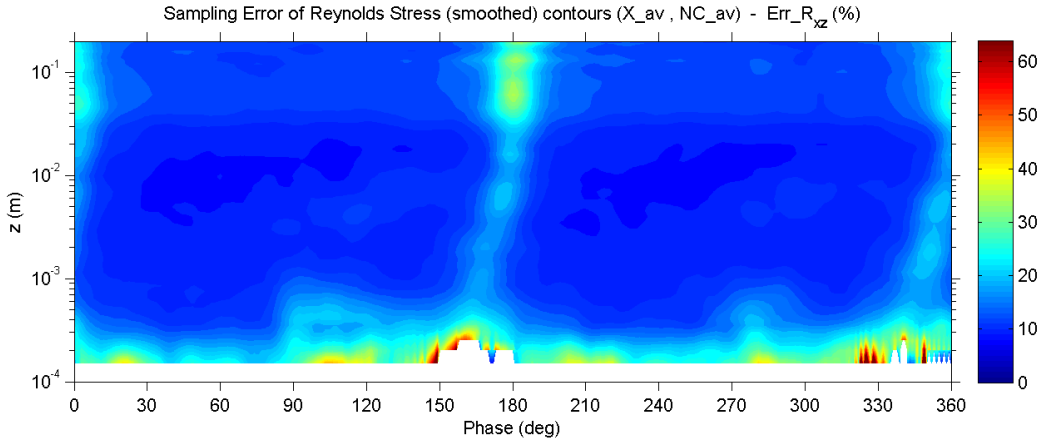
C.4.20 Percent error of mean velocity



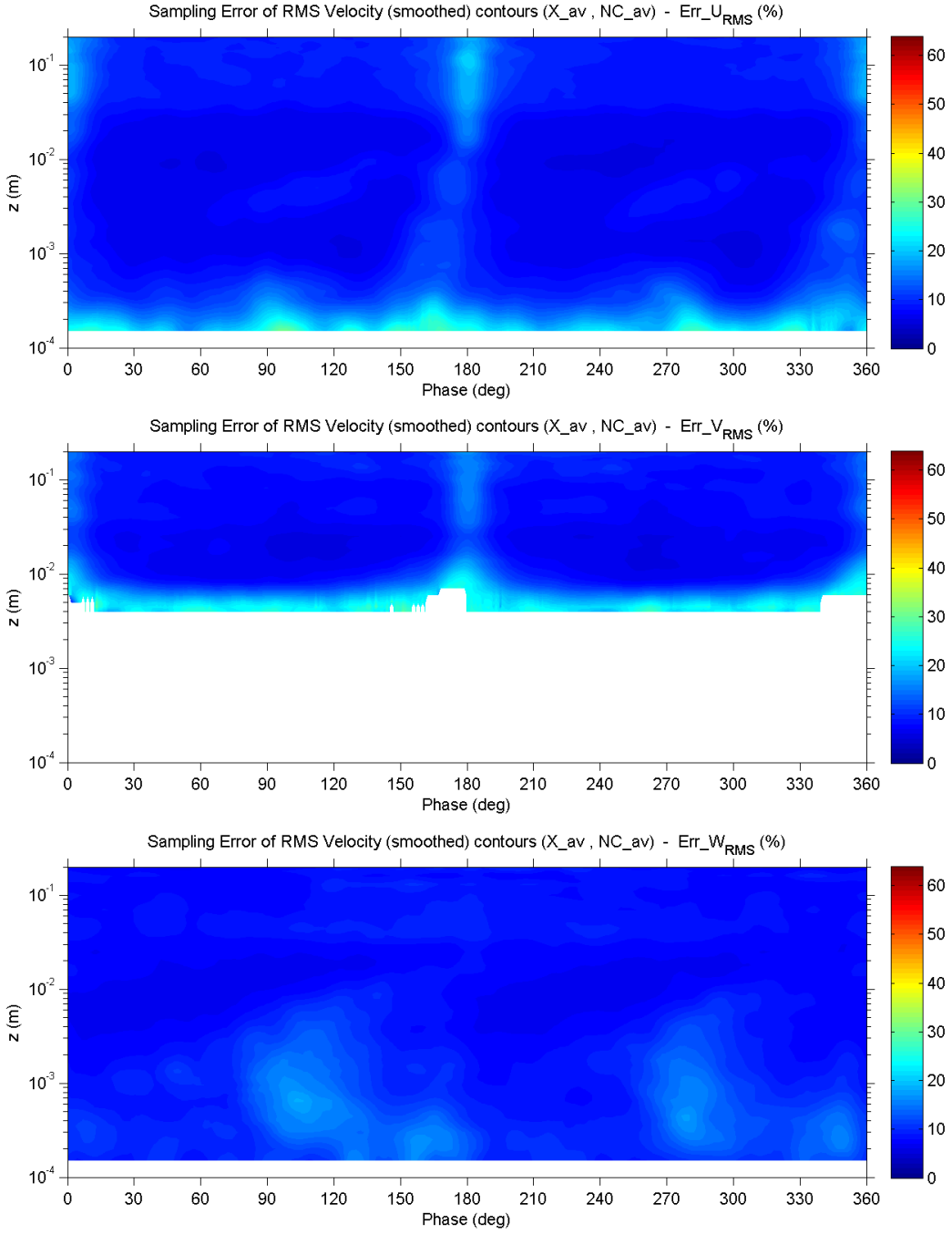
C.4.21 Percent error of turbulent kinetic energy



C.4.22 Percent error of Reynolds shear stress



C.4.23 Percent error of RMS velocity fluctuations



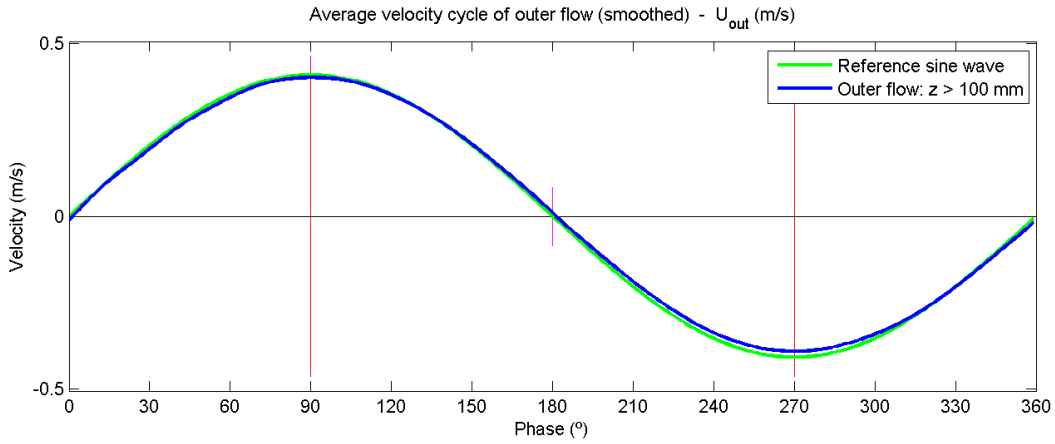
C.5 Experiment no. 5

C.5.1 Main parameters

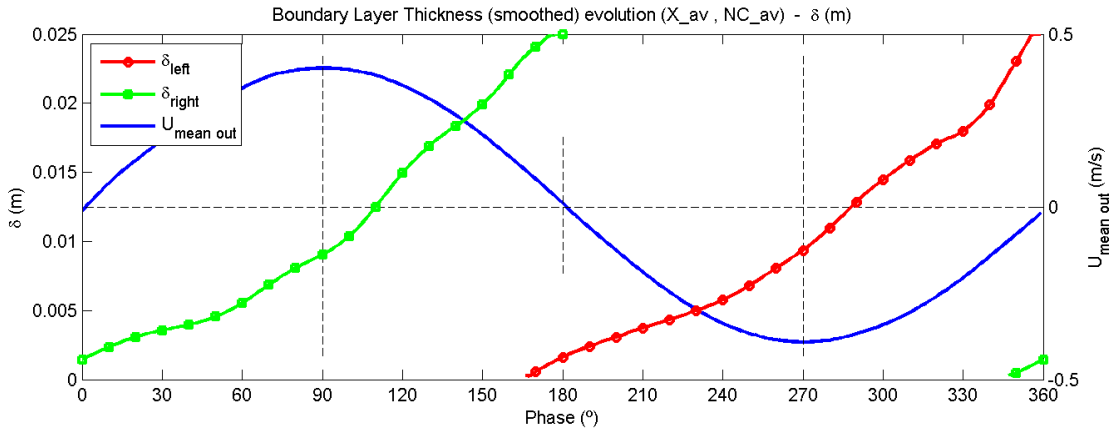
Exp no.	*Temp* (ºC)	ρ (kg/m^3)	ν (m^2/s)	*T* (s)	*2a* (m)	U_{out_max} (m/s)	Re_w (-)	N_{max} (cycles)
5	27.5	996.41	8.59E-07	10	1.261	0.396	2.9E+05	130

Exp no.	U_{*max} (m/s)	τ_{b_max} (N/m^2)	$\Delta\phi$ (deg)	f_{w_ref} (-)	Z_{v_min} (mm)	δ_{v_90} (mm)	δ_{90} (mm)	δ_{top} (mm)
5	0.018	0.32	-26.5	0.0041	0.048	-	9.2	25

C.5.2 Outer flow velocity

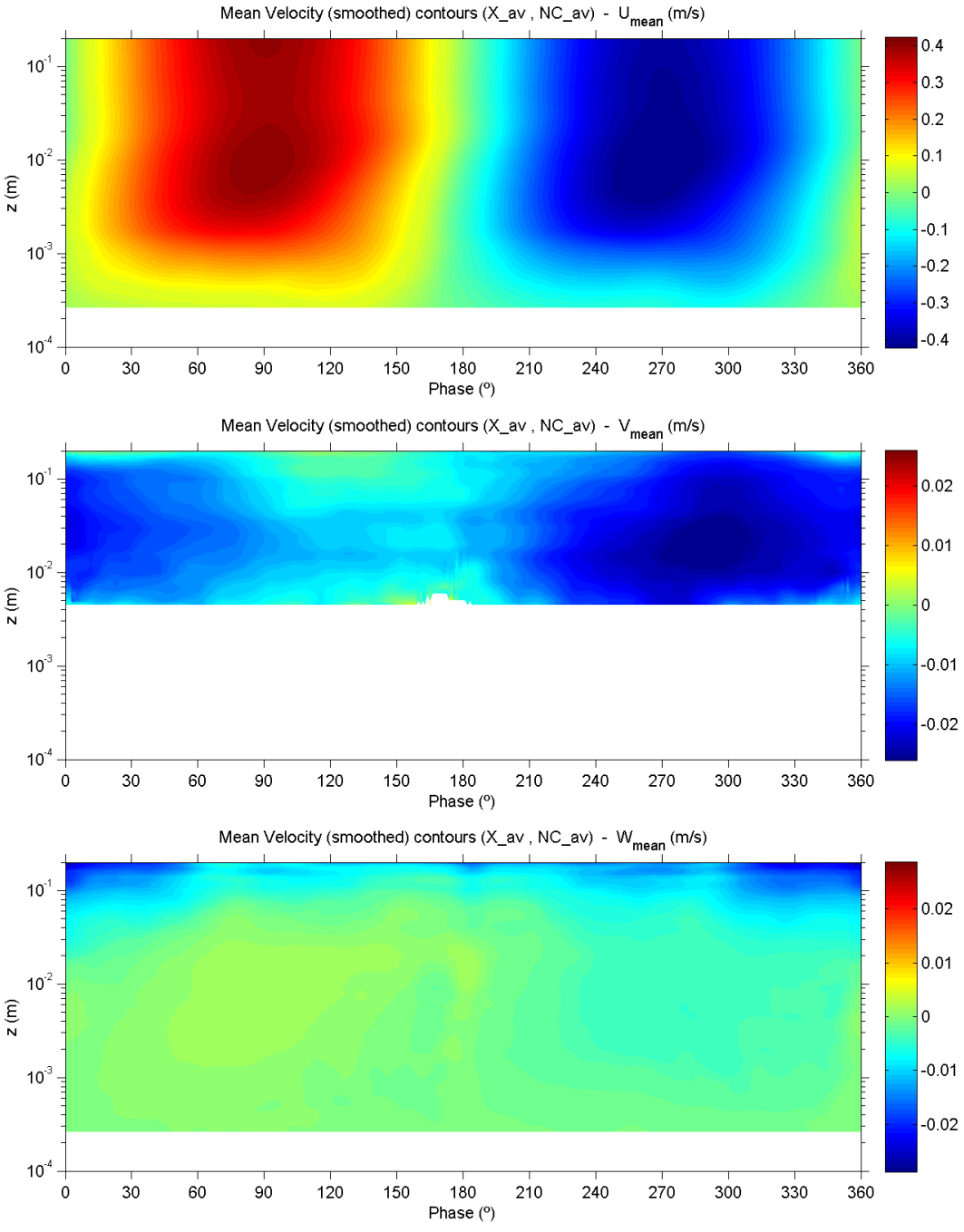


C.5.3 Boundary layer thickness

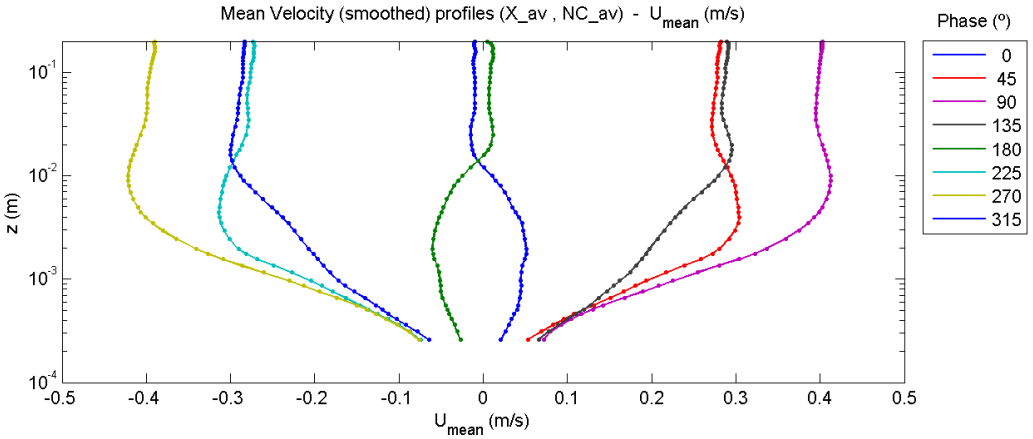


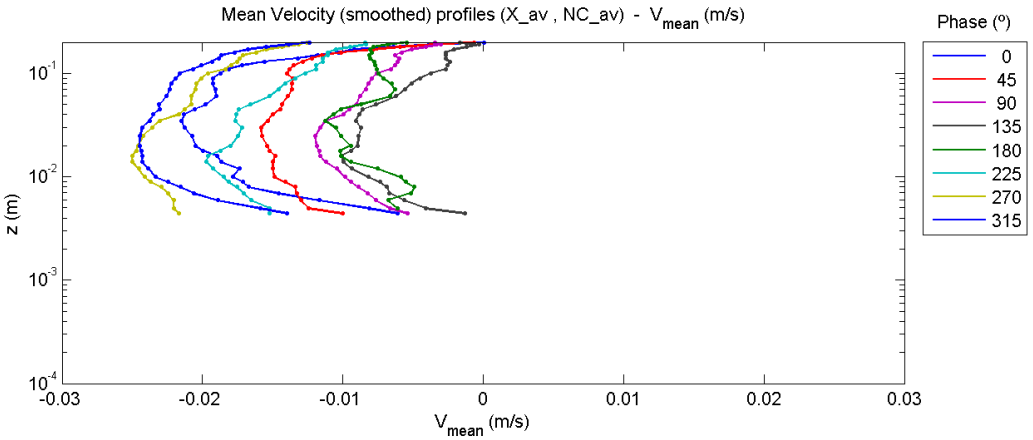
C.5.4 Mean velocities

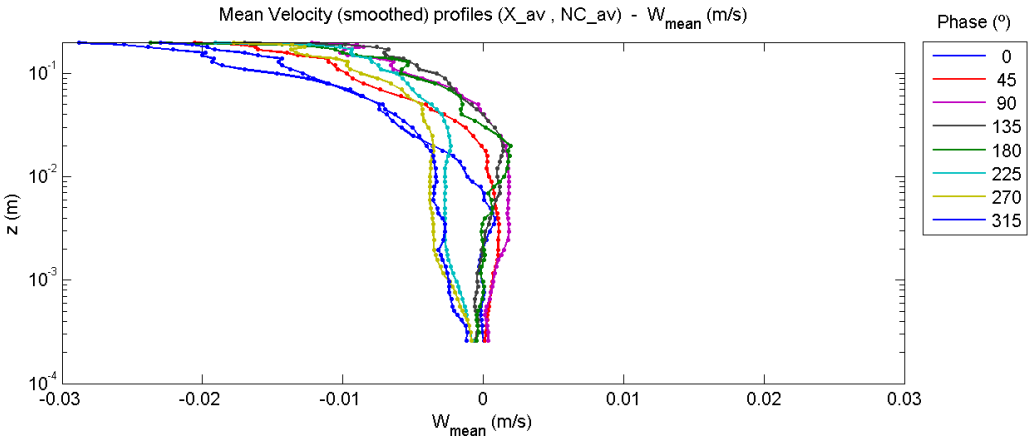
C.5.4.1 Contour plots



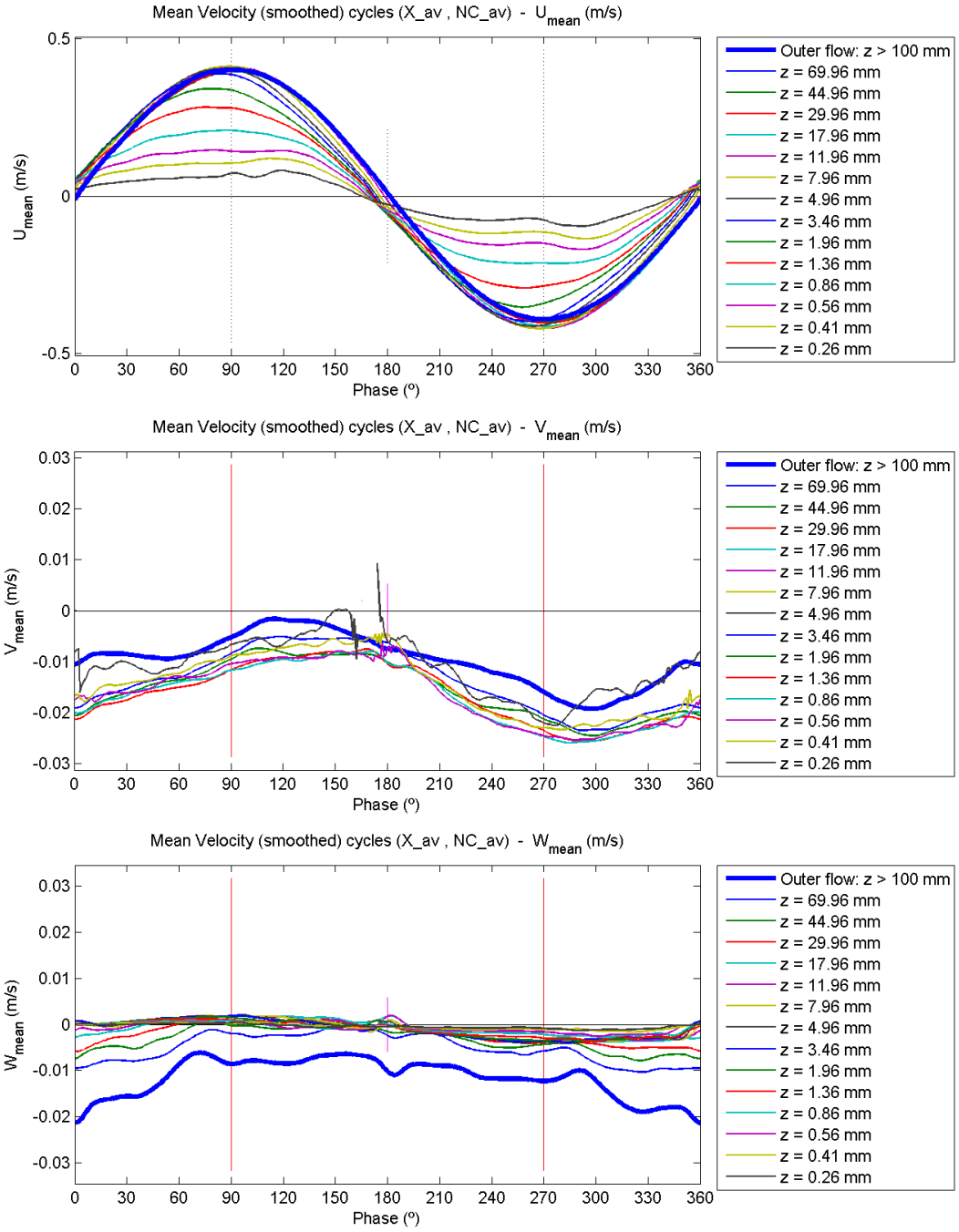
C.5.4.2 Profile plots





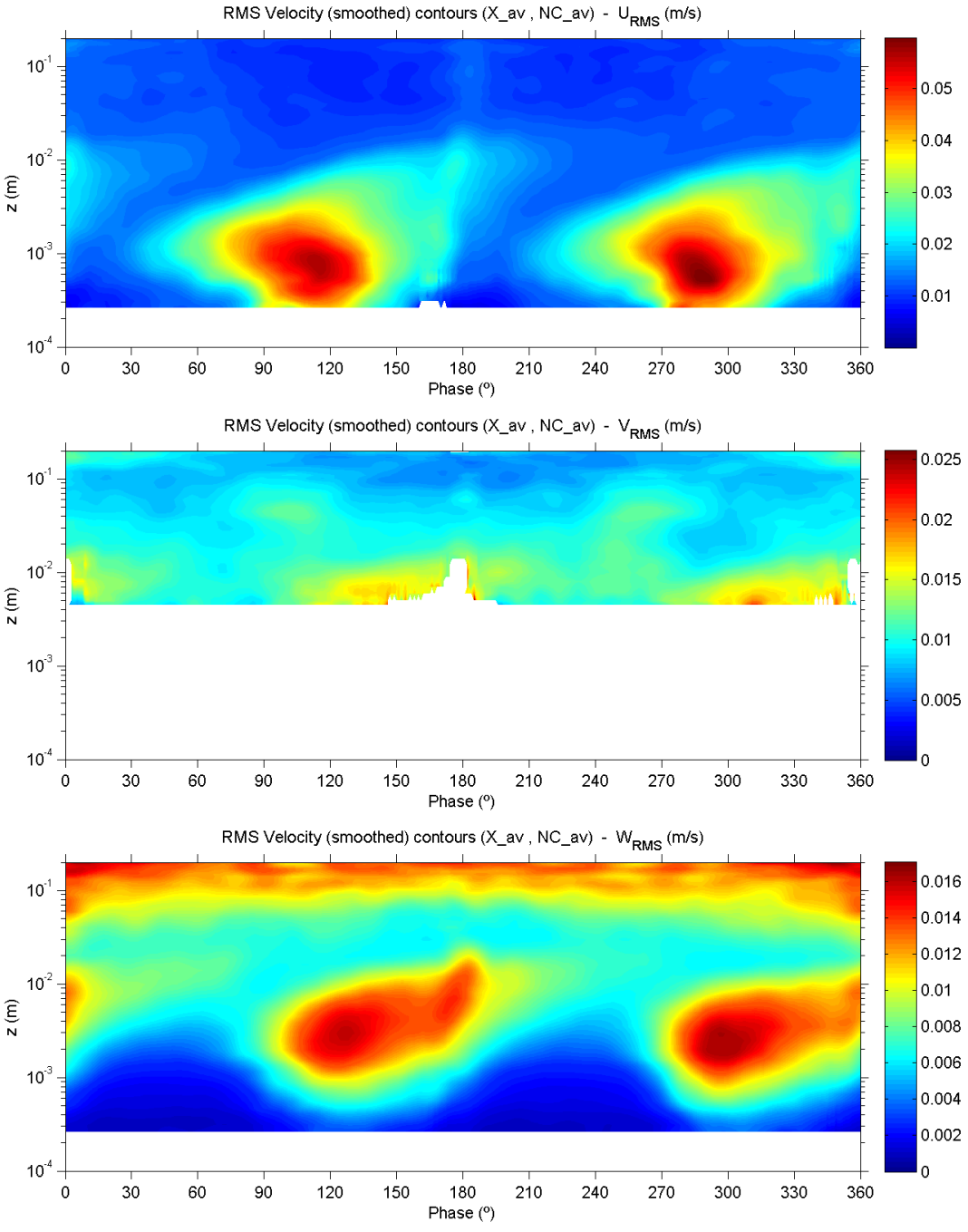


C.5.4.3 Evolution plots

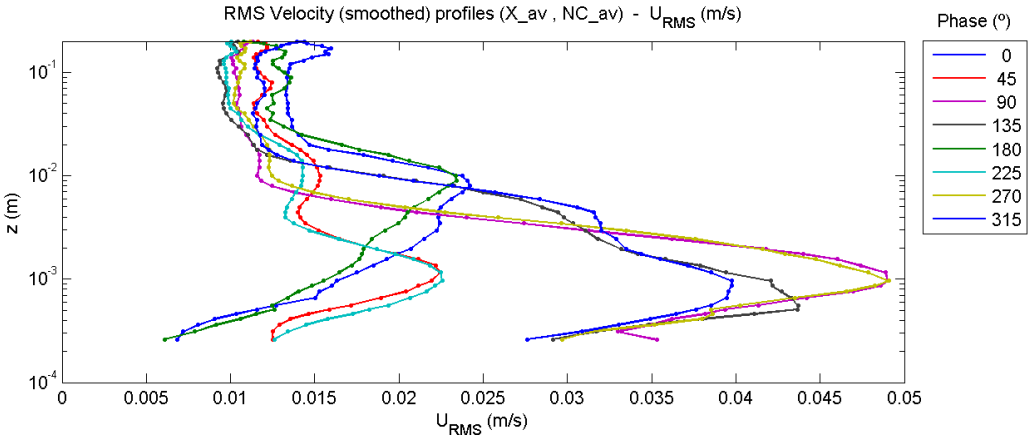


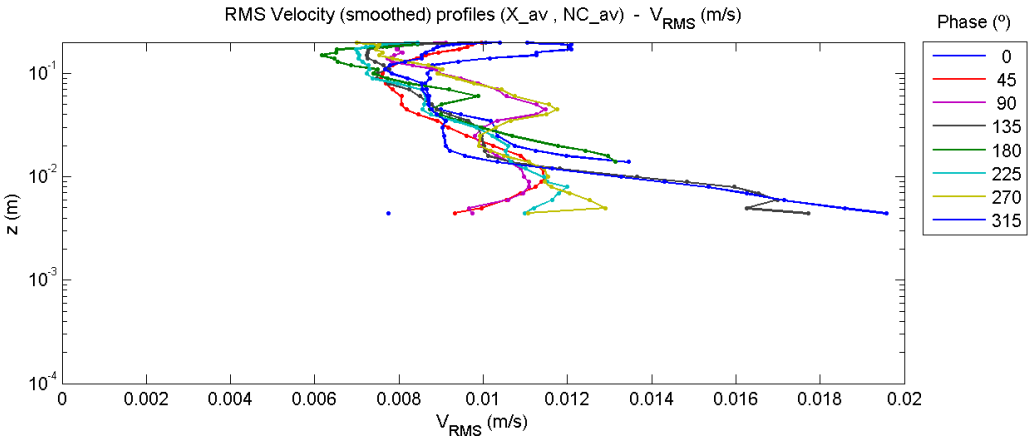
C.5.5 RMS velocity fluctuations

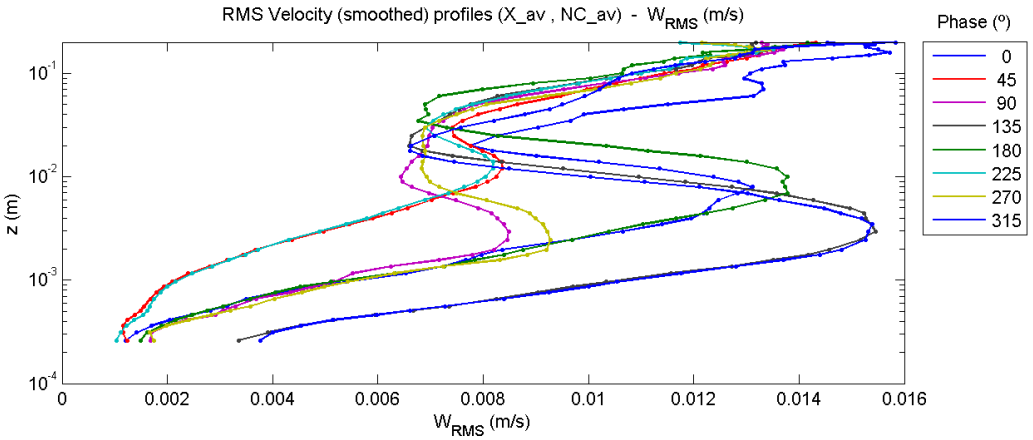
C.5.5.1 Contour plots



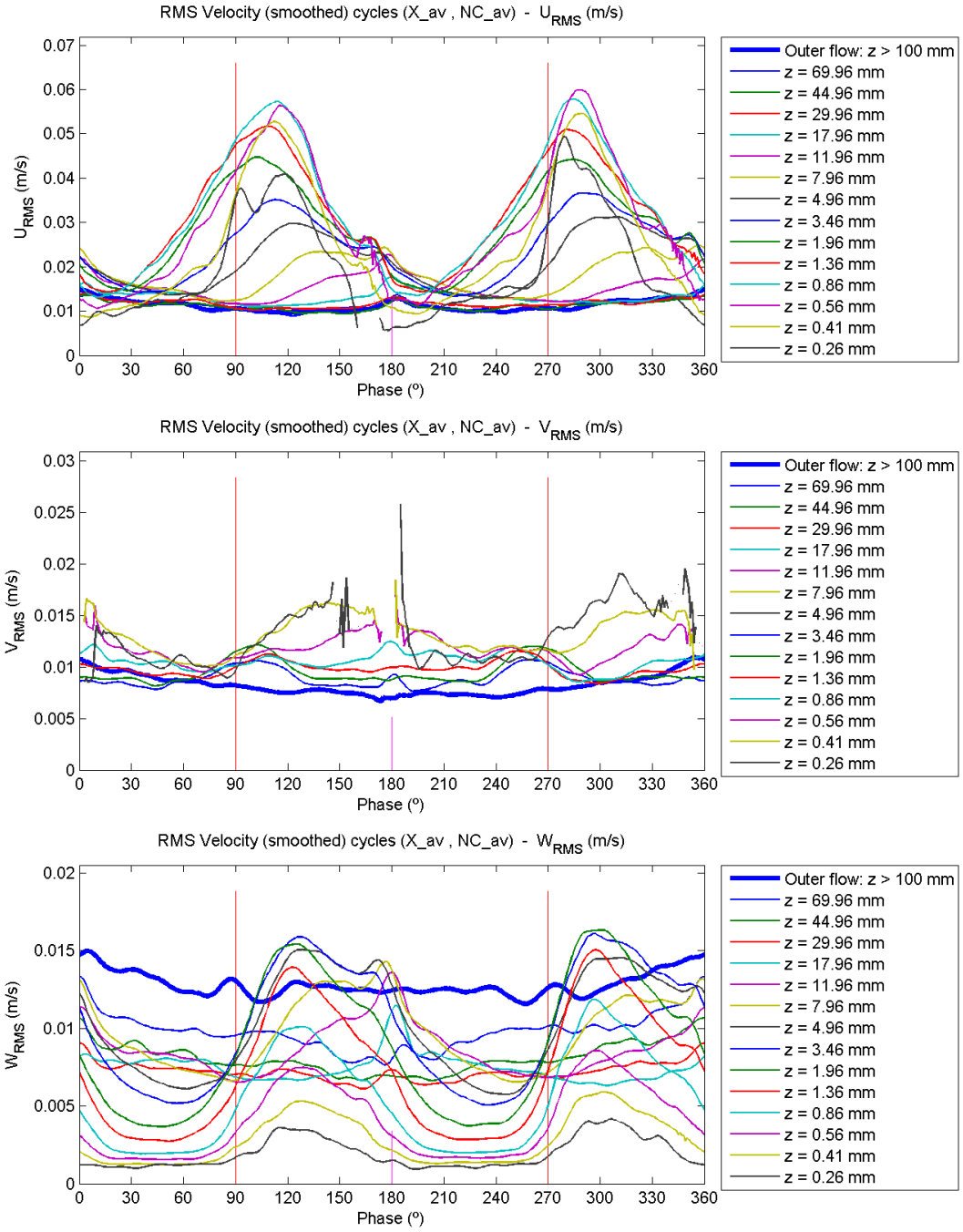
C.5.5.2 Profile plots





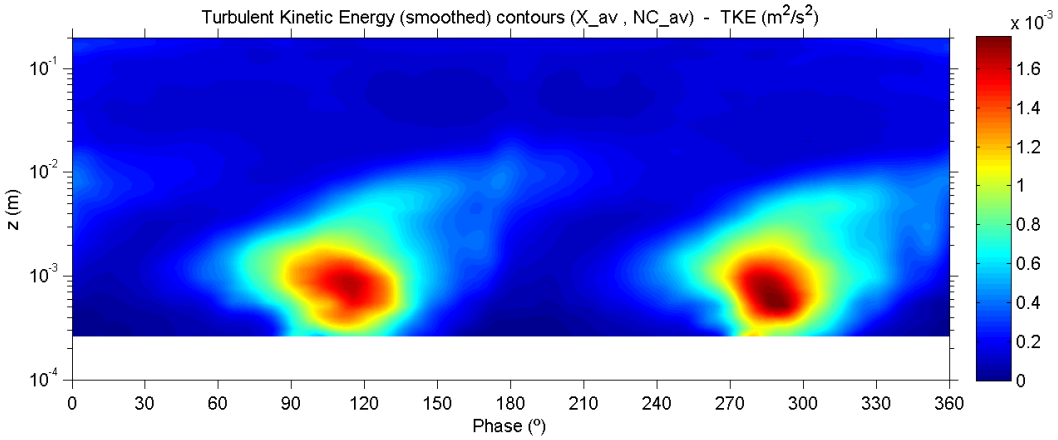


C.5.5.3 Evolution plots

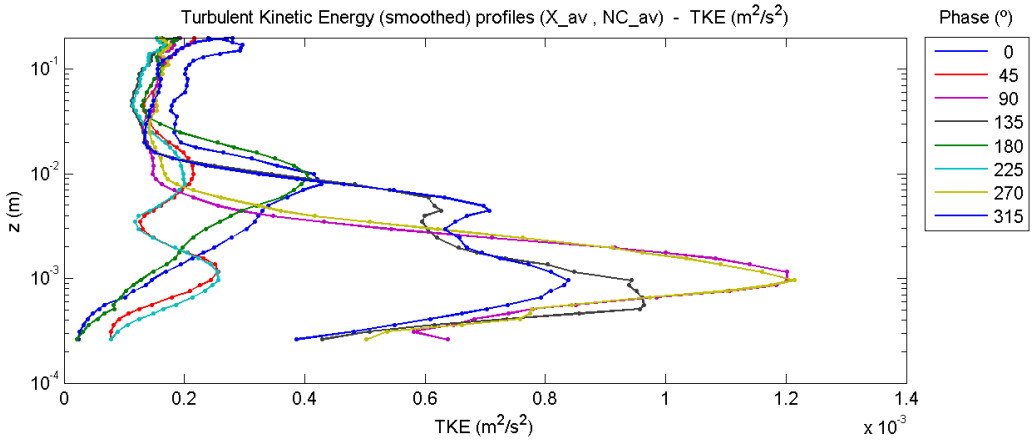


C.5.6 Turbulent kinetic energy

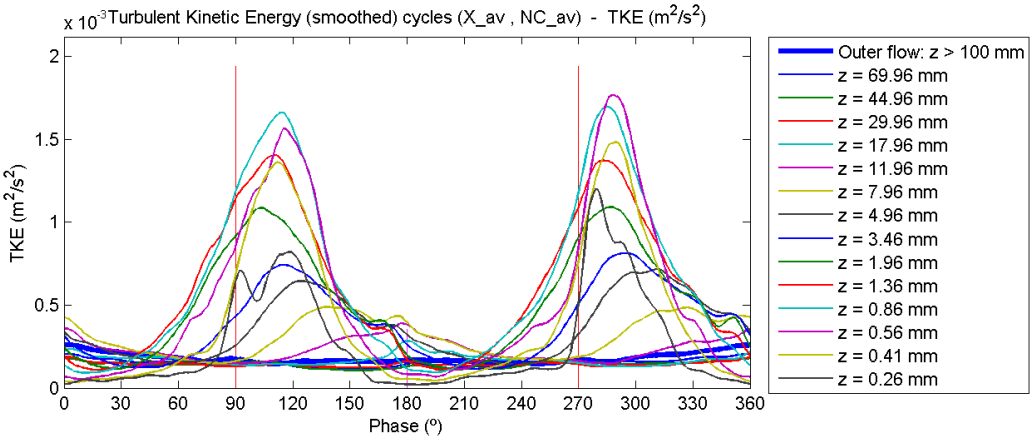
C.5.6.1 Contour plot



C.5.6.2 Profile plot

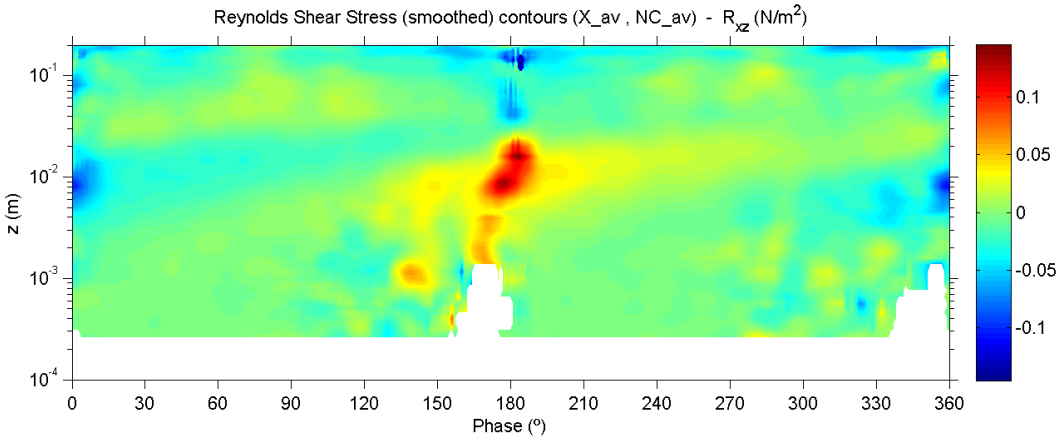


C.5.6.3 Evolution plot

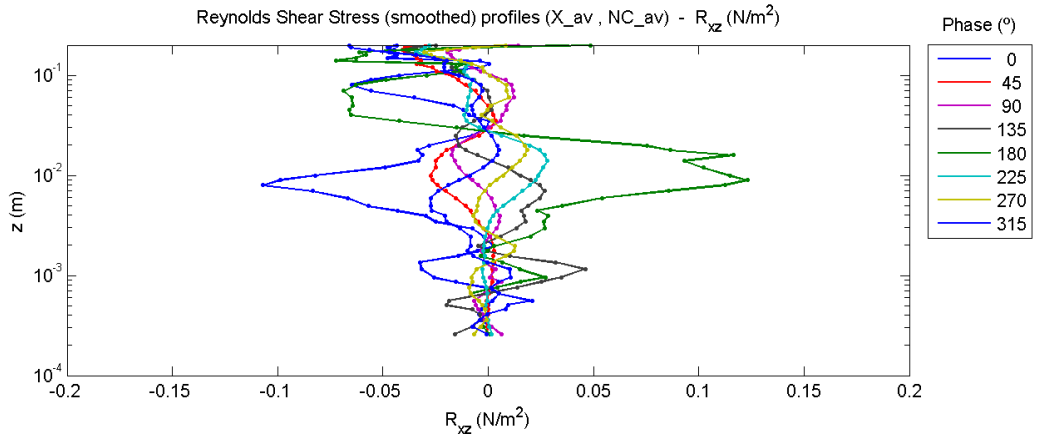


C.5.7 Turbulent shear stress (Reynolds shear stress)

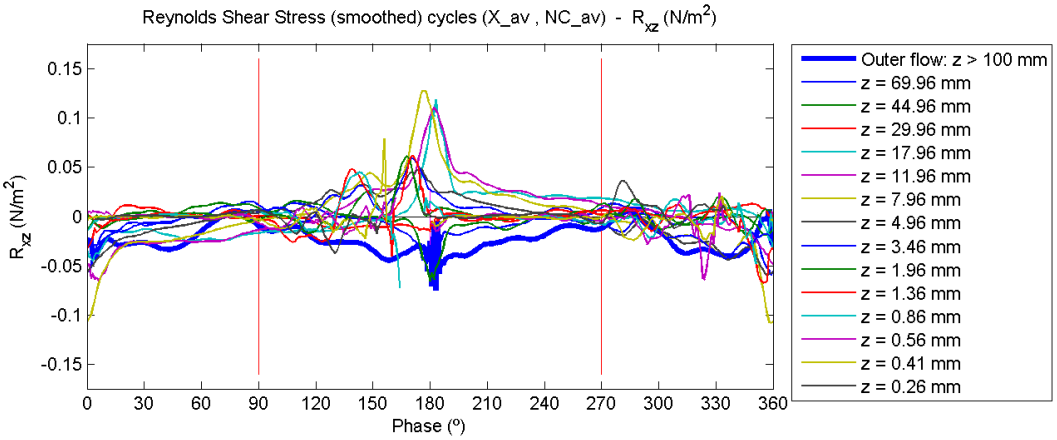
C.5.7.1 Contour plot



C.5.7.2 Profile plot

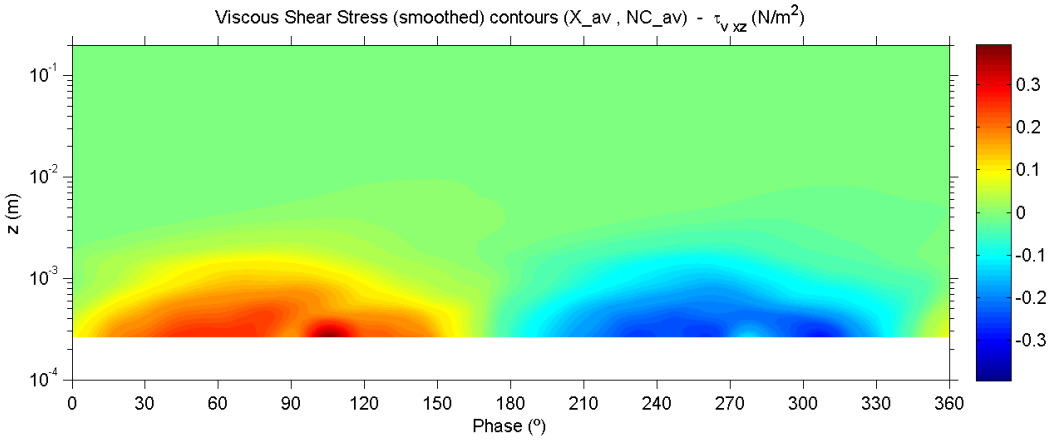


C.5.7.3 Evolution plot

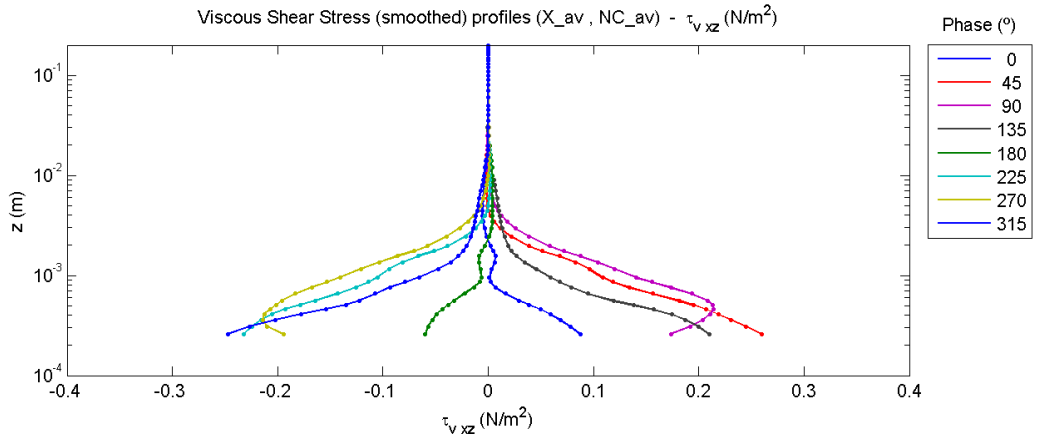


C.5.8 Viscous shear stress

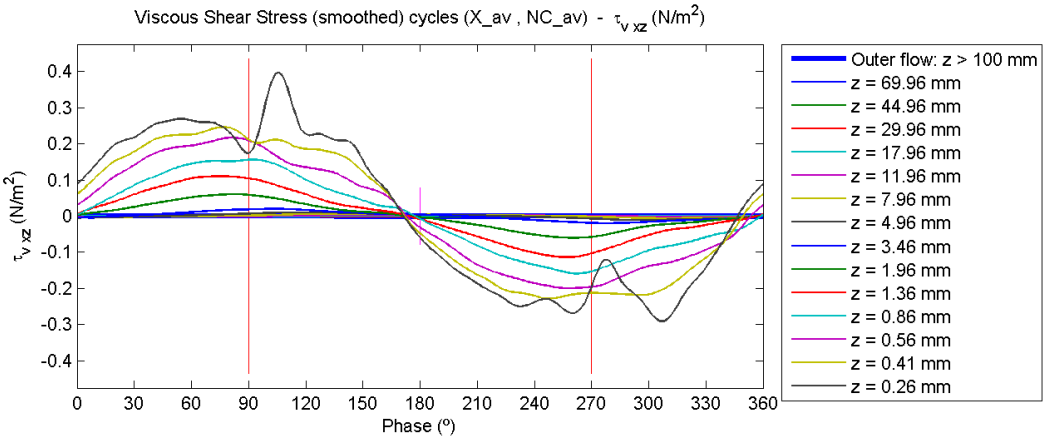
C.5.8.1 Contour plot



C.5.8.2 Profile plot

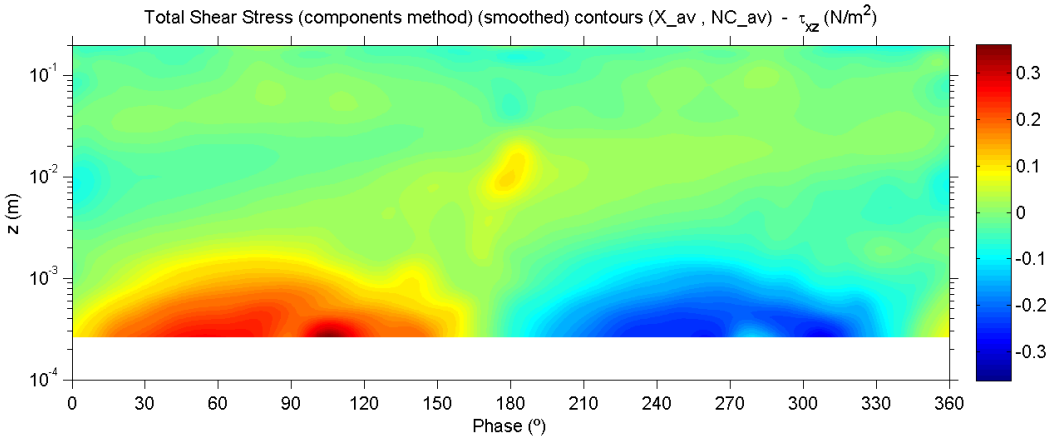


C.5.8.3 Evolution plot

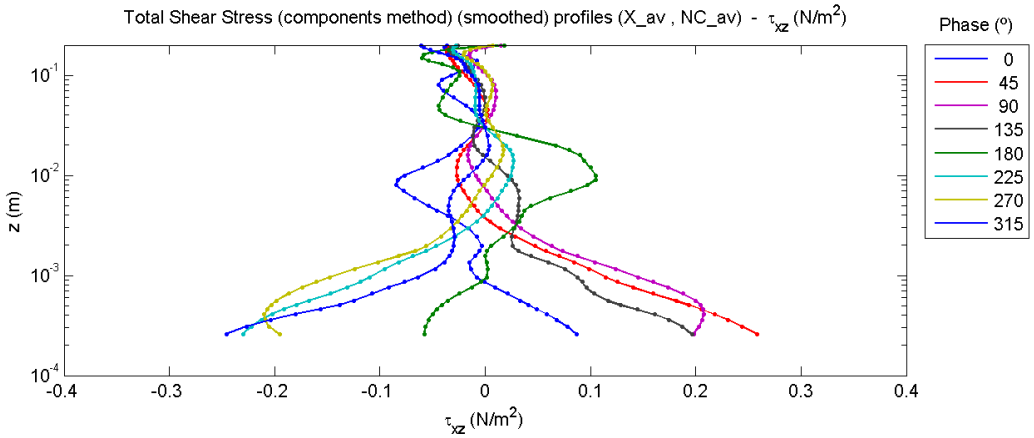


C.5.9 Total shear stress

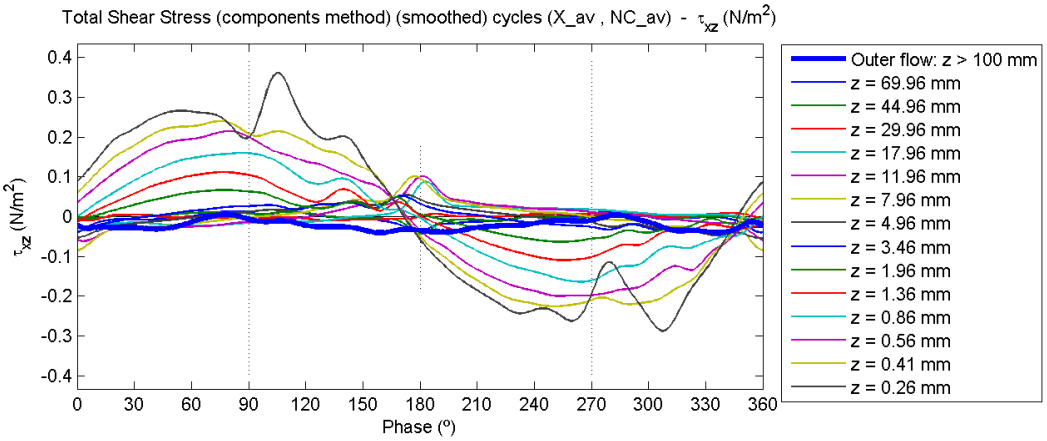
C.5.9.1 Contour plot



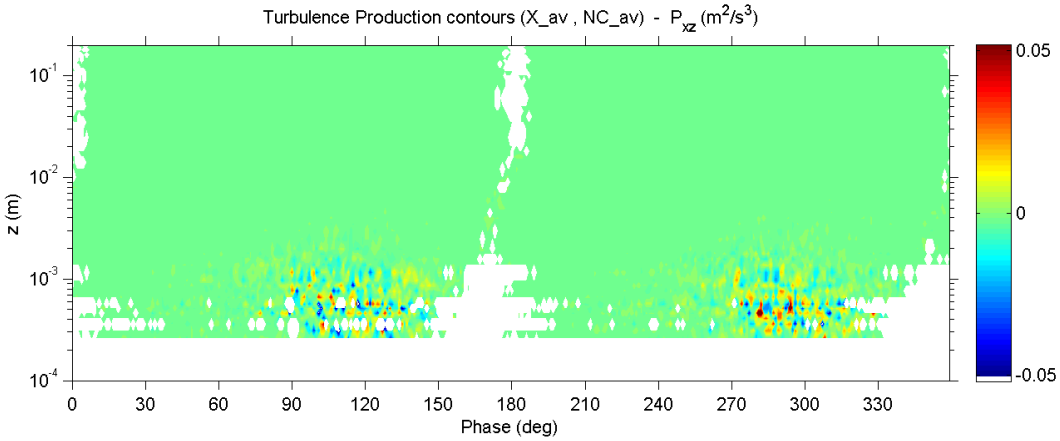
C.5.9.2 Profile plot



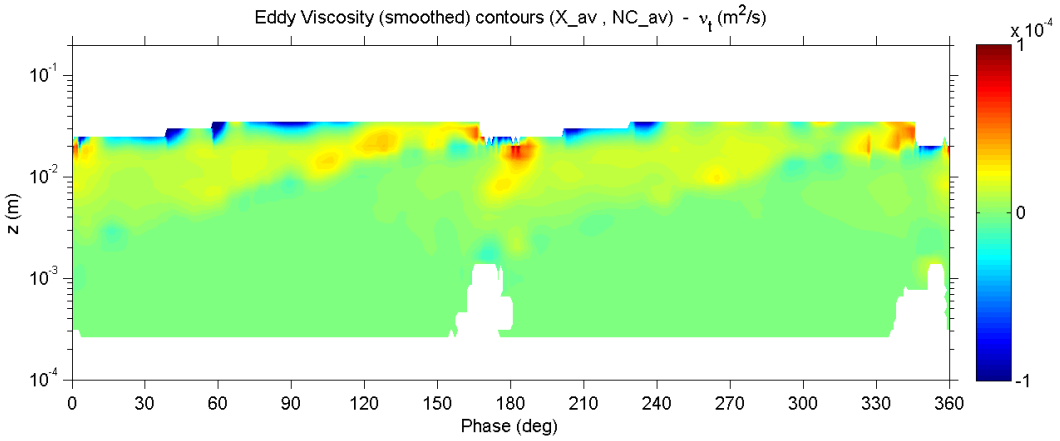
C.5.9.3 Evolution plot



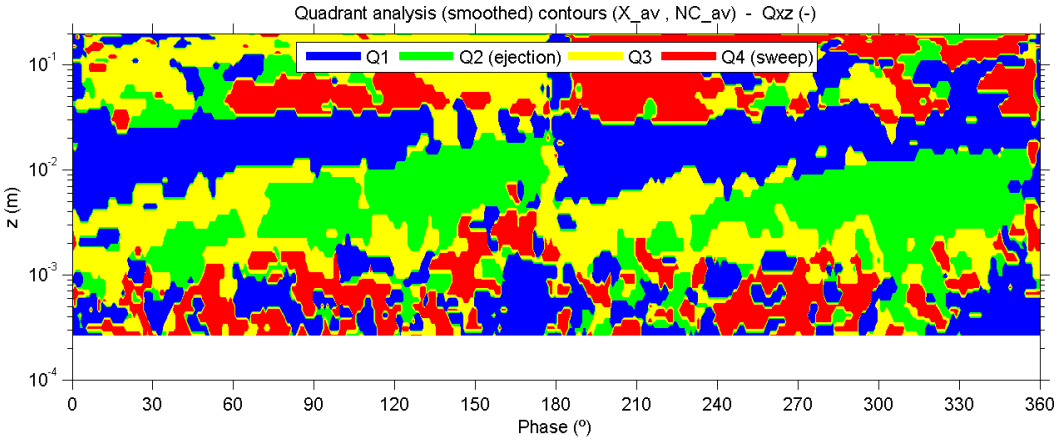
C.5.10 Turbulence production



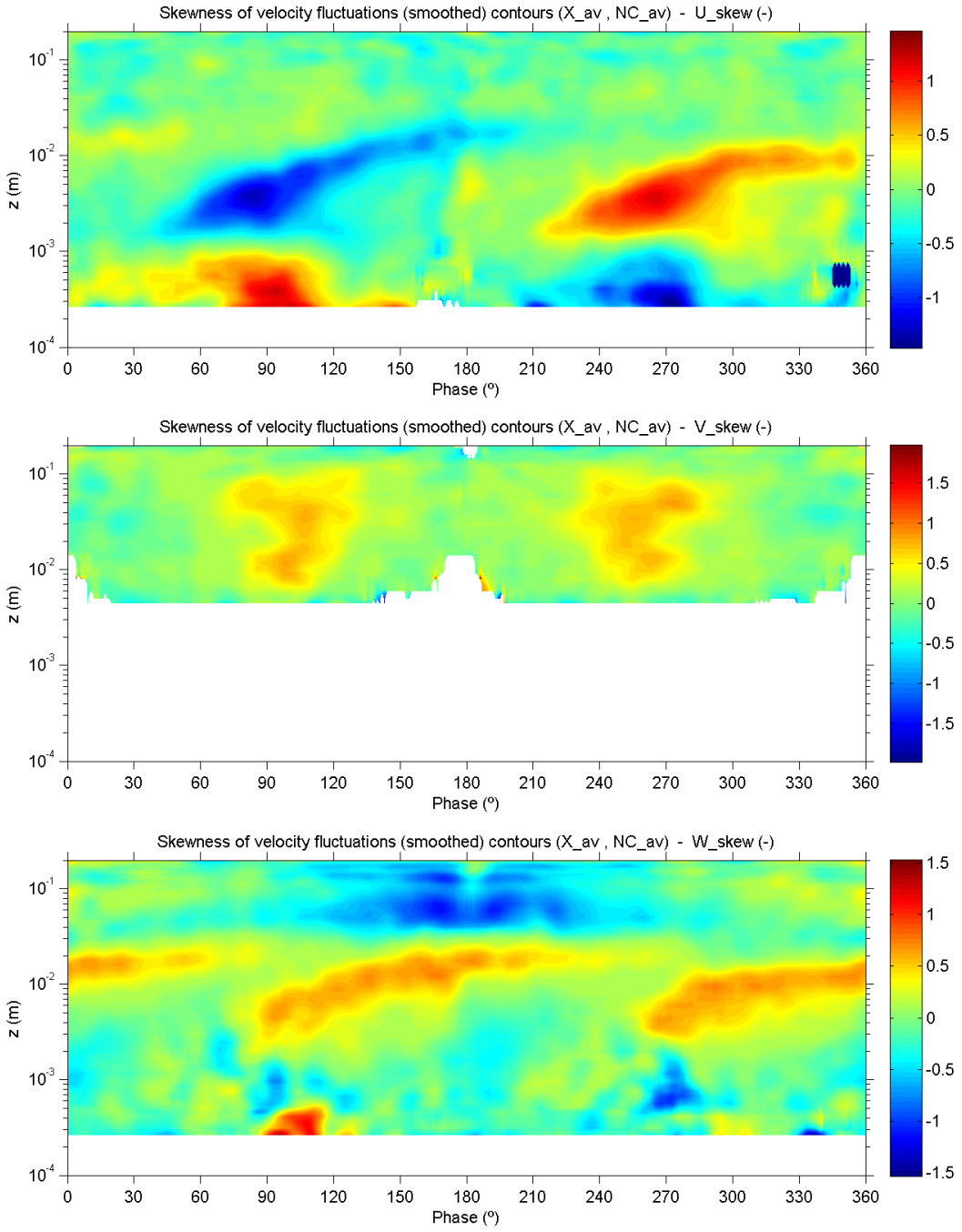
C.5.11 Turbulent viscosity (Eddy viscosity)



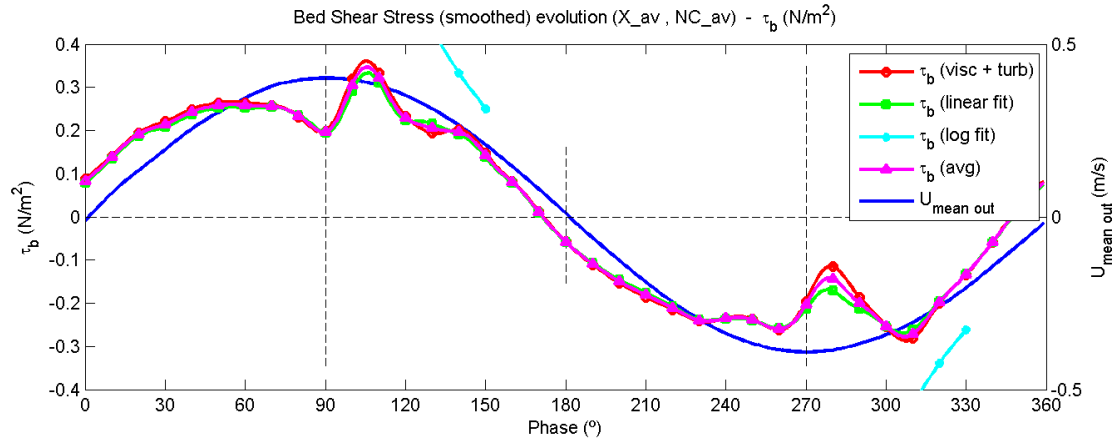
C.5.12 Quadrant analysis



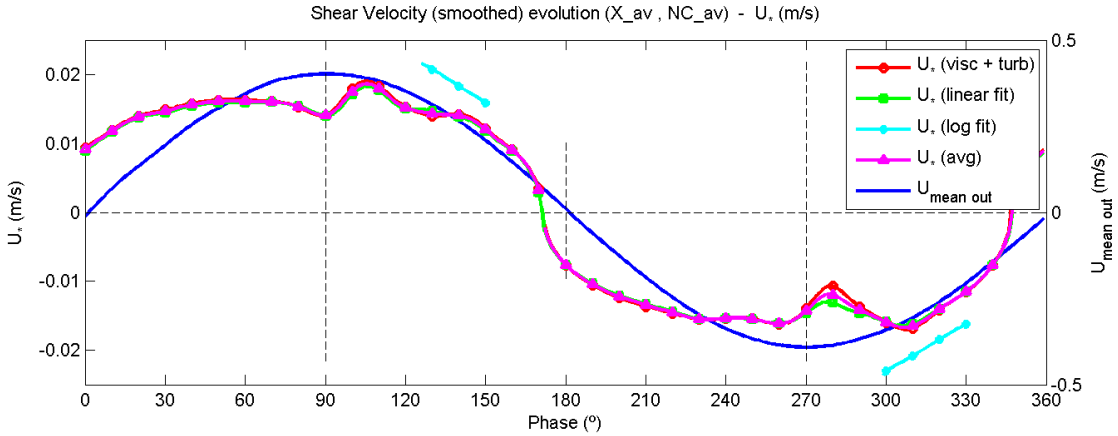
C.5.13 Skewness of velocity fluctuations



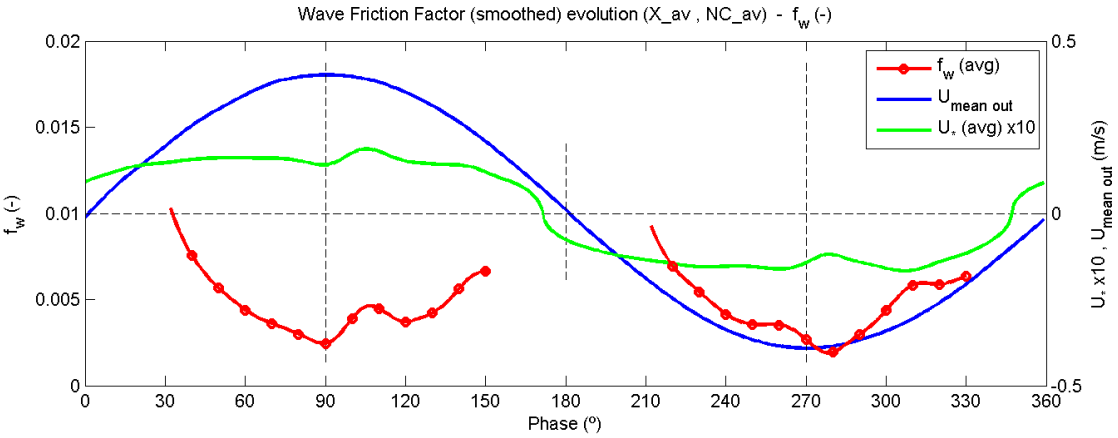
C.5.14 Bed shear stress



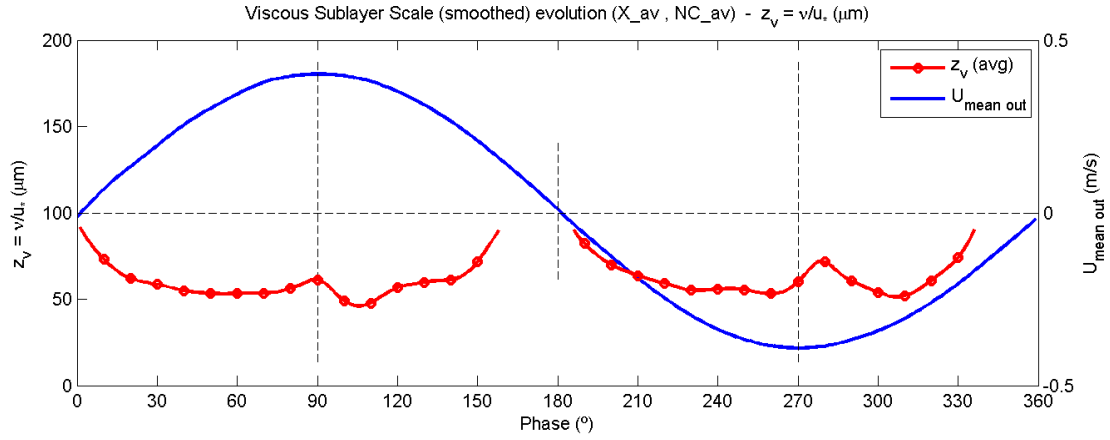
C.5.15 Shear velocity



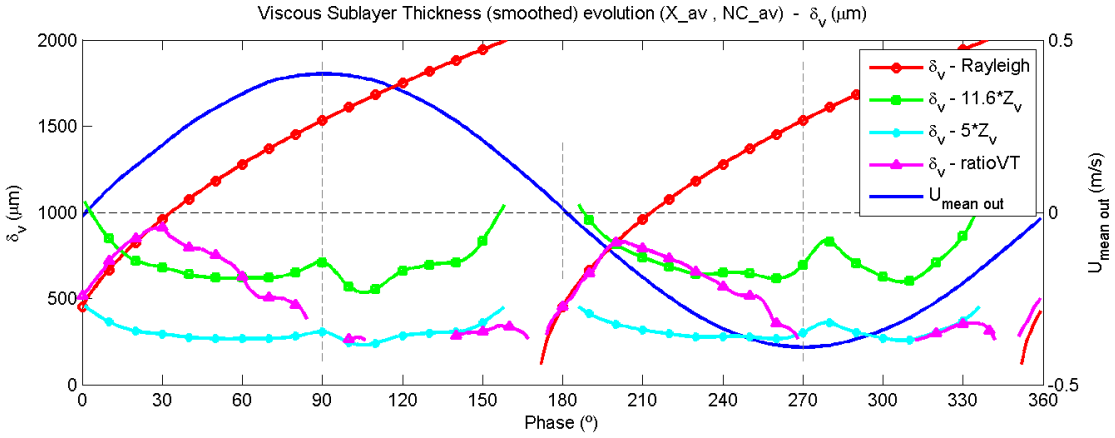
C.5.16 Wave friction factor



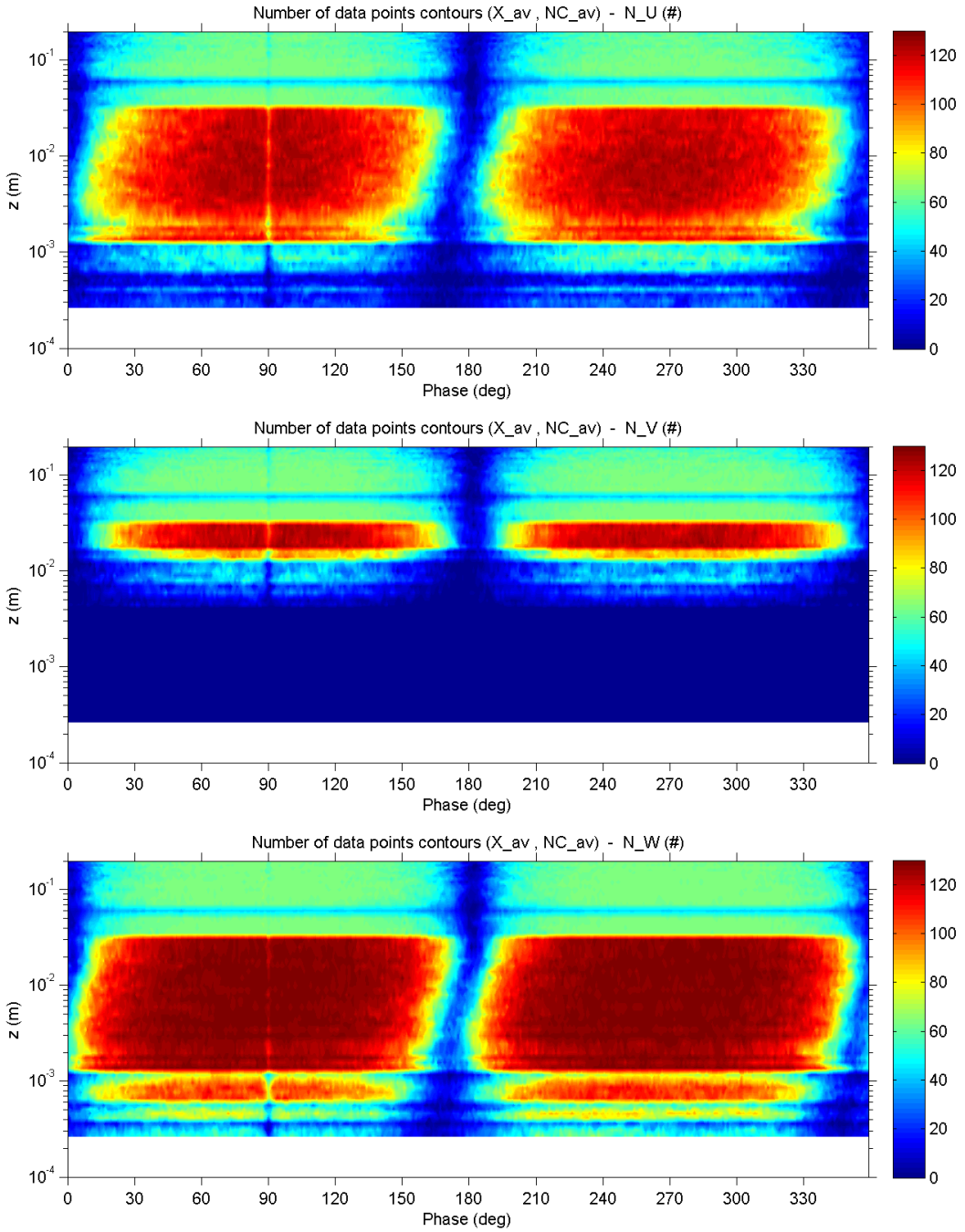
C.5.17 Viscous length scale



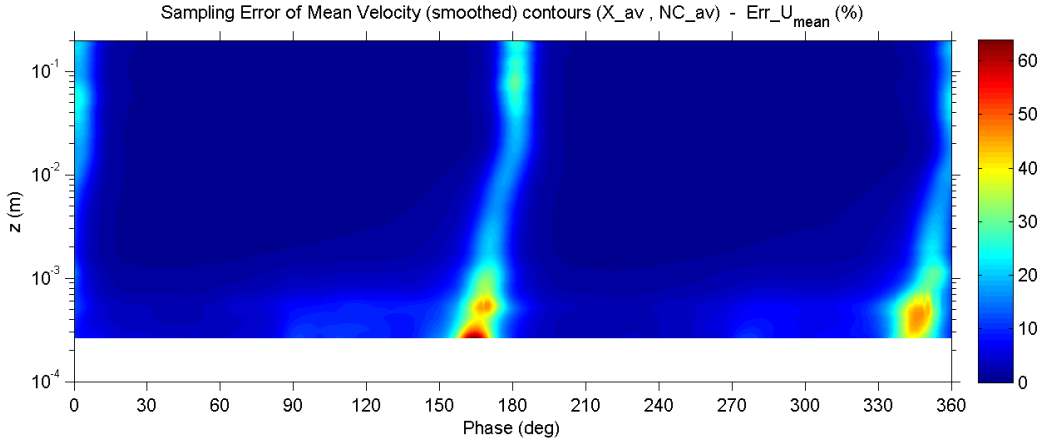
C.5.18 Viscous sublayer thickness



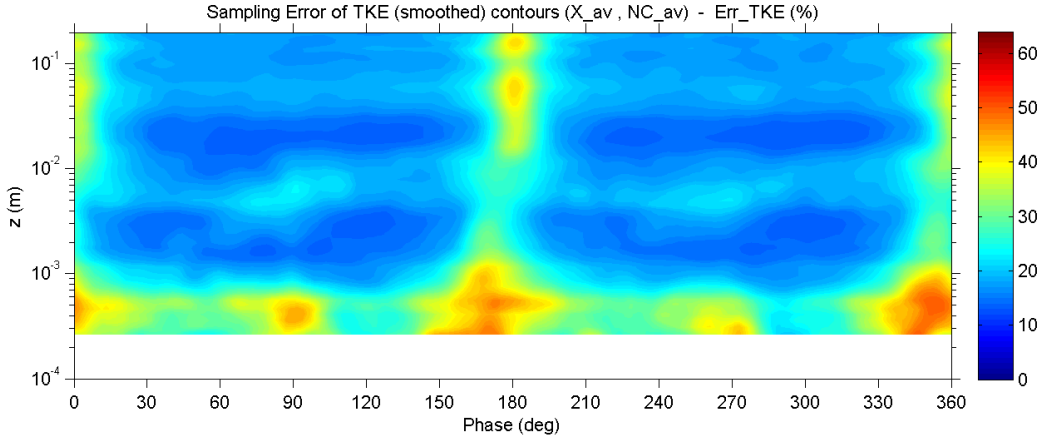
C.5.19 Number of valid data points



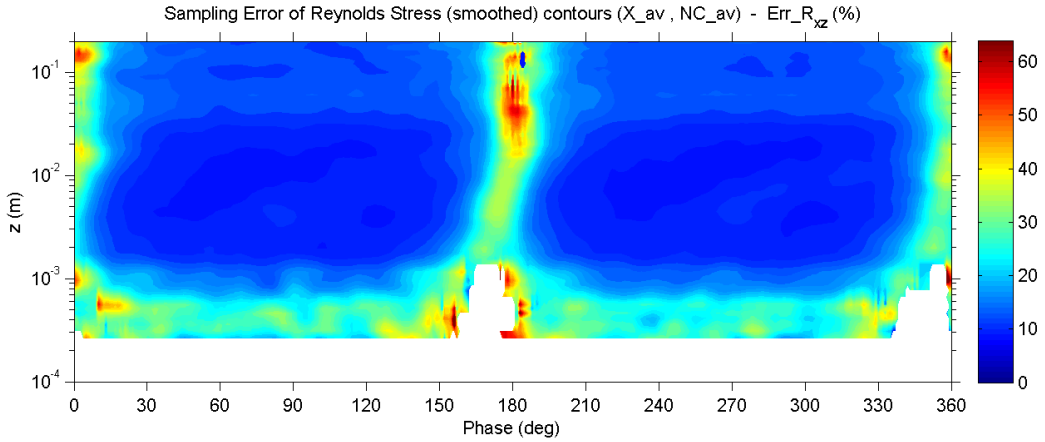
C.5.20 Percent error of mean velocity



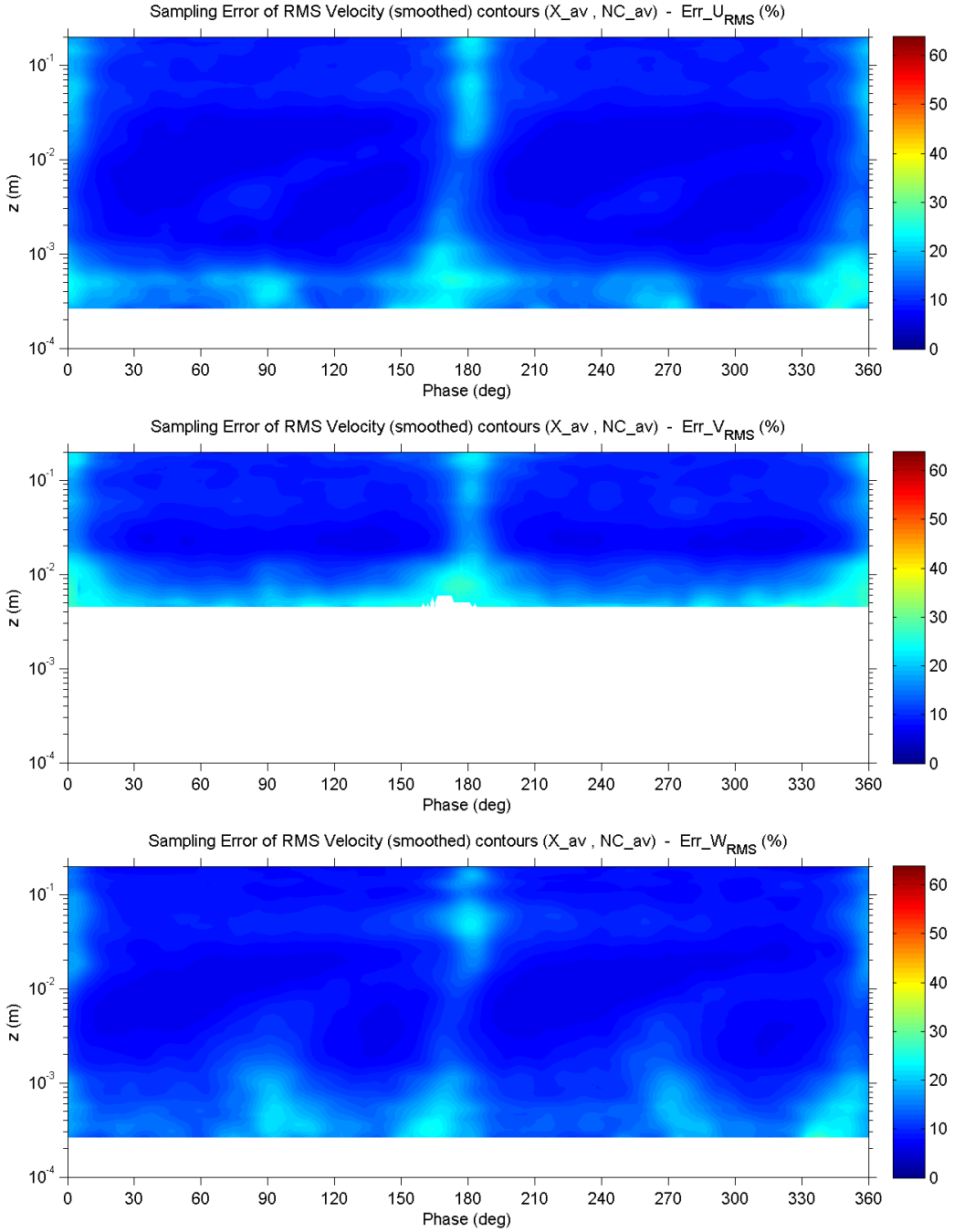
C.5.21 Percent error of turbulent kinetic energy



C.5.22 Percent error of Reynolds shear stress



C.5.23 Percent error of RMS velocity fluctuations



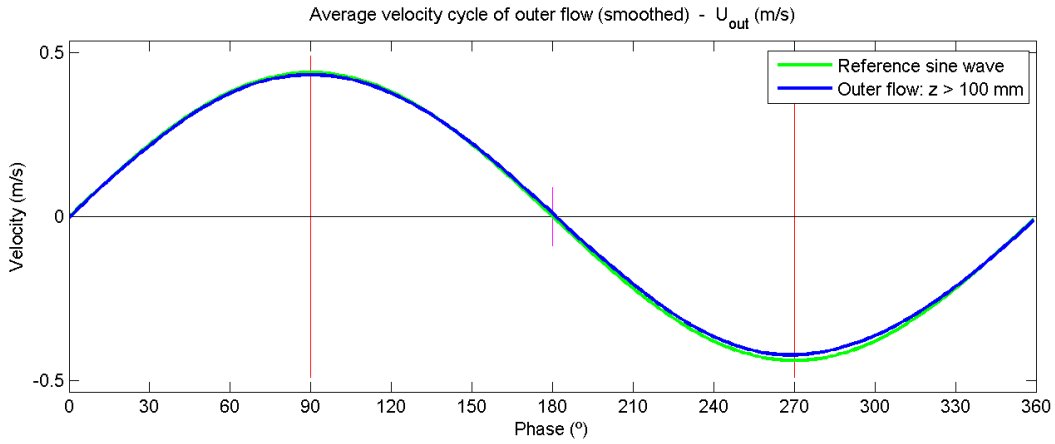
C.6 Experiment no. 6

C.6.1 Main parameters

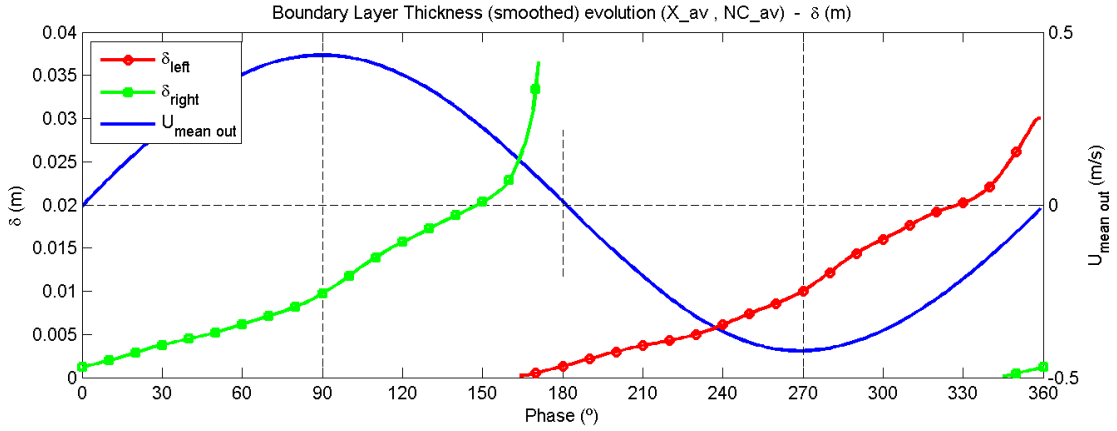
Exp no.	*Temp* (ºC)	ρ (kg/m^3)	ν (m^2/s)	*T* (s)	*2a* (m)	U_{out_max} (m/s)	Re_w (-)	N_{max} (cycles)
6	27.0	996.54	8.68E-07	10	1.362	0.428	3.4E+05	130

Exp no.	U_{*max} (m/s)	τ_{b_max} (N/m^2)	$\Delta\phi$ (deg)	f_{w_ref} (-)	Z_{v_min} (mm)	δ_{v_90} (mm)	δ_{90} (mm)	δ_{top} (mm)
6	0.023	0.52	-16.0	0.0057	0.038	0.172	9.8	30

C.6.2 Outer flow velocity

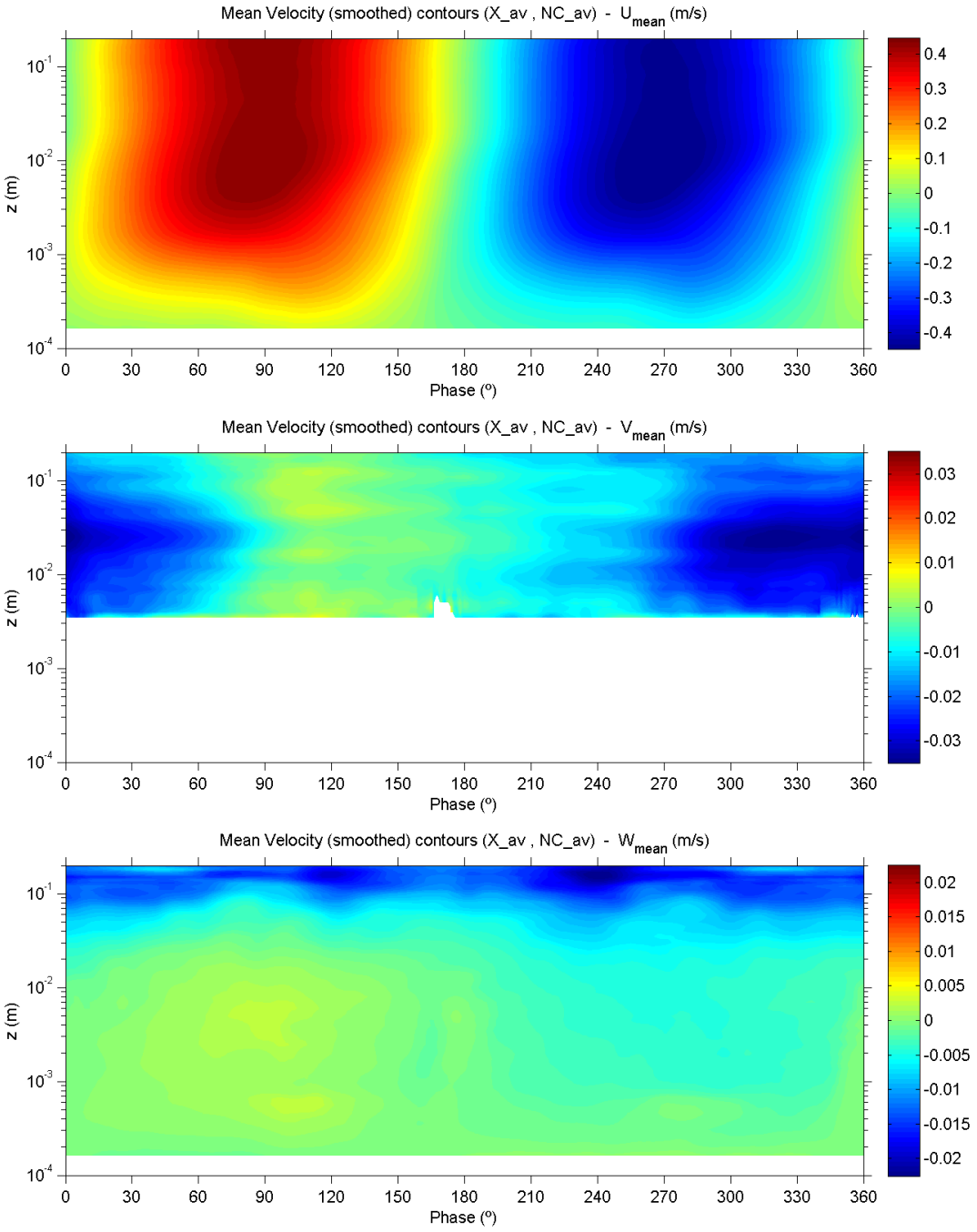


C.6.3 Boundary layer thickness

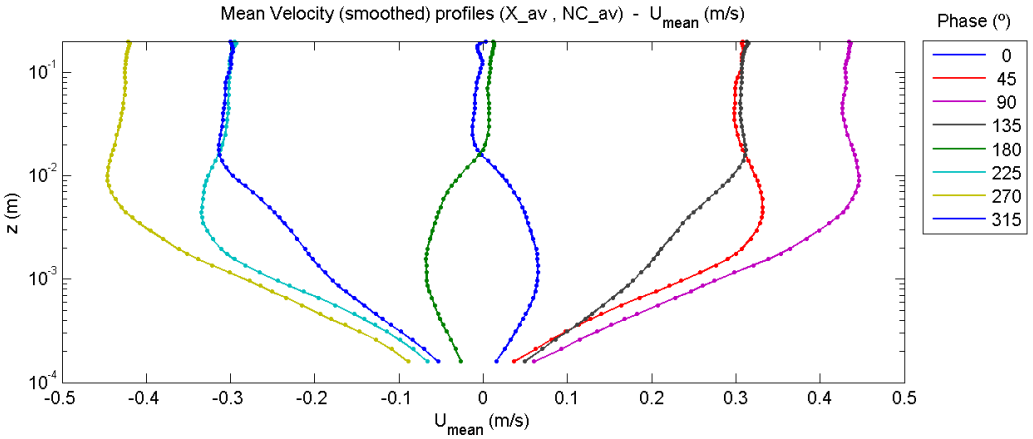


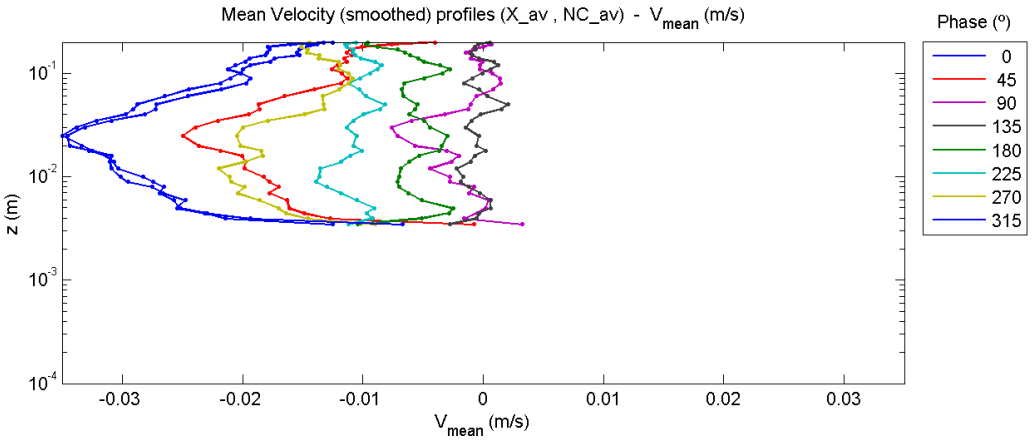
C.6.4 Mean velocities

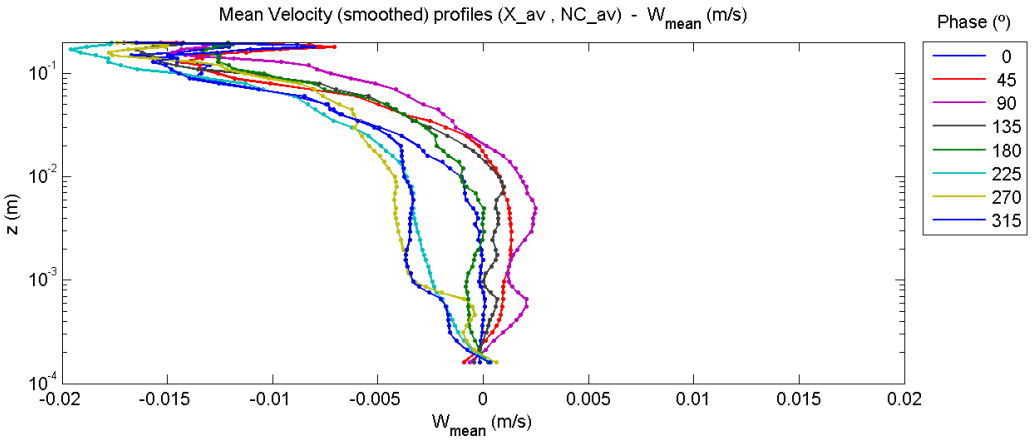
C.6.4.1 Contour plots



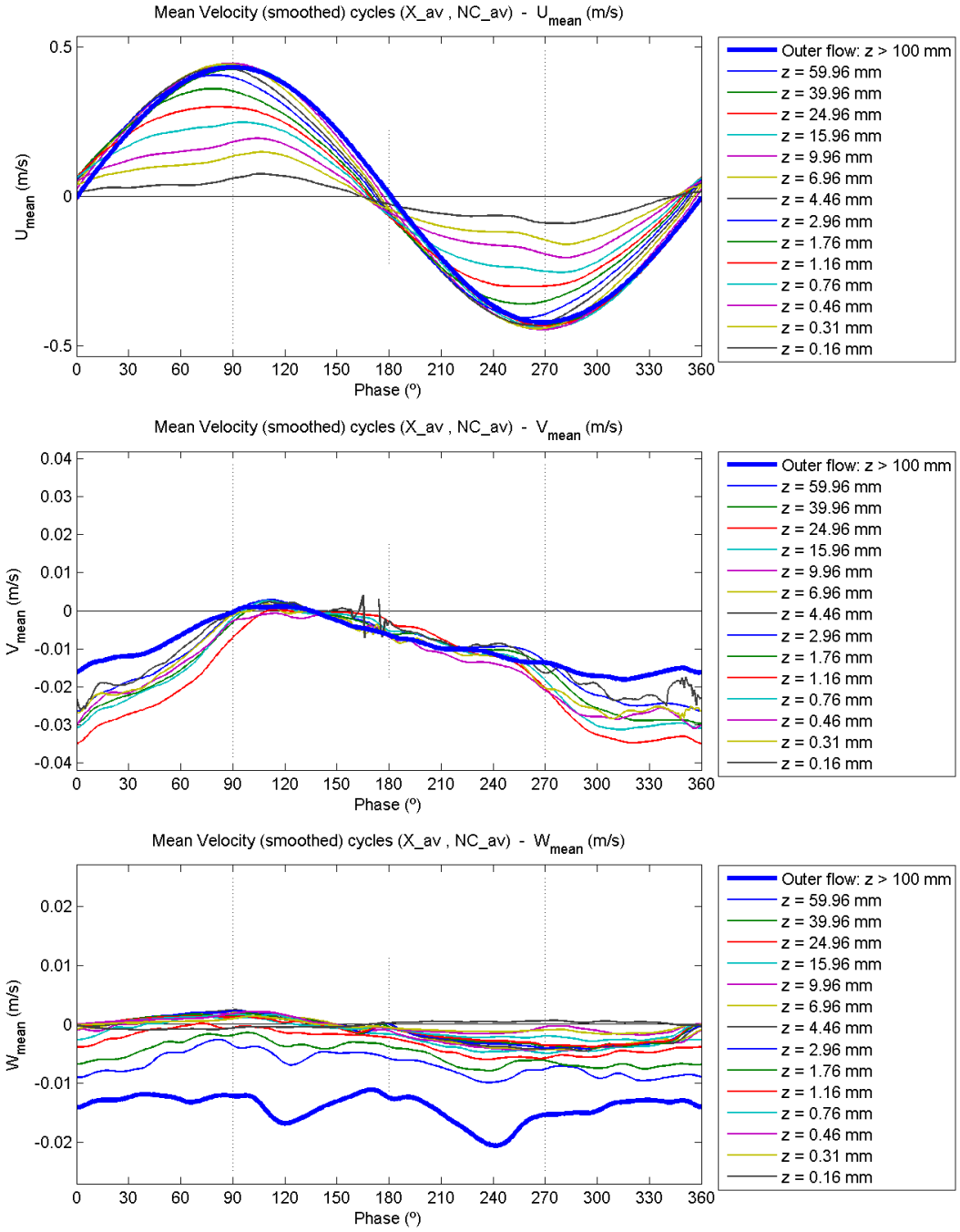
C.6.4.2 Profile plots





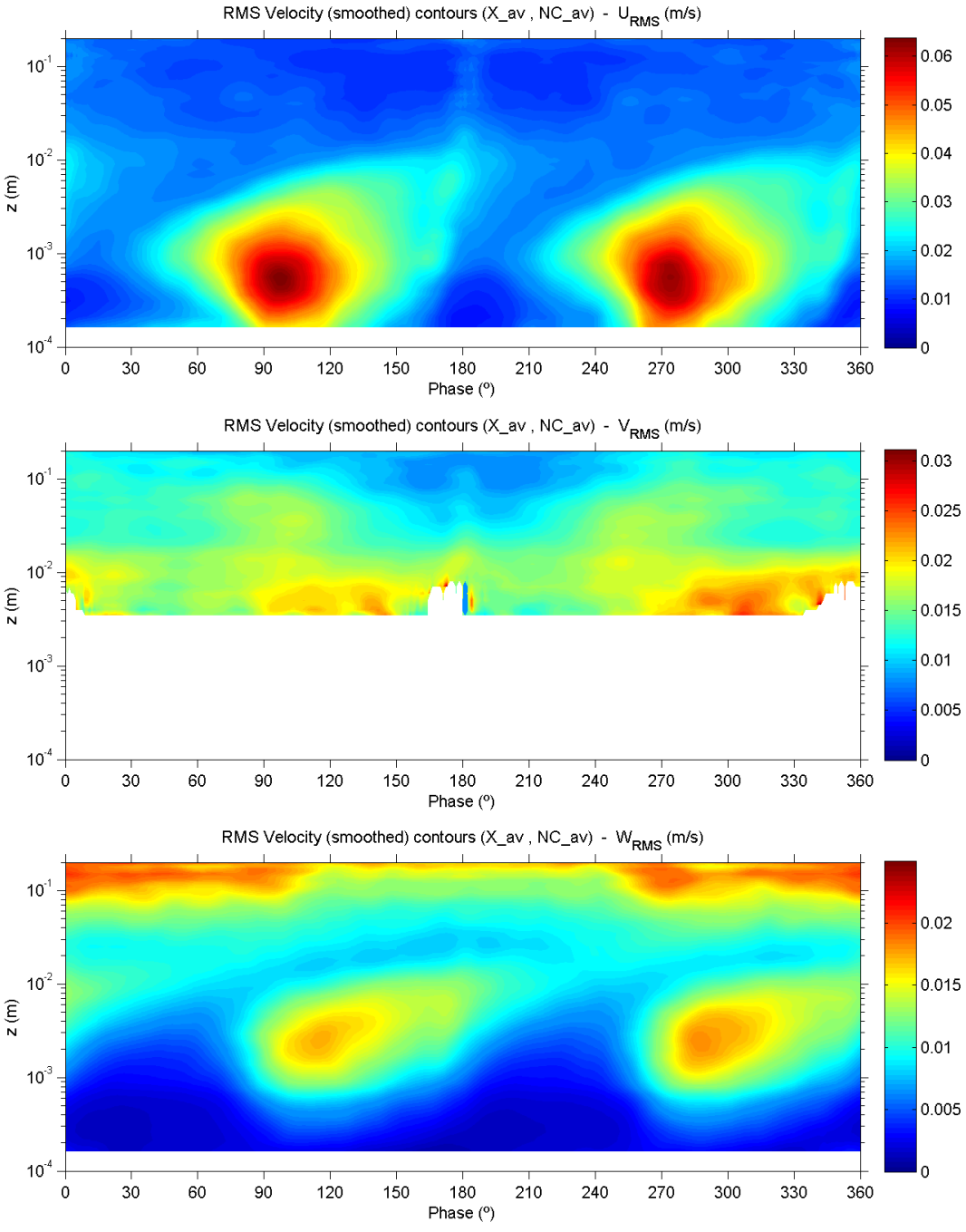


C.6.4.3 Evolution plots

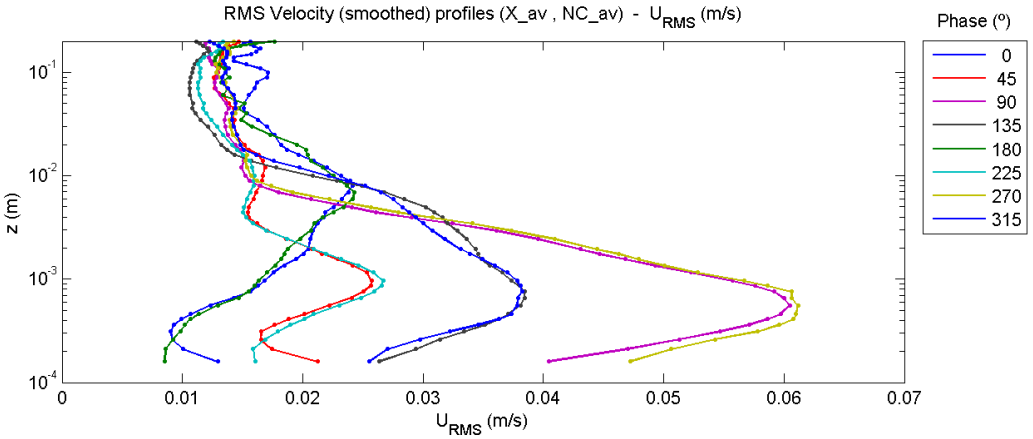


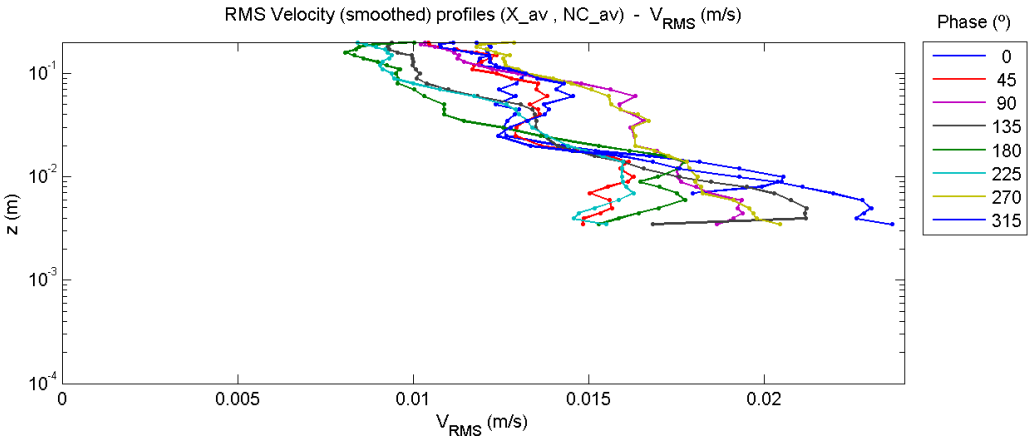
C.6.5 RMS velocity fluctuations

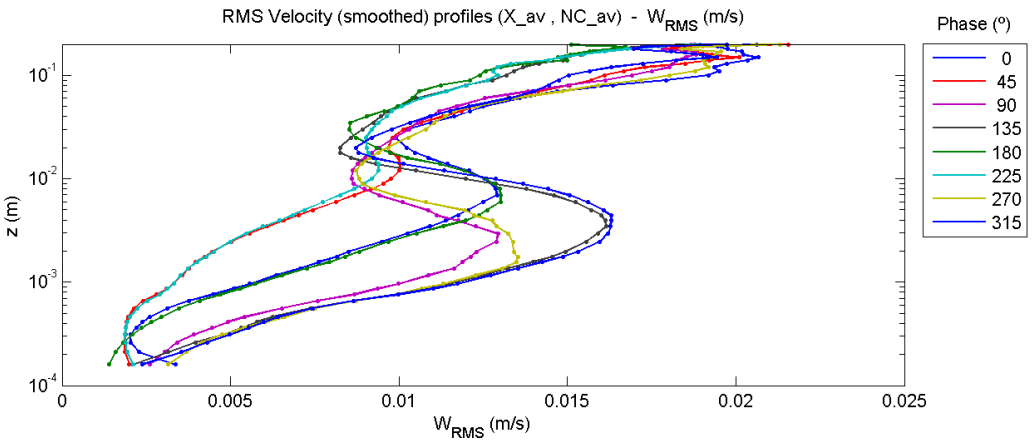
C.6.5.1 Contour plots



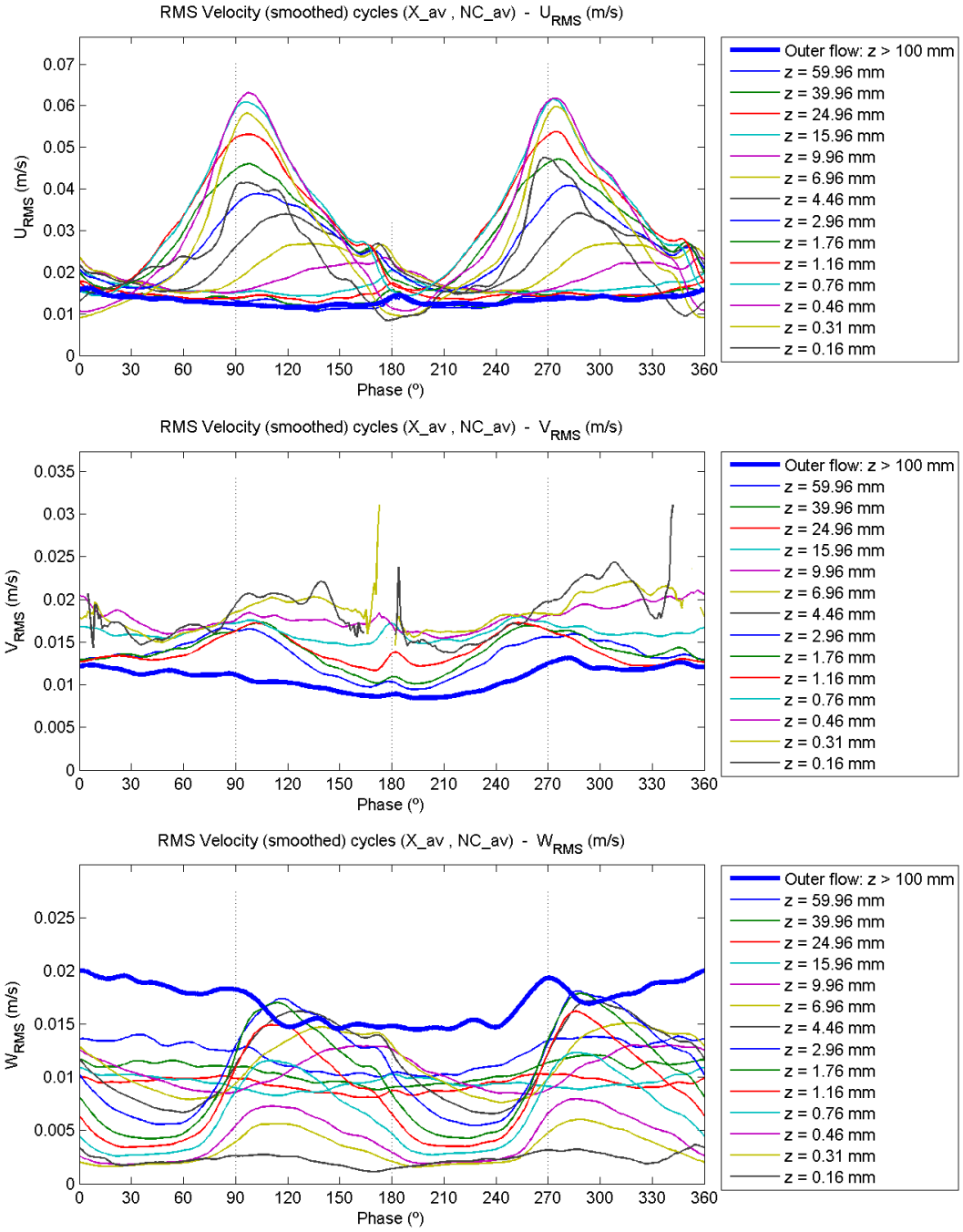
C.6.5.2 Profile plots





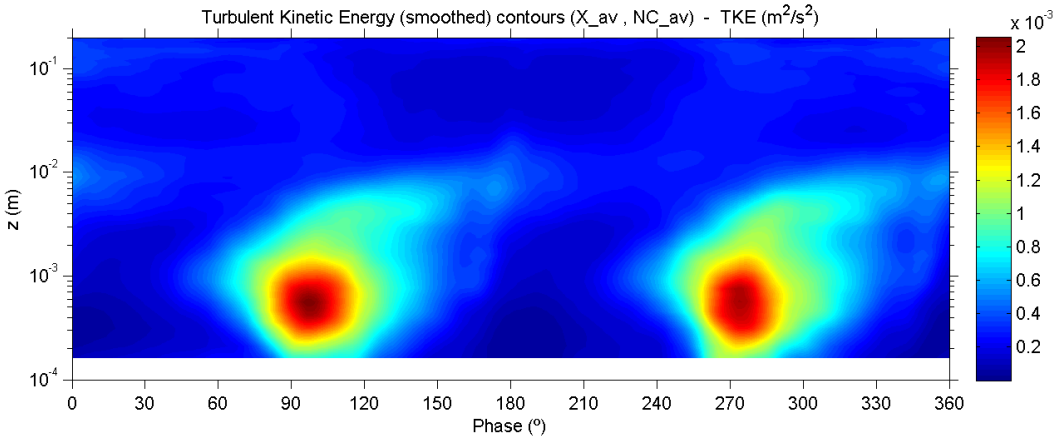


C.6.5.3 Evolution plots

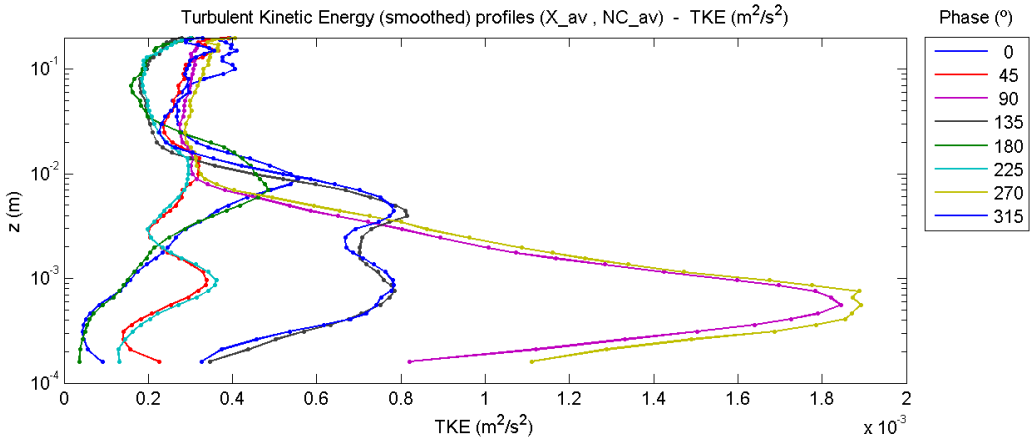


C.6.6 Turbulent kinetic energy

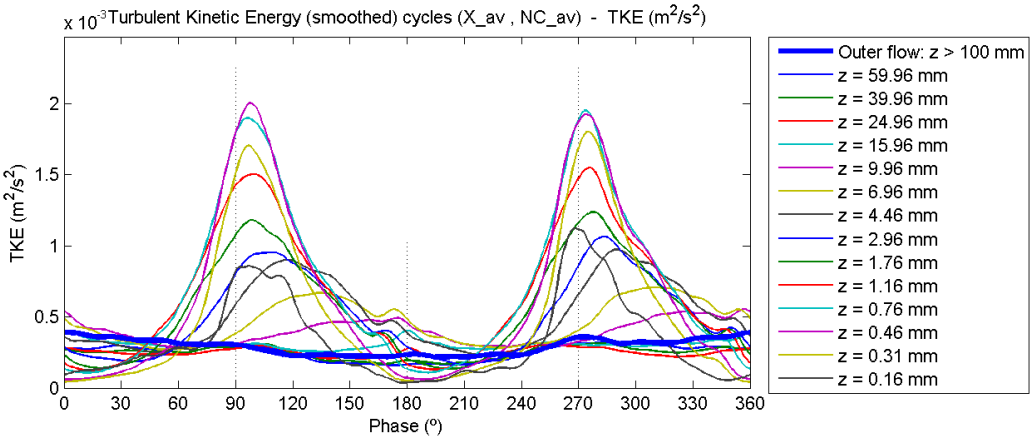
C.6.6.1 Contour plot



C.6.6.2 Profile plot

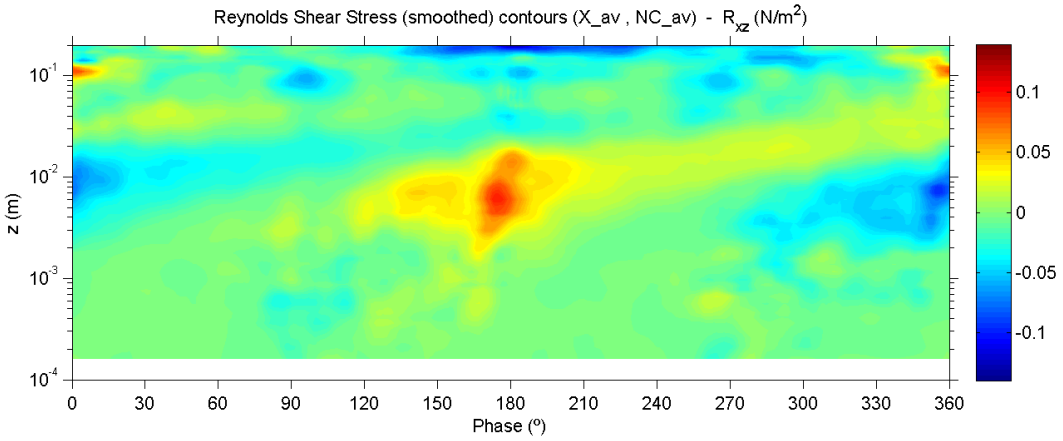


C.6.6.3 Evolution plot

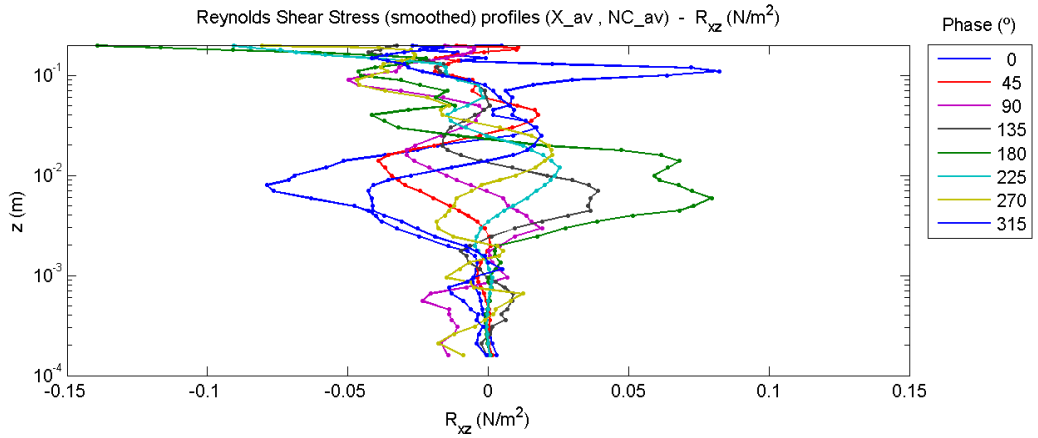


C.6.7 Turbulent shear stress (Reynolds shear stress)

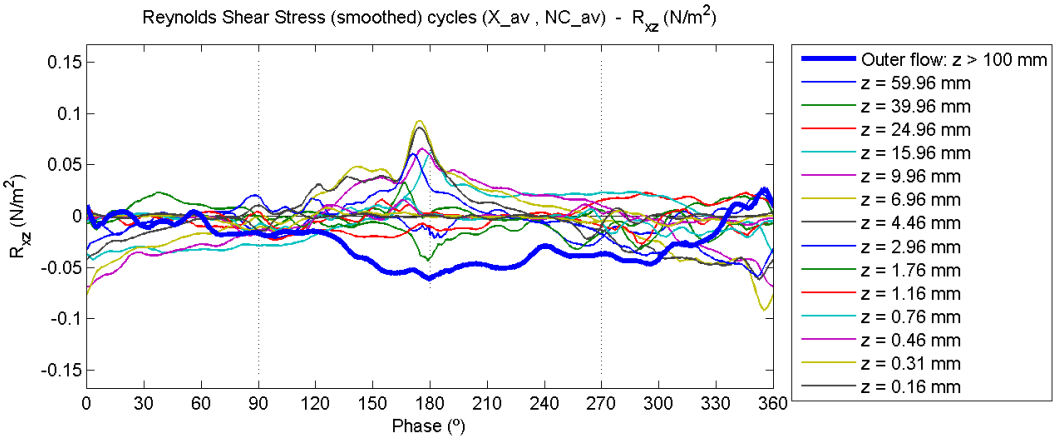
C.6.7.1 Contour plot



C.6.7.2 Profile plot

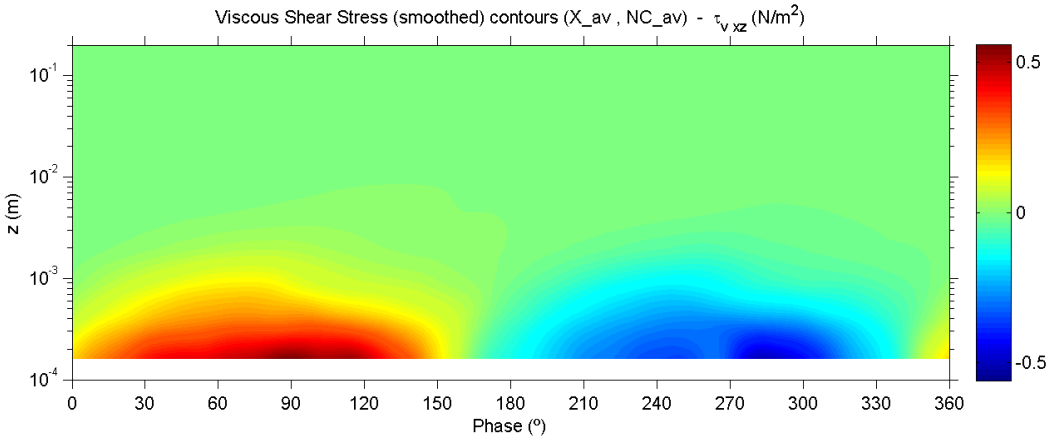


C.6.7.3 Evolution plot

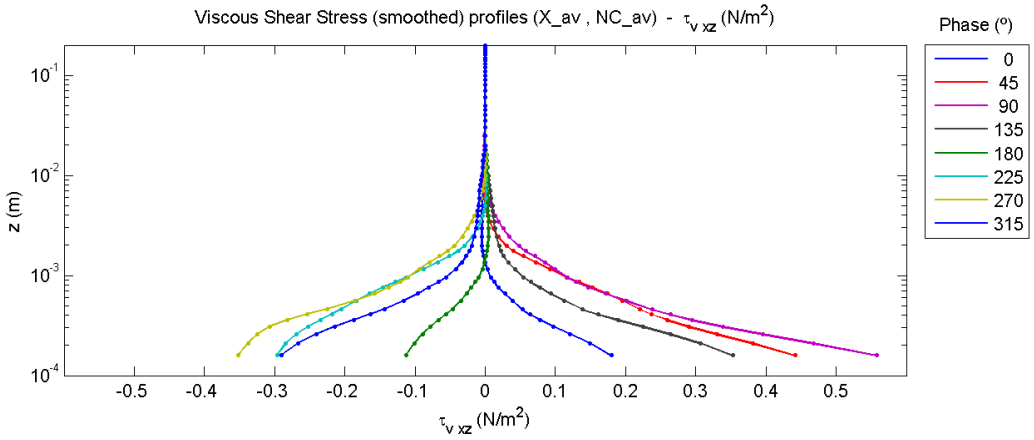


C.6.8 Viscous shear stress

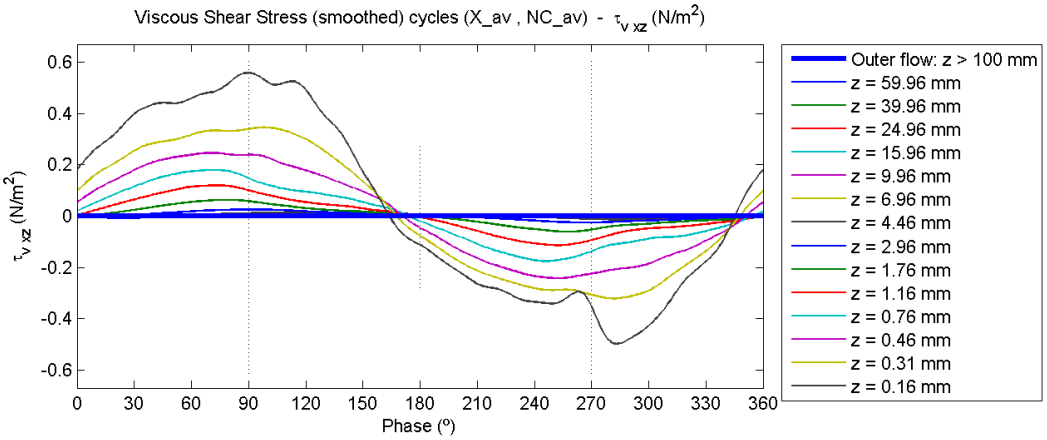
C.6.8.1 Contour plot



C.6.8.2 Profile plot

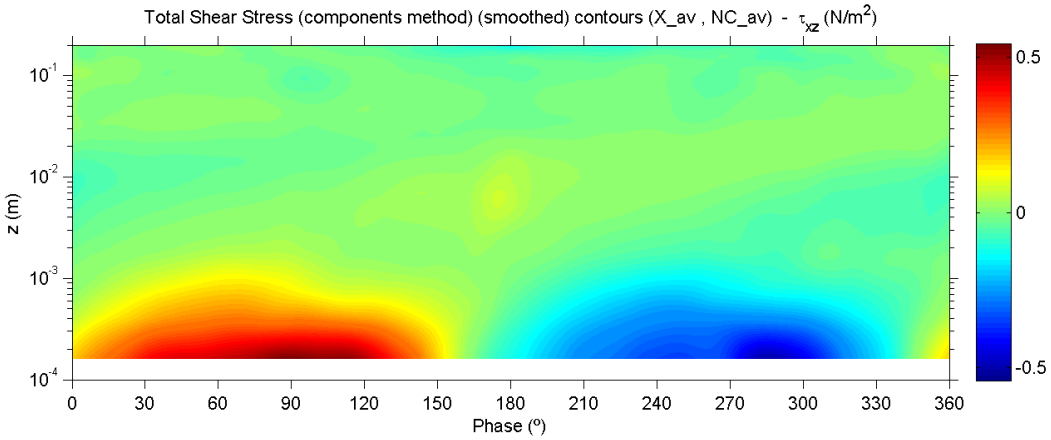


C.6.8.3 Evolution plot

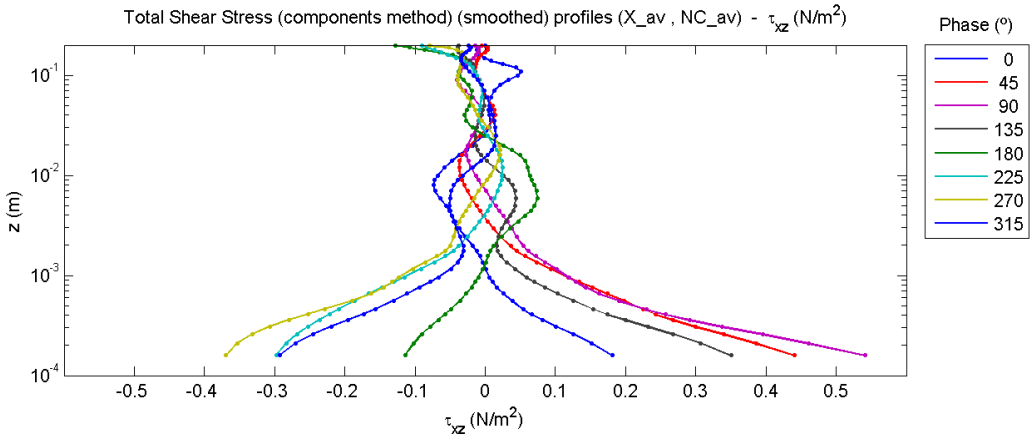


C.6.9 Total shear stress

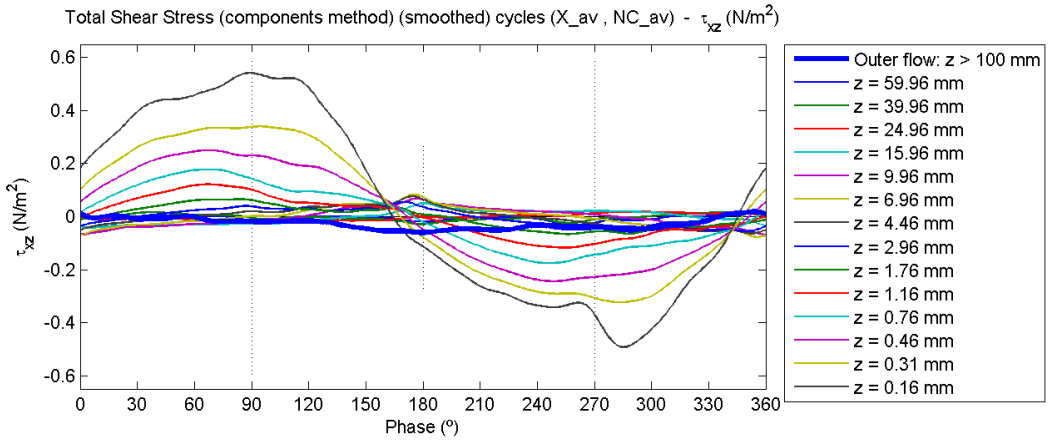
C.6.9.1 Contour plot



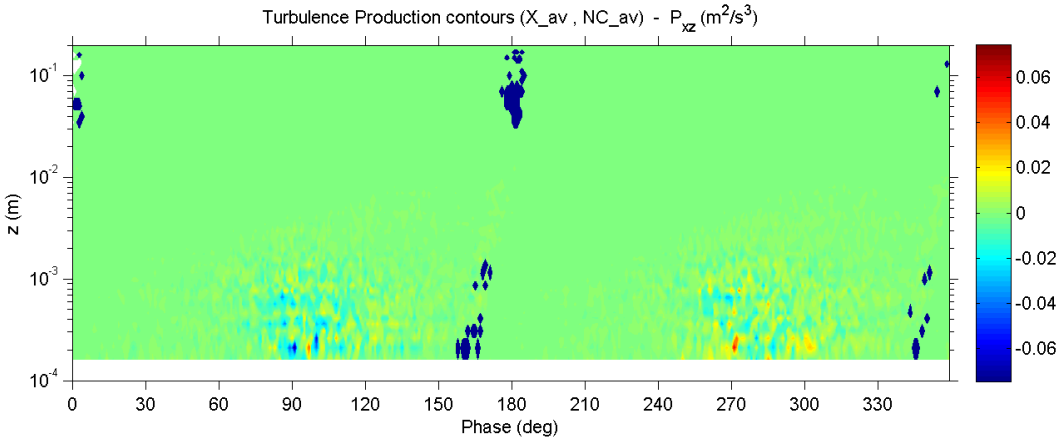
C.6.9.2 Profile plot



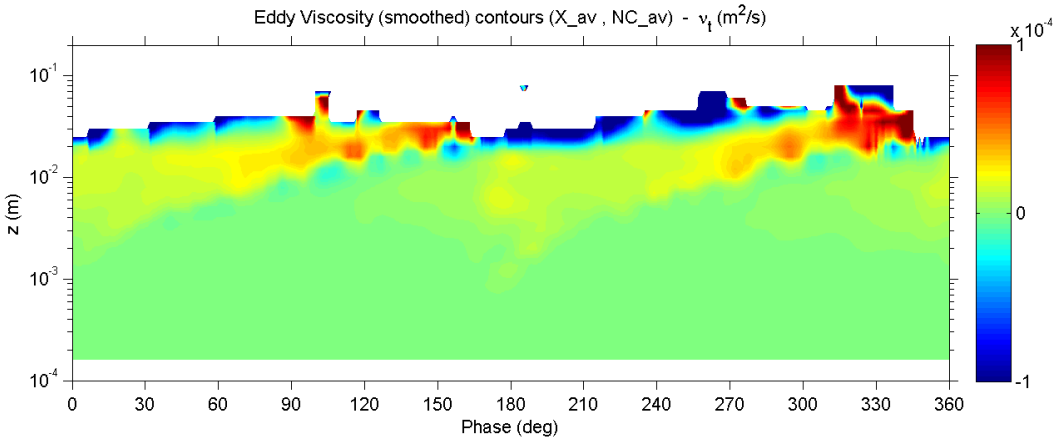
C.6.9.3 Evolution plot



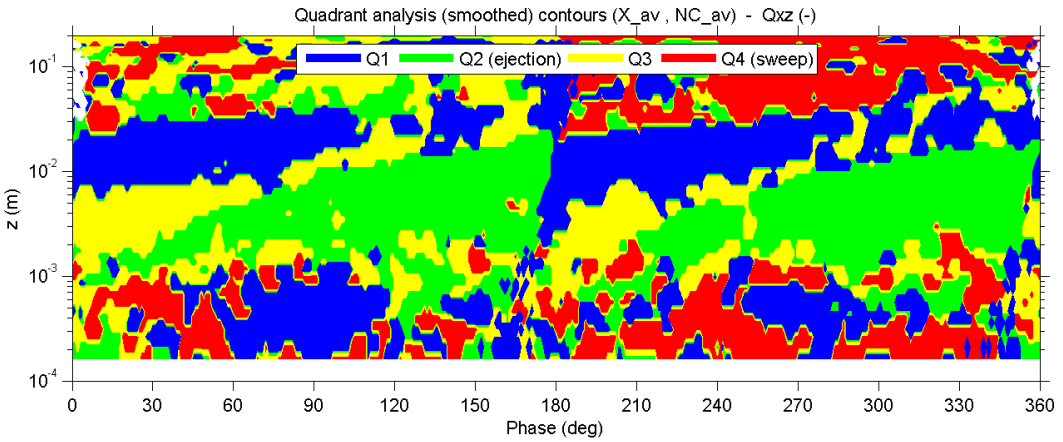
C.6.10 Turbulence production



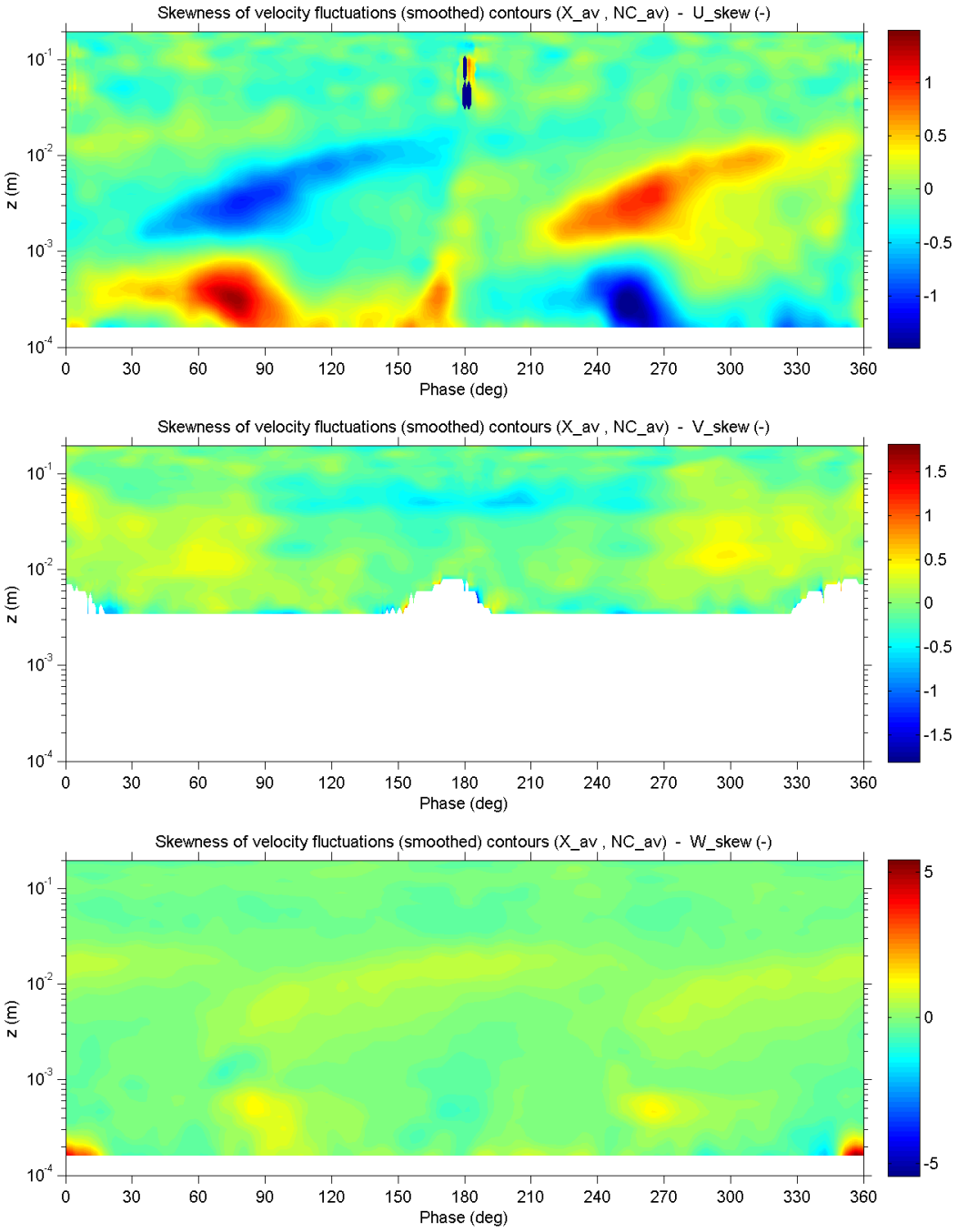
C.6.11 Turbulent viscosity (Eddy viscosity)



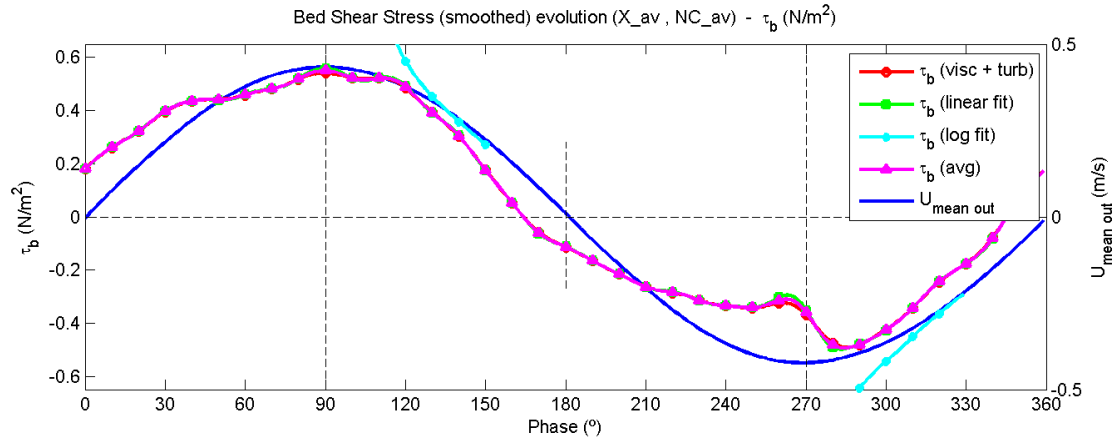
C.6.12 Quadrant analysis



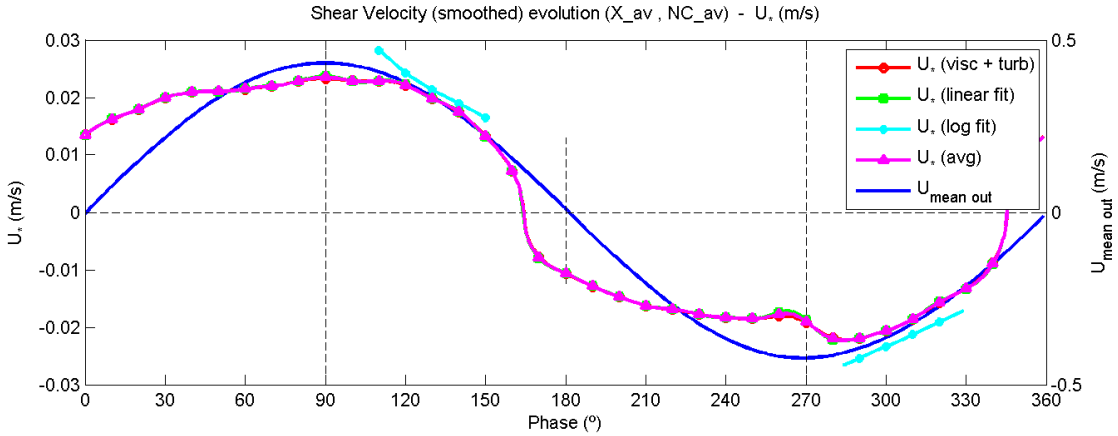
C.6.13 Skewness of velocity fluctuations



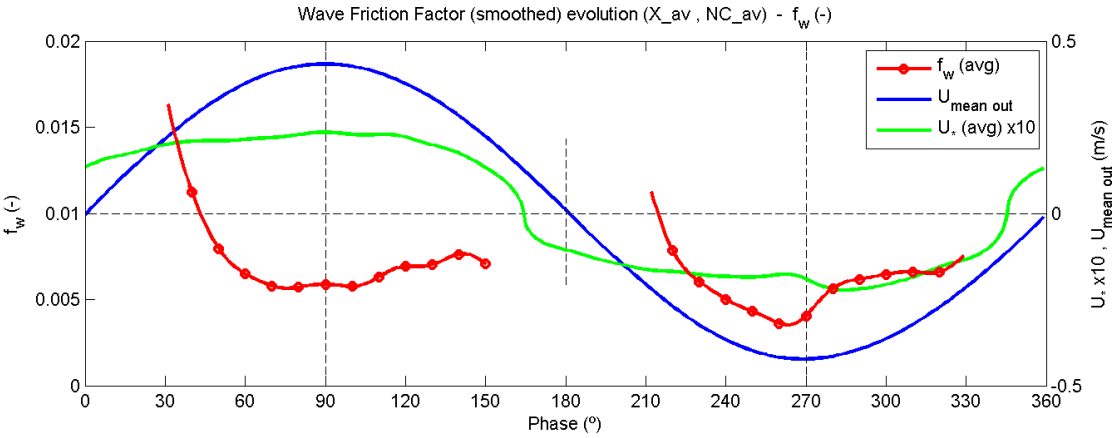
C.6.14 Bed shear stress



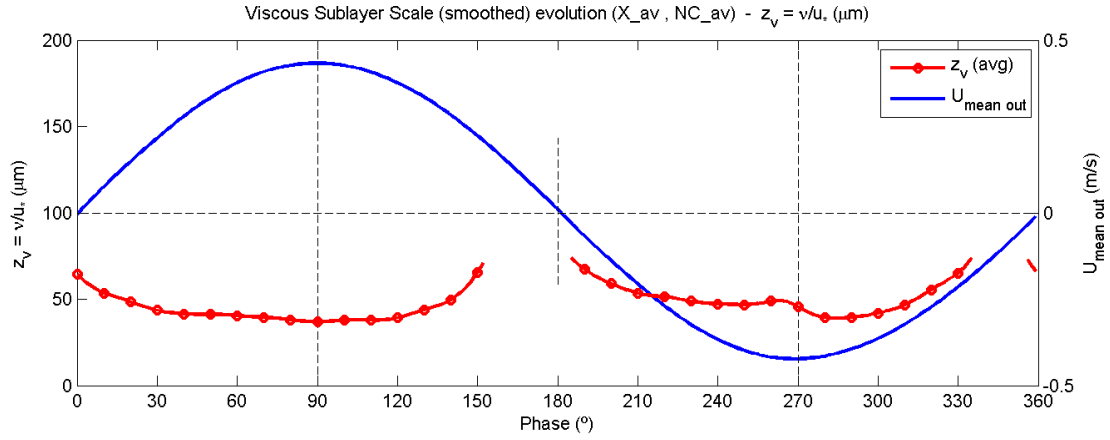
C.6.15 Shear velocity



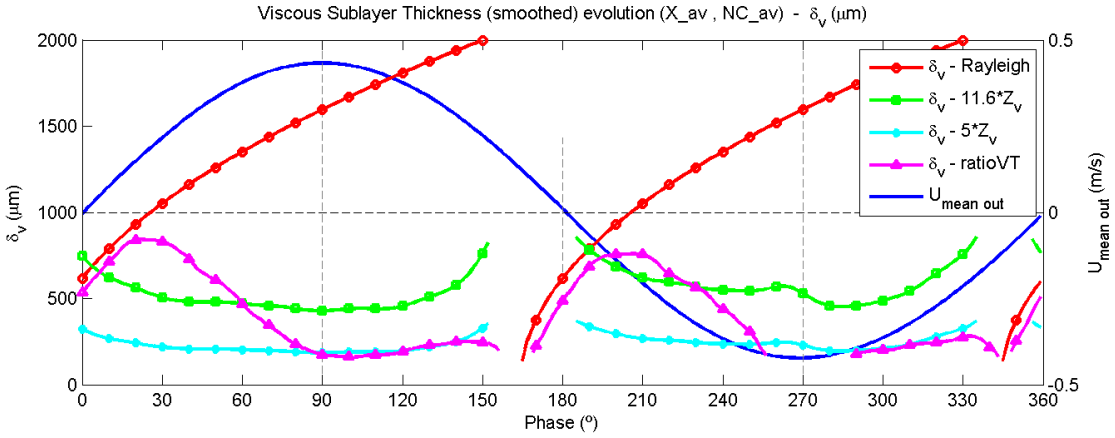
C.6.16 Wave friction factor



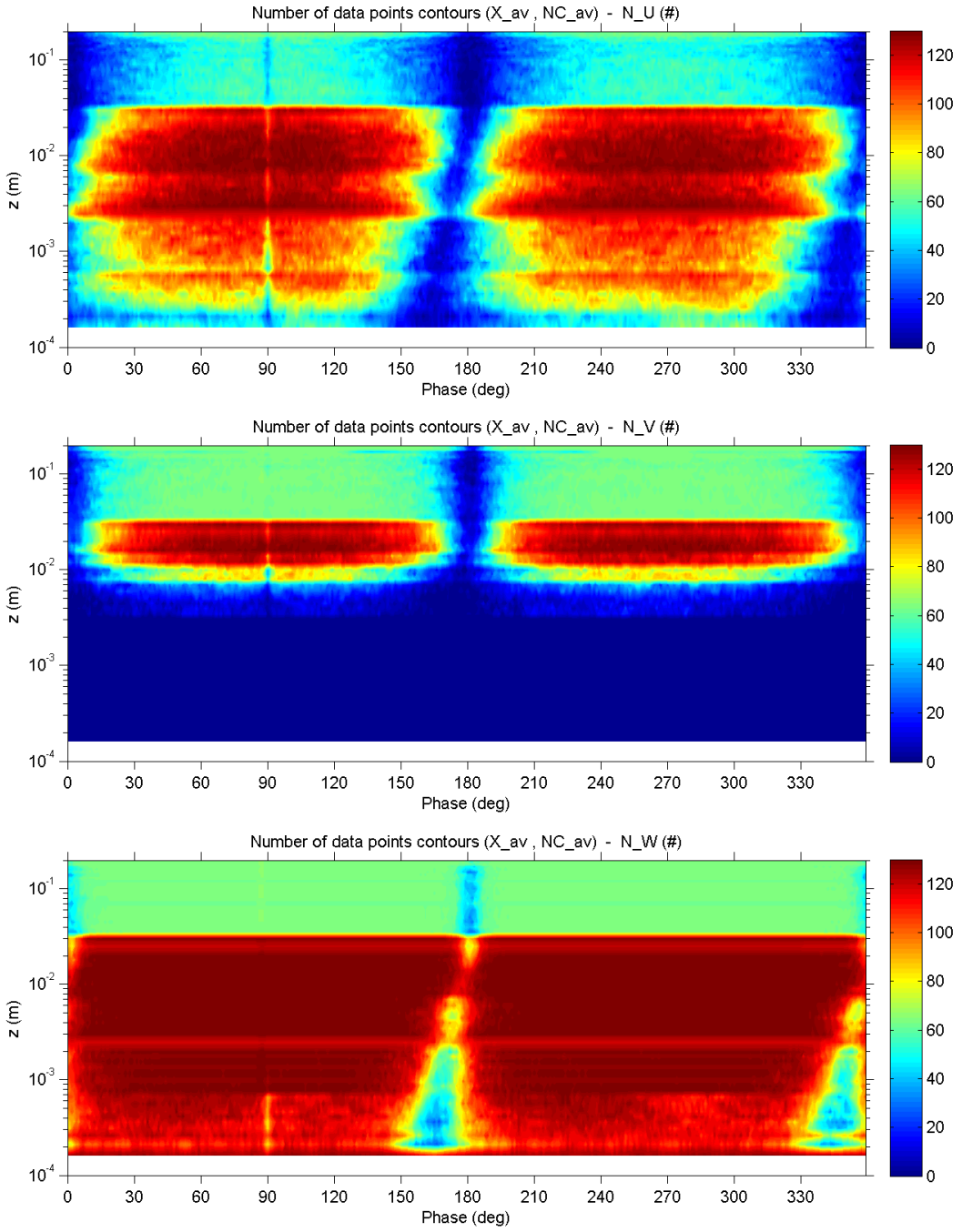
C.6.17 Viscous length scale



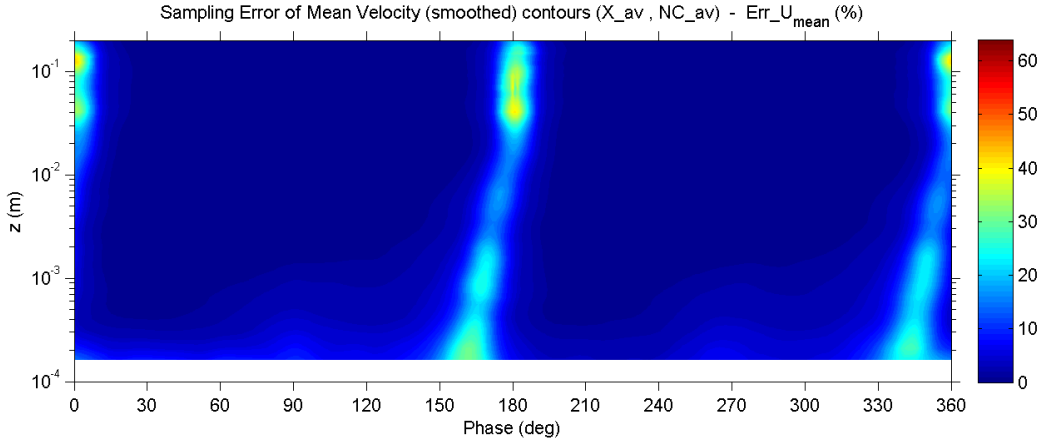
C.6.18 Viscous sublayer thickness



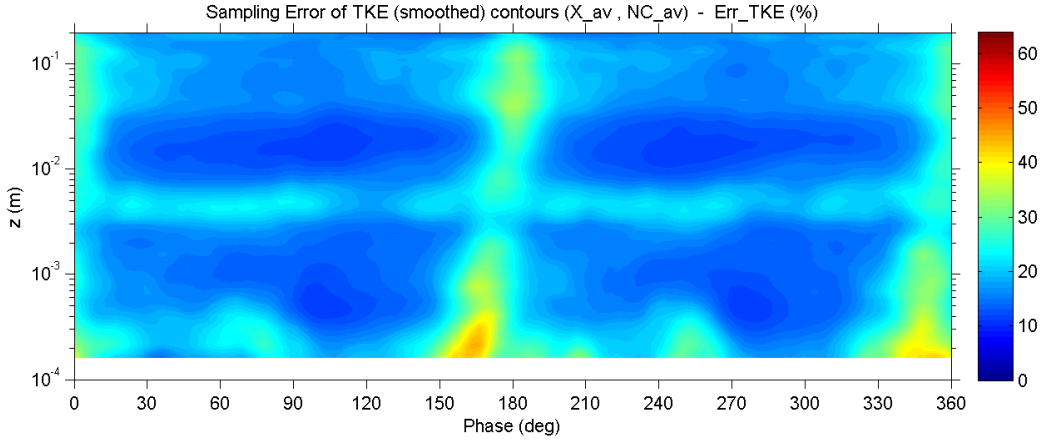
C.6.19 Number of valid data points



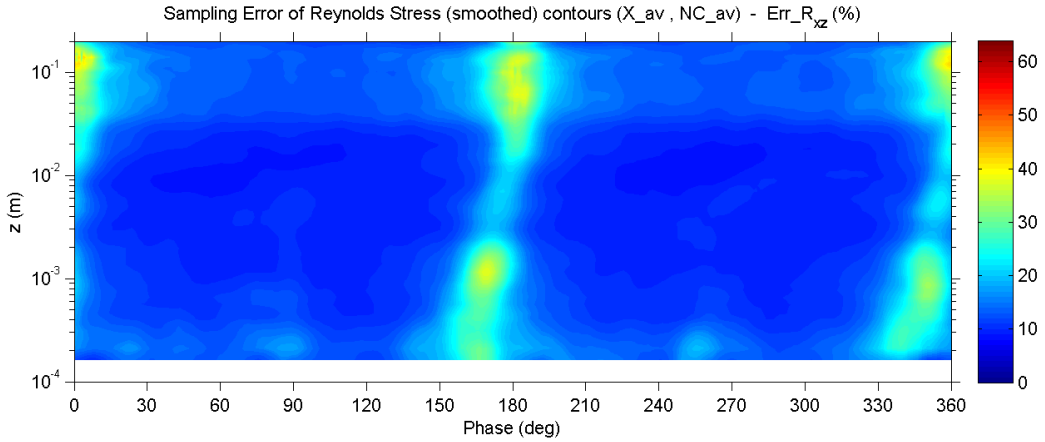
C.6.20 Percent error of mean velocity



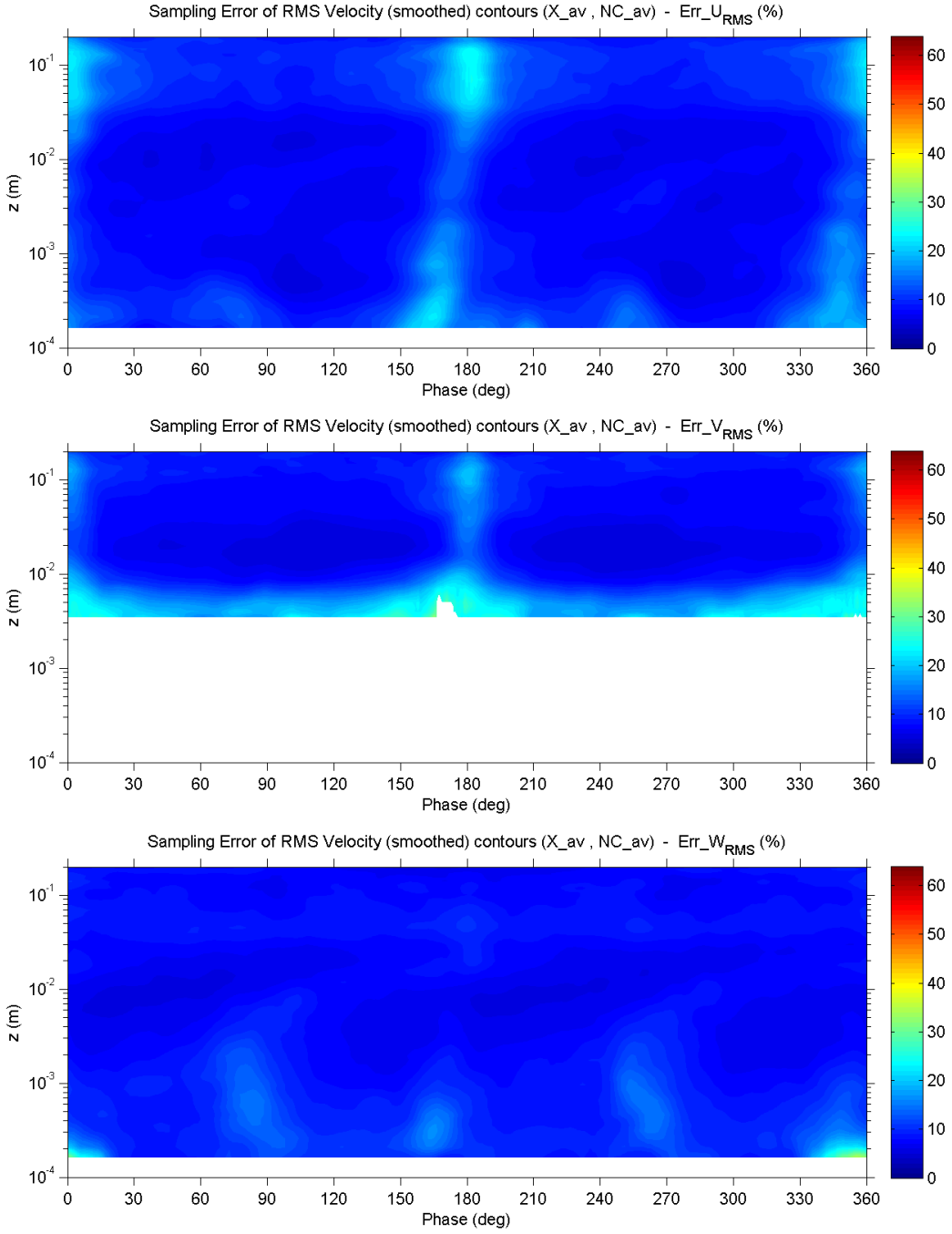
C.6.21 Percent error of turbulent kinetic energy



C.6.22 Percent error of Reynolds shear stress



C.6.23 Percent error of RMS velocity fluctuations



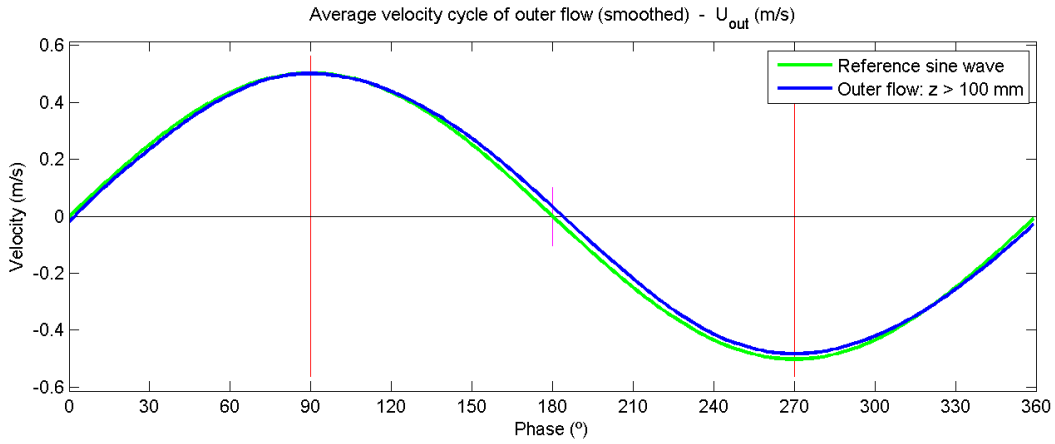
C.7 Experiment no. 7

C.7.1 Main parameters

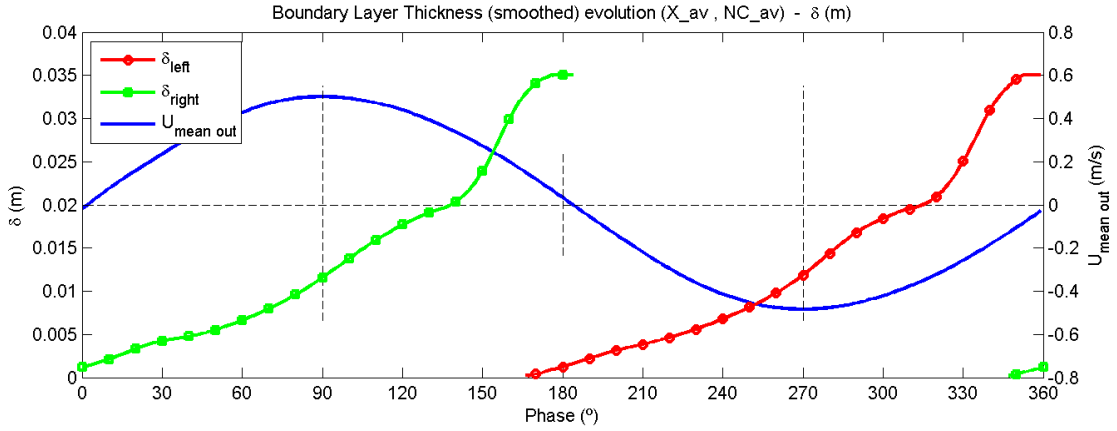
Exp no.	*Temp* (ºC)	ρ (kg/m^3)	ν (m^2/s)	*T* (s)	*2a* (m)	U_{out_max} (m/s)	Re_w (-)	N_{max} (cycles)
7	26.5	996.68	8.77E-07	10	1.566	0.492	4.4E+05	130

Exp no.	U_{*max} (m/s)	τ_{b_max} (N/m^2)	$\Delta\phi$ (deg)	f_{w_ref} (-)	Z_{v_min} (mm)	δ_{v_90} (mm)	δ_{90} (mm)	δ_{top} (mm)
7	0.024	0.58	-7.0	0.0048	0.036	-	11.7	35

C.7.2 Outer flow velocity

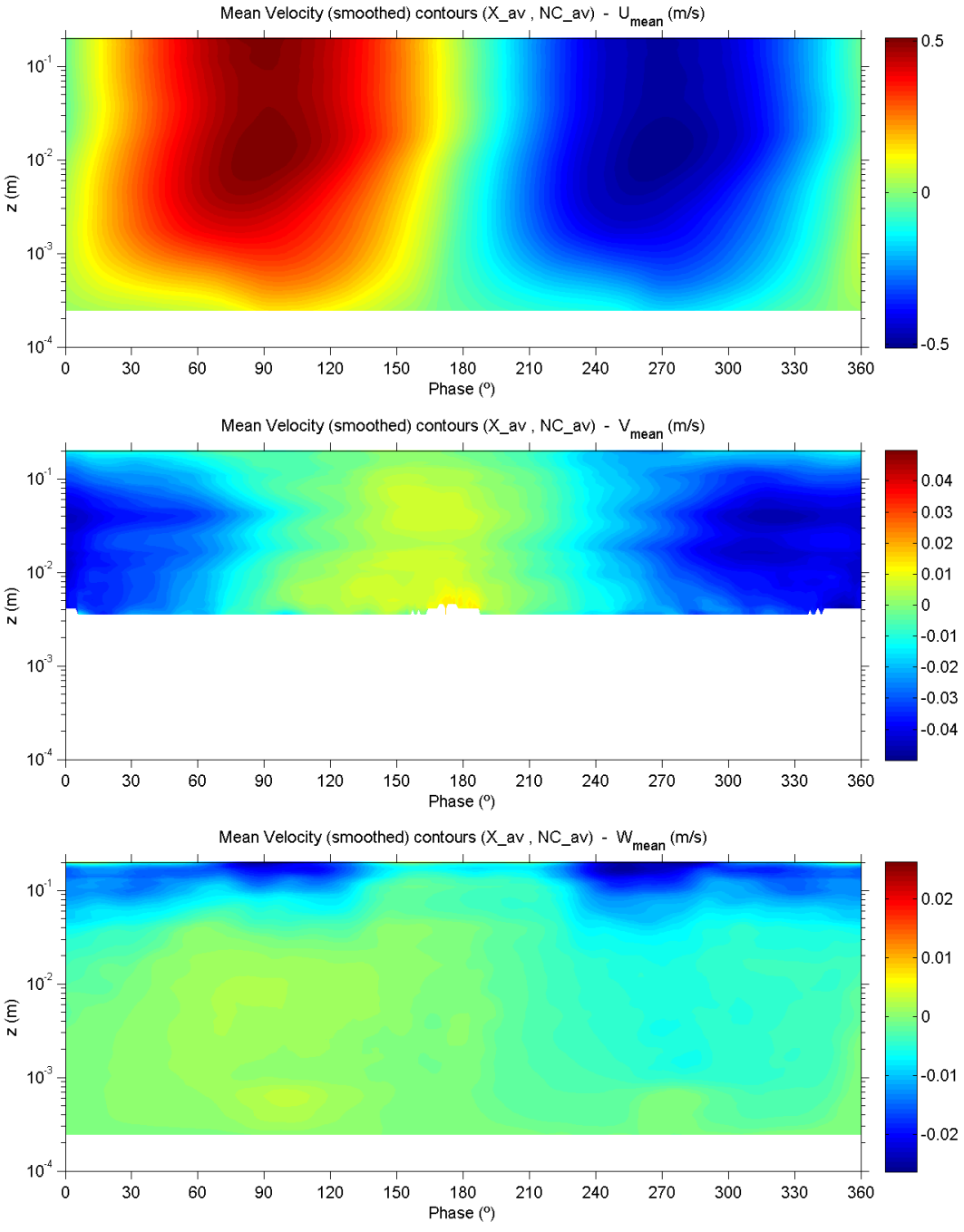


C.7.3 Boundary layer thickness

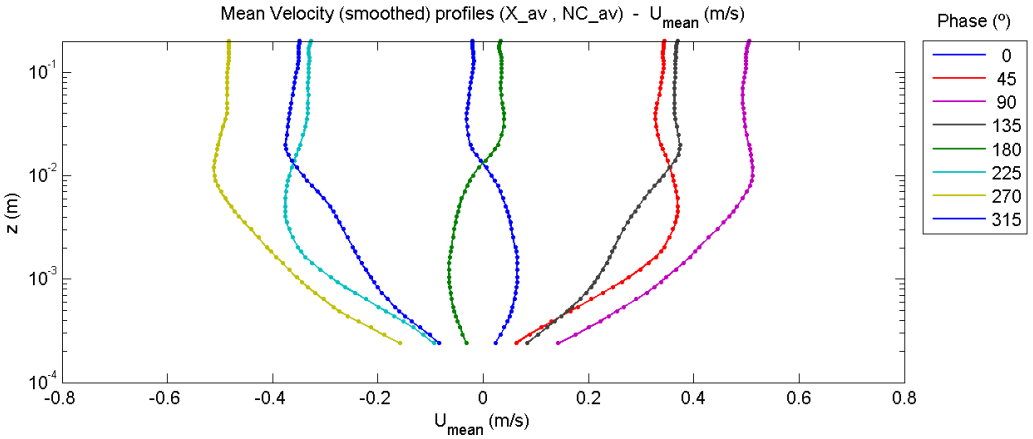


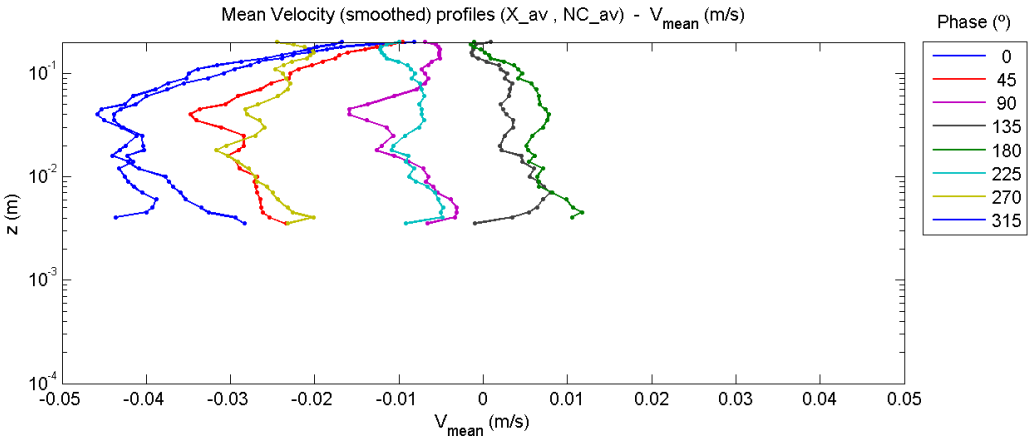
C.7.4 Mean velocities

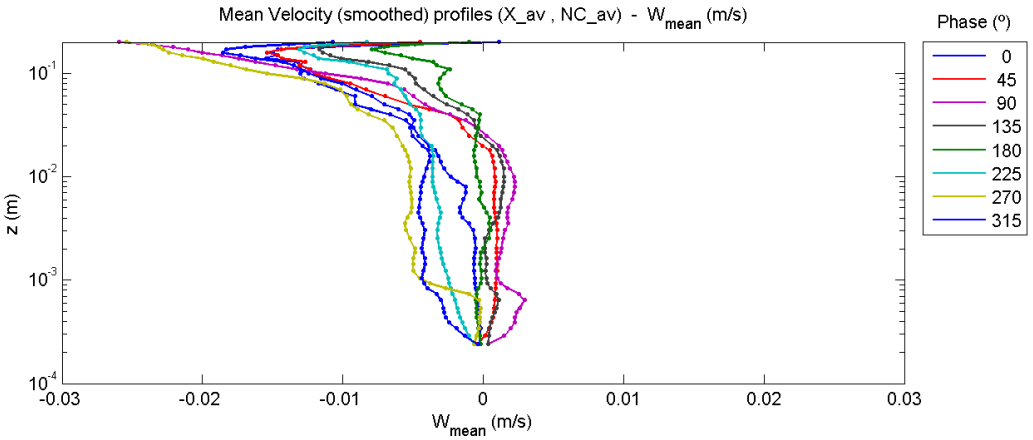
C.7.4.1 Contour plots



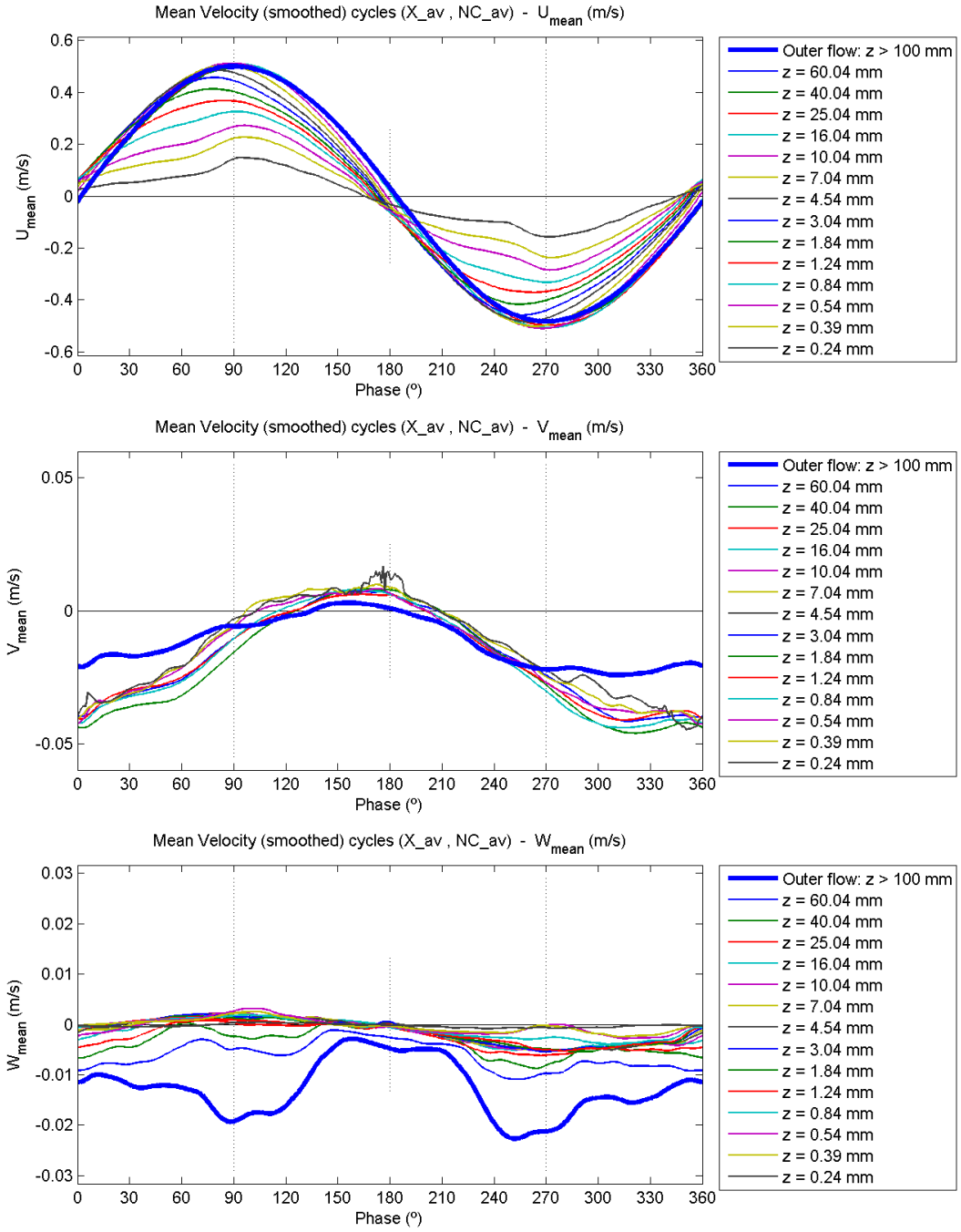
C.7.4.2 Profile plots





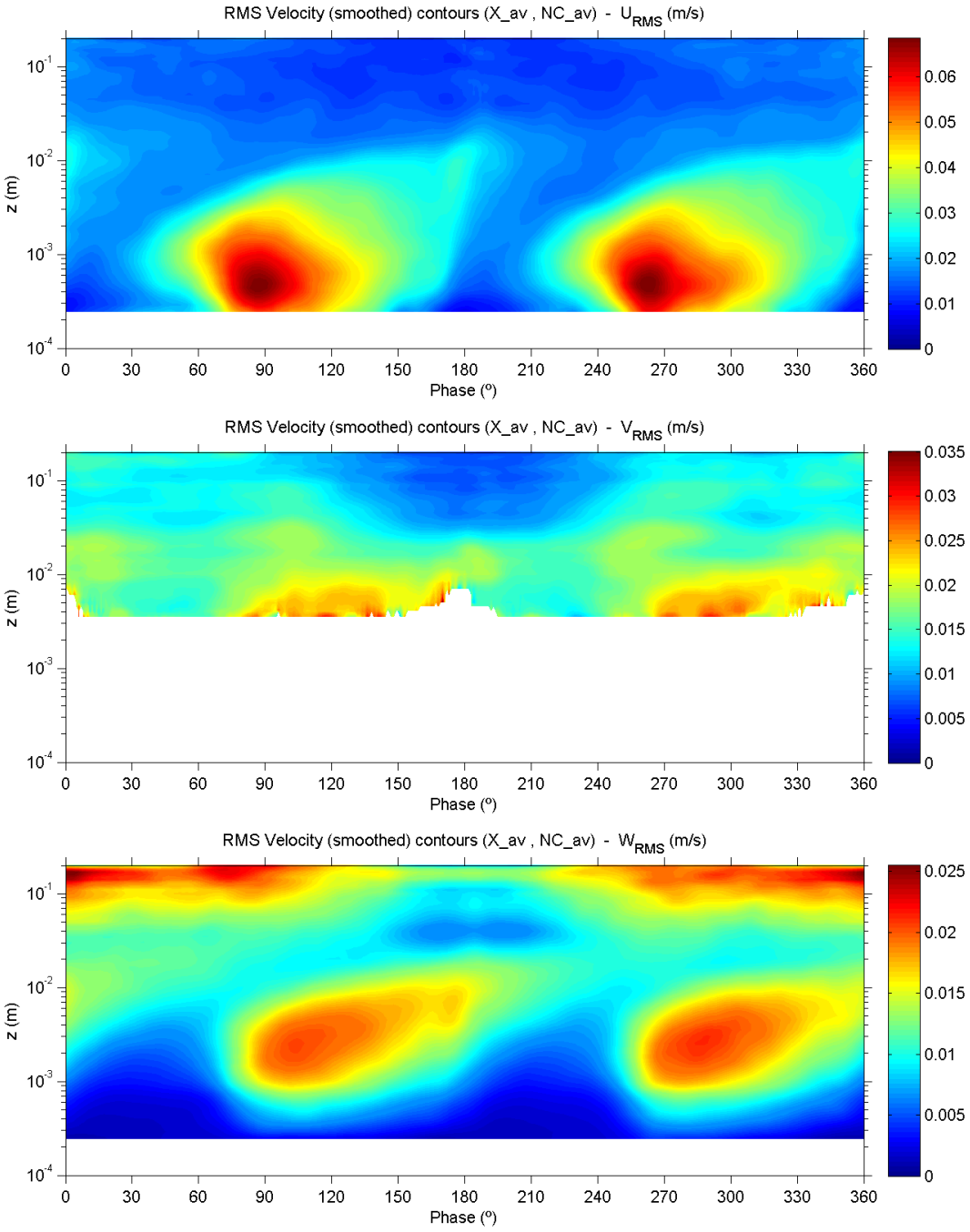


C.7.4.3 Evolution plots

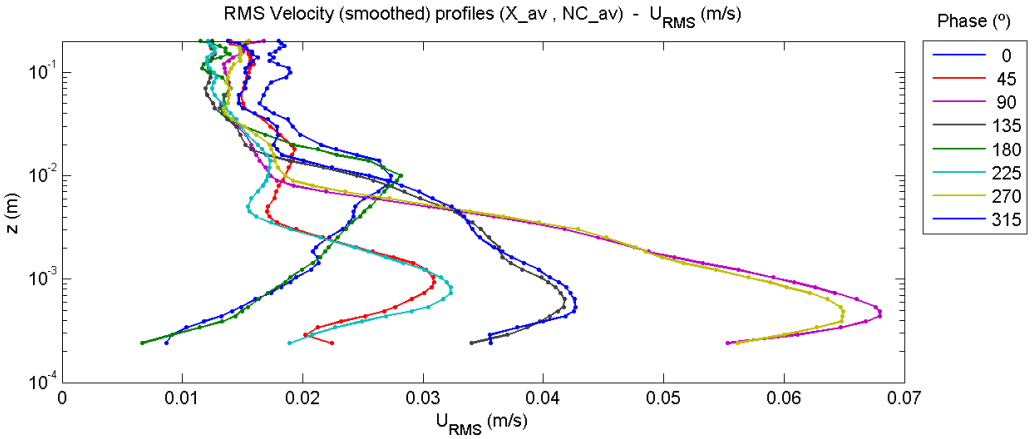


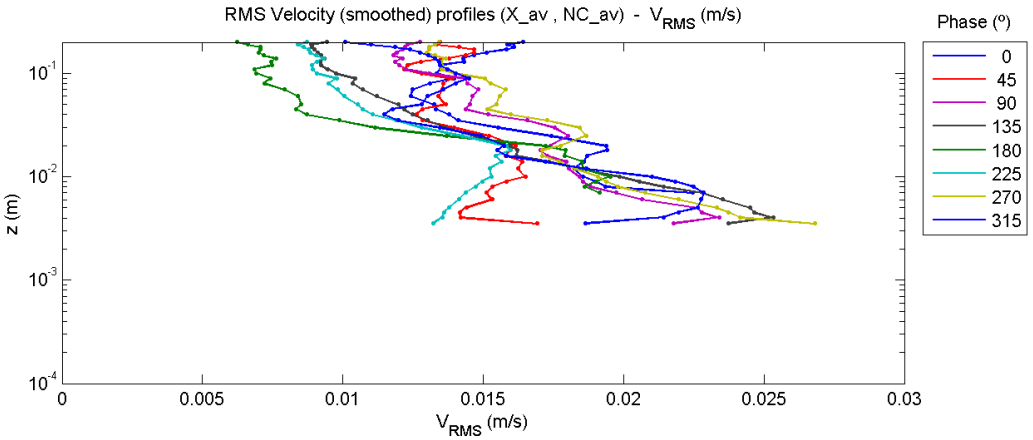
C.7.5 RMS velocity fluctuations

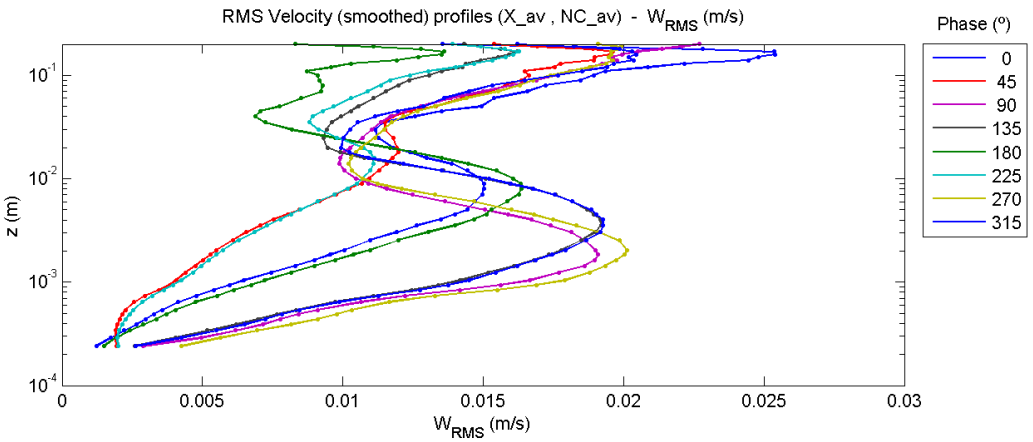
C.7.5.1 Contour plots



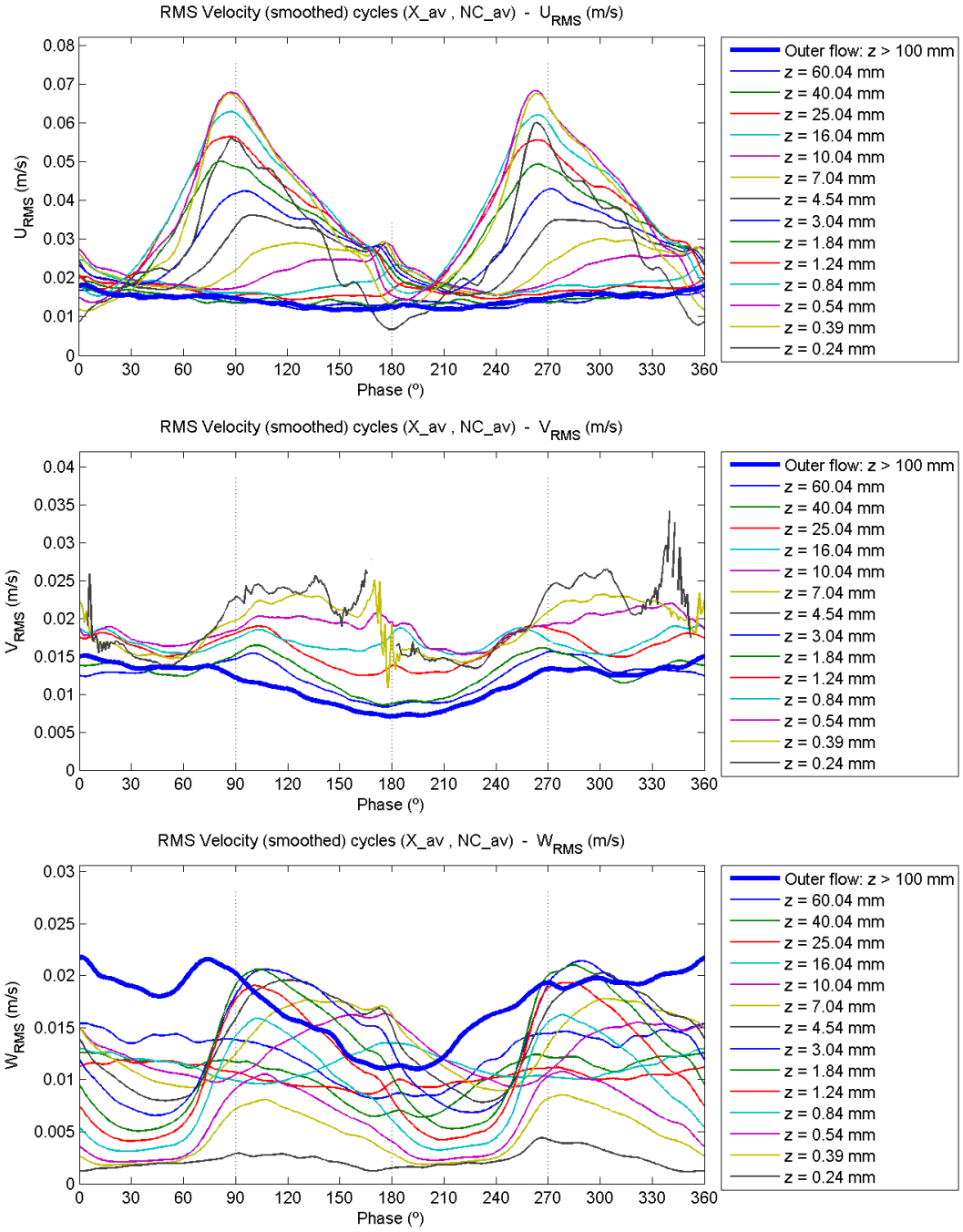
C.7.5.2 Profile plots





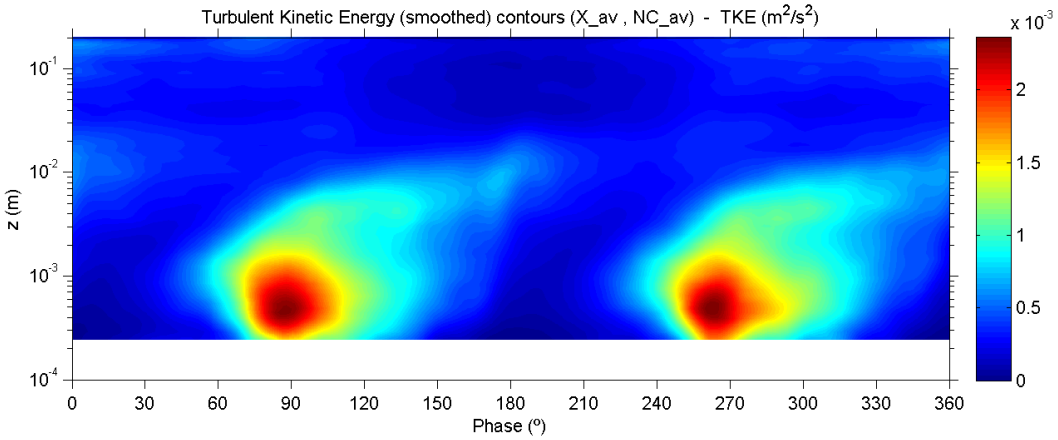


C.7.5.3 Evolution plots

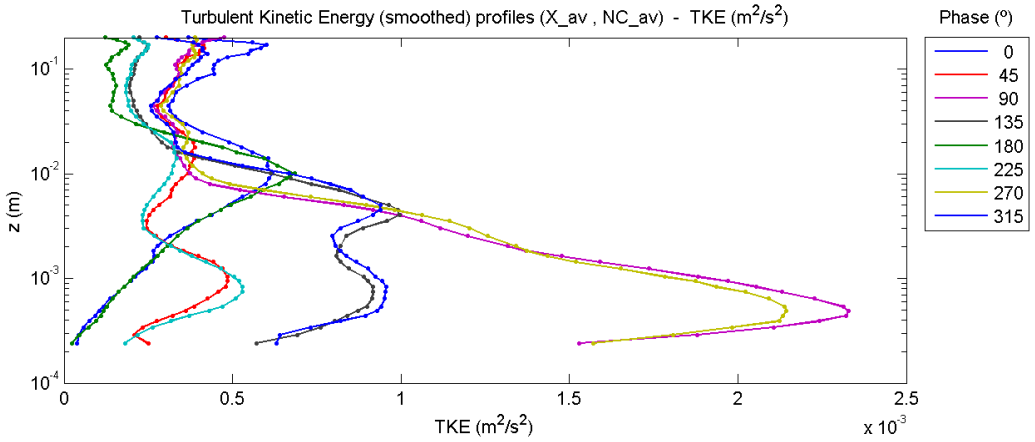


C.7.6 Turbulent kinetic energy

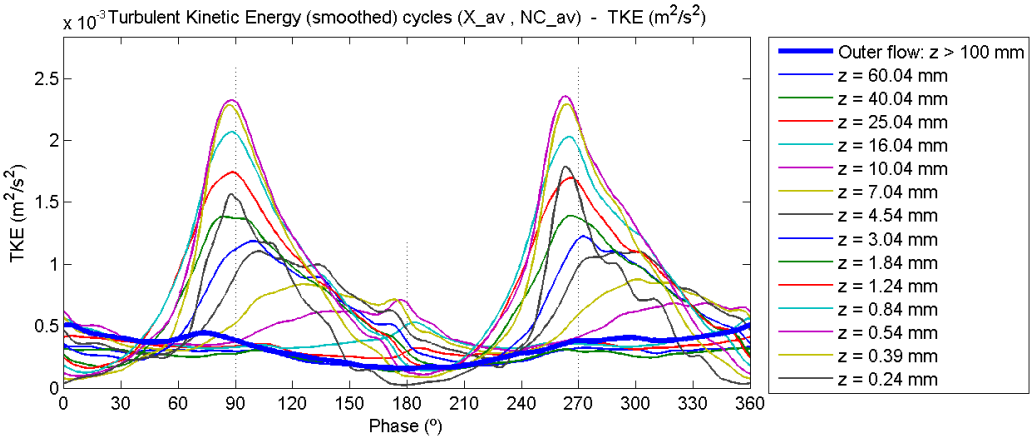
C.7.6.1 Contour plot



C.7.6.2 Profile plot

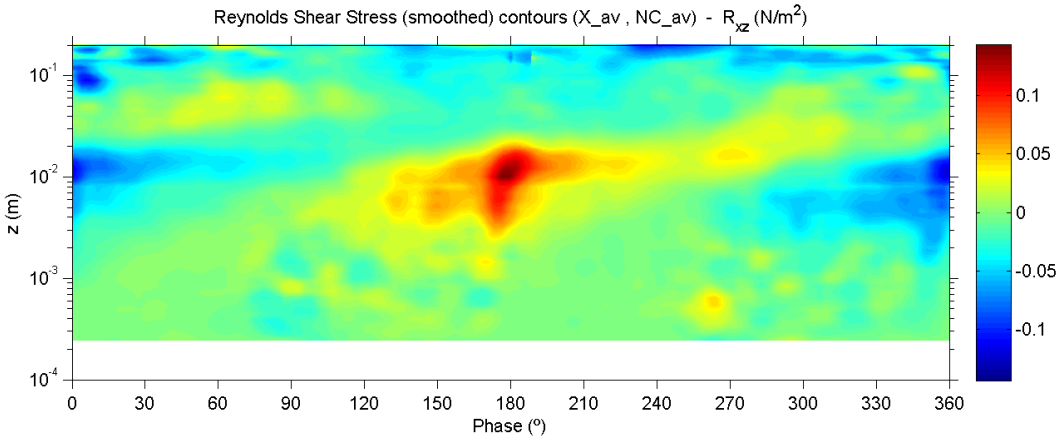


C.7.6.3 Evolution plot

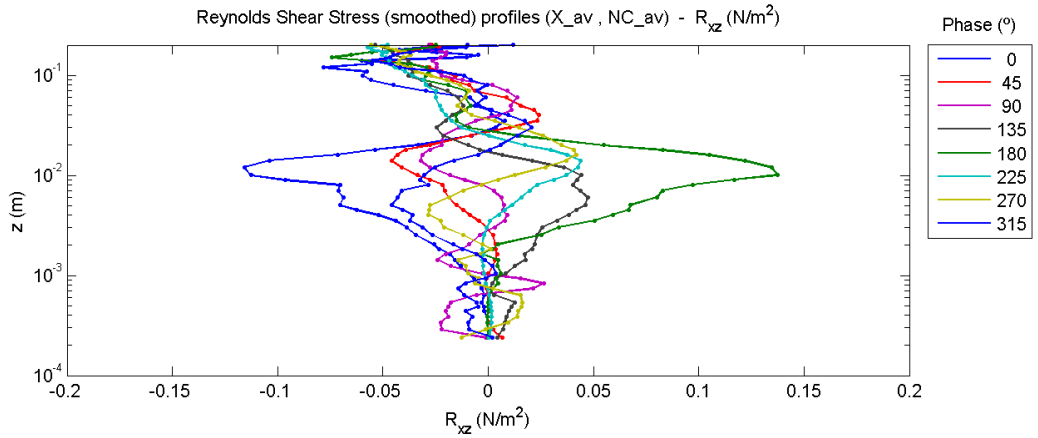


C.7.7 Turbulent shear stress (Reynolds shear stress)

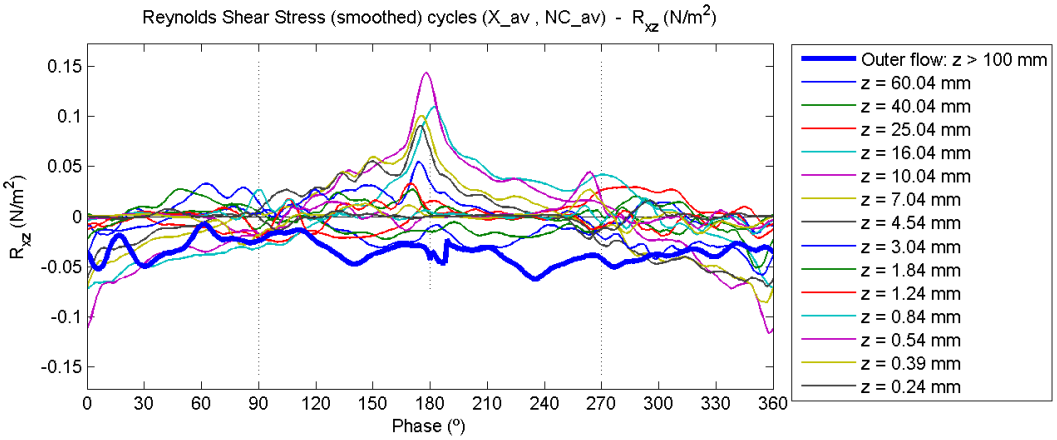
C.7.7.1 Contour plot



C.7.7.2 Profile plot

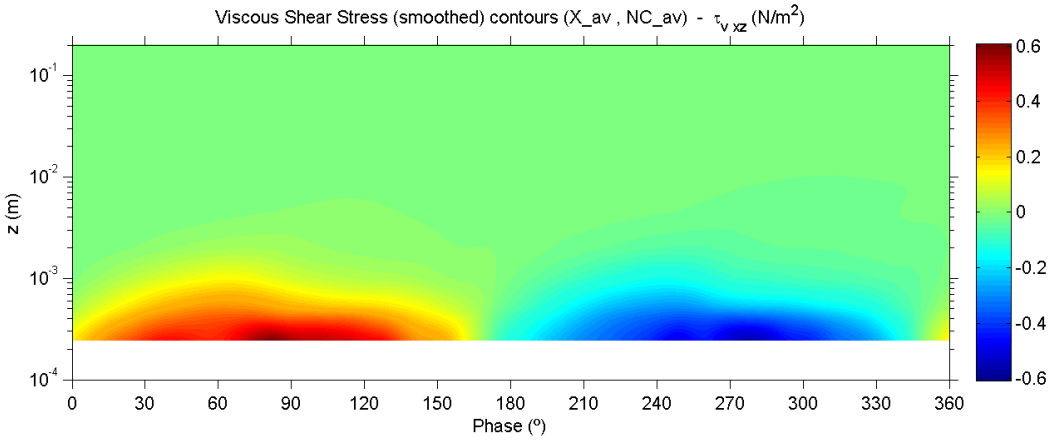


C.7.7.3 Evolution plot

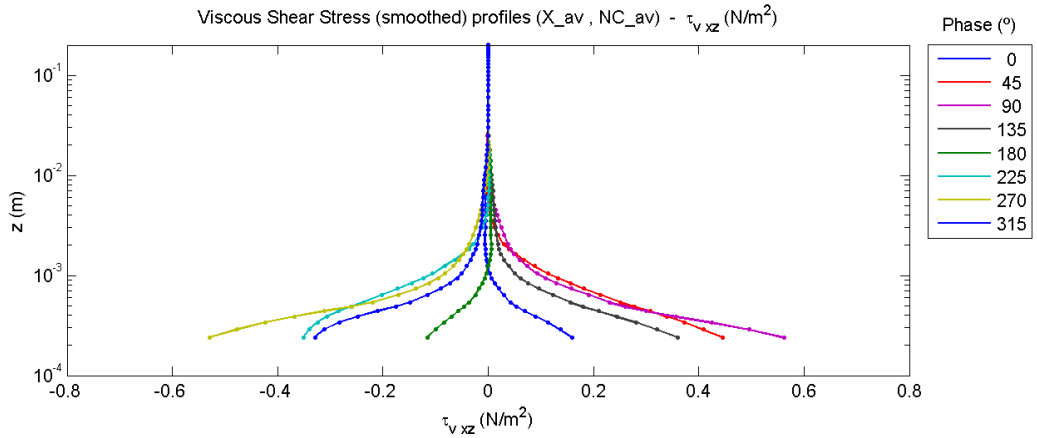


C.7.8 Viscous shear stress

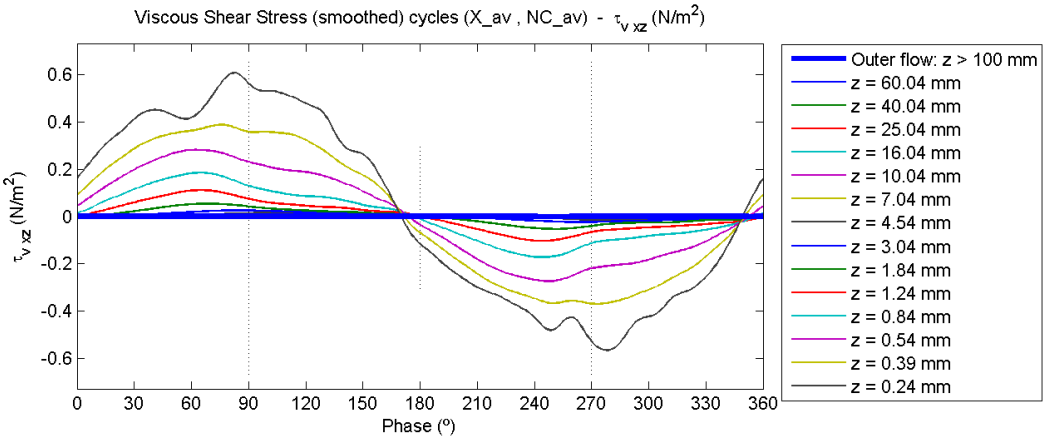
C.7.8.1 Contour plot



C.7.8.2 Profile plot

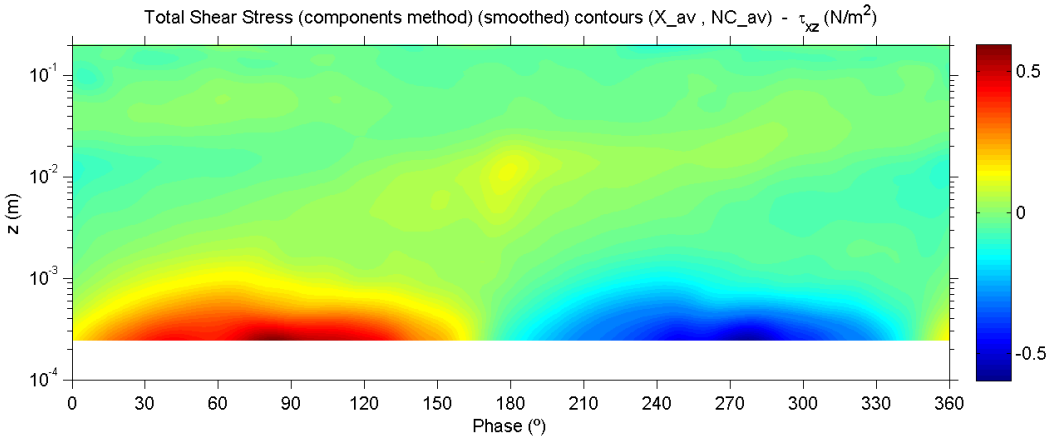


C.7.8.3 Evolution plot

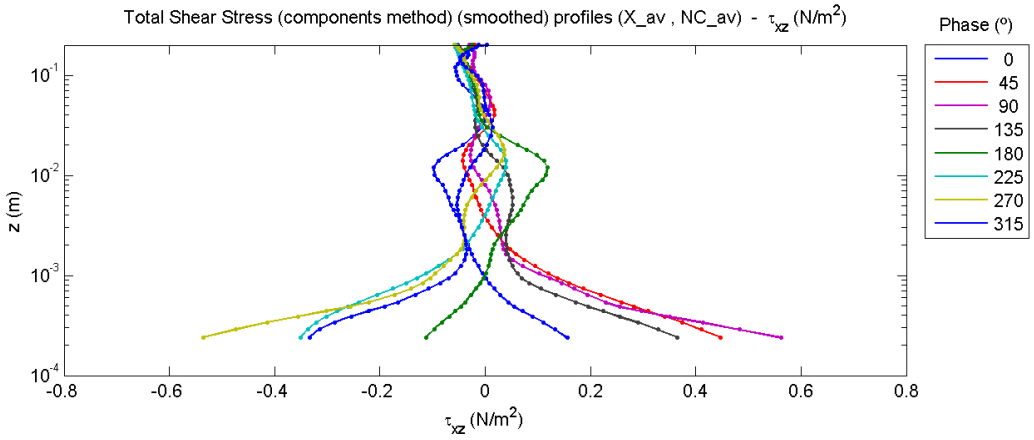


C.7.9 Total shear stress

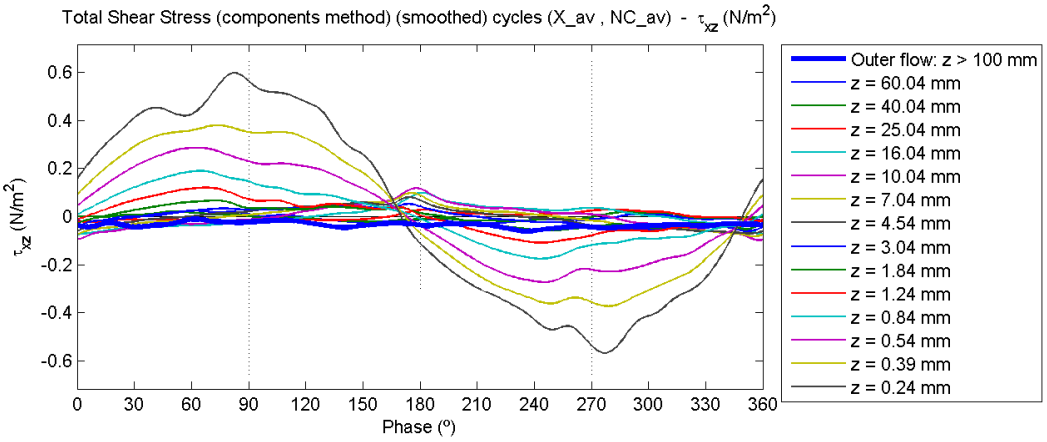
C.7.9.1 Contour plot



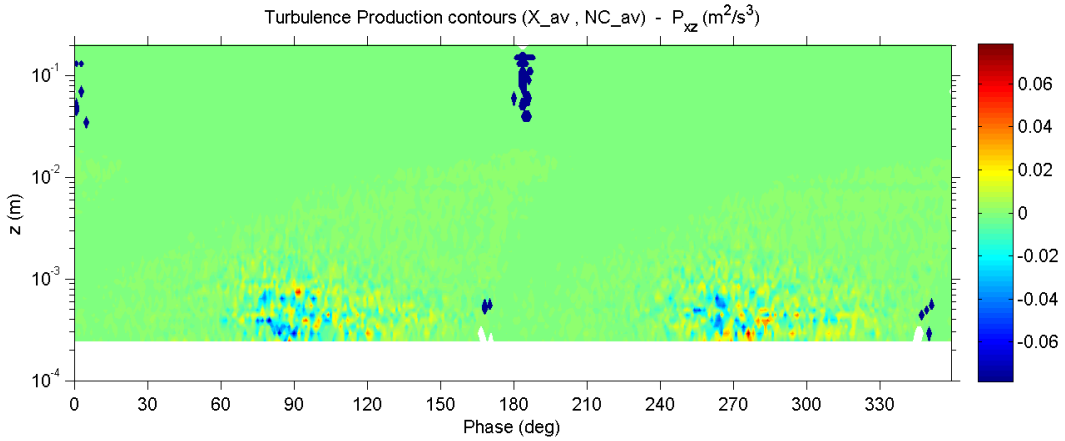
C.7.9.2 Profile plot



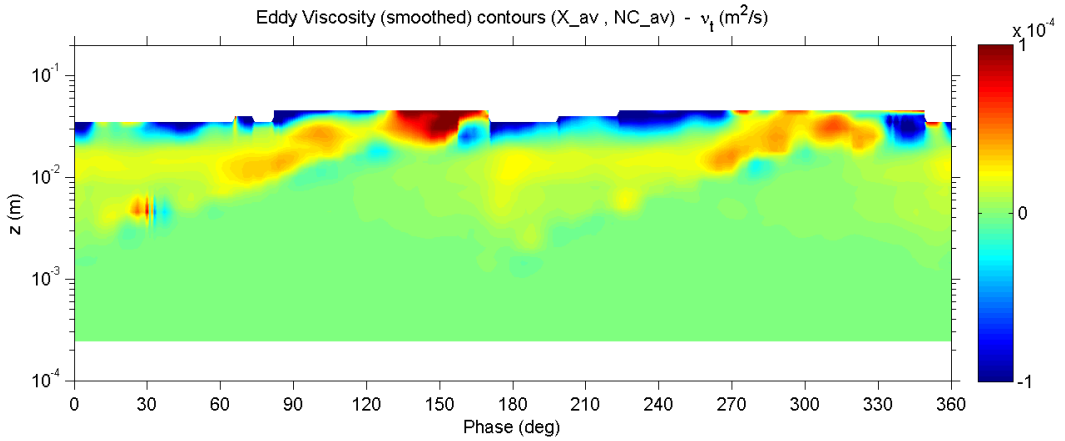
C.7.9.3 Evolution plot



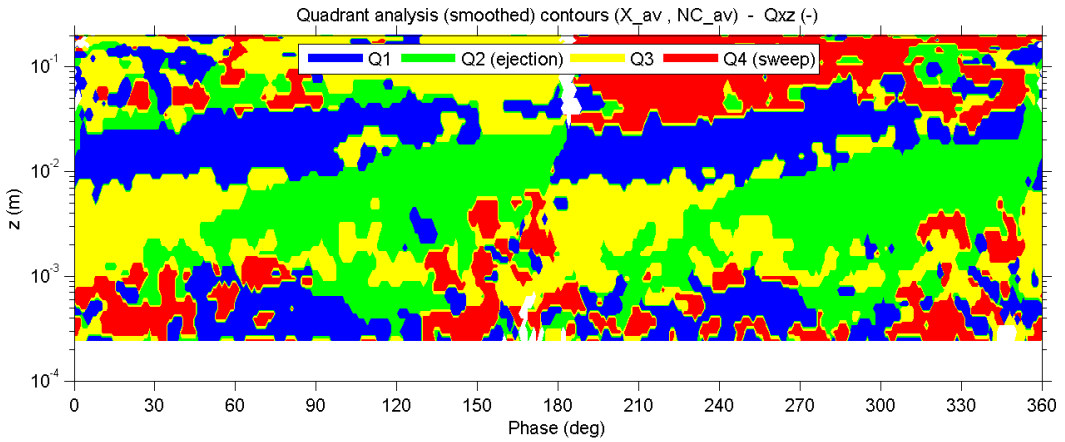
C.7.10 Turbulence production



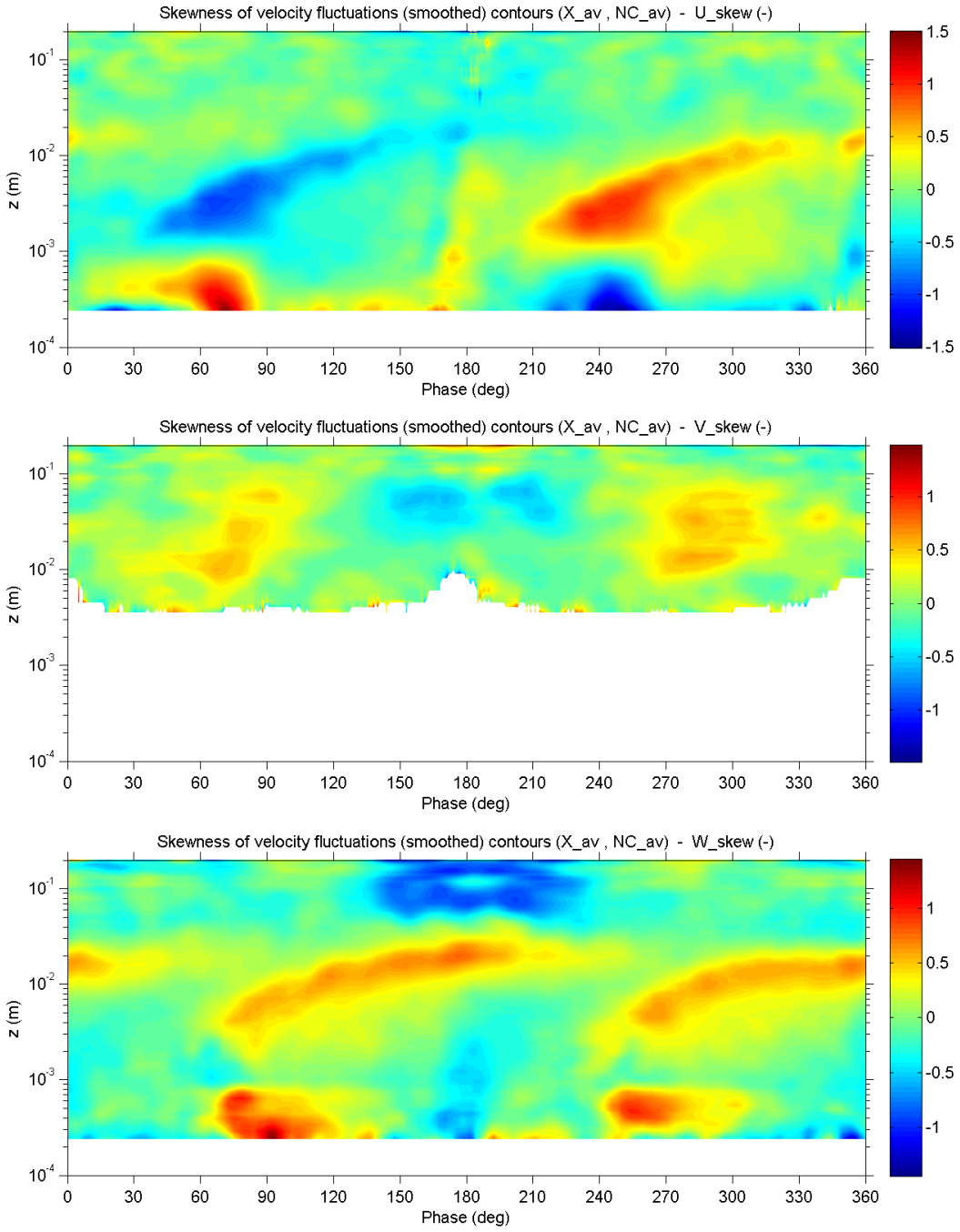
C.7.11 Turbulent viscosity (Eddy viscosity)



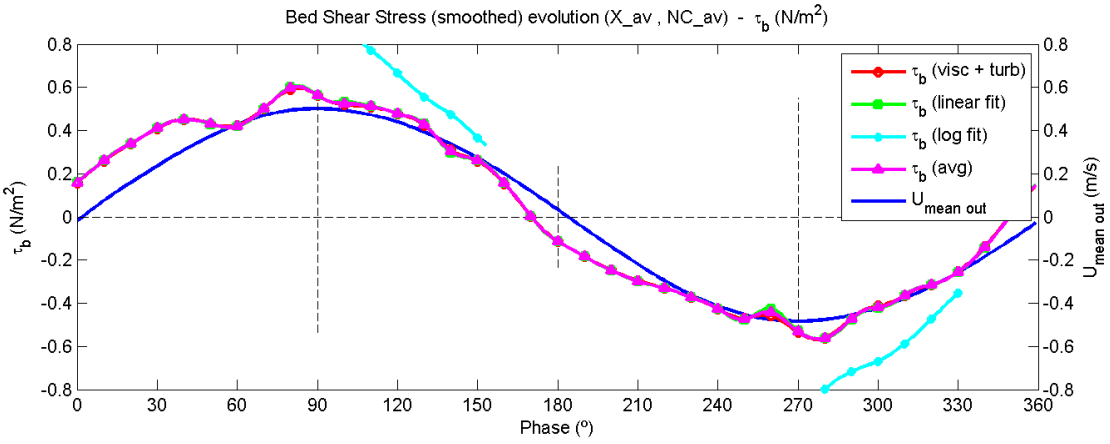
C.7.12 Quadrant analysis



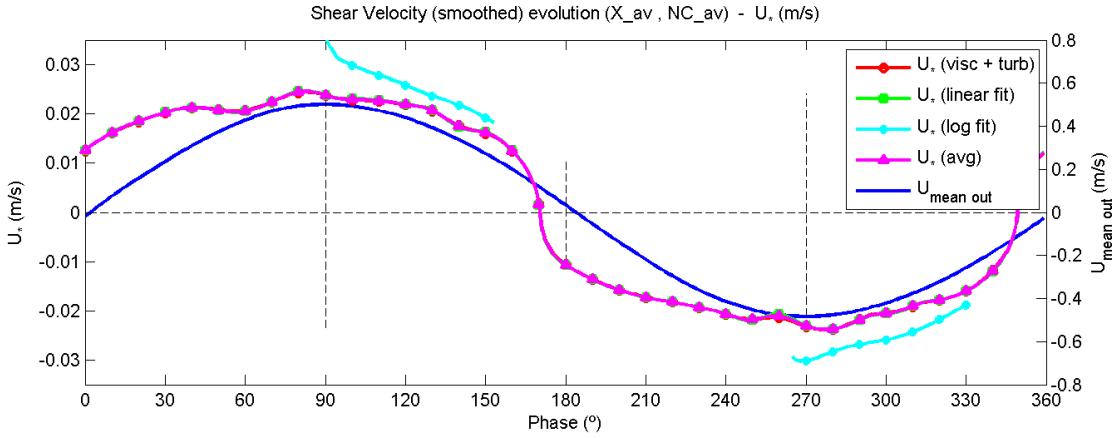
C.7.13 Skewness of velocity fluctuations



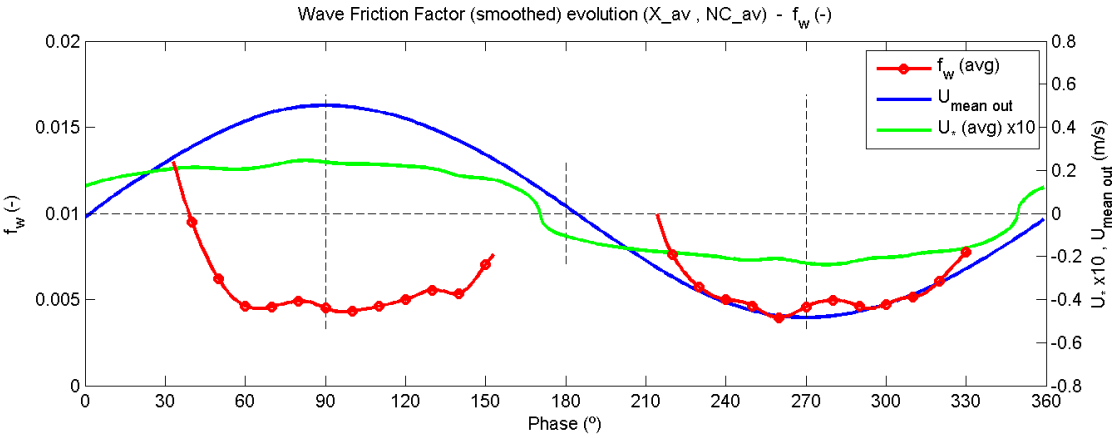
C.7.14 Bed shear stress



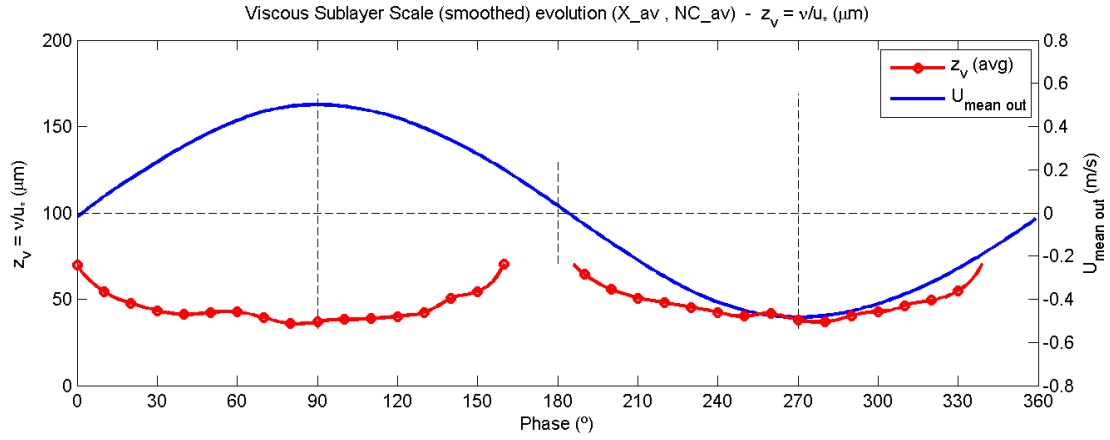
C.7.15 Shear velocity



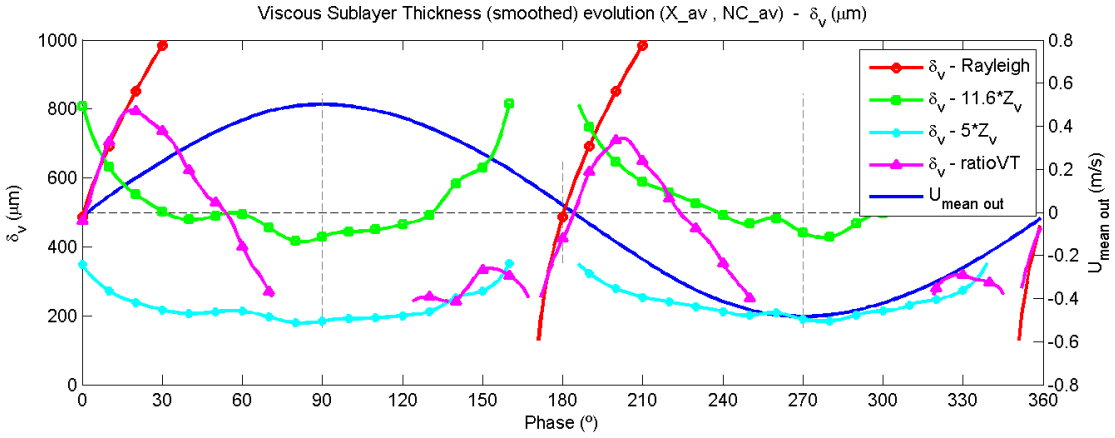
C.7.16 Wave friction factor



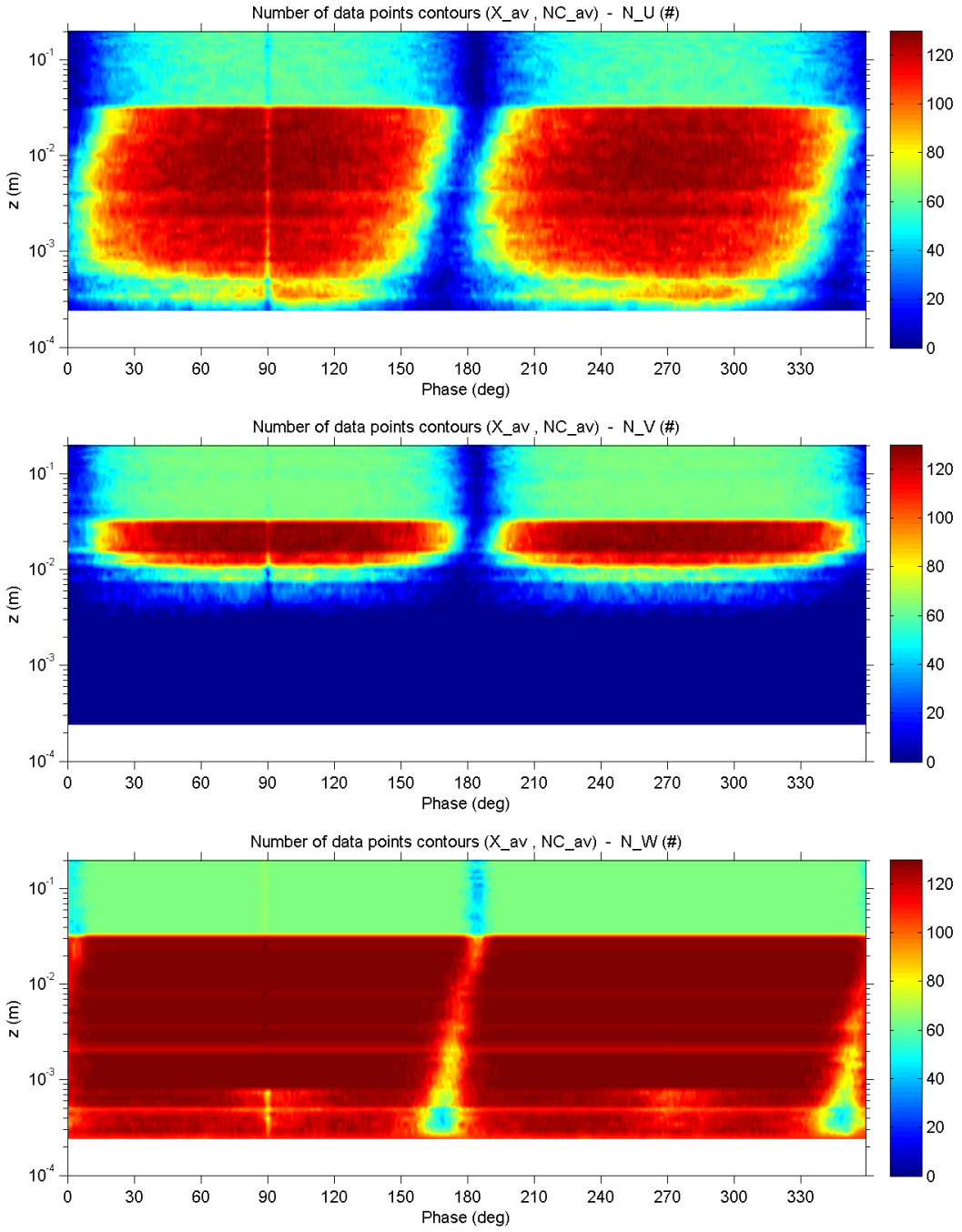
C.7.17 Viscous length scale



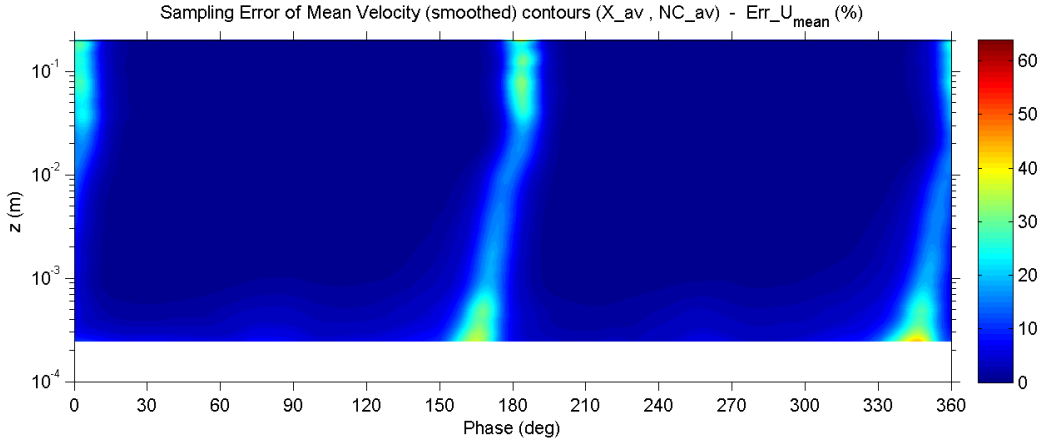
C.7.18 Viscous sublayer thickness



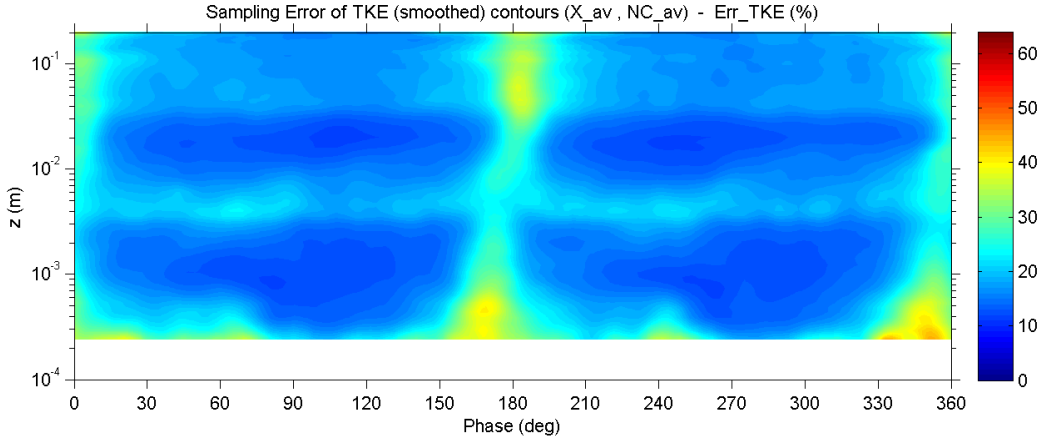
C.7.19 Number of valid data points



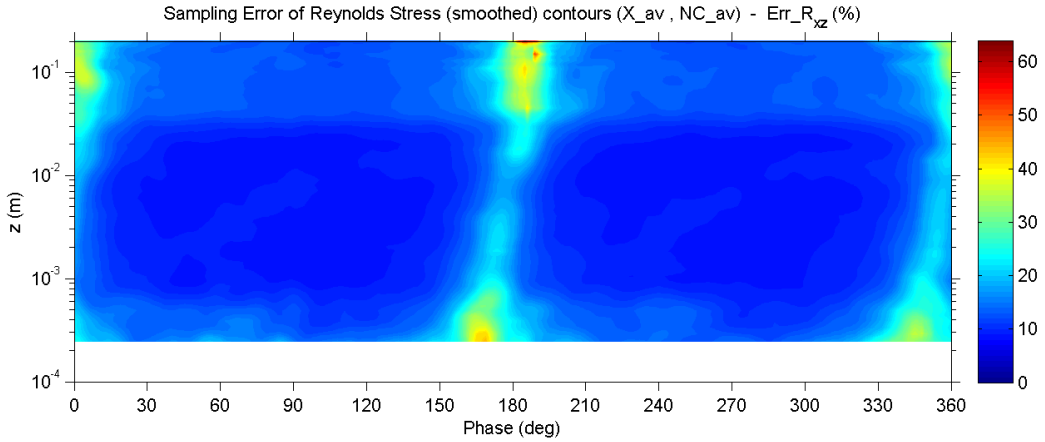
C.7.20 Percent error of mean velocity



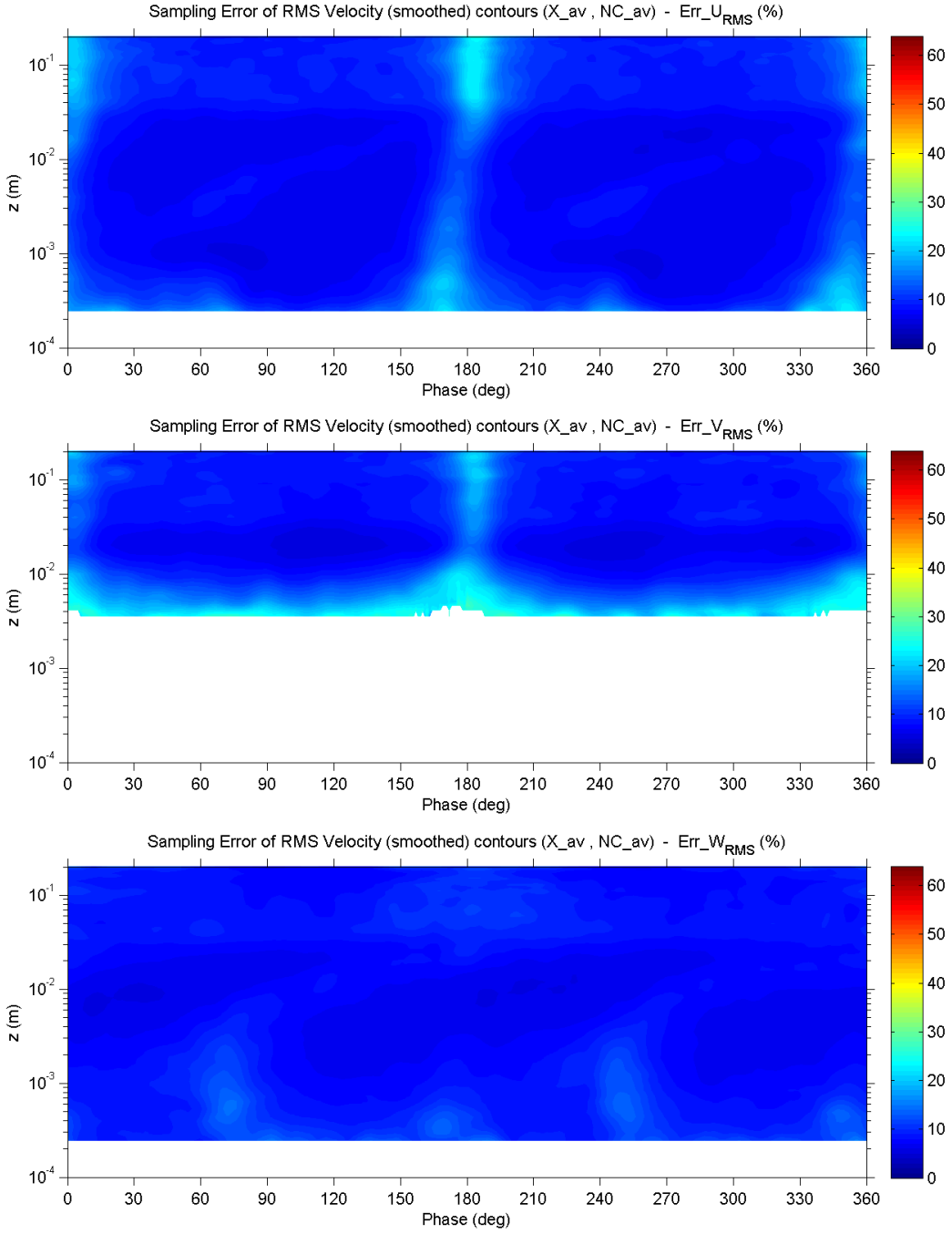
C.7.21 Percent error of turbulent kinetic energy



C.7.22 Percent error of Reynolds shear stress



C.7.23 Percent error of RMS velocity fluctuations



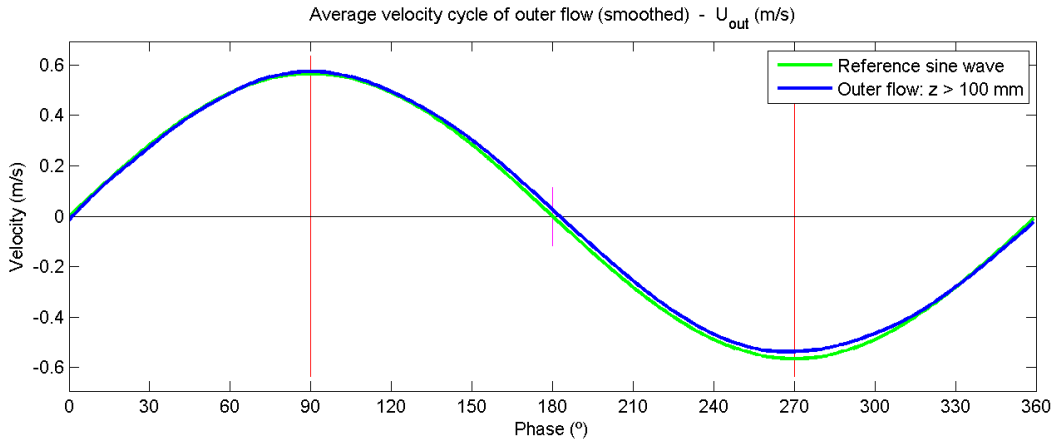
C.8 Experiment no. 8

C.8.1 Main parameters

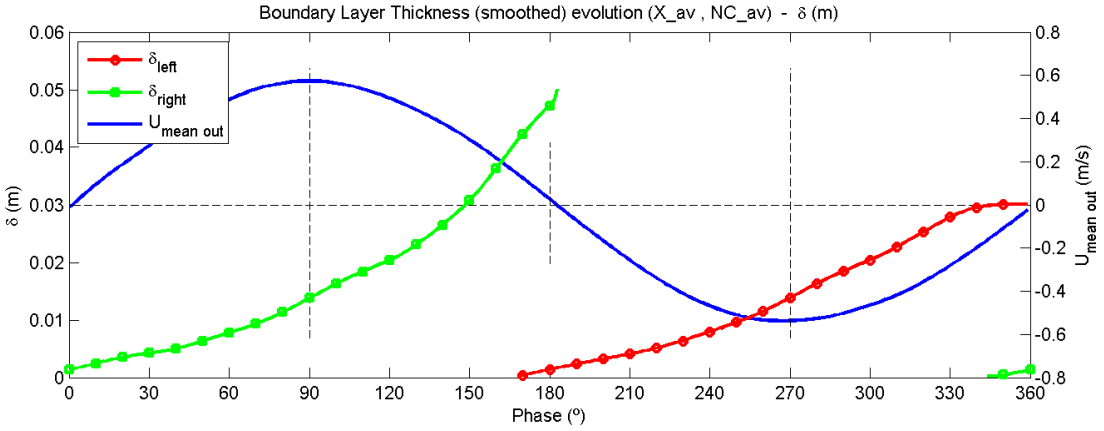
Exp no.	*Temp* (ºC)	ρ (kg/m^3)	ν (m^2/s)	*T* (s)	*2a* (m)	U_{out_max} (m/s)	Re_w (-)	N_{max} (cycles)
8	24.5	997.20	9.17E-07	10	1.770	0.556	5.4E+05	130

Exp no.	U_{*max} (m/s)	τ_{b_max} (N/m^2)	$\Delta\phi$ (deg)	f_{w_ref} (-)	Z_{v_min} (mm)	δ_{v_90} (mm)	δ_{90} (mm)	δ_{top} (mm)
8	0.026	0.70	-7.5	0.0045	0.035	-	13.4	40

C.8.2 Outer flow velocity

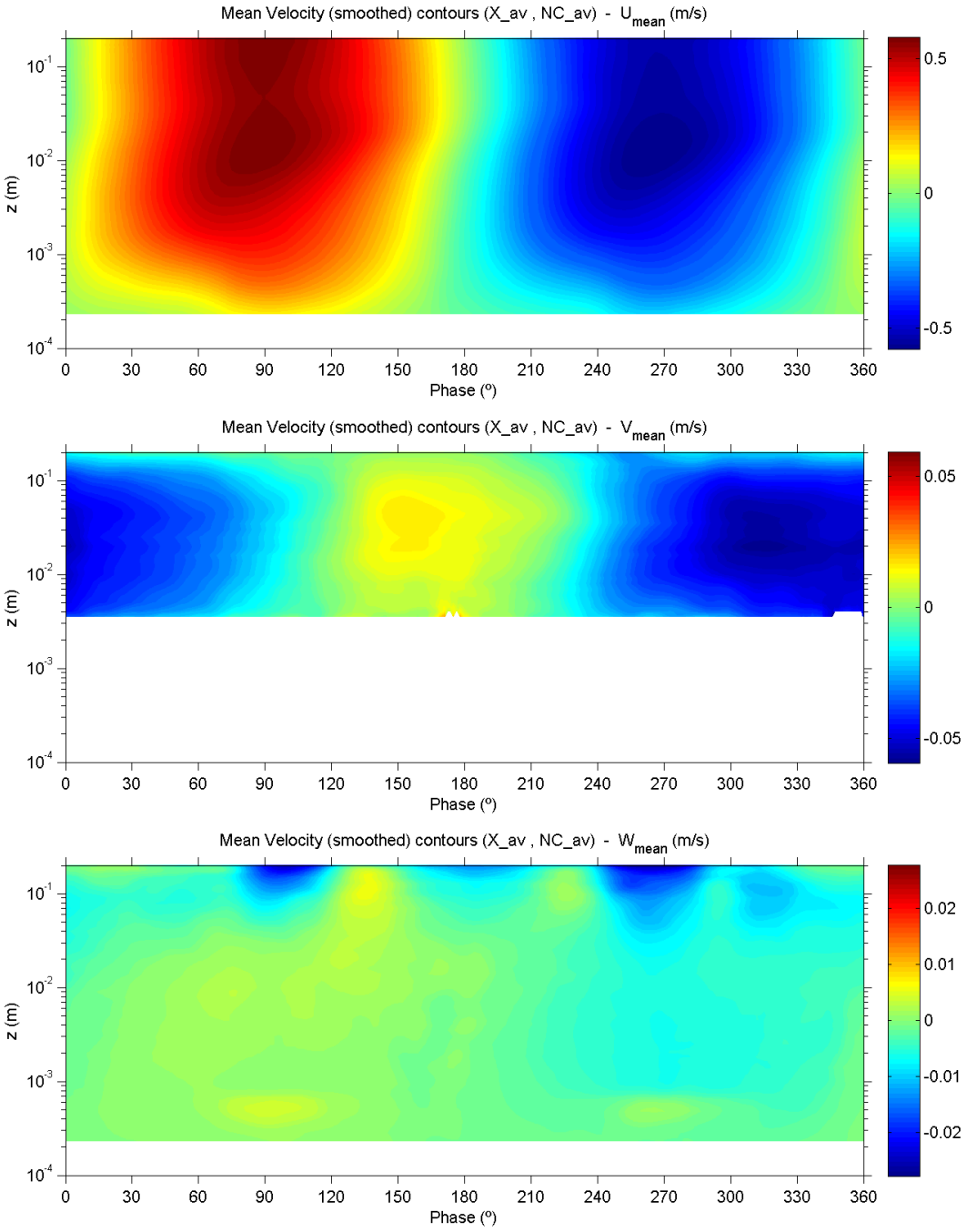


C.8.3 Boundary layer thickness

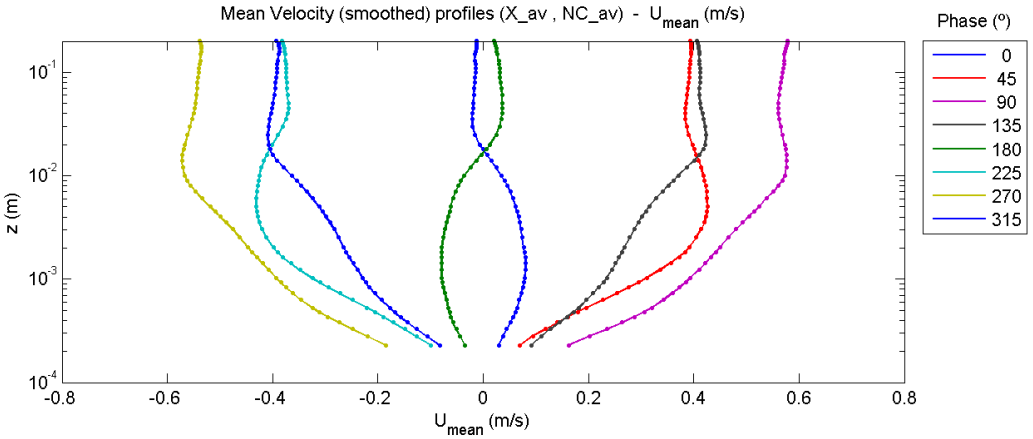


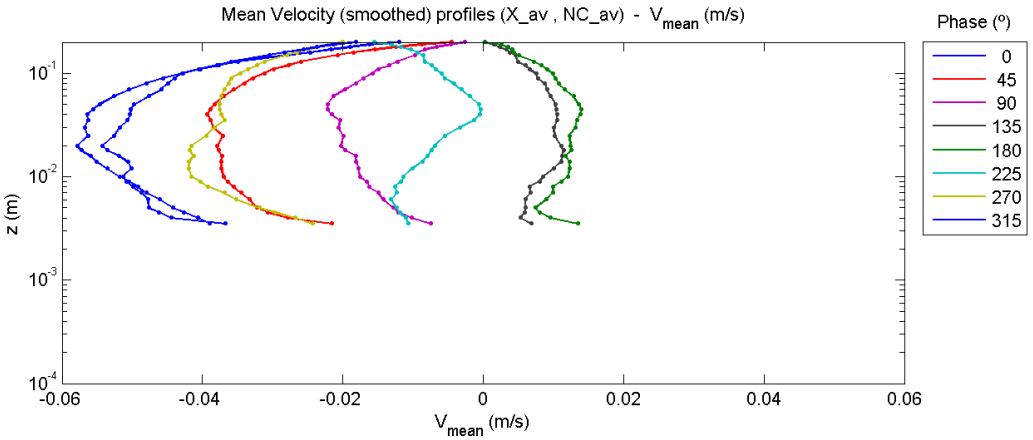
C.8.4 Mean velocities

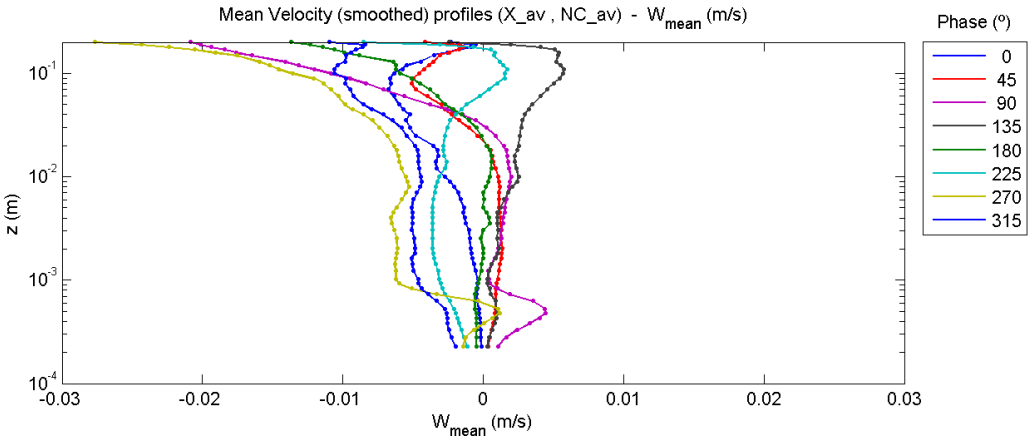
C.8.4.1 Contour plots



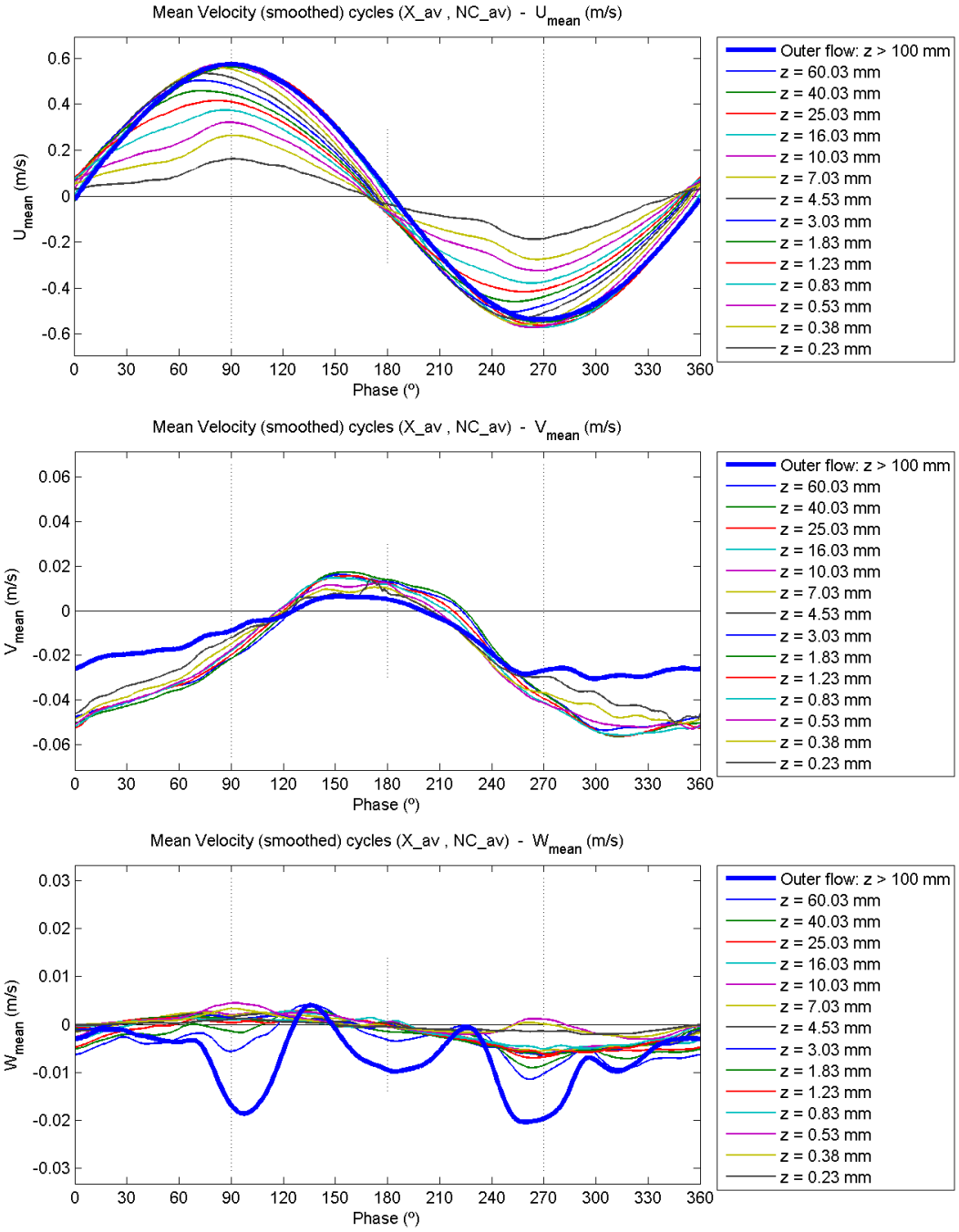
C.8.4.2 Profile plots





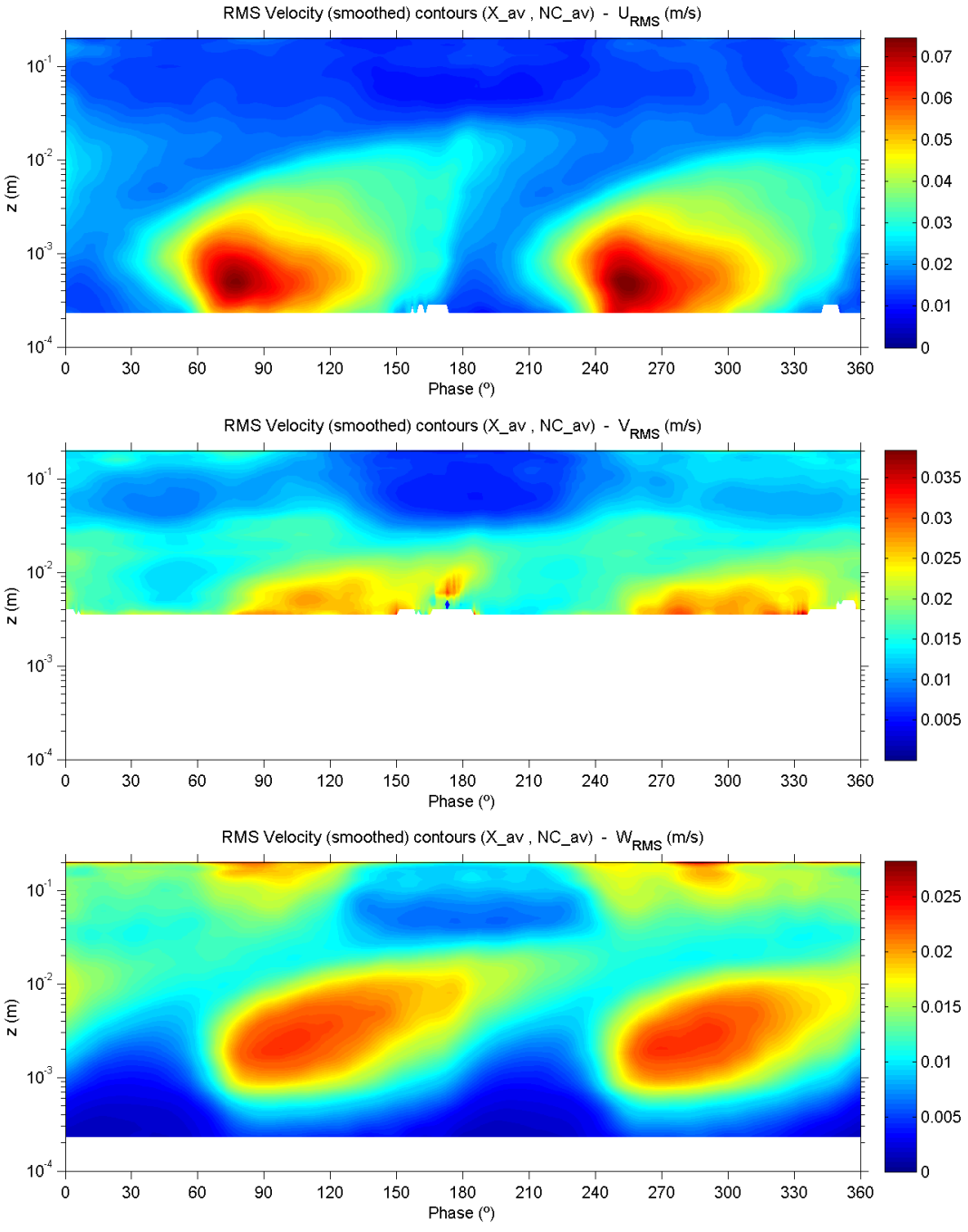


C.8.4.3 Evolution plots

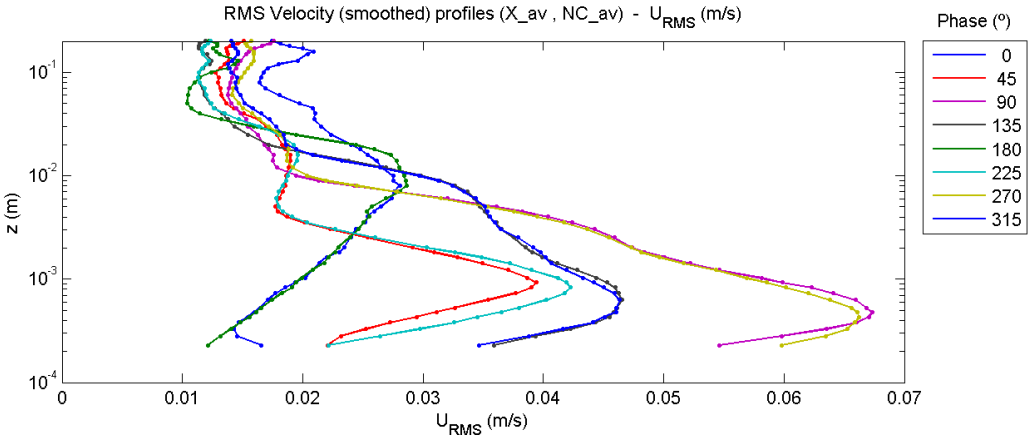


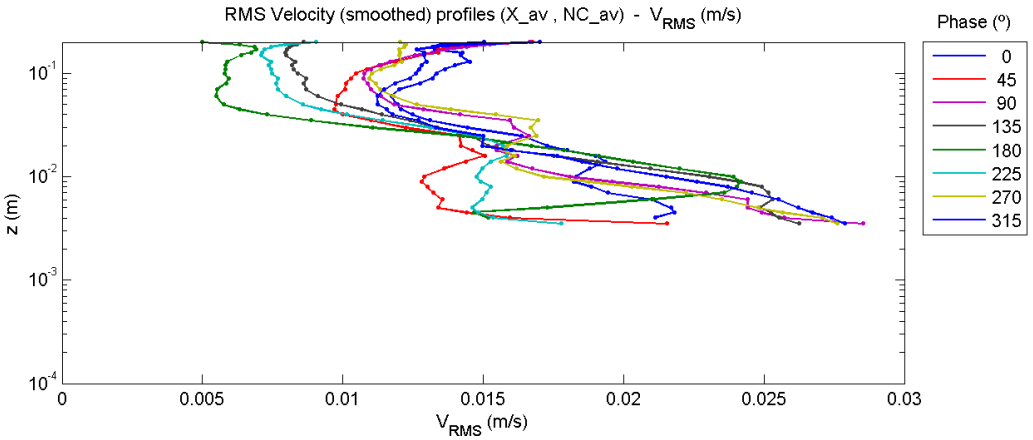
C.8.5 RMS velocity fluctuations

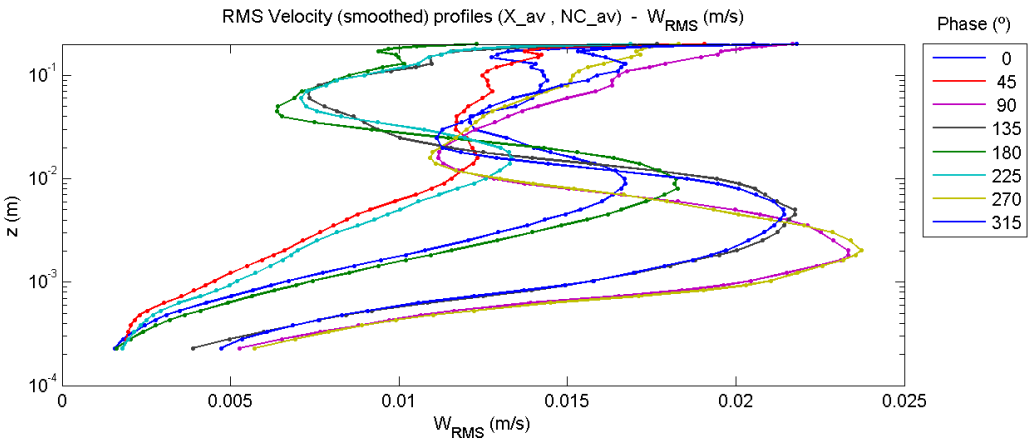
C.8.5.1 Contour plots



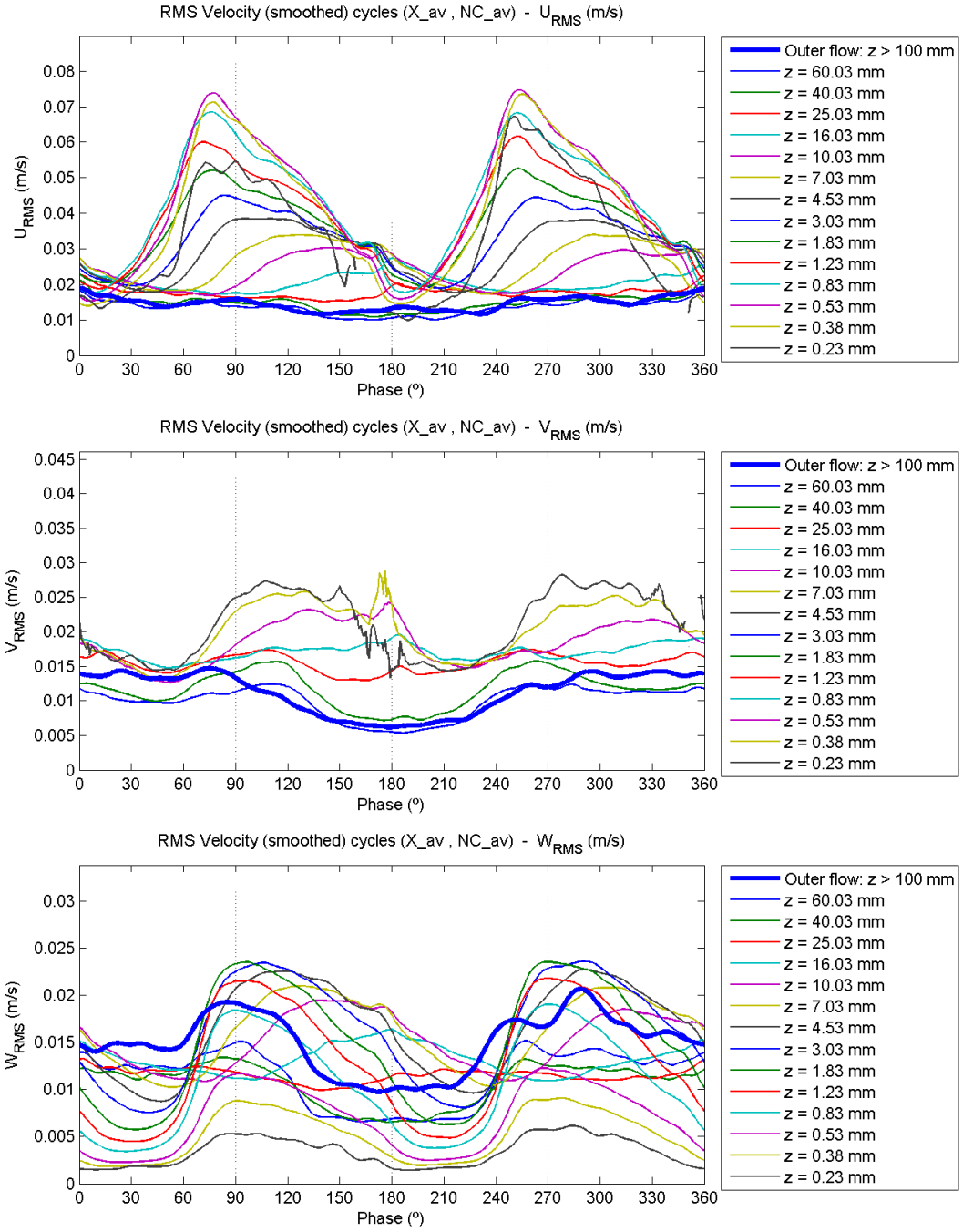
C.8.5.2 Profile plots





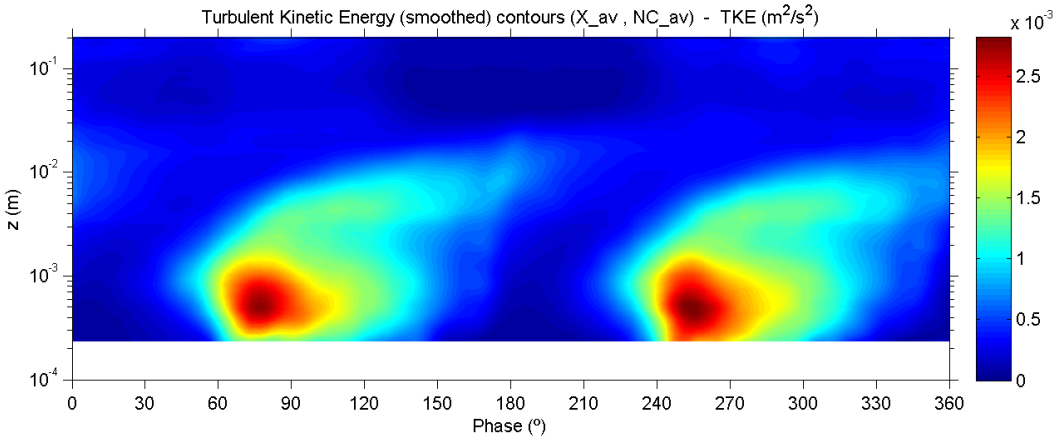


C.8.5.3 Evolution plots

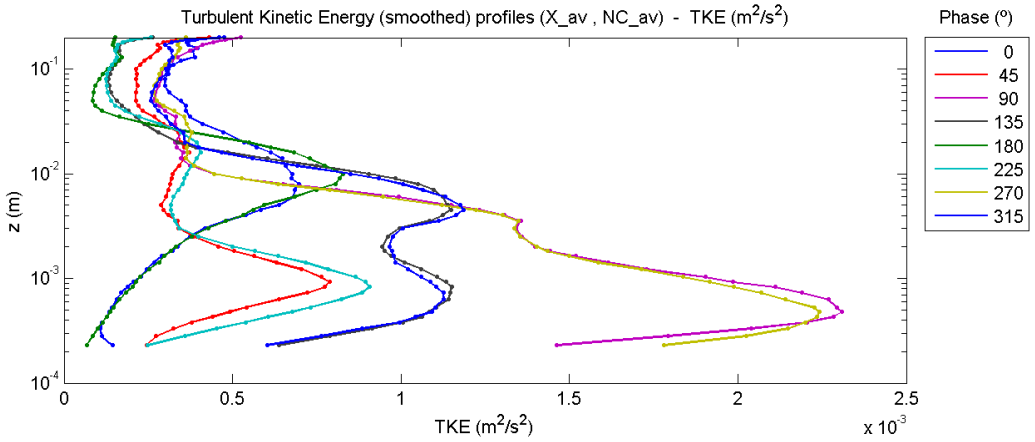


C.8.6 Turbulent kinetic energy

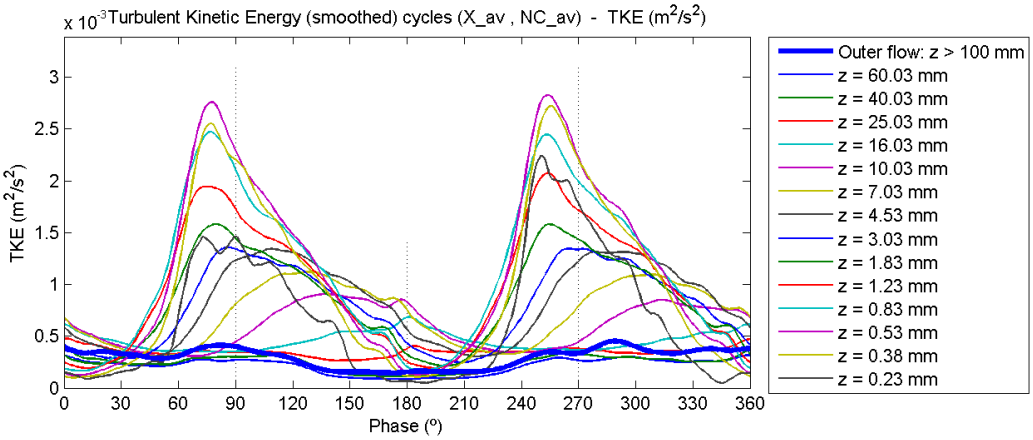
C.8.6.1 Contour plot



C.8.6.2 Profile plot

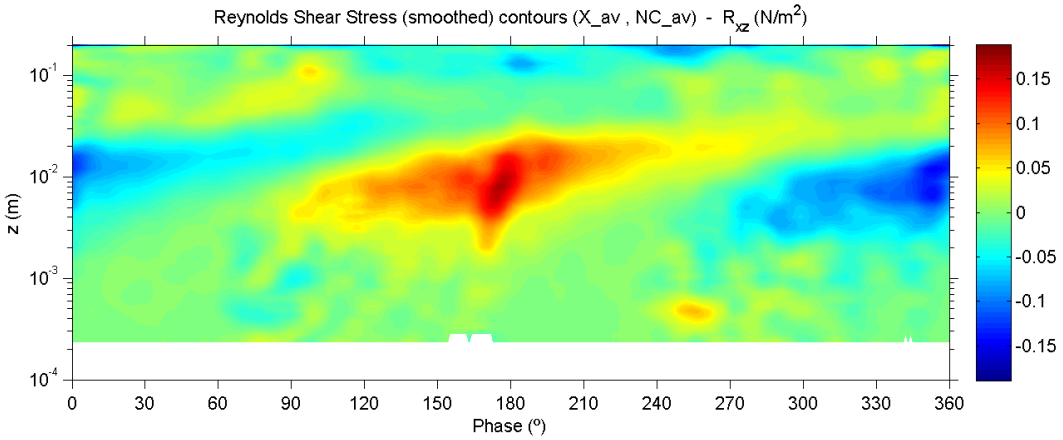


C.8.6.3 Evolution plot

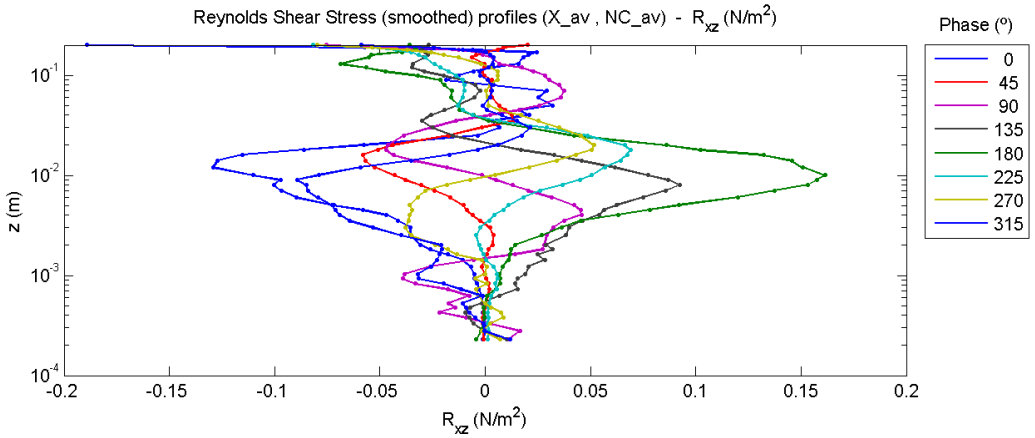


C.8.7 Turbulent shear stress (Reynolds shear stress)

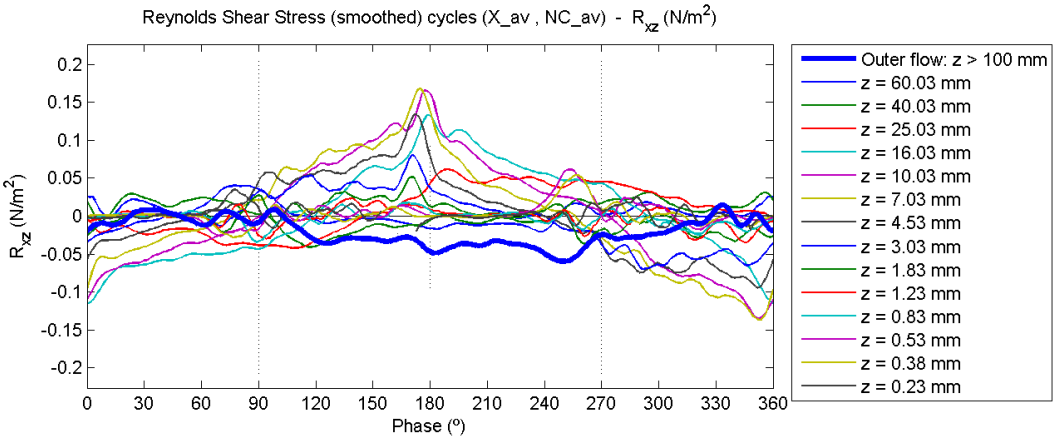
C.8.7.1 Contour plot



C.8.7.2 Profile plot

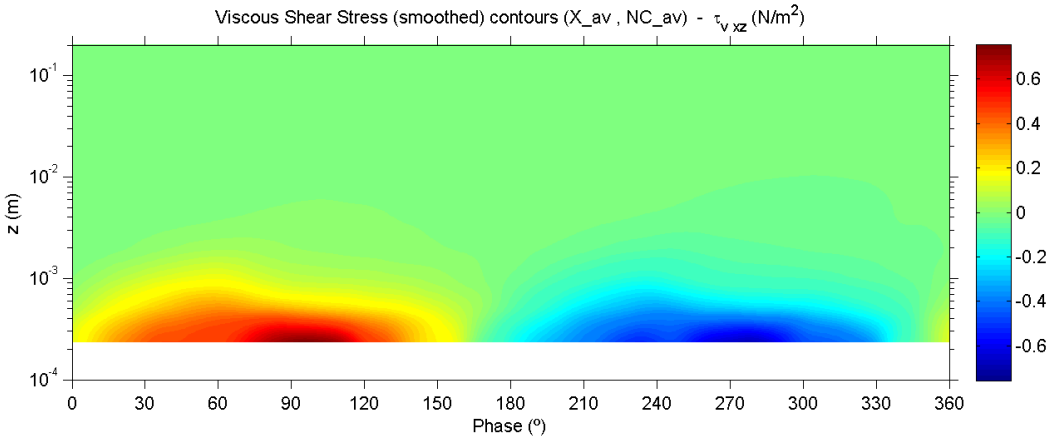


C.8.7.3 Evolution plot

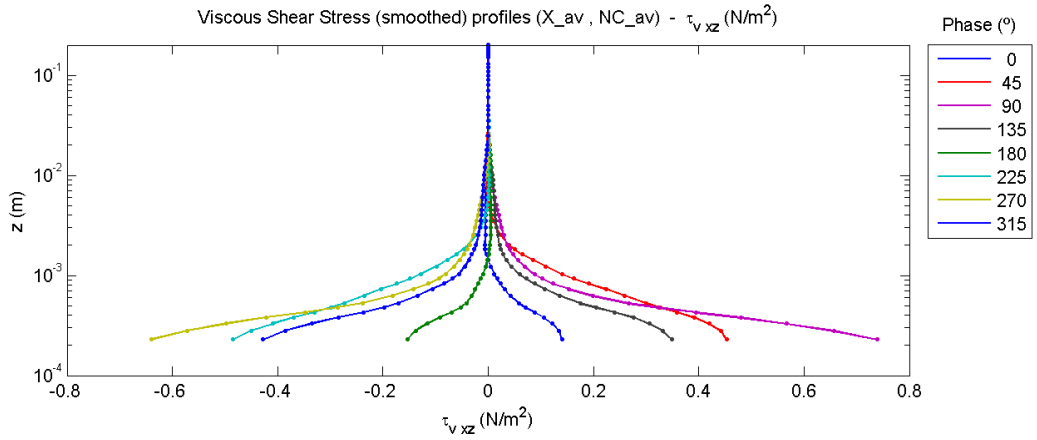


C.8.8 Viscous shear stress

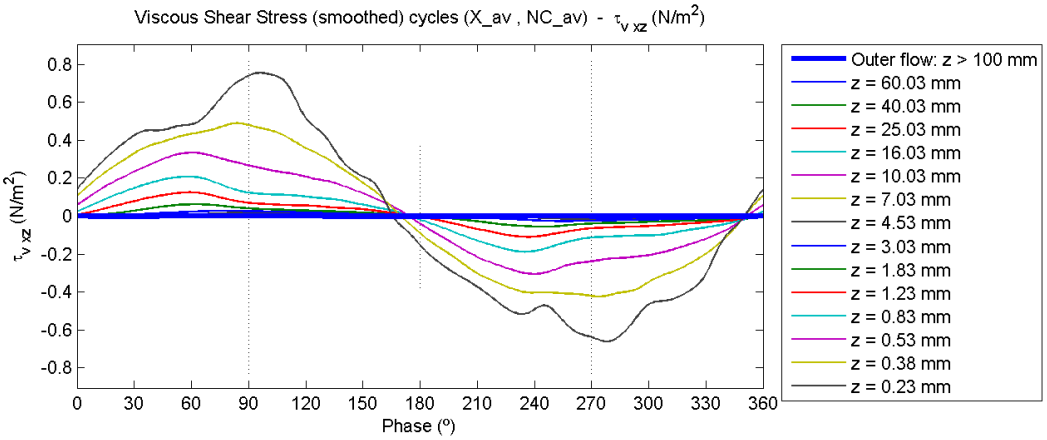
C.8.8.1 Contour plot



C.8.8.2 Profile plot

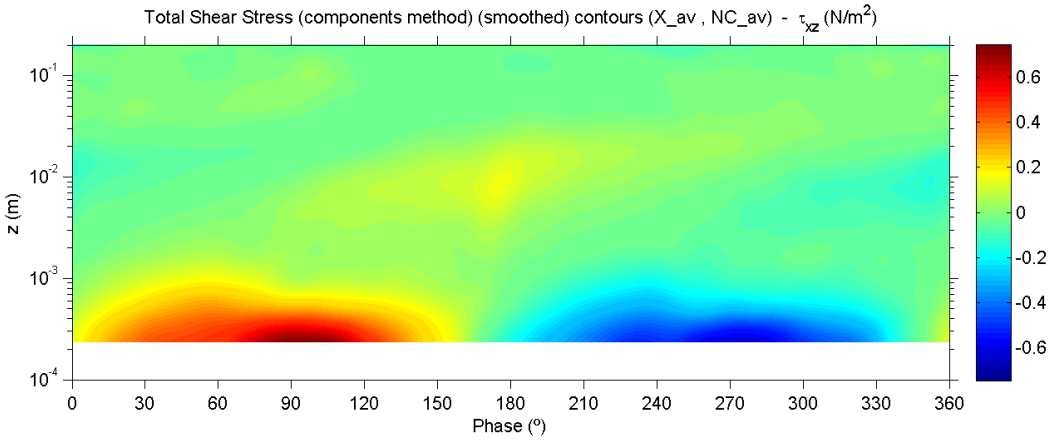


C.8.8.3 Evolution plot

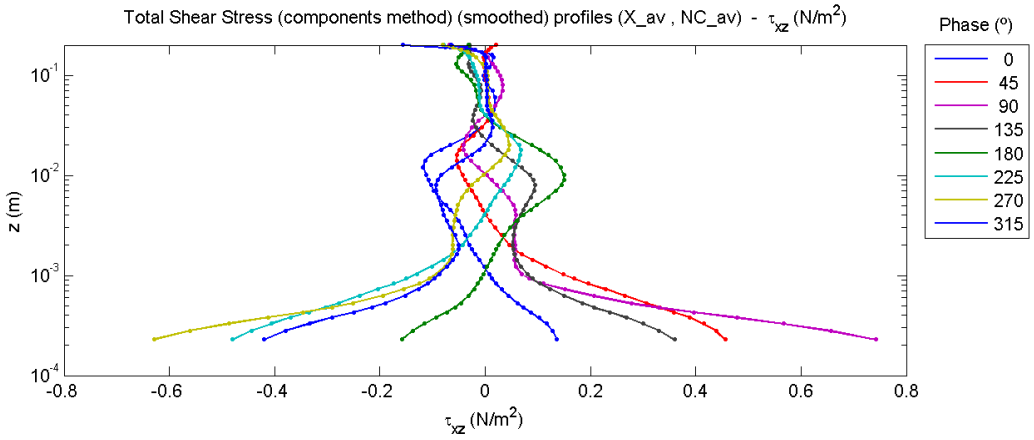


C.8.9 Total shear stress

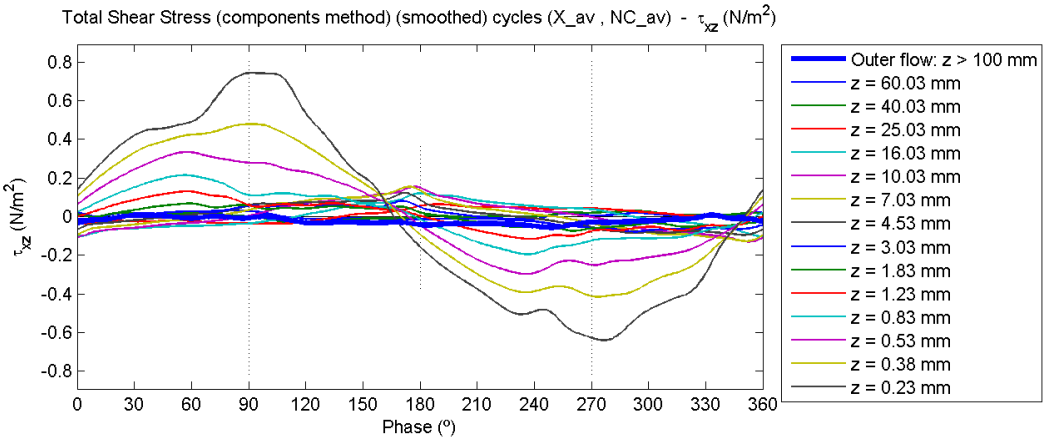
C.8.9.1 Contour plot



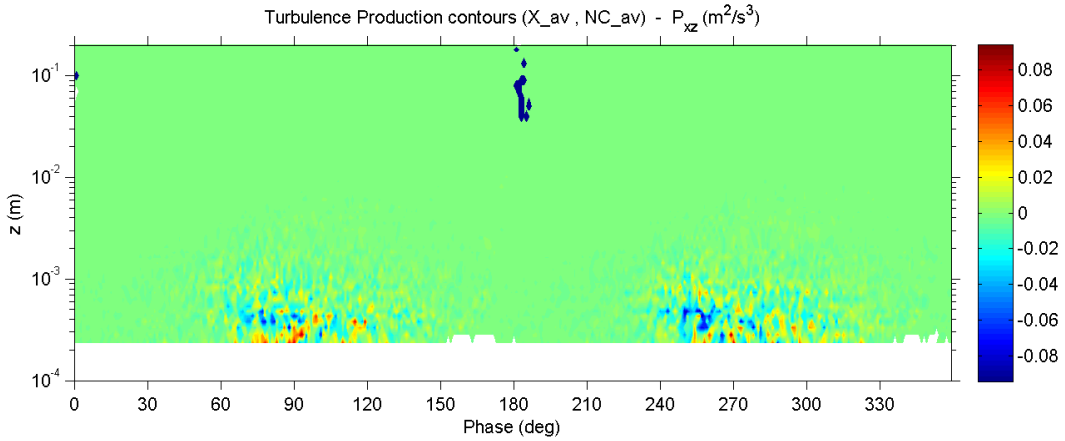
C.8.9.2 Profile plot



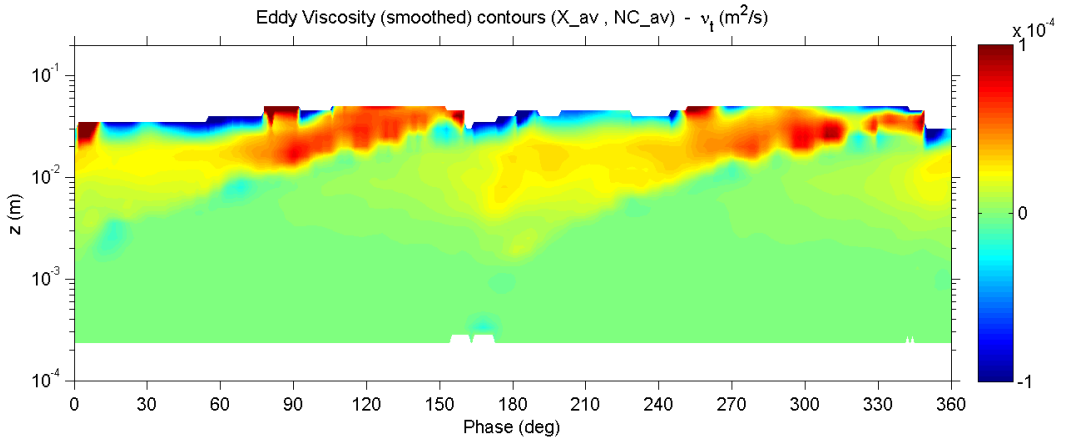
C.8.9.3 Evolution plot



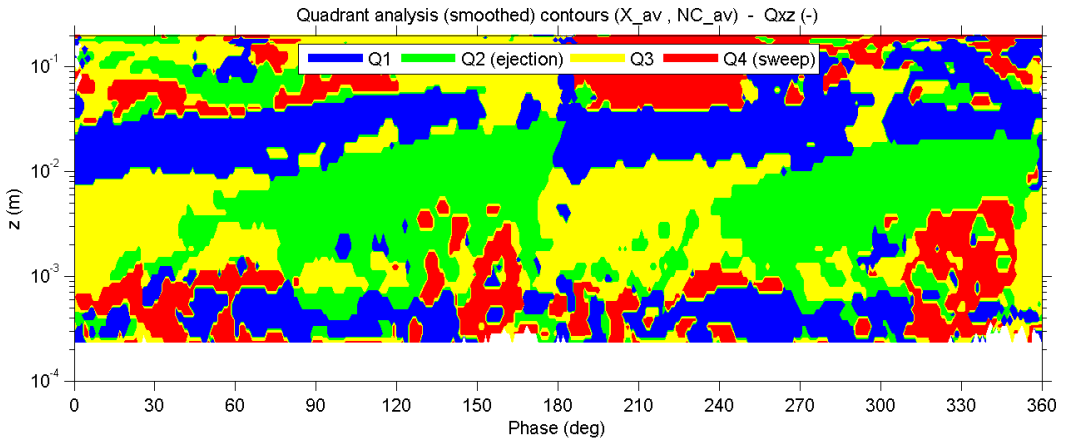
C.8.10 Turbulence production



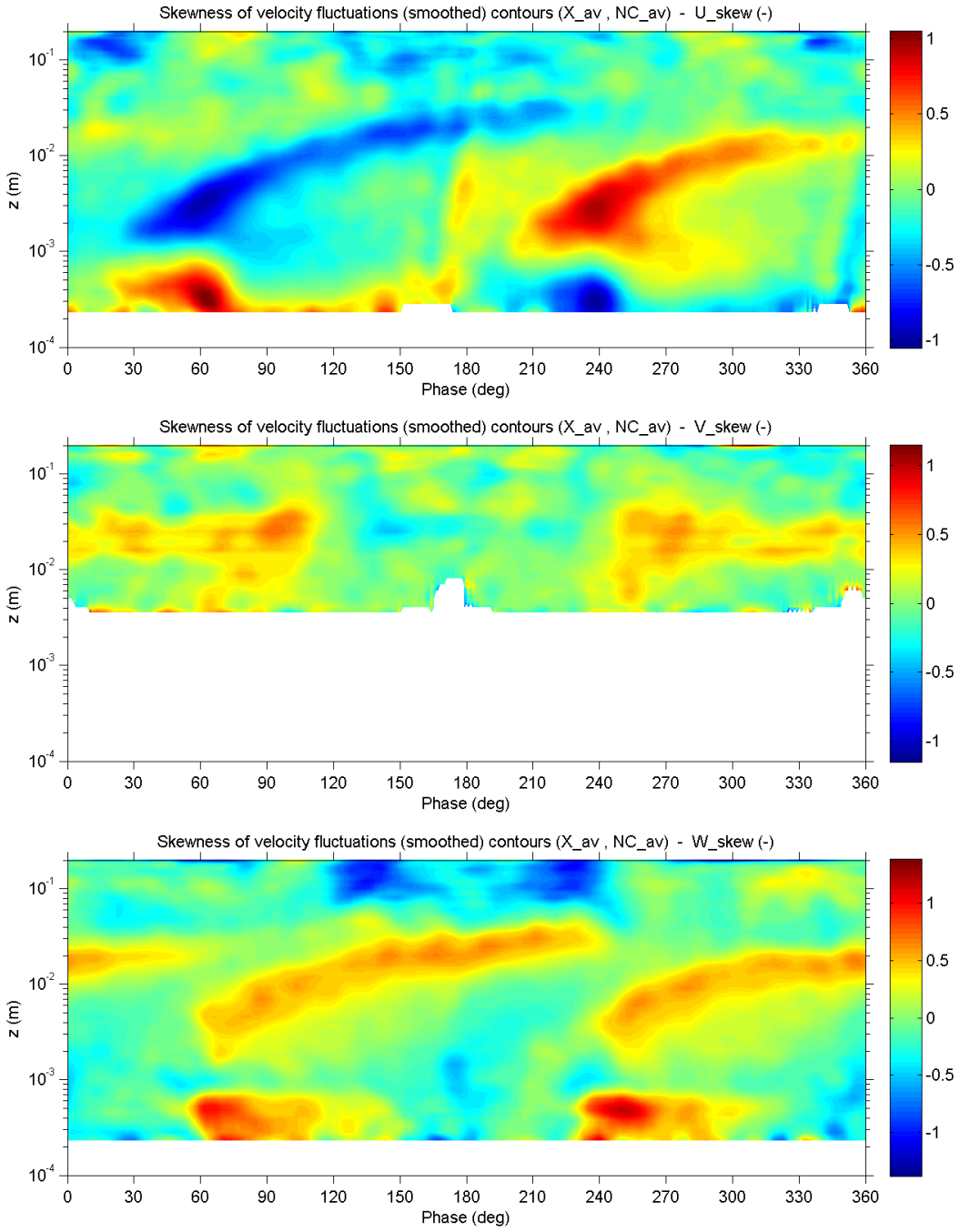
C.8.11 Turbulent viscosity (Eddy viscosity)



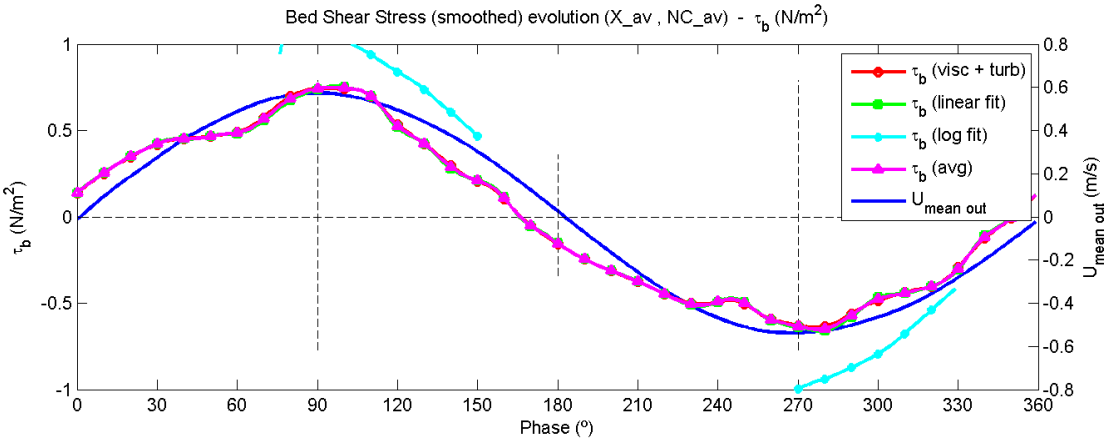
C.8.12 Quadrant analysis



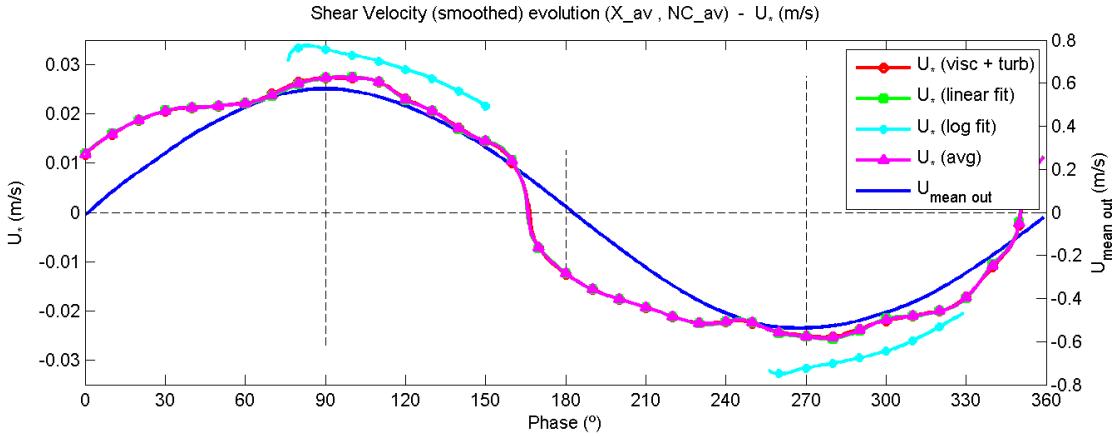
C.8.13 Skewness of velocity fluctuations



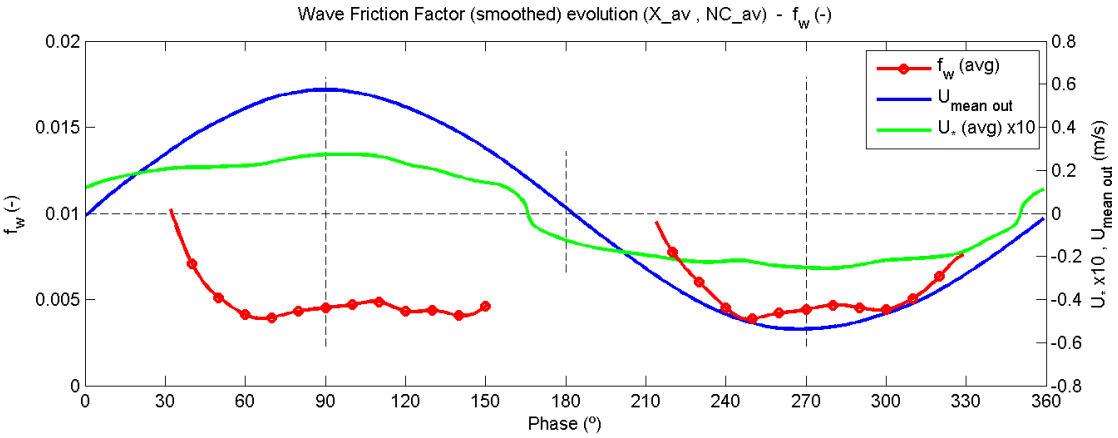
C.8.14 Bed shear stress



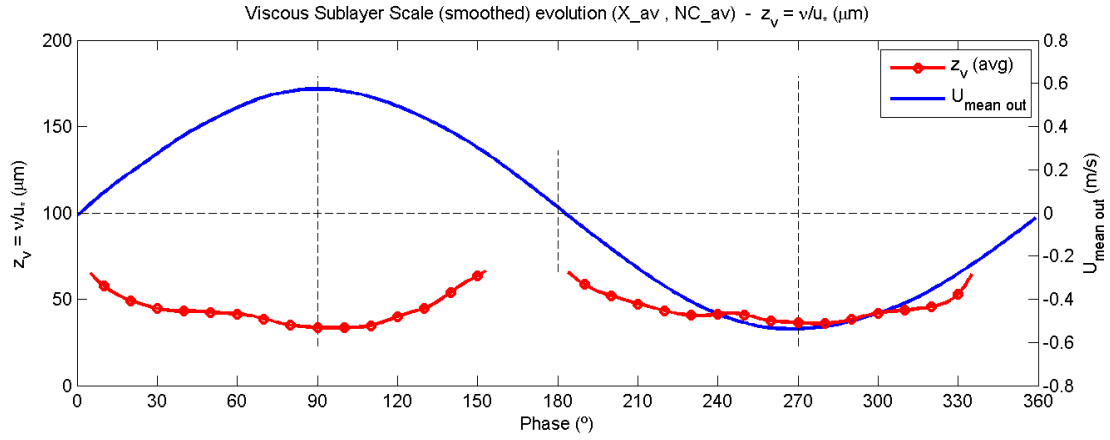
C.8.15 Shear velocity



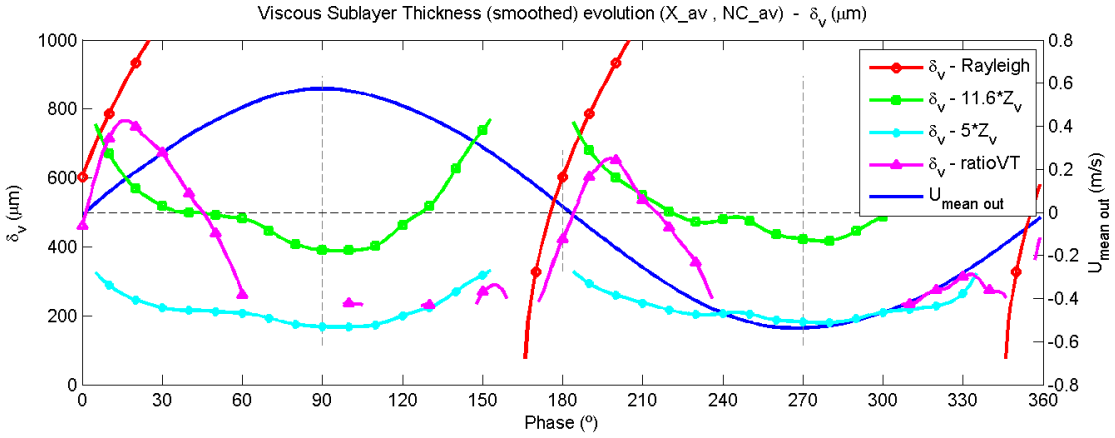
C.8.16 Wave friction factor



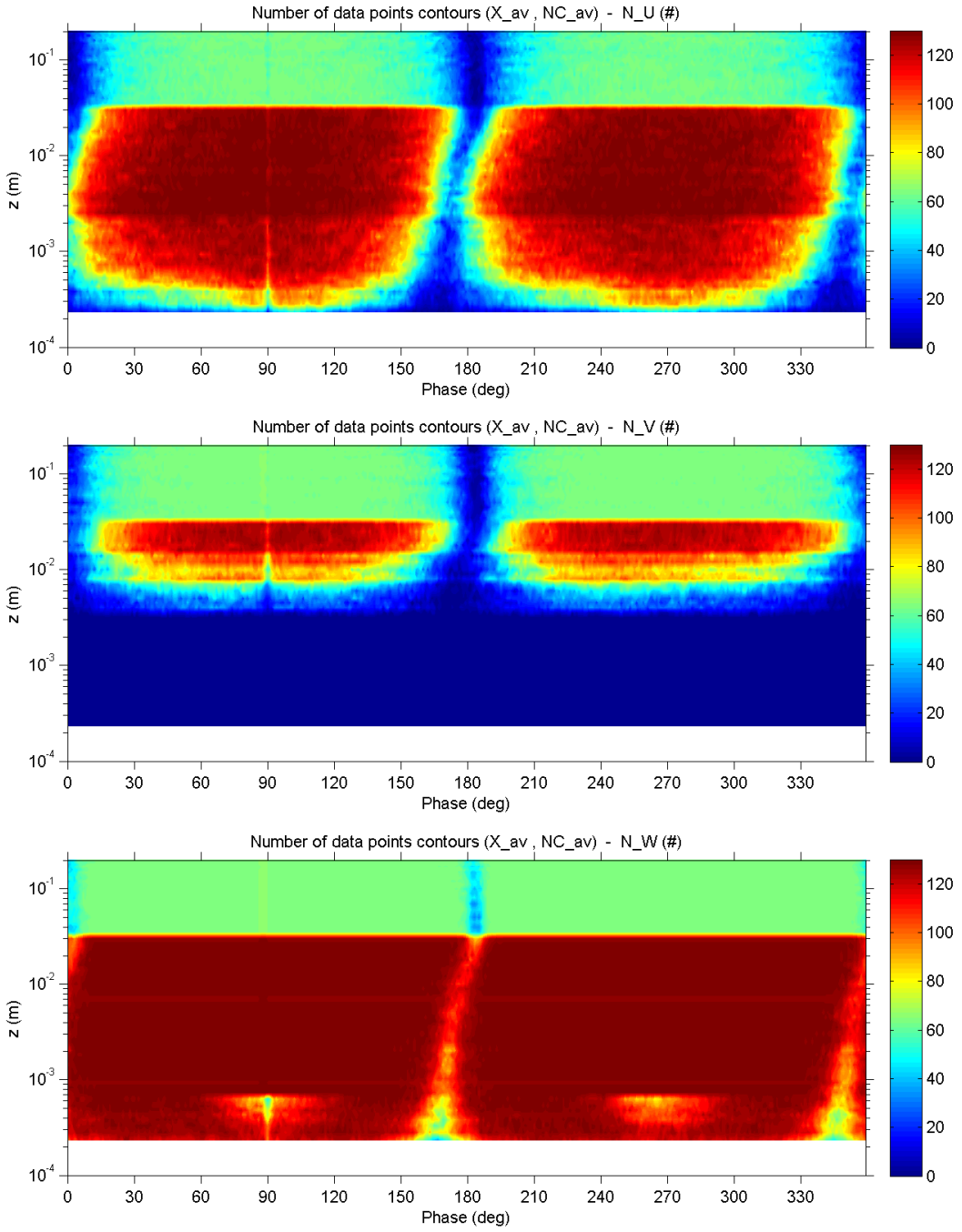
C.8.17 Viscous length scale



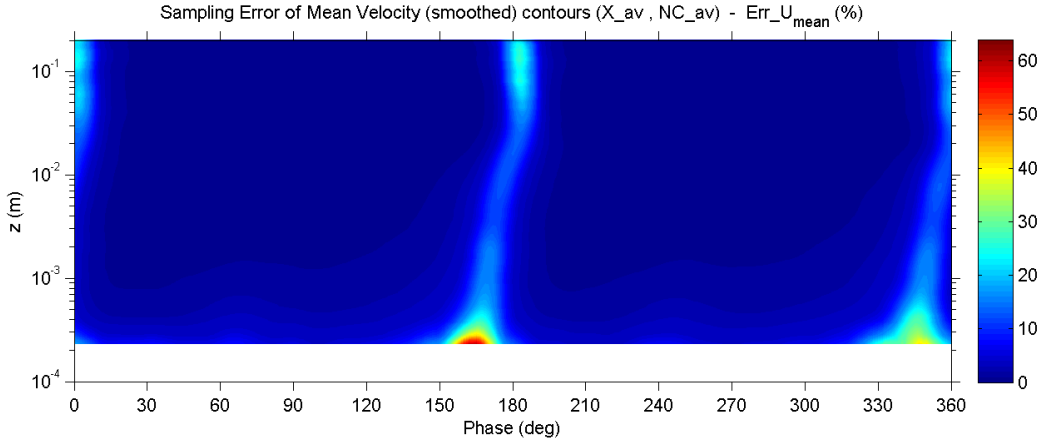
C.8.18 Viscous sublayer thickness



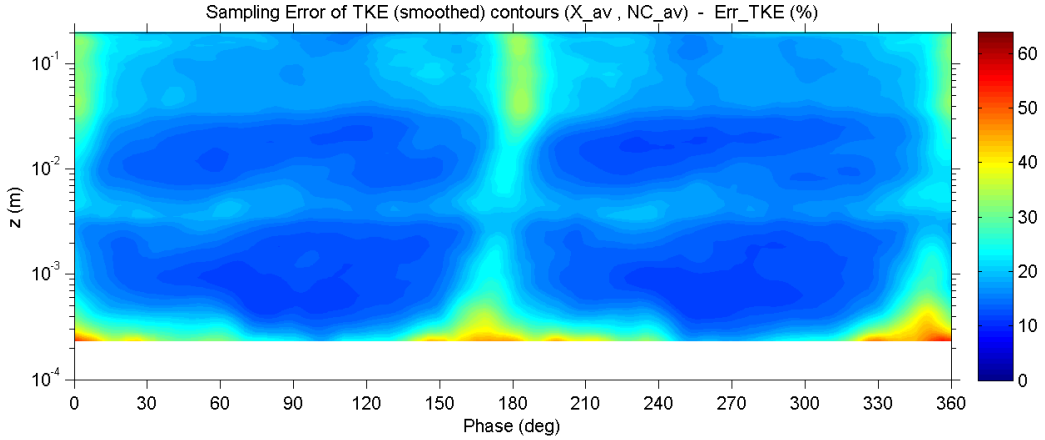
C.8.19 Number of valid data points



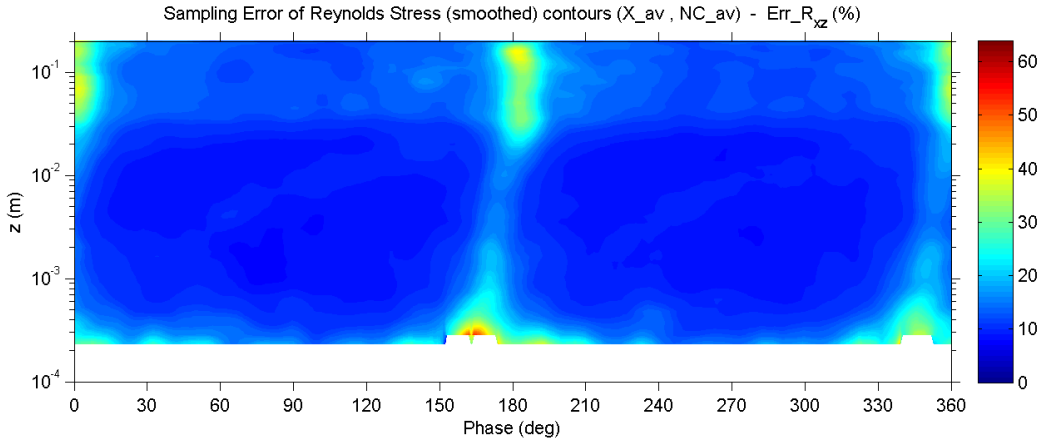
C.8.20 Percent error of mean velocity



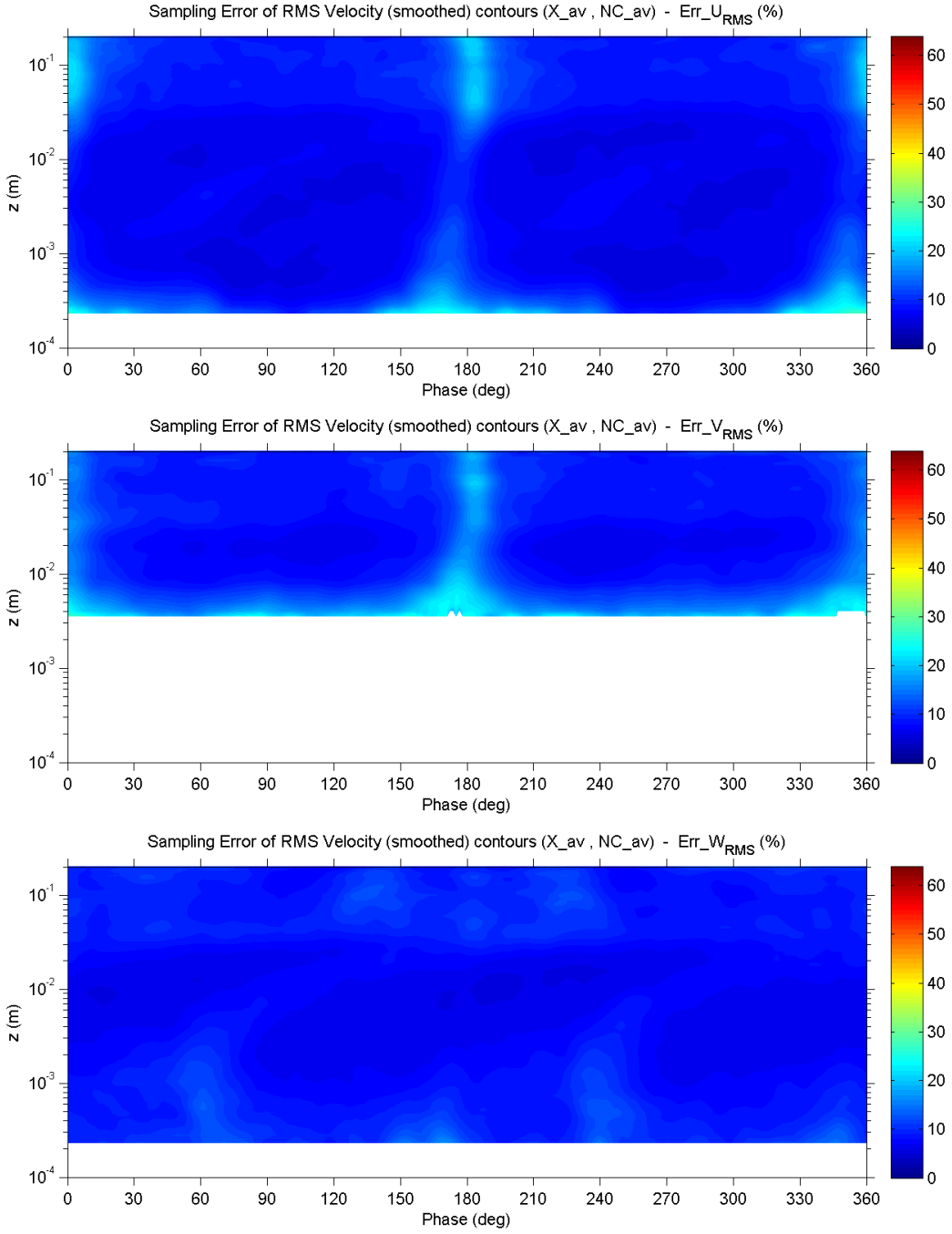
C.8.21 Percent error of turbulent kinetic energy



C.8.22 Percent error of Reynolds shear stress



C.8.23 Percent error of RMS velocity fluctuations



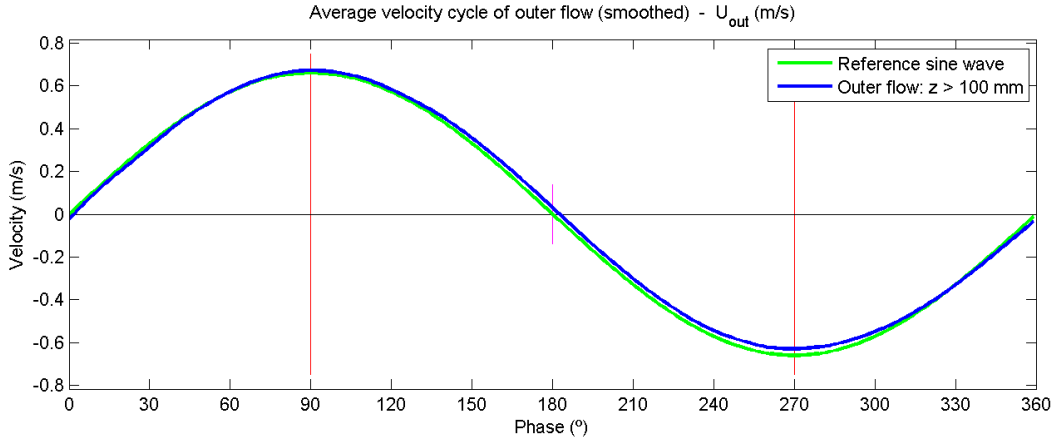
C.9 Experiment no. 9

C.9.1 Main parameters

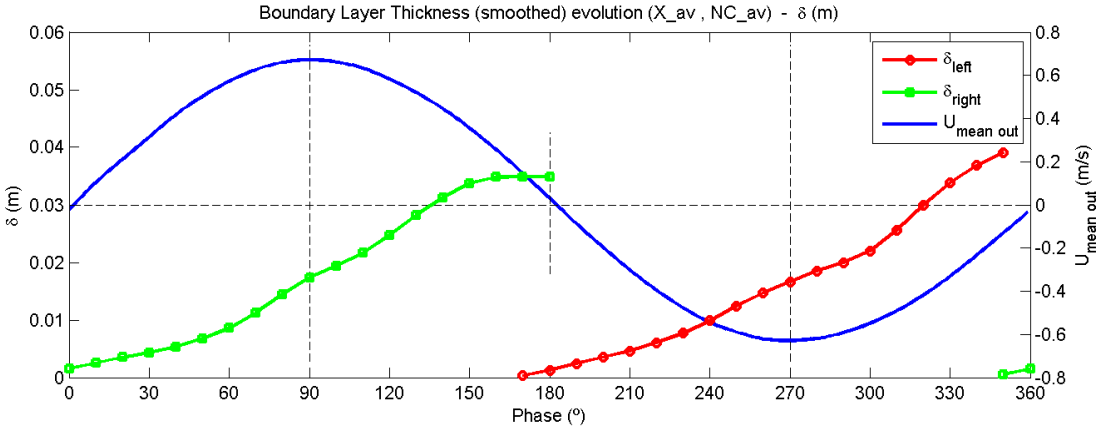
Exp no.	*Temp* (ºC)	ρ (kg/m^3)	ν (m^2/s)	*T* (s)	*2a* (m)	U_{out_max} (m/s)	Re_w (-)	N_{max} (cycles)
9	18.1	998.61	1.07E-06	10	2.069	0.650	6.3E+05	130

Exp no.	U_{*max} (m/s)	τ_{b_max} (N/m^2)	$\Delta\phi$ (deg)	f_{w_ref} (-)	Z_{v_min} (mm)	δ_{v_90} (mm)	δ_{90} (mm)	δ_{top} (mm)
9	0.037	1.34	6.0	0.0063	0.029	0.169	17.0	45

C.9.2 Outer flow velocity

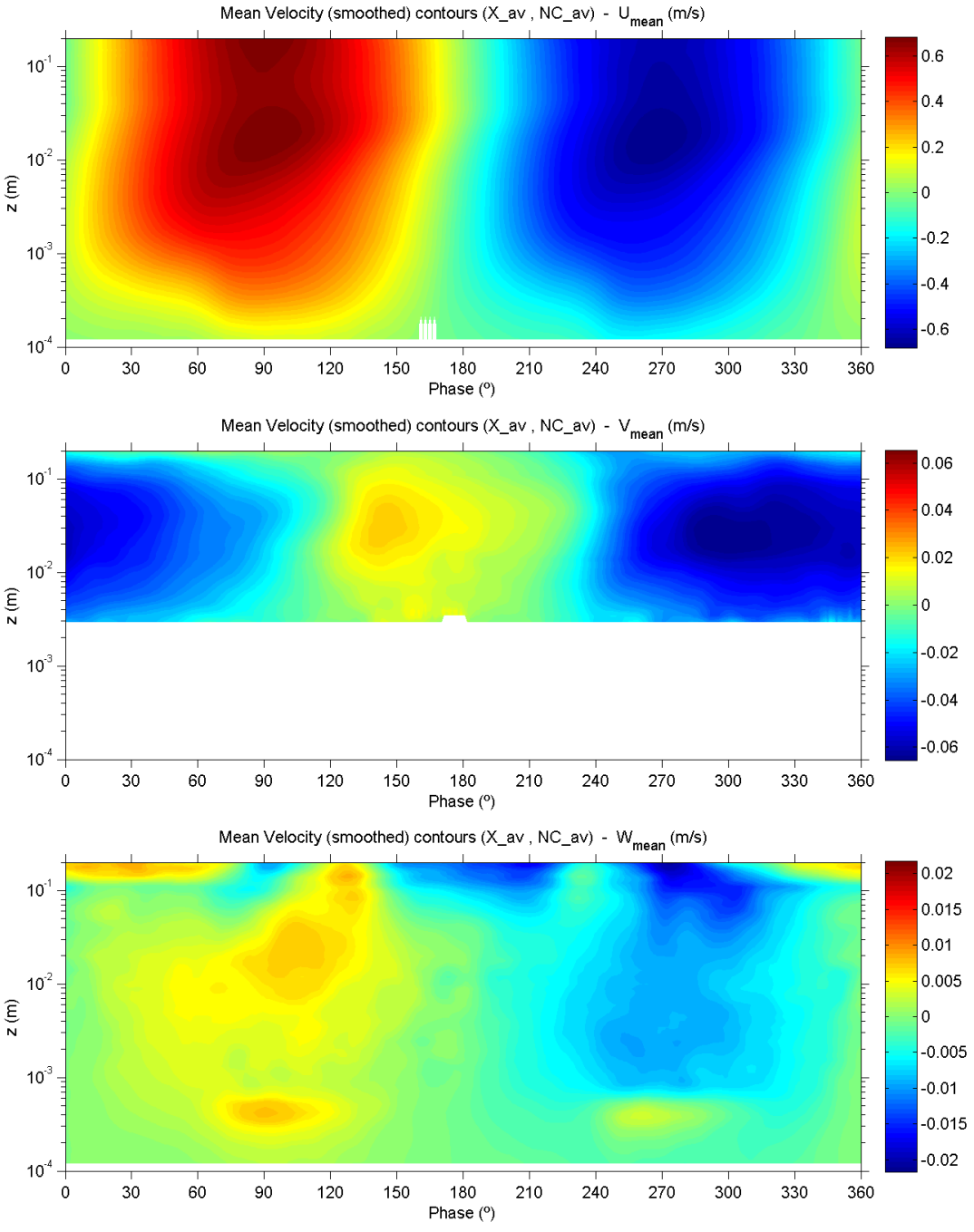


C.9.3 Boundary layer thickness

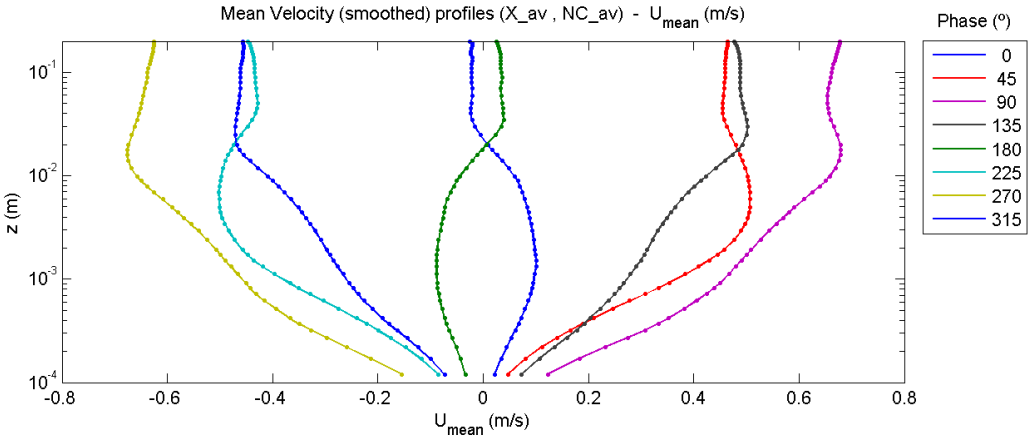


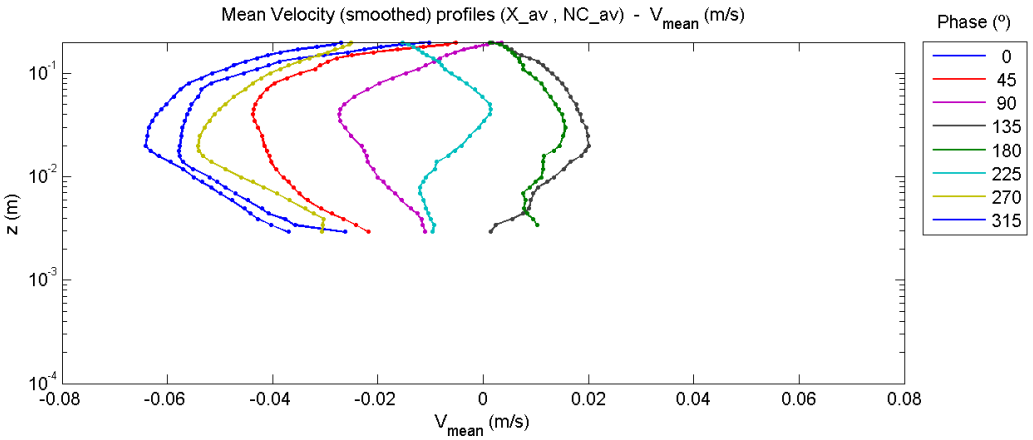
C.9.4 Mean velocities

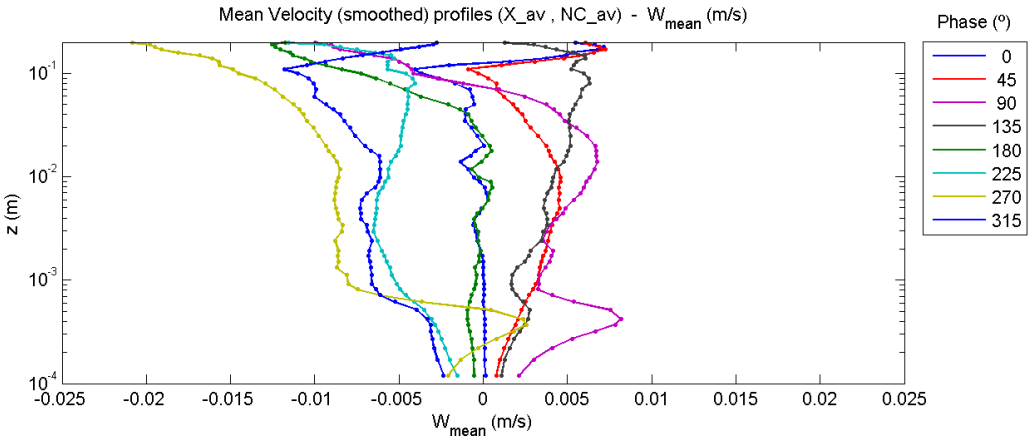
C.9.4.1 Contour plots



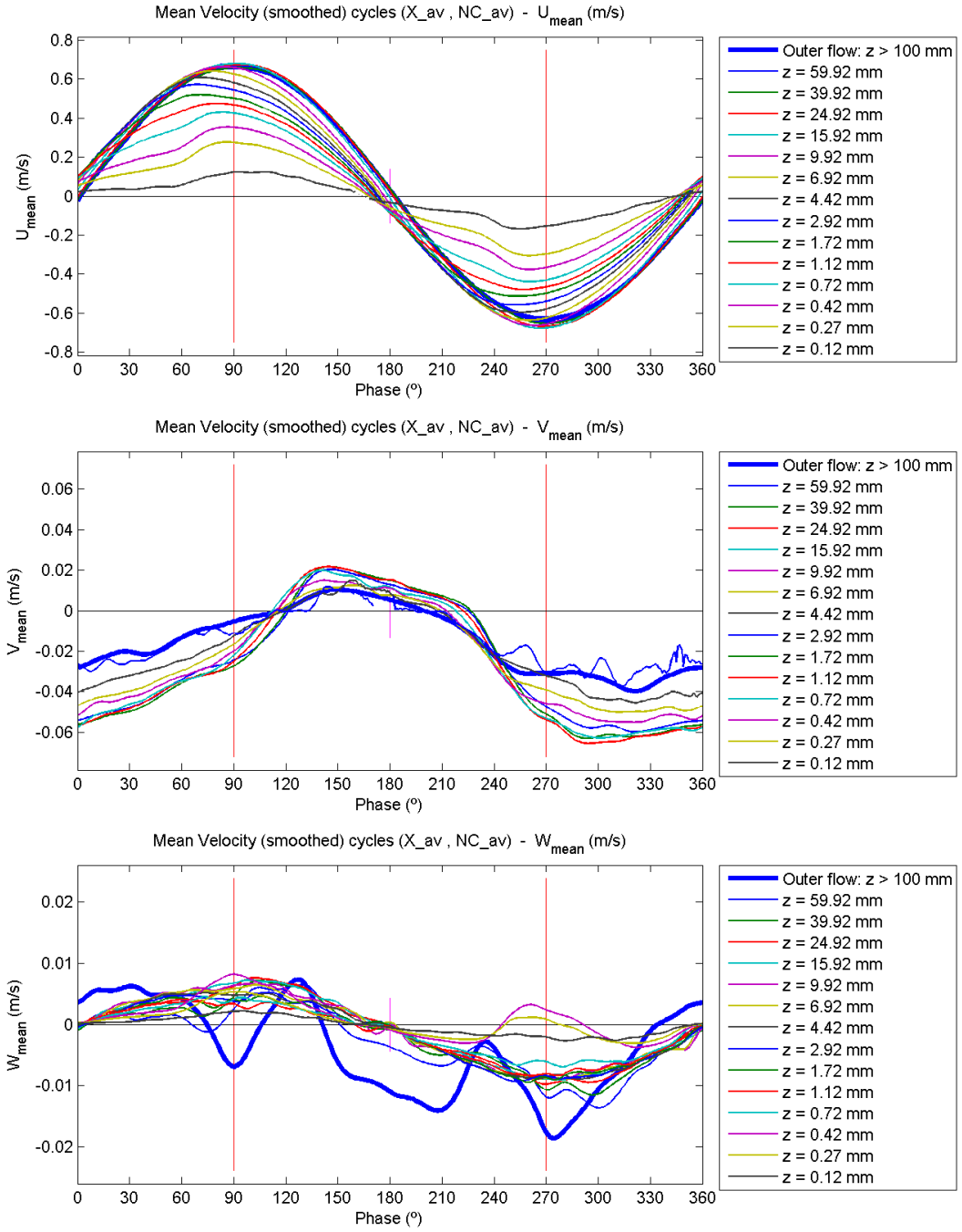
C.9.4.2 Profile plots





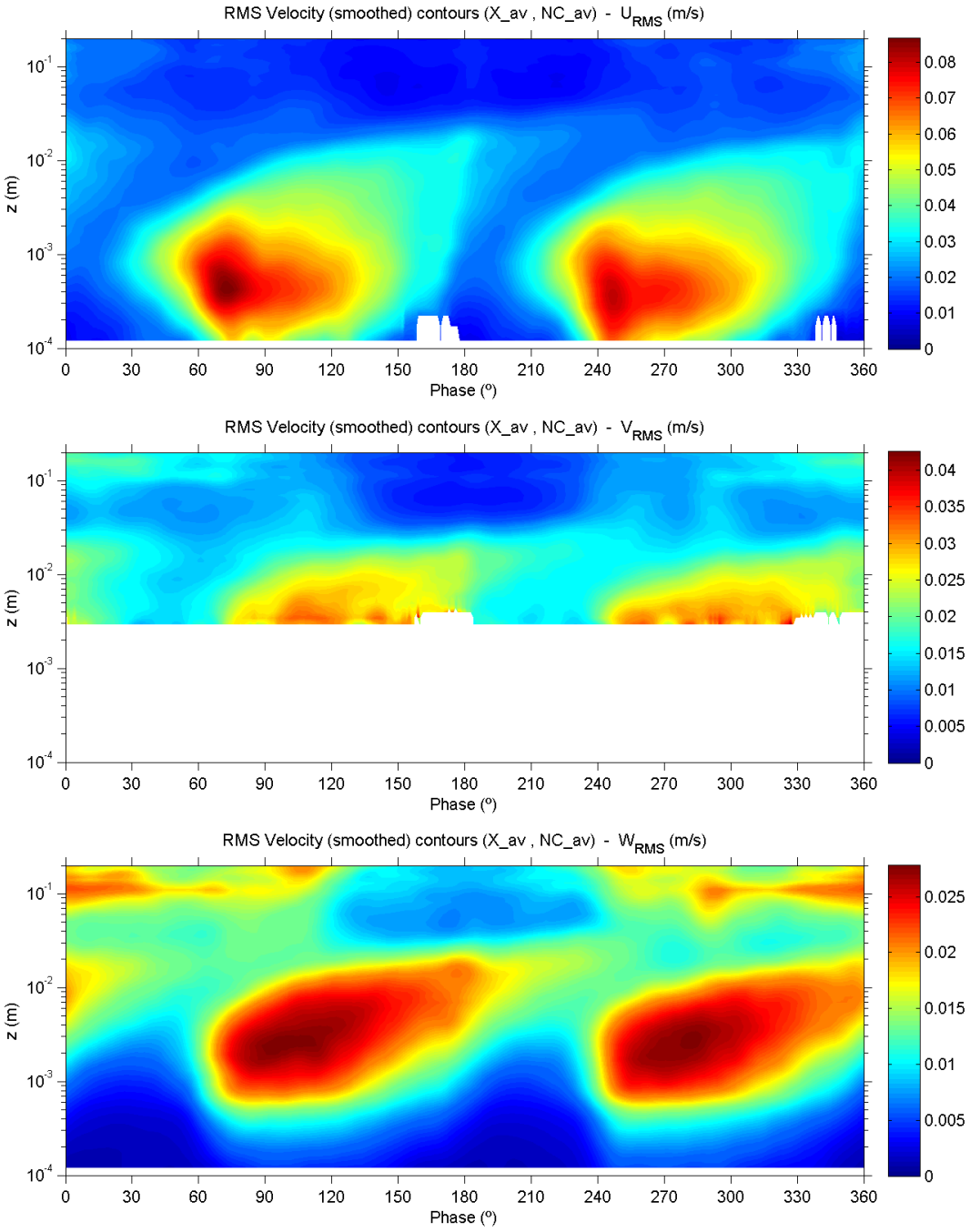


C.9.4.3 Evolution plots

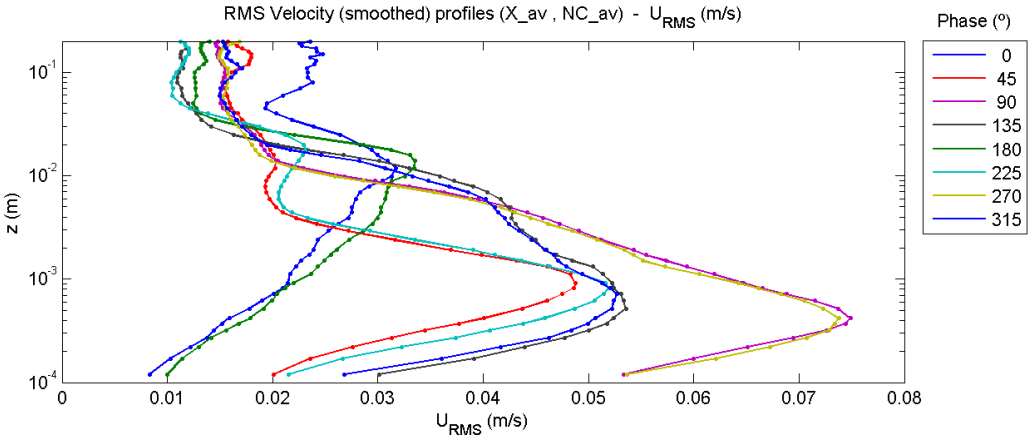


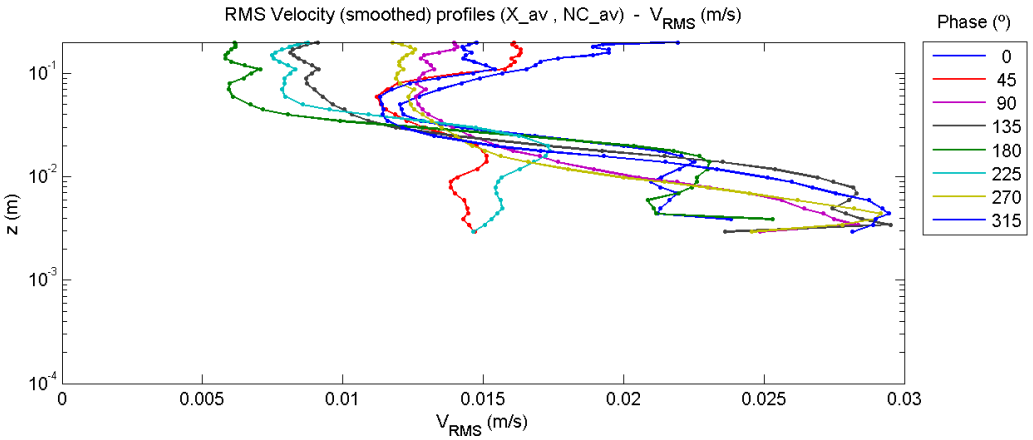
C.9.5 RMS velocity fluctuations

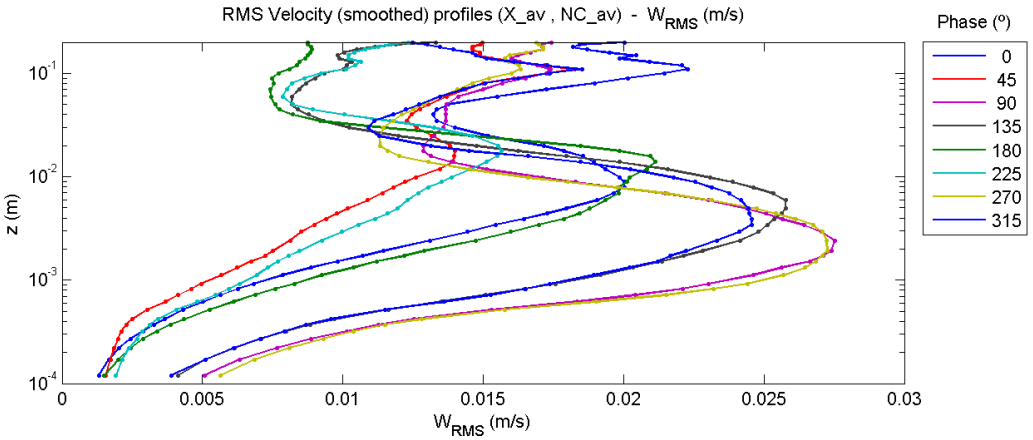
C.9.5.1 Contour plots



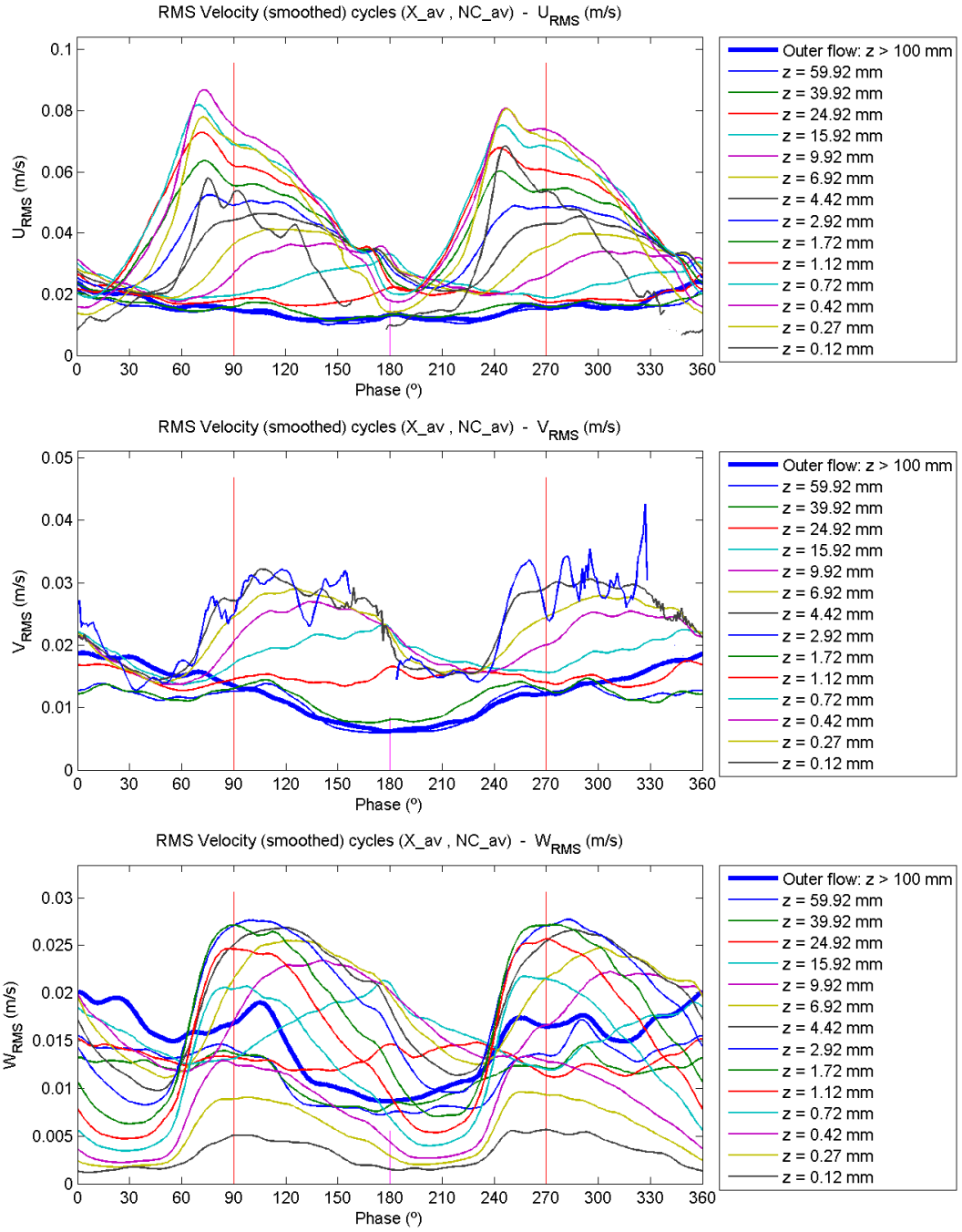
C.9.5.2 Profile plots





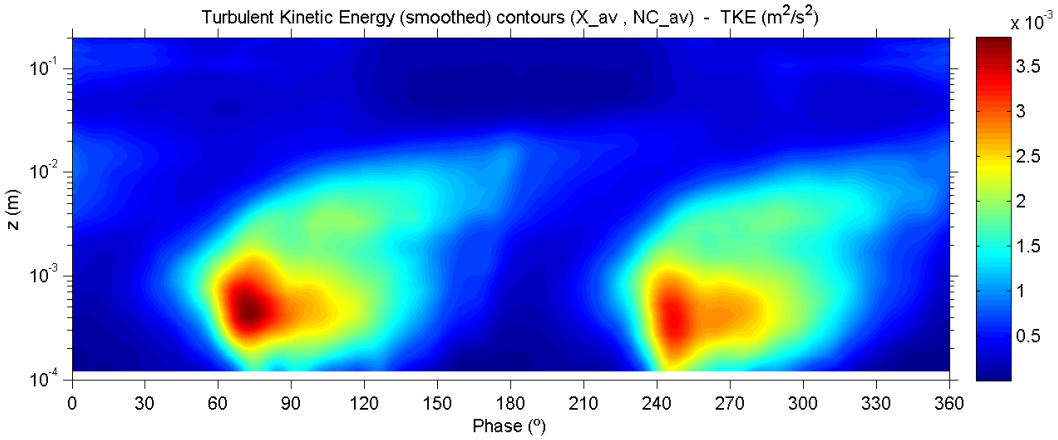


C.9.5.3 Evolution plots

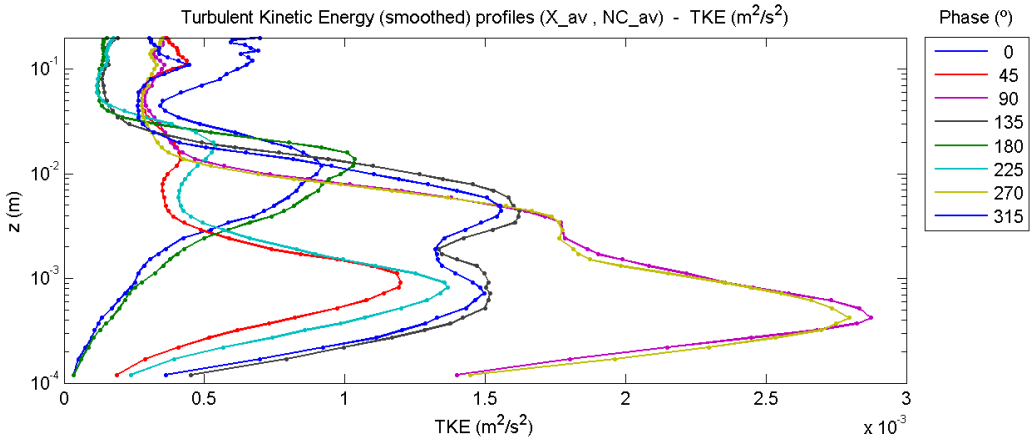


C.9.6 Turbulent kinetic energy

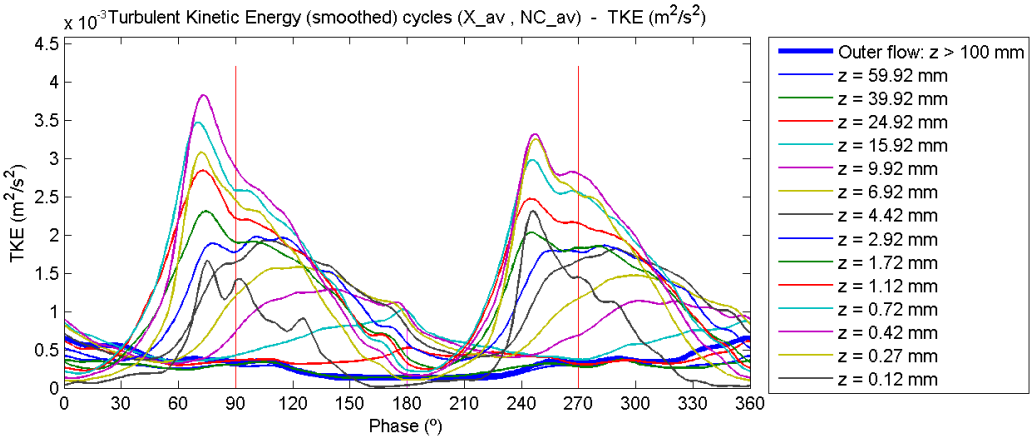
C.9.6.1 Contour plot



C.9.6.2 Profile plot

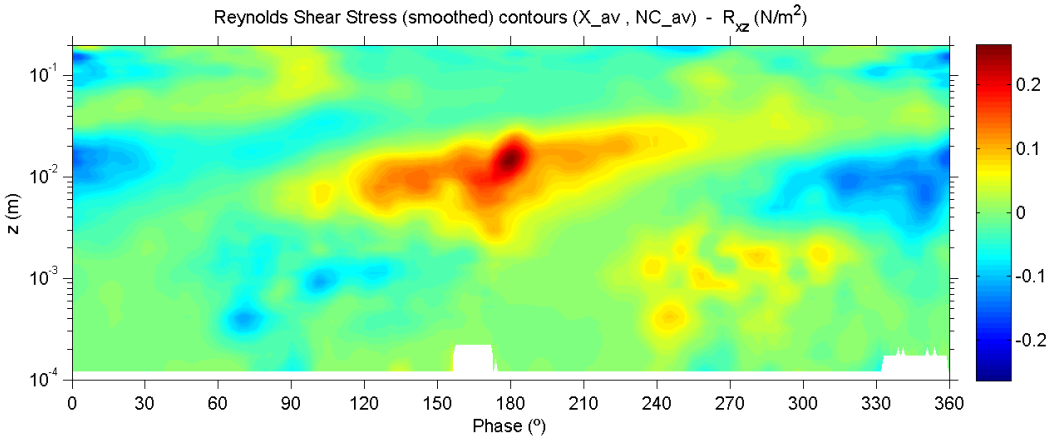


C.9.6.3 Evolution plot

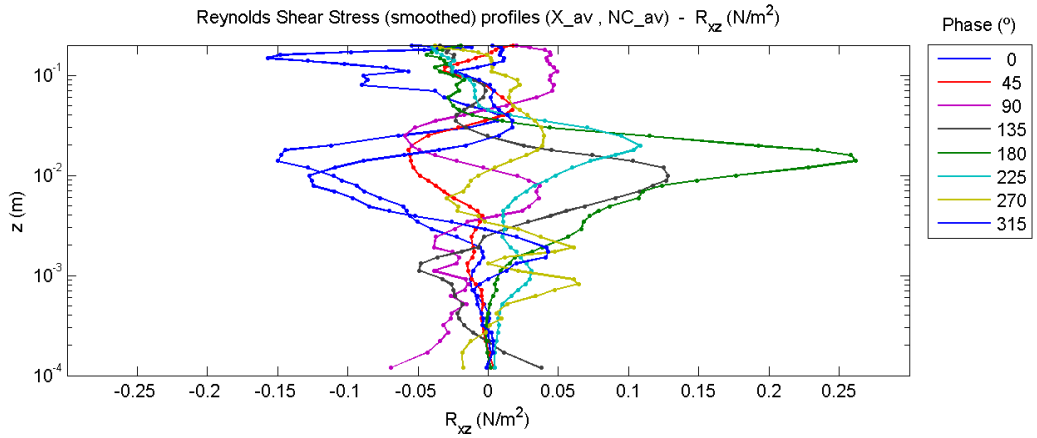


C.9.7 Turbulent shear stress (Reynolds shear stress)

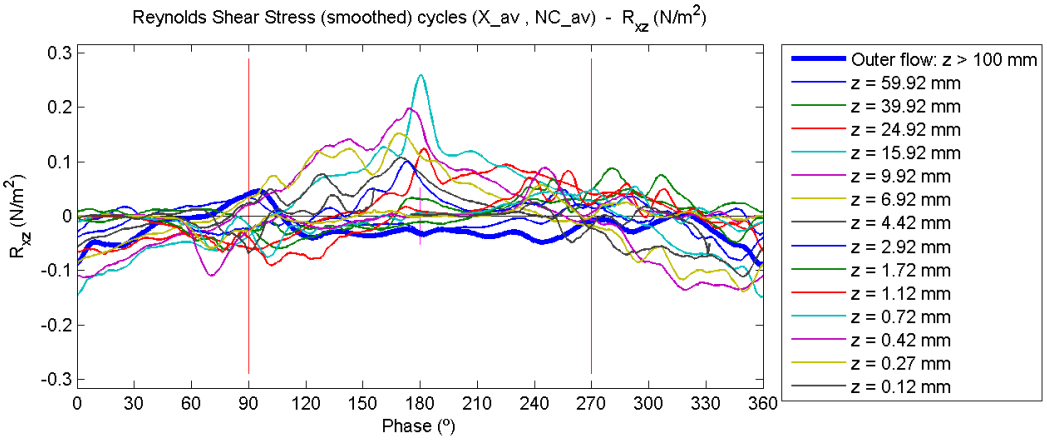
C.9.7.1 Contour plot



C.9.7.2 Profile plot

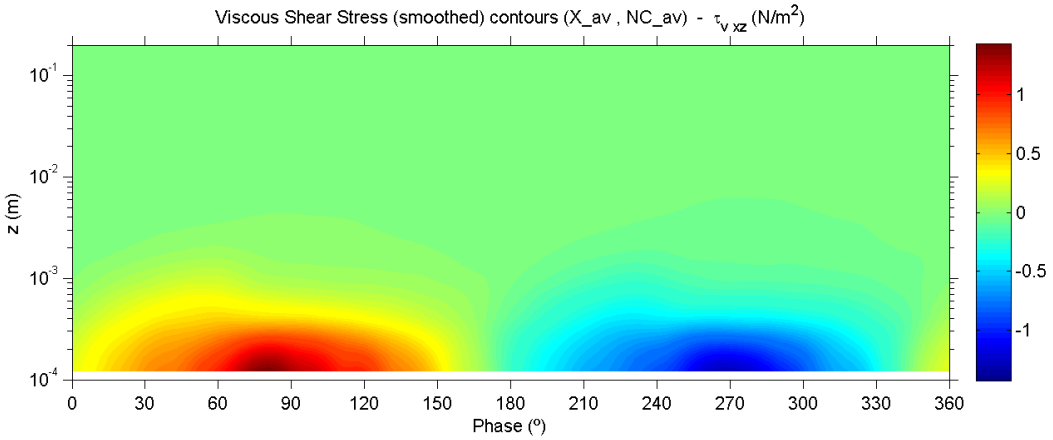


C.9.7.3 Evolution plot

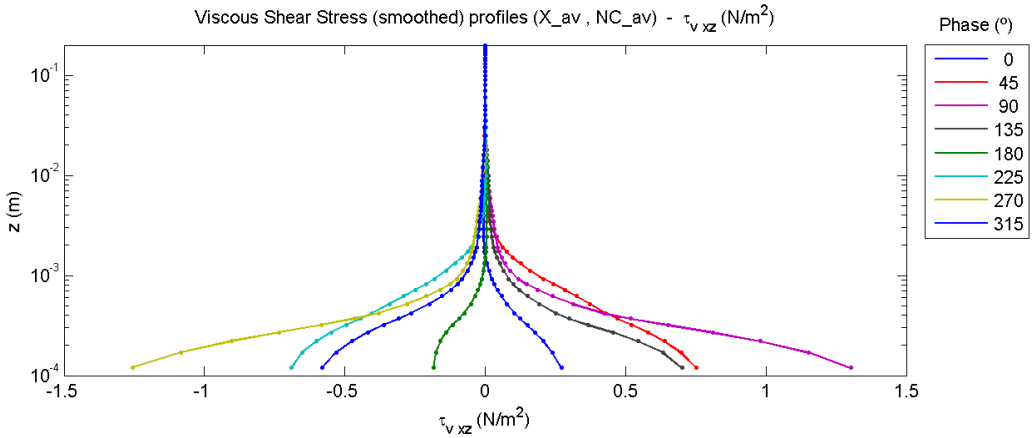


C.9.8 Viscous shear stress

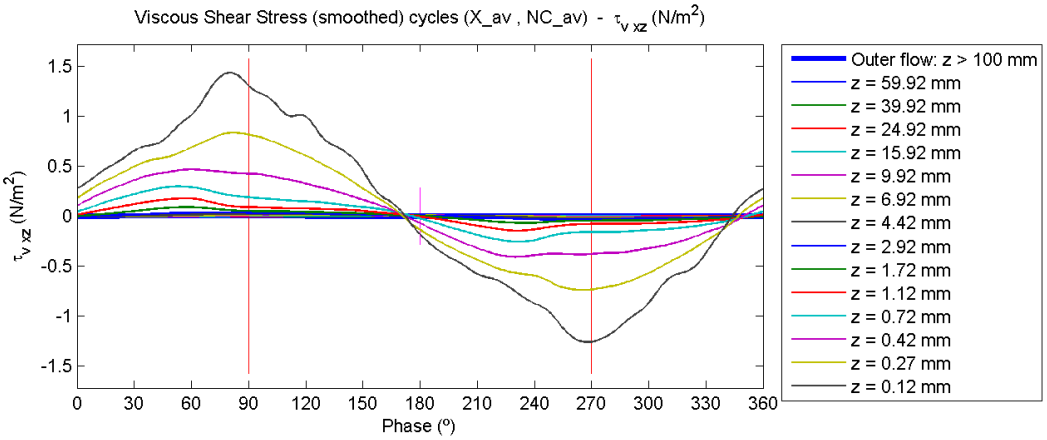
C.9.8.1 Contour plot



C.9.8.2 Profile plot

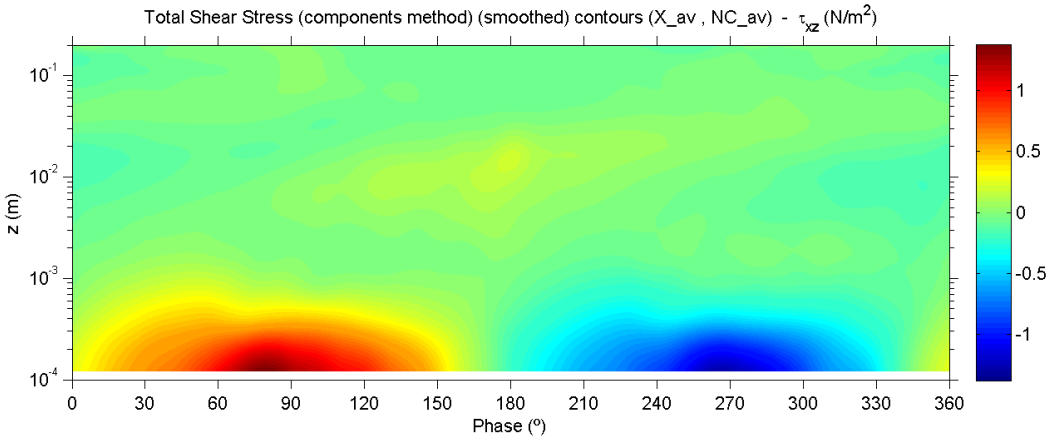


C.9.8.3 Evolution plot

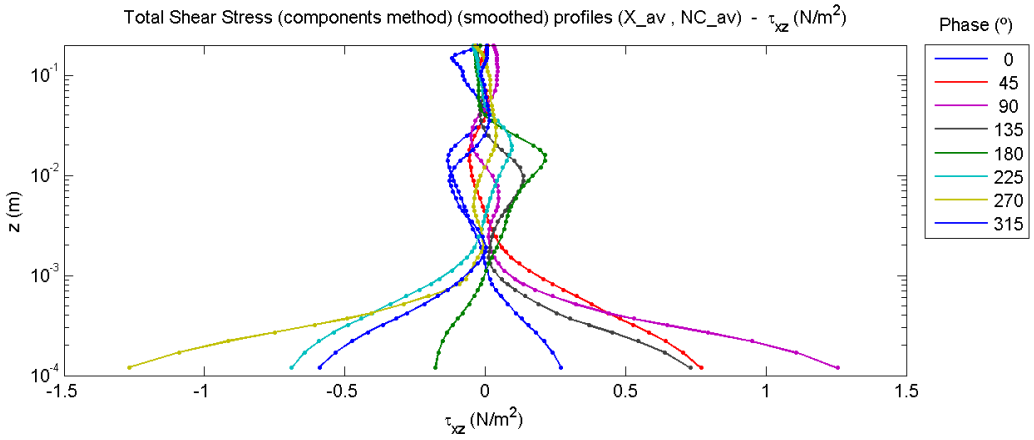


C.9.9 Total shear stress

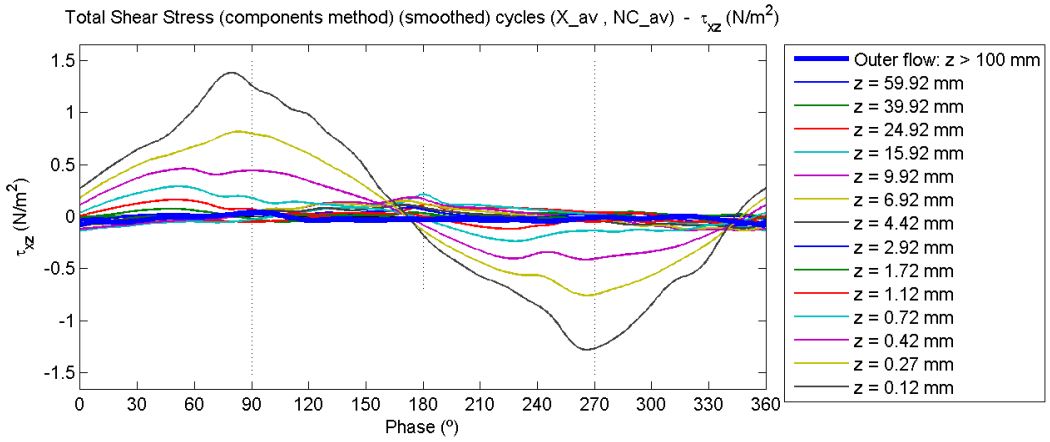
C.9.9.1 Contour plot



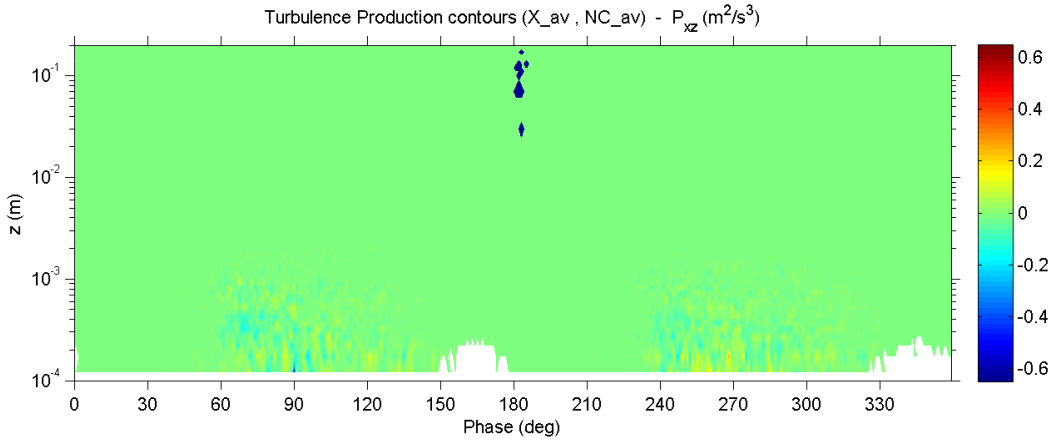
C.9.9.2 Profile plot



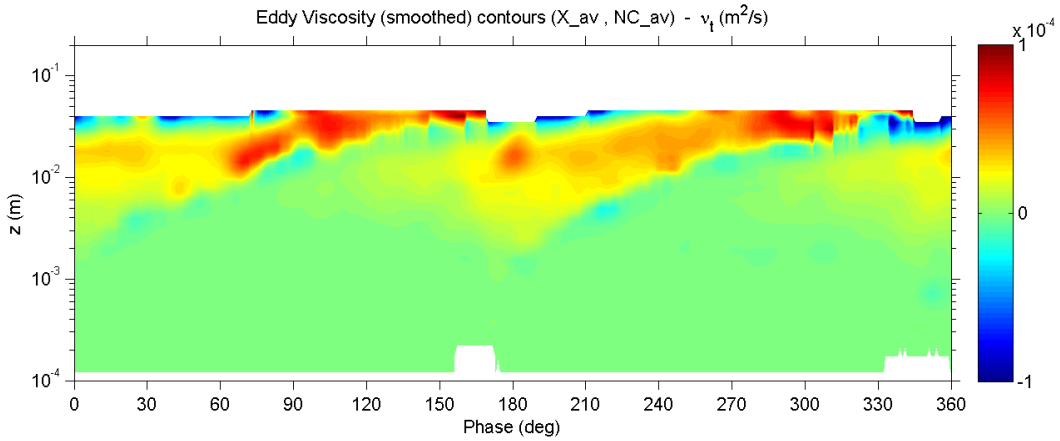
C.9.9.3 Evolution plot



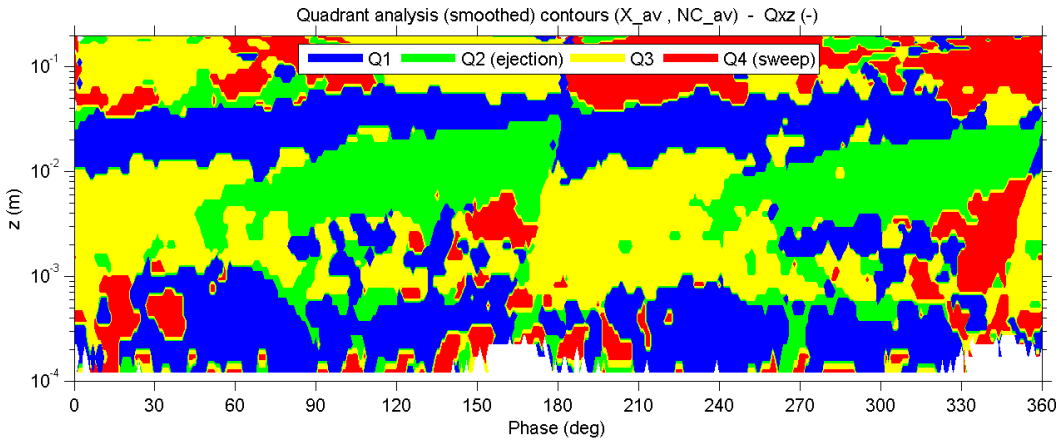
C.9.10 Turbulence production



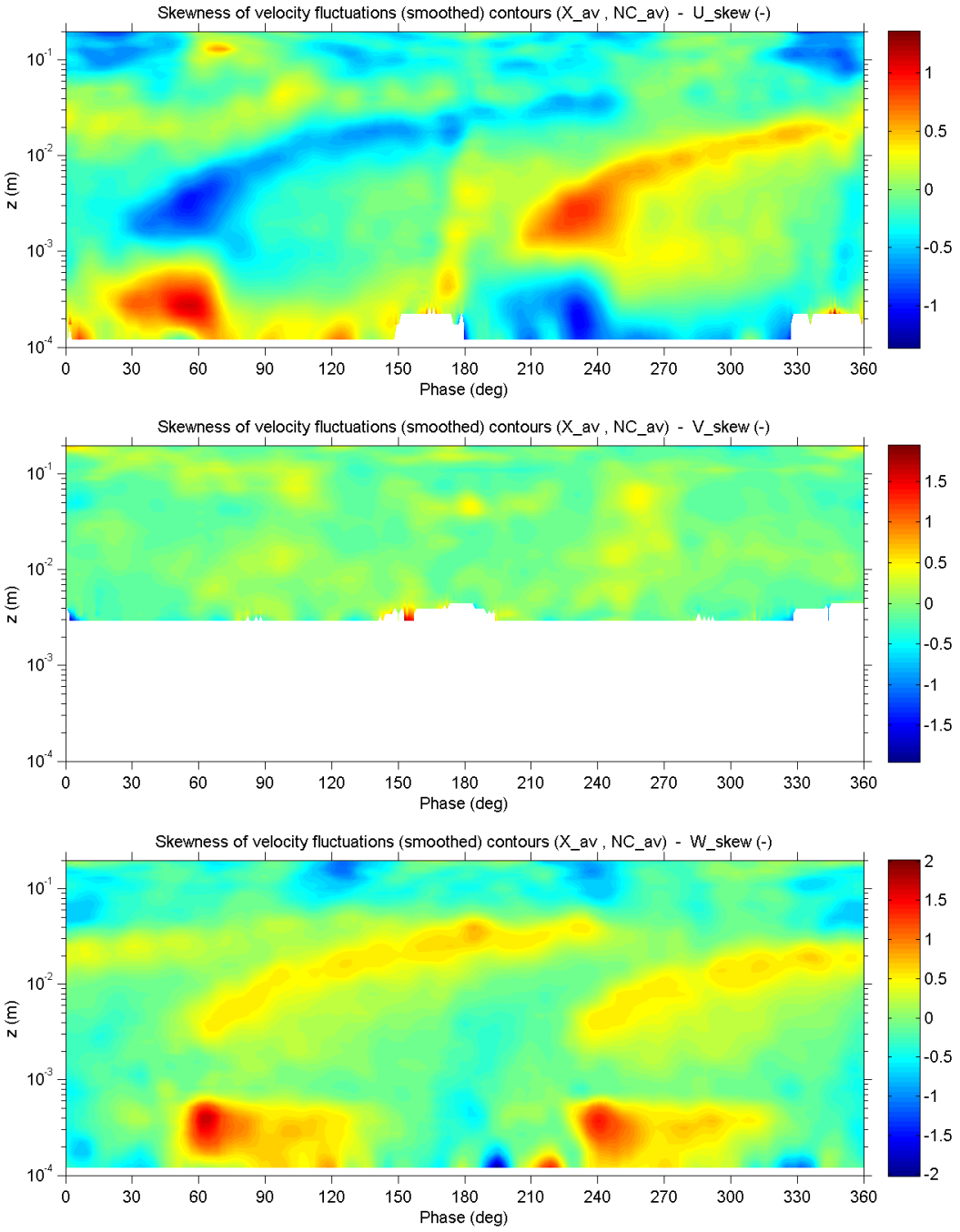
C.9.11 Turbulent viscosity (Eddy viscosity)



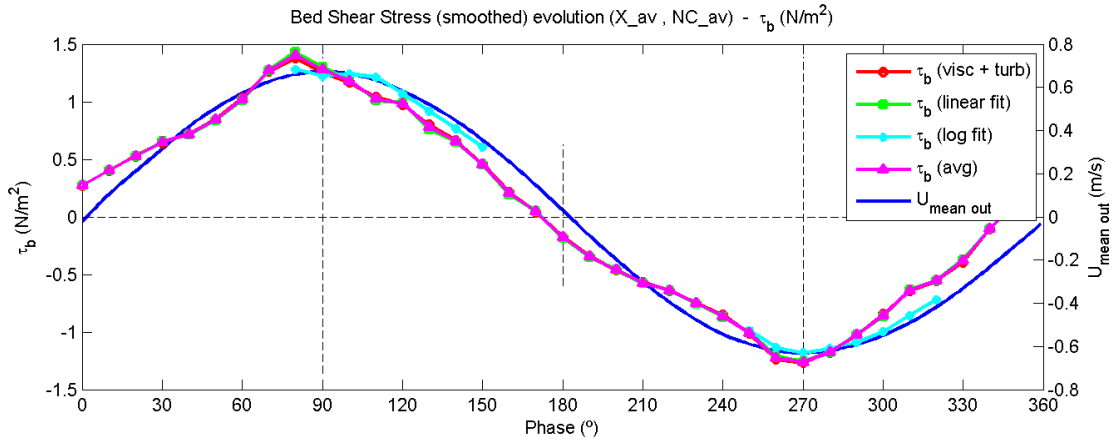
C.9.12 Quadrant analysis



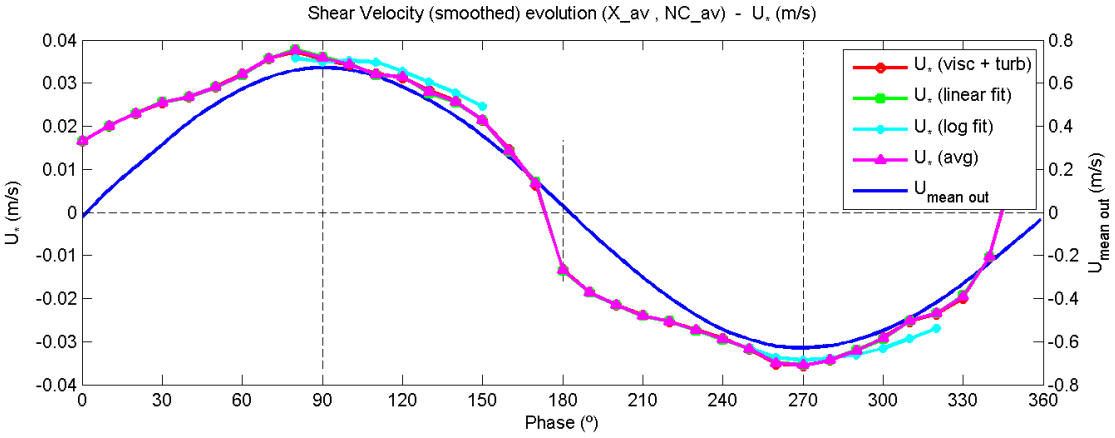
C.9.13 Skewness of velocity fluctuations



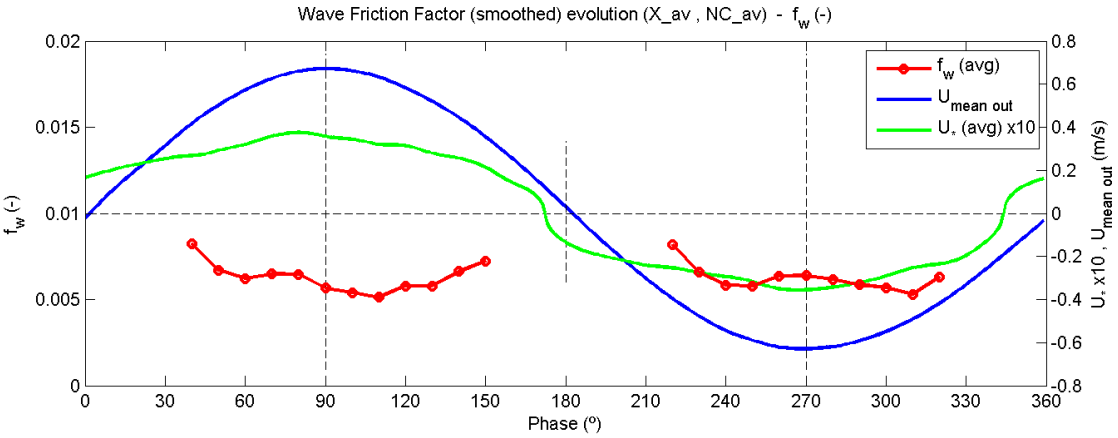
C.9.14 Bed shear stress



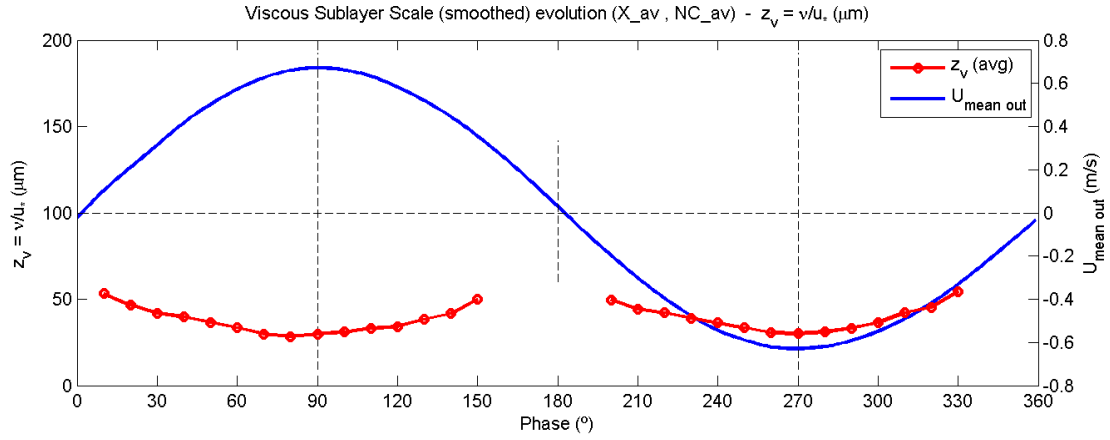
C.9.15 Shear velocity



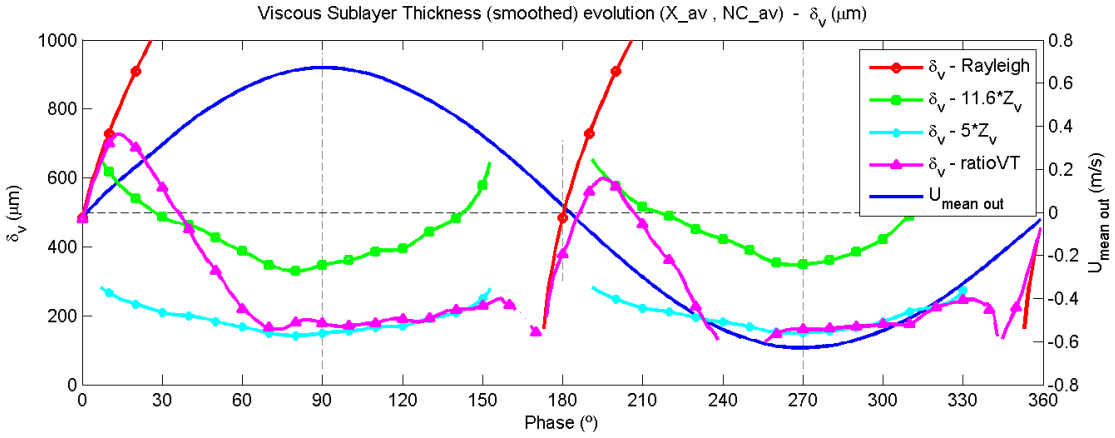
C.9.16 Wave friction factor



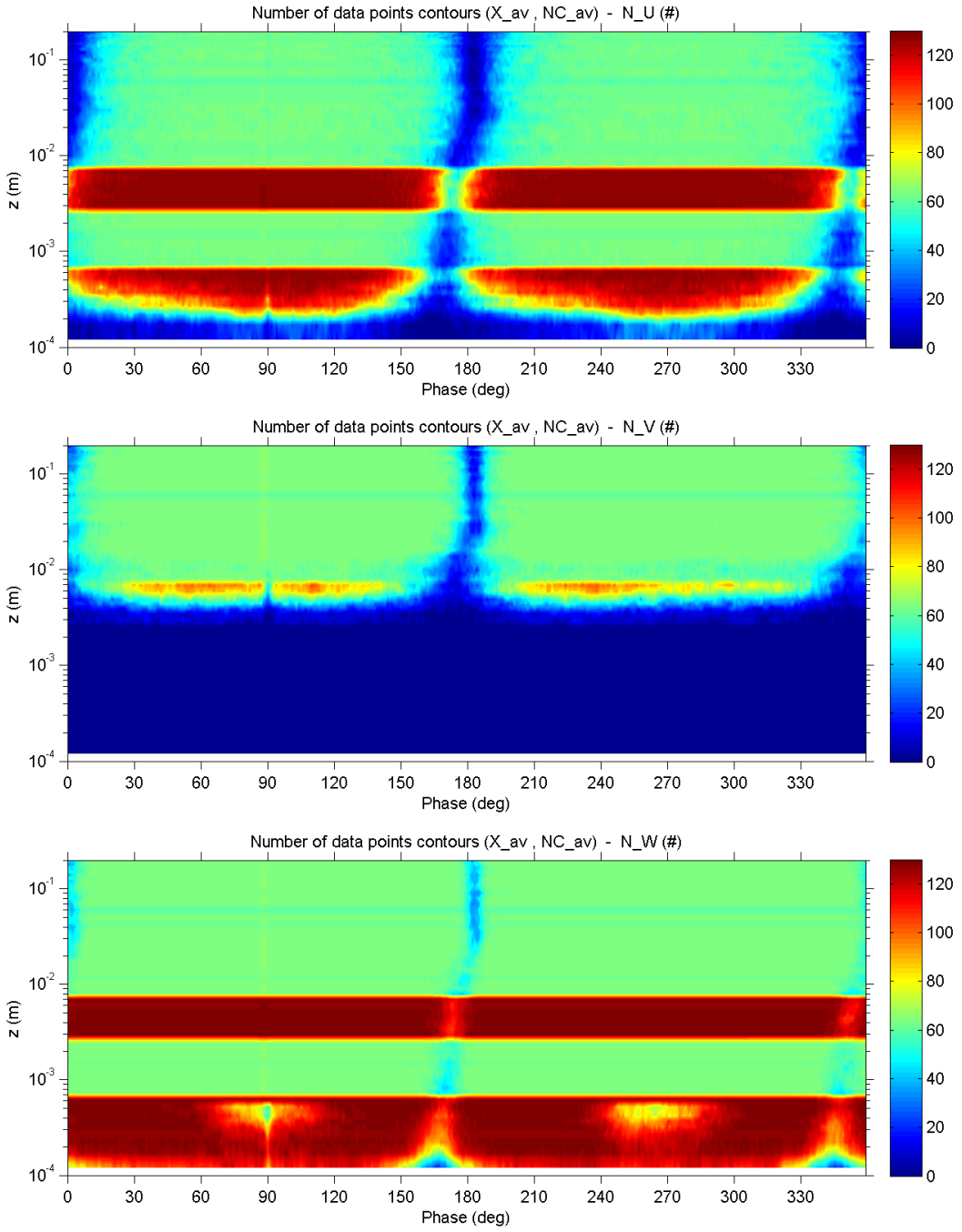
C.9.17 Viscous length scale



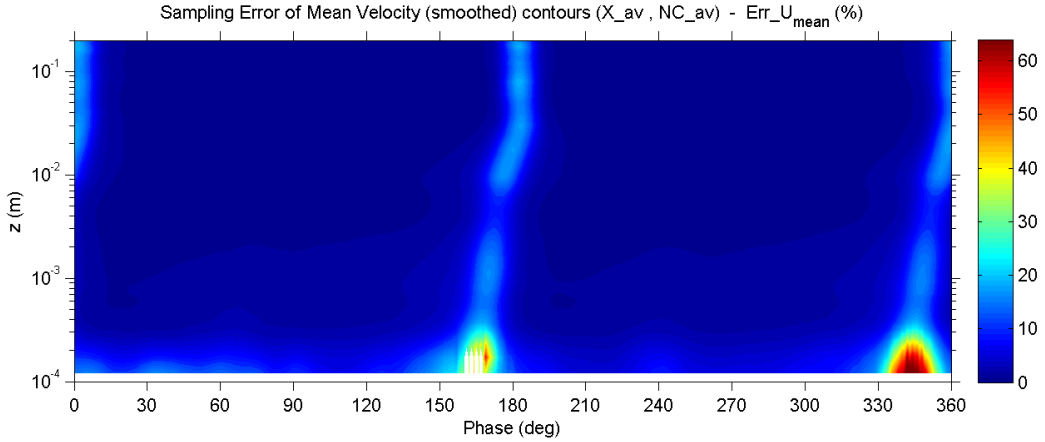
C.9.18 Viscous sublayer thickness



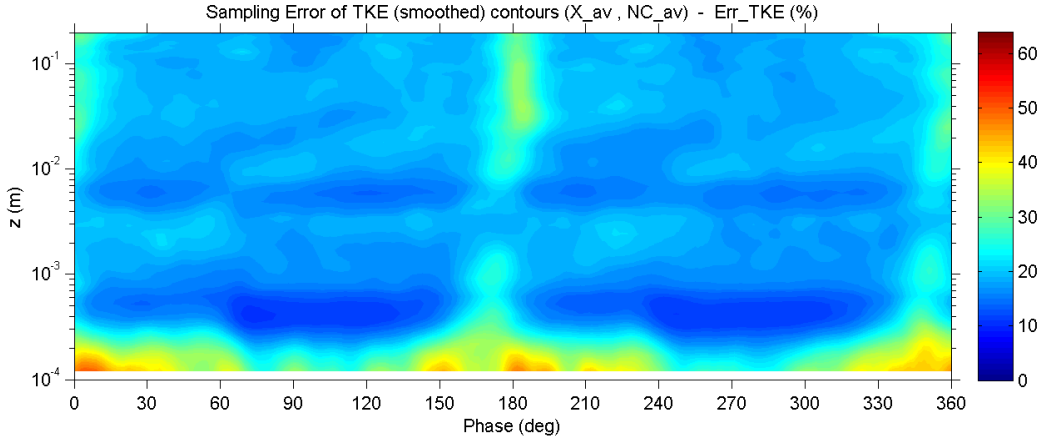
C.9.19 Number of valid data points



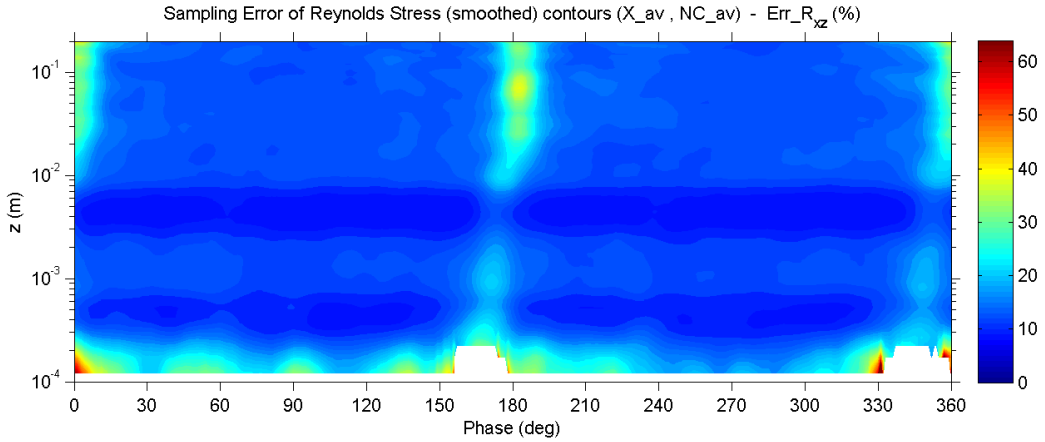
C.9.20 Percent error of mean velocity



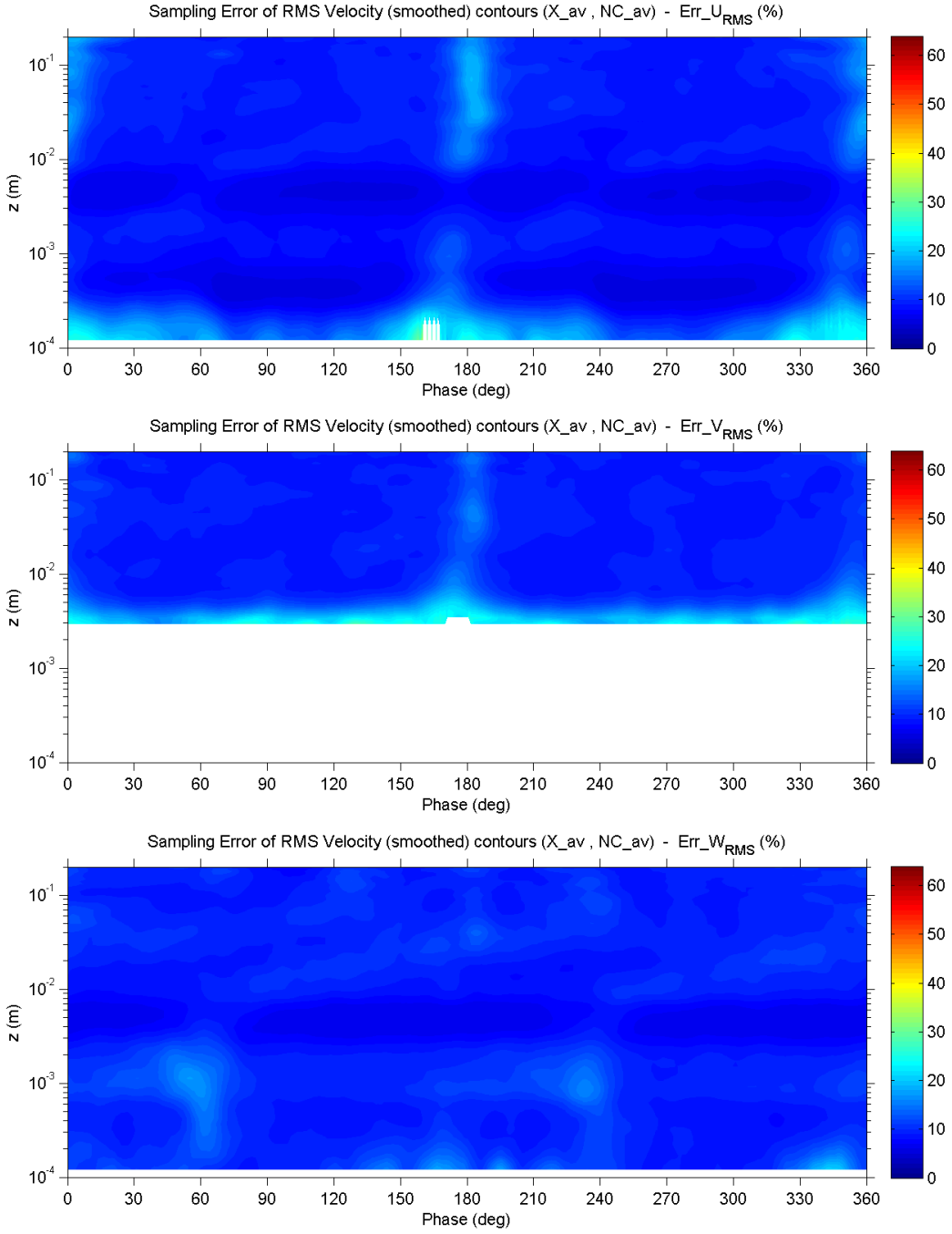
C.9.21 Percent error of turbulent kinetic energy



C.9.22 Percent error of Reynolds shear stress



C.9.23 Percent error of RMS velocity fluctuations



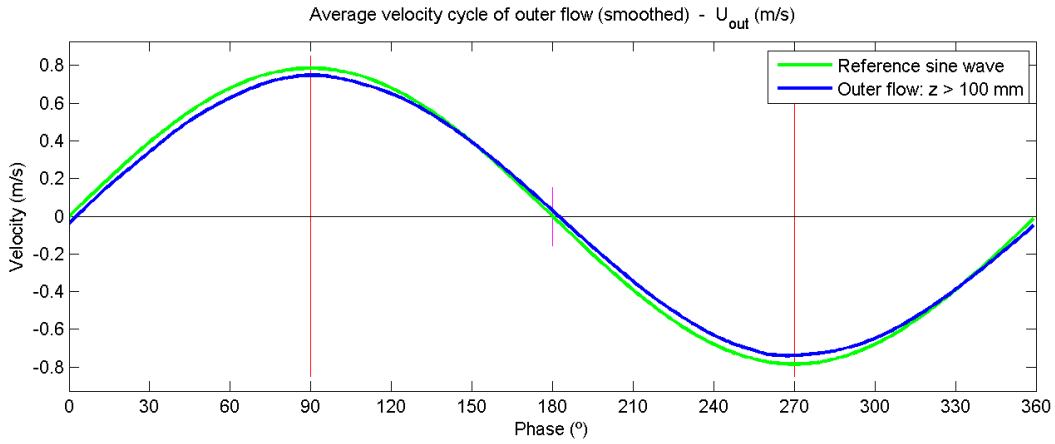
C.10 Experiment no. 10

C.10.1 Main parameters

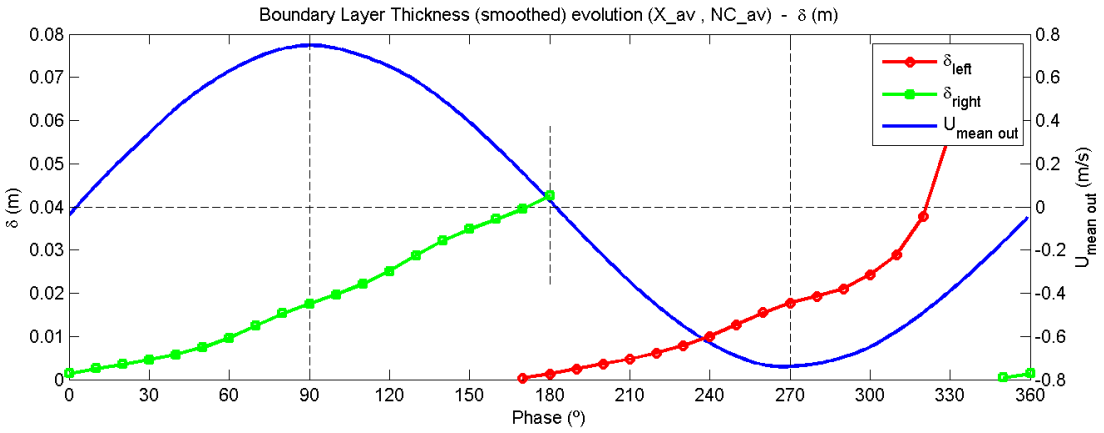
Exp no.	*Temp* (ºC)	ρ (kg/m^3)	ν (m^2/s)	*T* (s)	*2a* (m)	U_{out_max} (m/s)	Re_w (-)	N_{max} (cycles)
10	20.0	998.23	1.02E-06	10	2.368	0.744	8.7E+05	130

Exp no.	U_{*max} (m/s)	τ_{b_max} (N/m^2)	$\Delta\phi$ (deg)	f_{w_ref} (-)	Z_{v_min} (mm)	δ_{v_90} (mm)	δ_{90} (mm)	δ_{top} (mm)
10	0.039	1.51	2.0	0.0054	0.026	0.150	17.7	45

C.10.2 Outer flow velocity

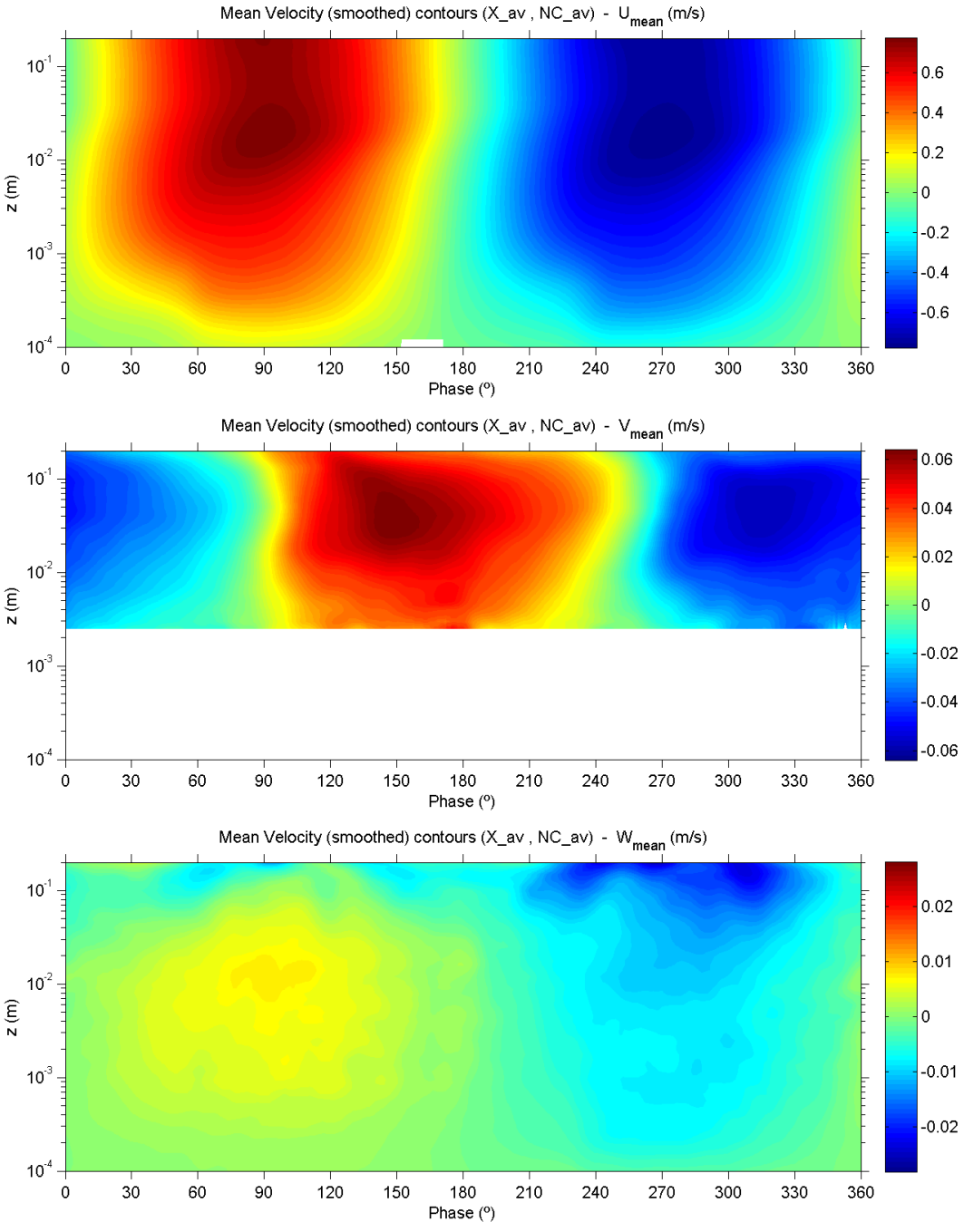


C.10.3 Boundary layer thickness

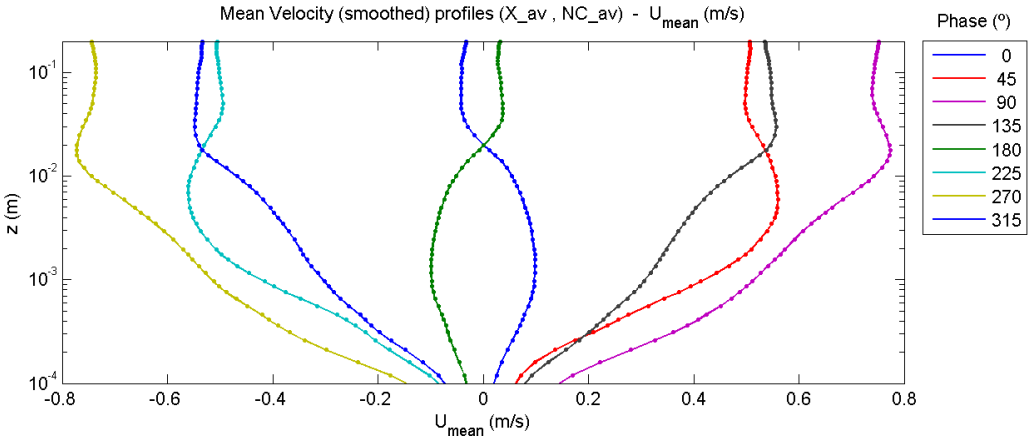


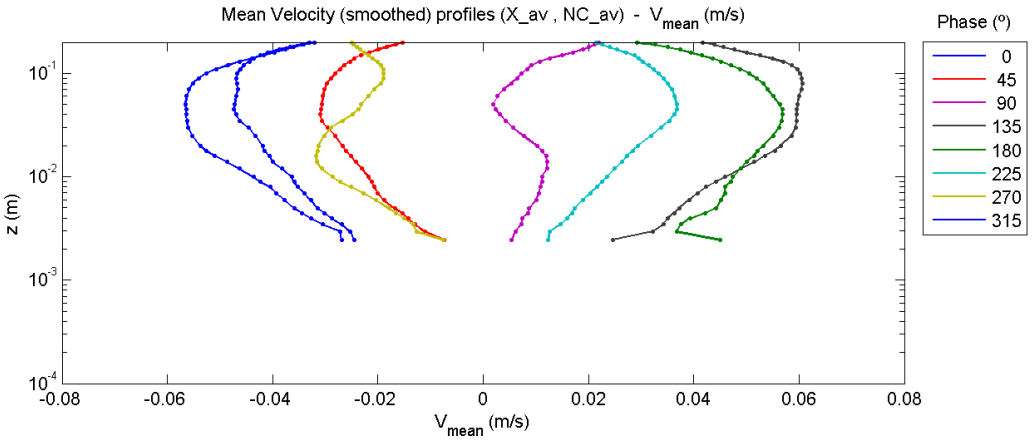
C.10.4 Mean velocities

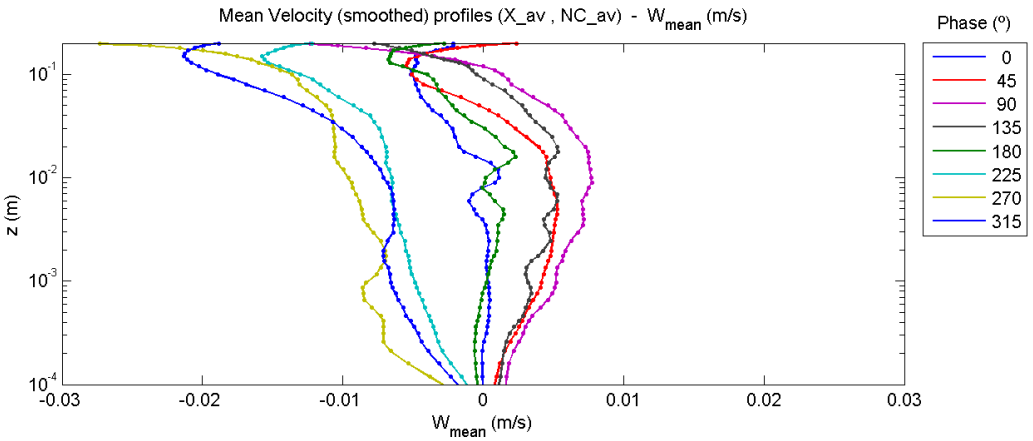
C.10.4.1 Contour plots



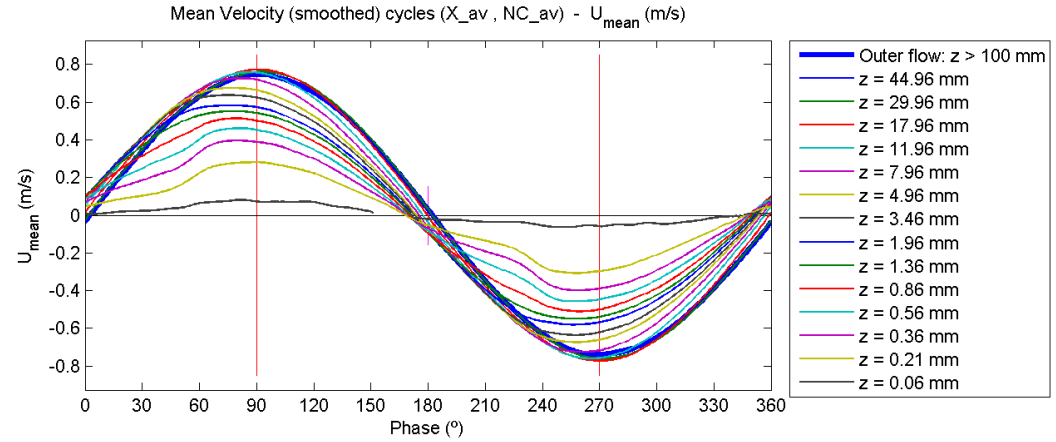
C.10.4.2 Profile plots

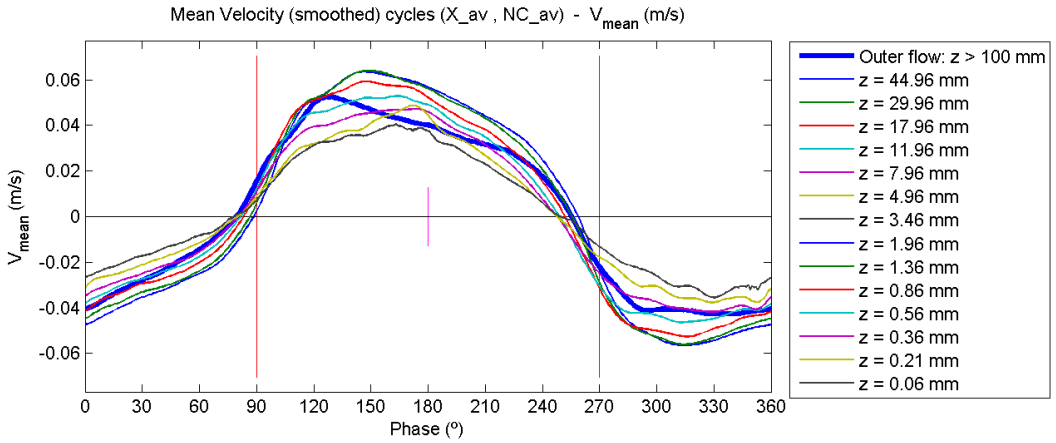


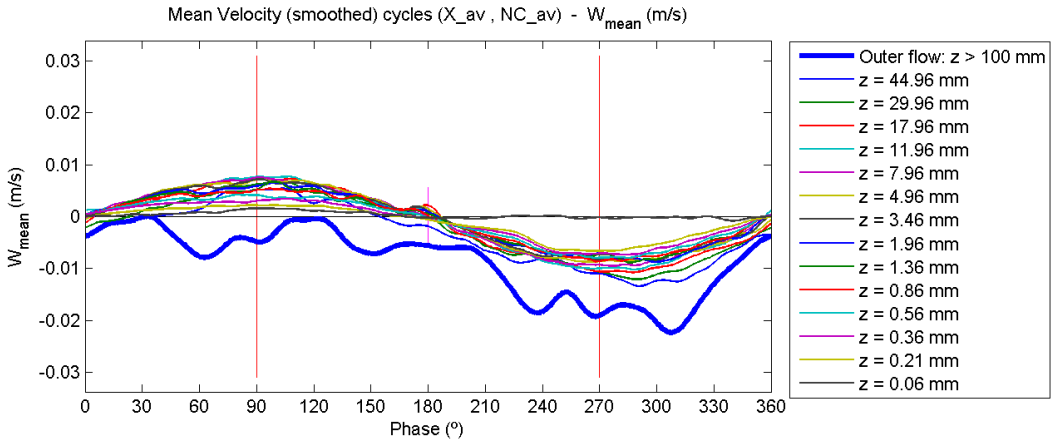




C.10.4.3 Evolution plots

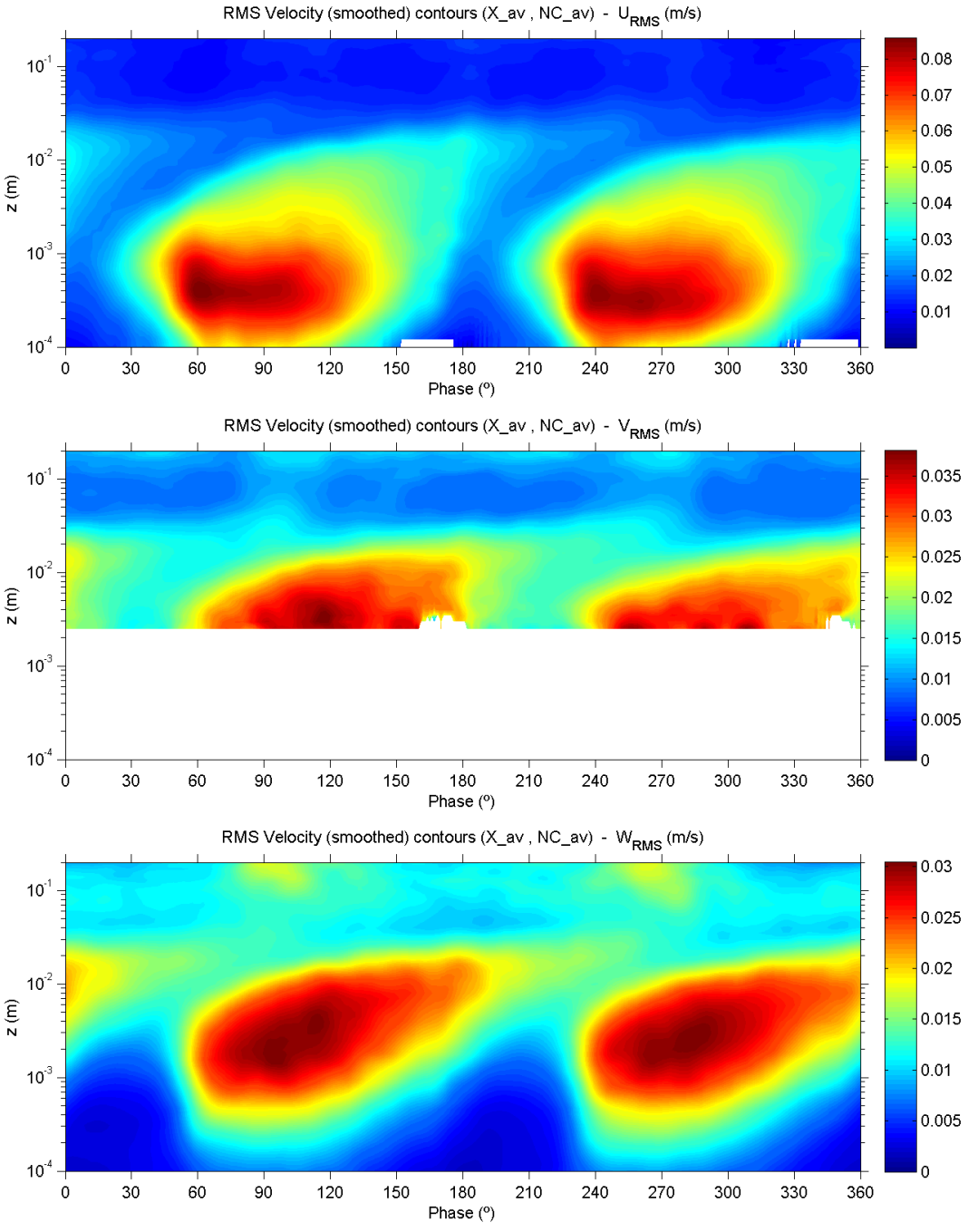




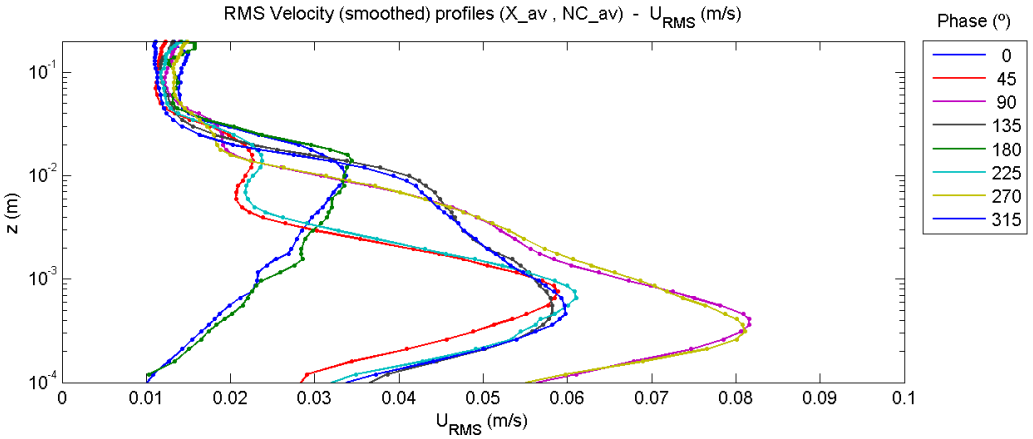


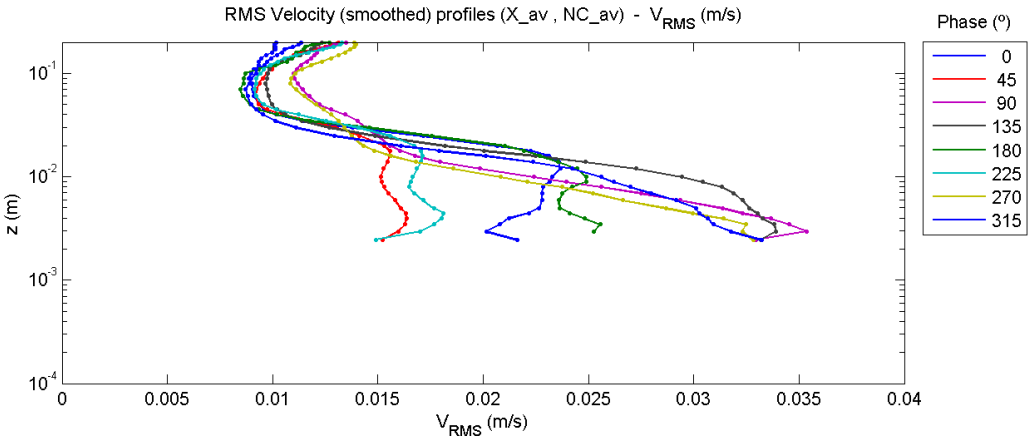
C.10.5 RMS velocity fluctuations

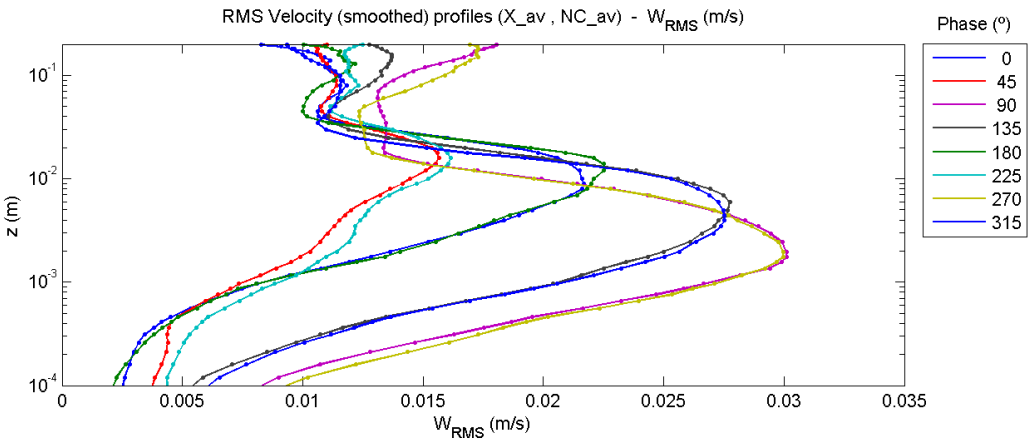
C.10.5.1 Contour plots



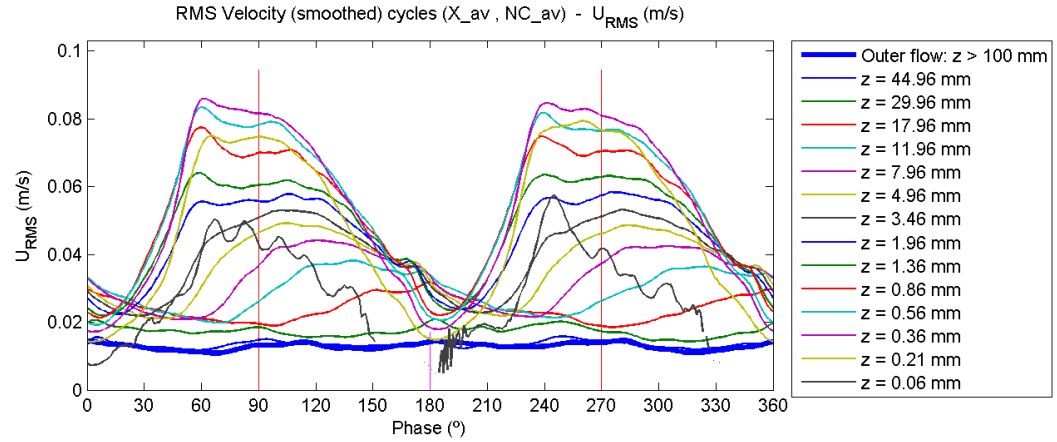
C.10.5.2 Profile plots

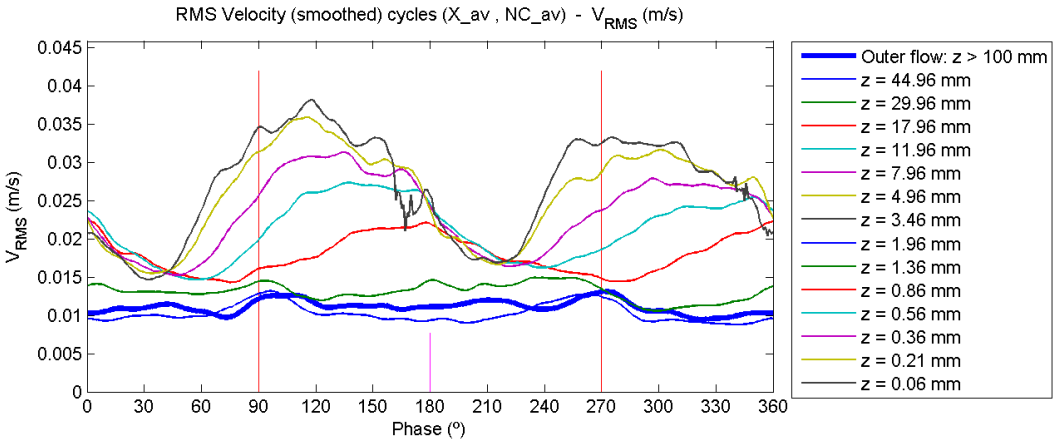


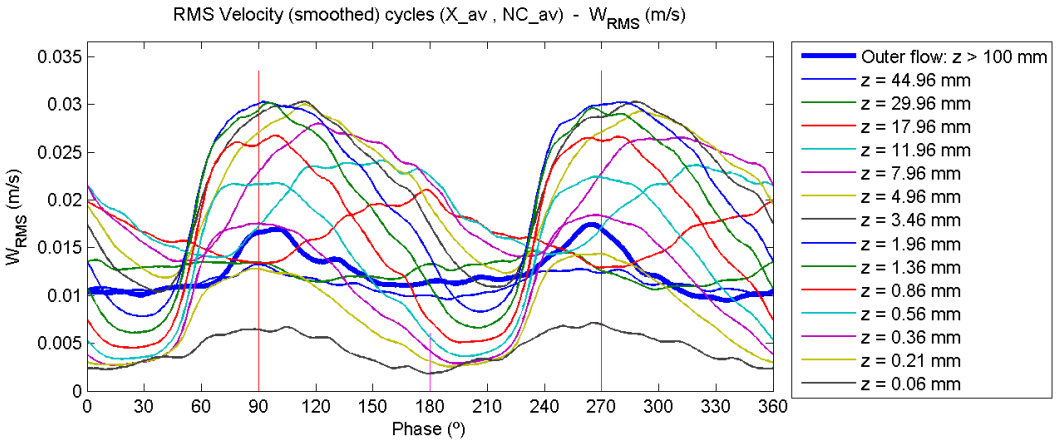




C.10.5.3 Evolution plots

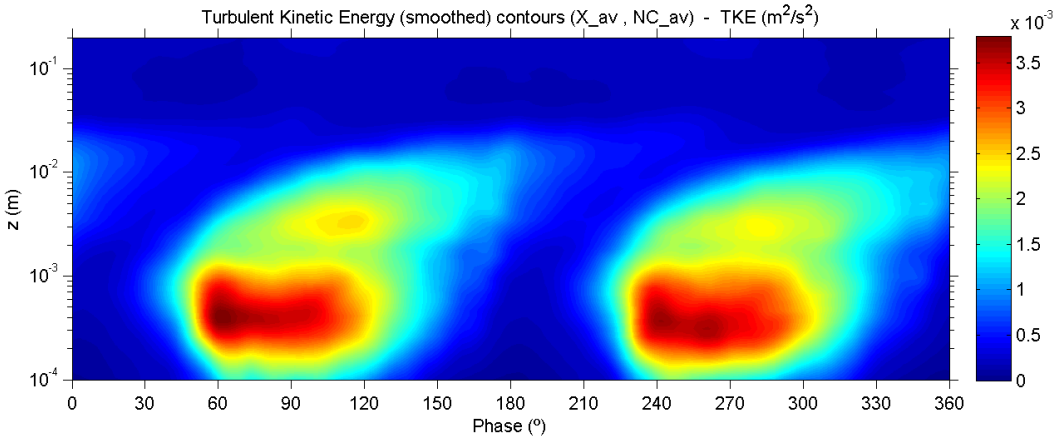




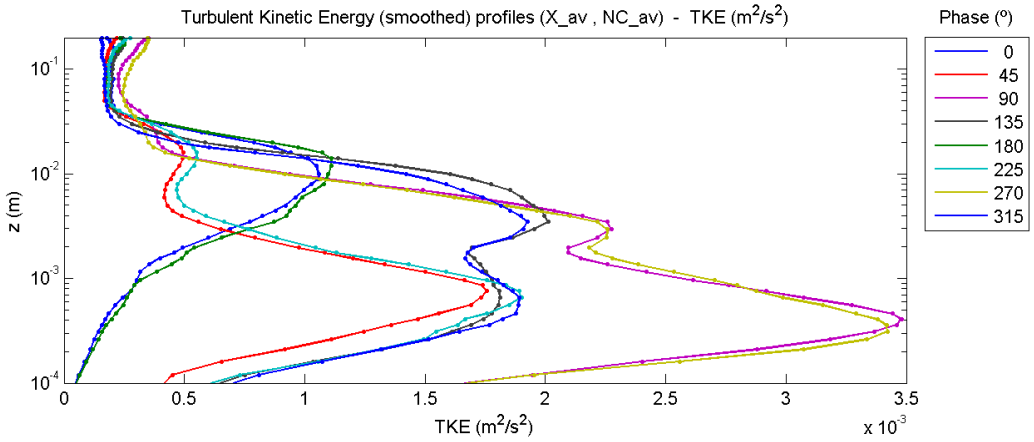


C.10.6 Turbulent kinetic energy

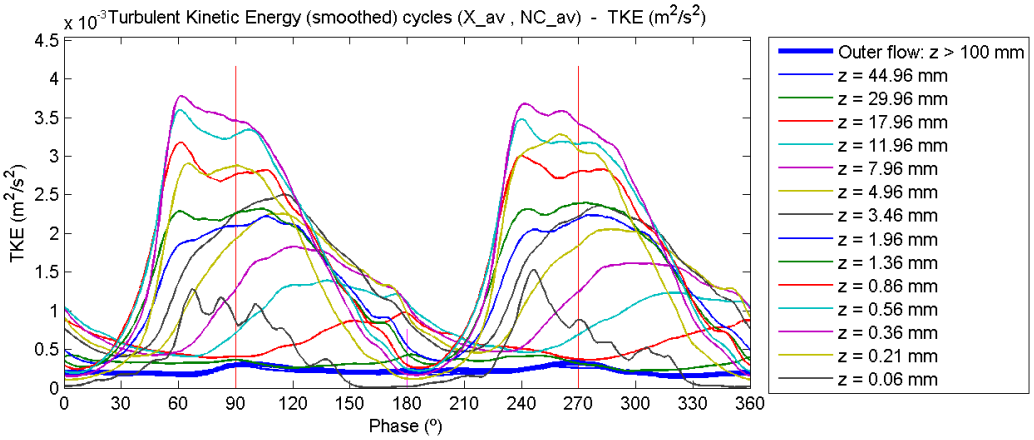
C.10.6.1 Contour plot



C.10.6.2 Profile plot

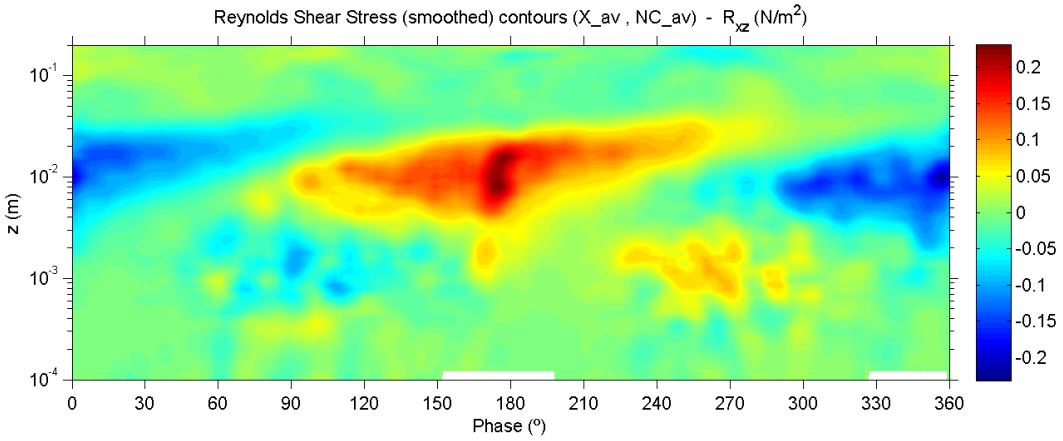


C.10.6.3 Evolution plot

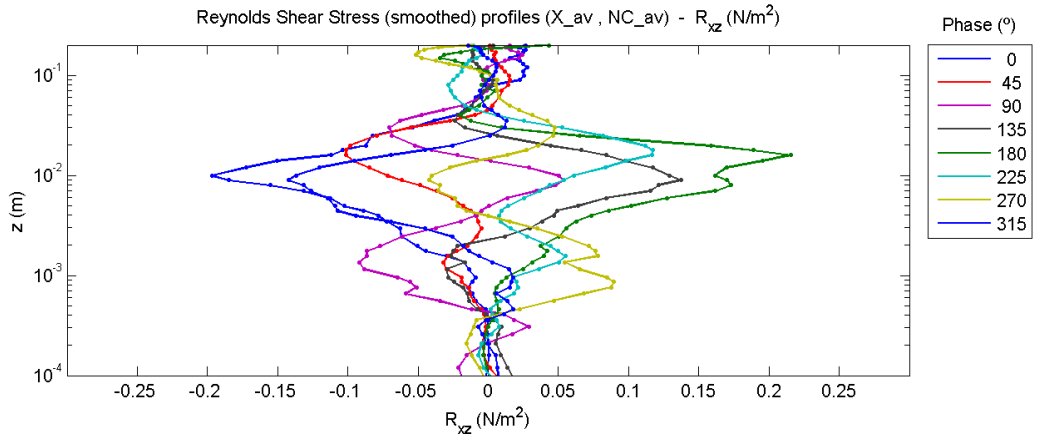


C.10.7 Turbulent shear stress (Reynolds shear stress)

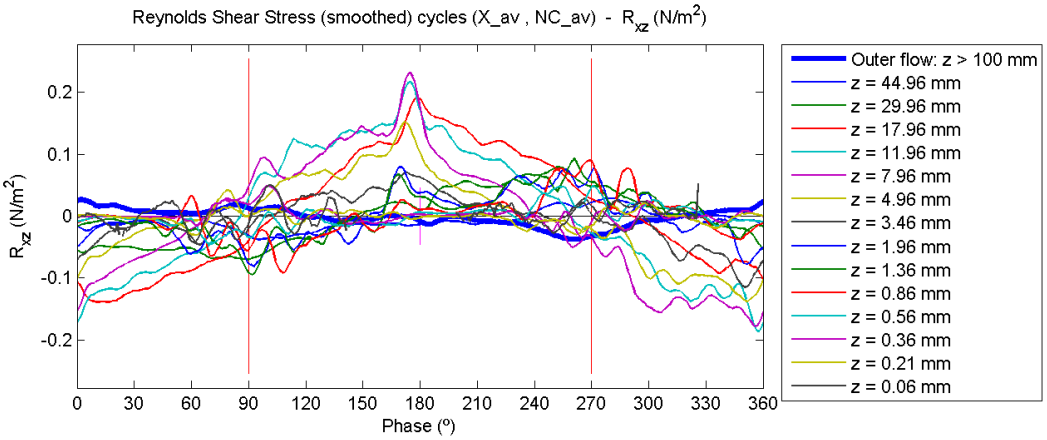
C.10.7.1 Contour plot



C.10.7.2 Profile plot

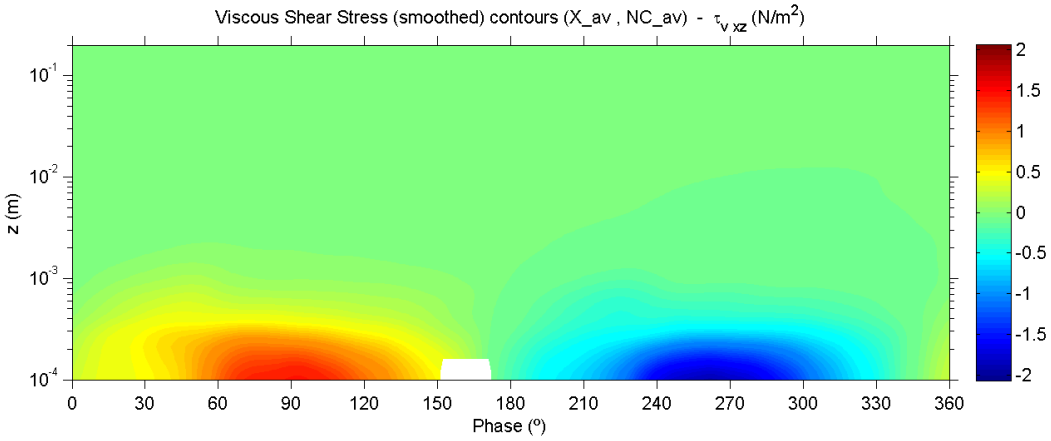


C.10.7.3 Evolution plot

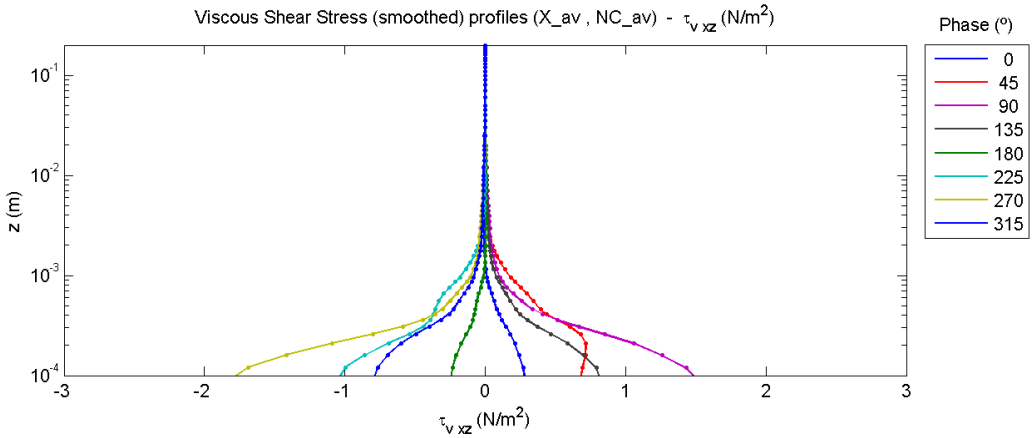


C.10.8 Viscous shear stress

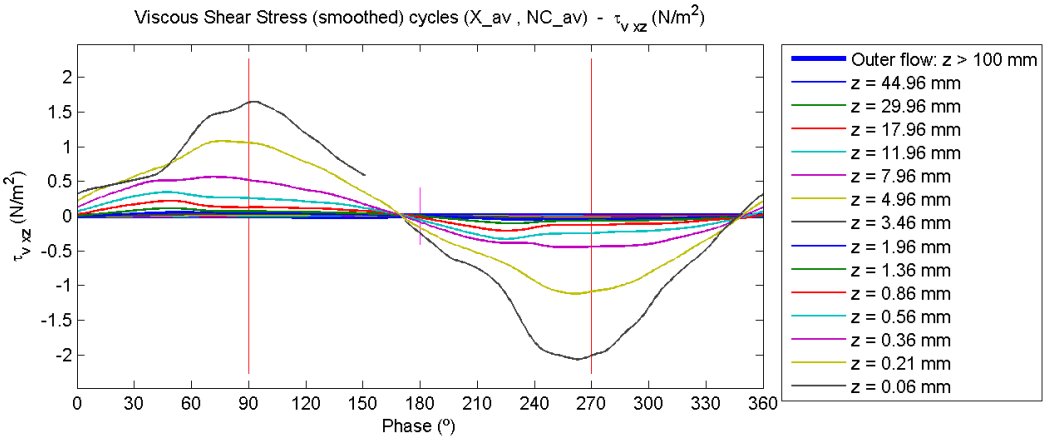
C.10.8.1 Contour plot



C.10.8.2 Profile plot

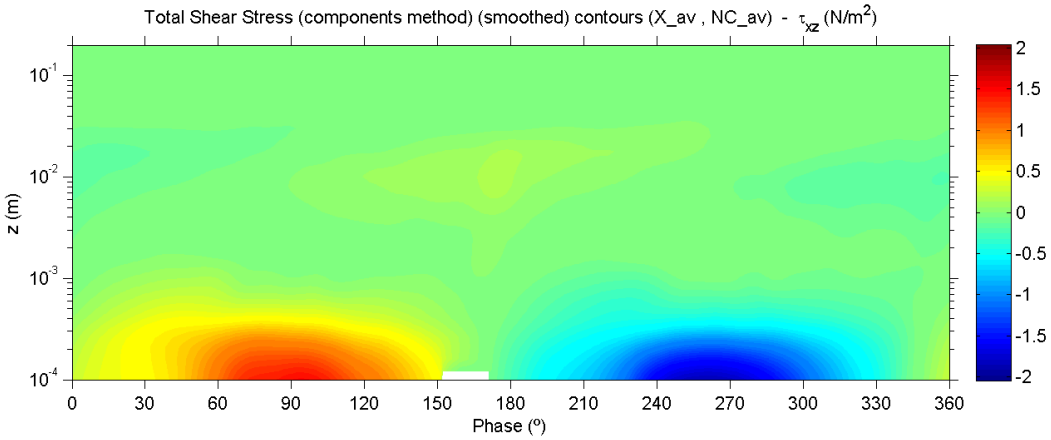


C.10.8.3 Evolution plot

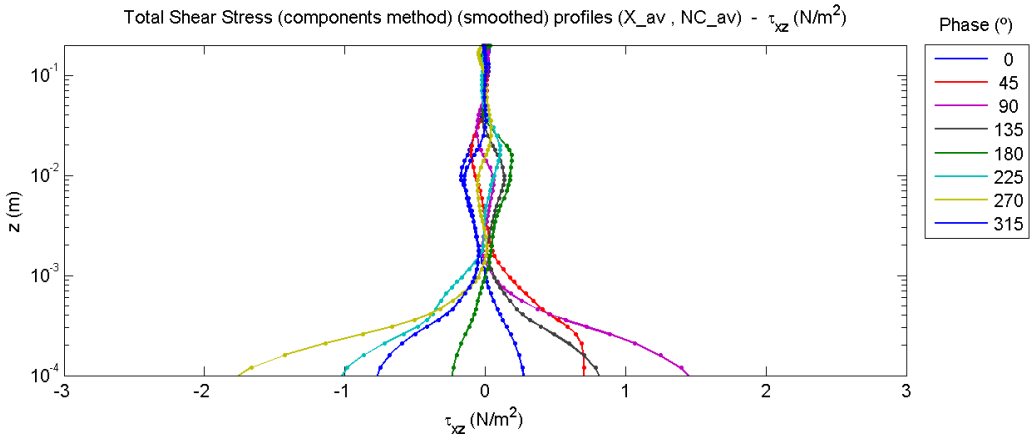


C.10.9 Total shear stress

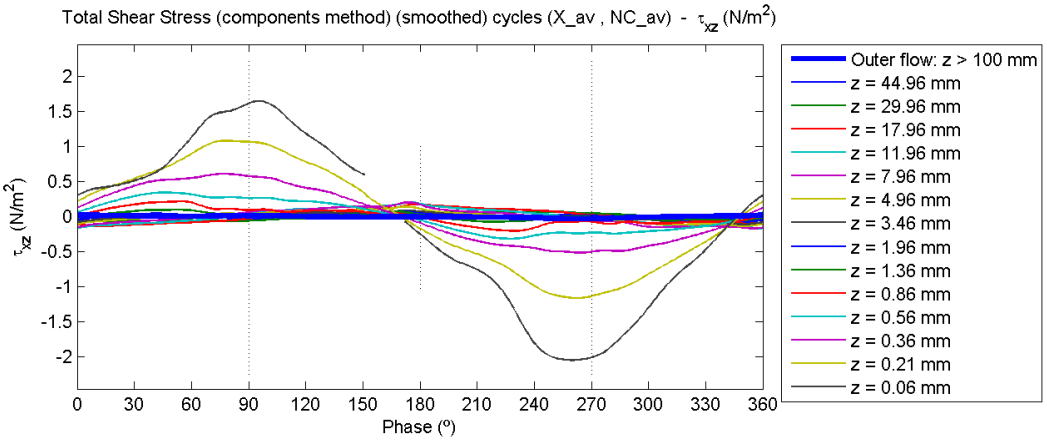
C.10.9.1 Contour plot



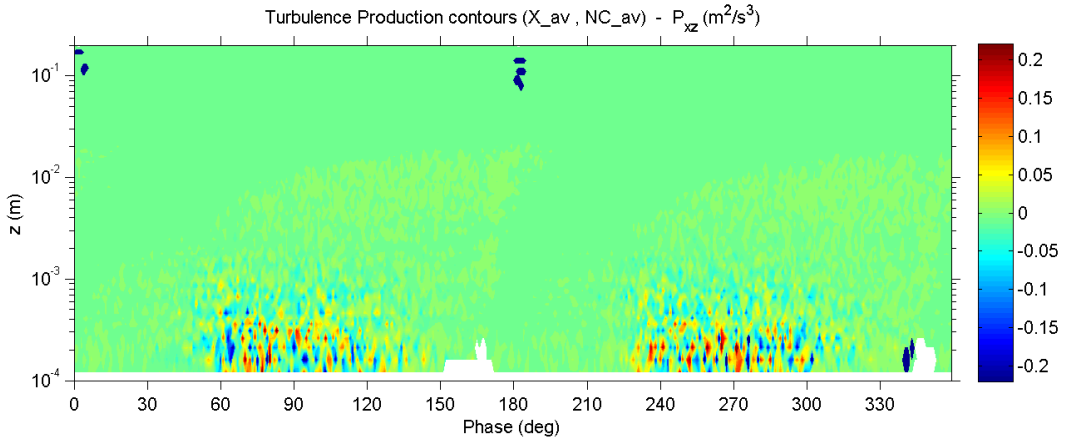
C.10.9.2 Profile plot



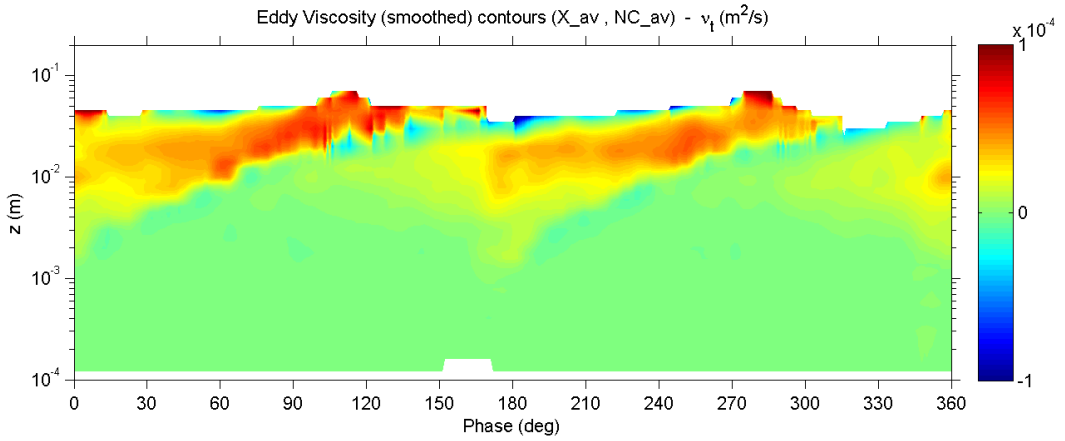
C.10.9.3 Evolution plot



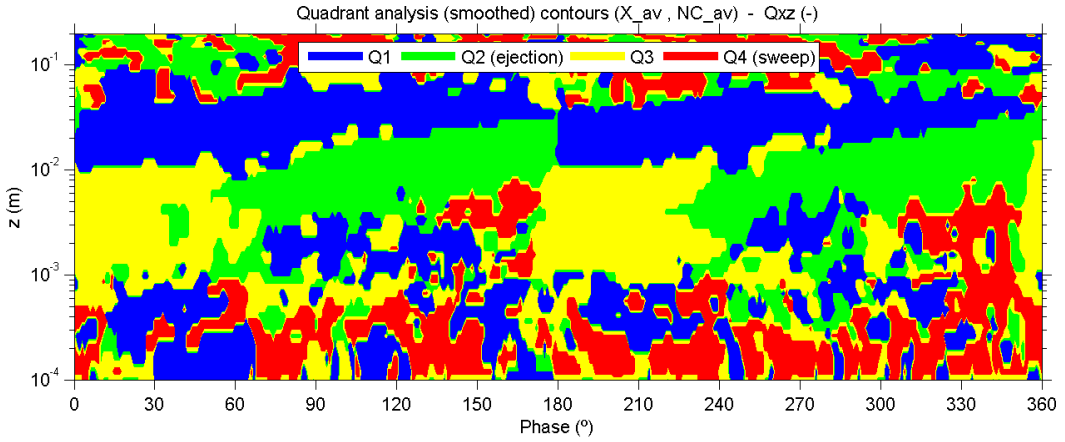
C.10.10 Turbulence production



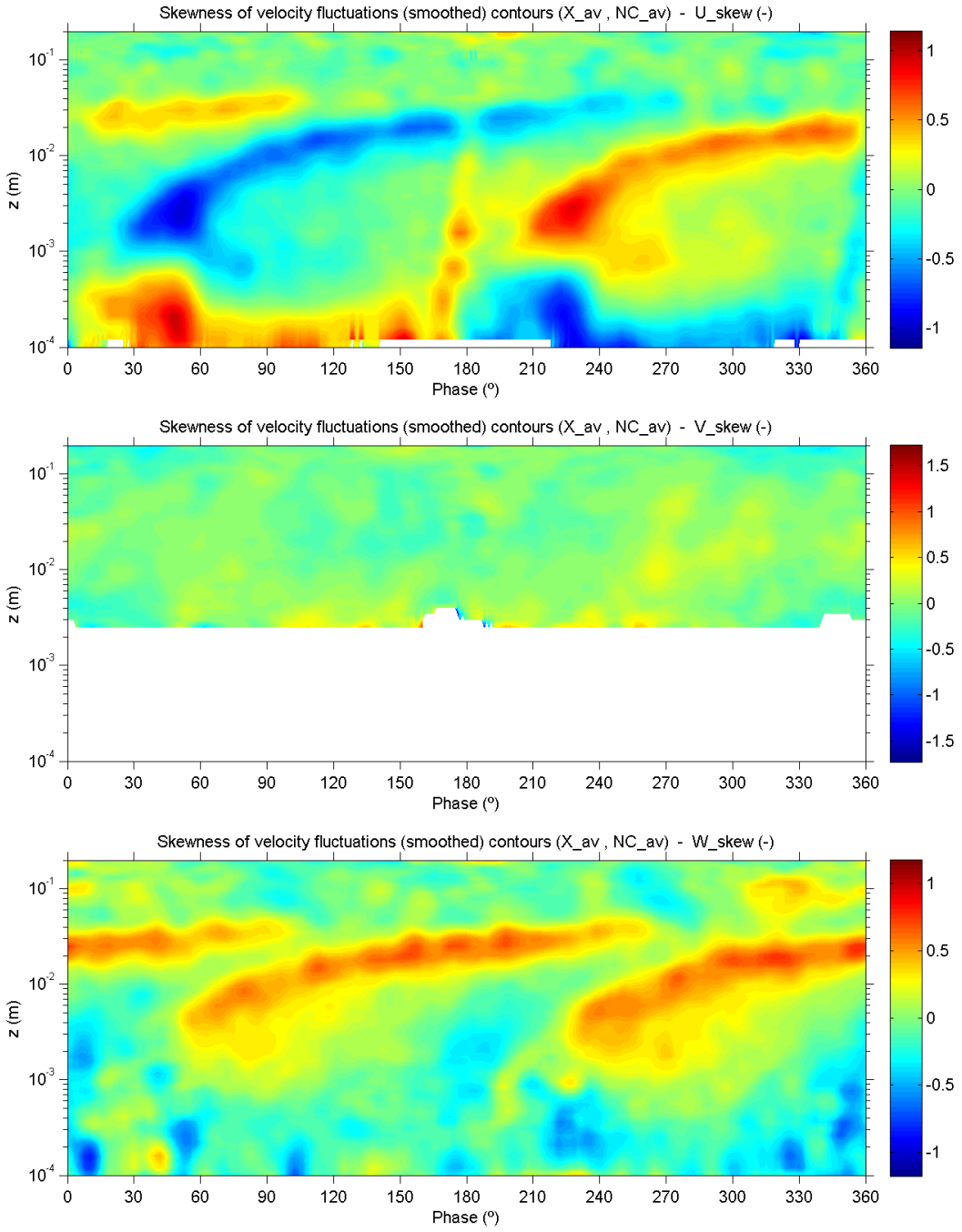
C.10.11 Turbulent viscosity (Eddy viscosity)



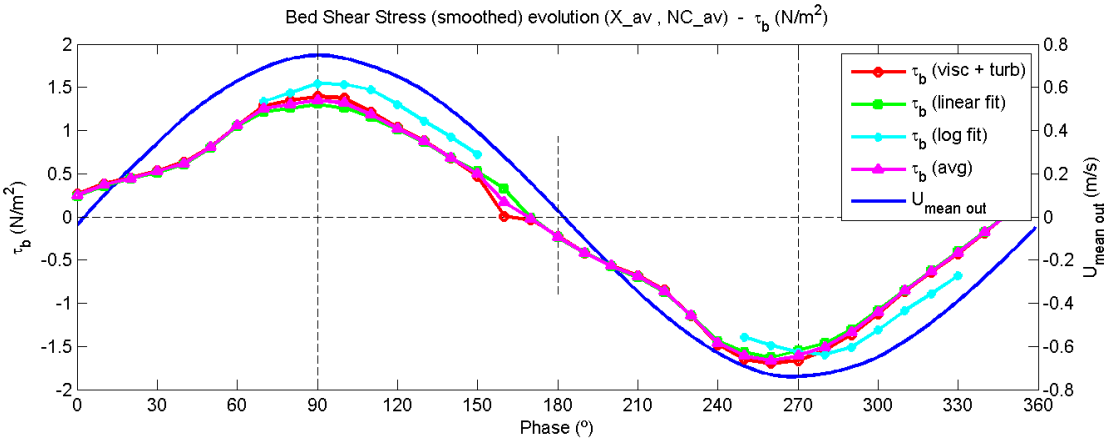
C.10.12 Quadrant analysis



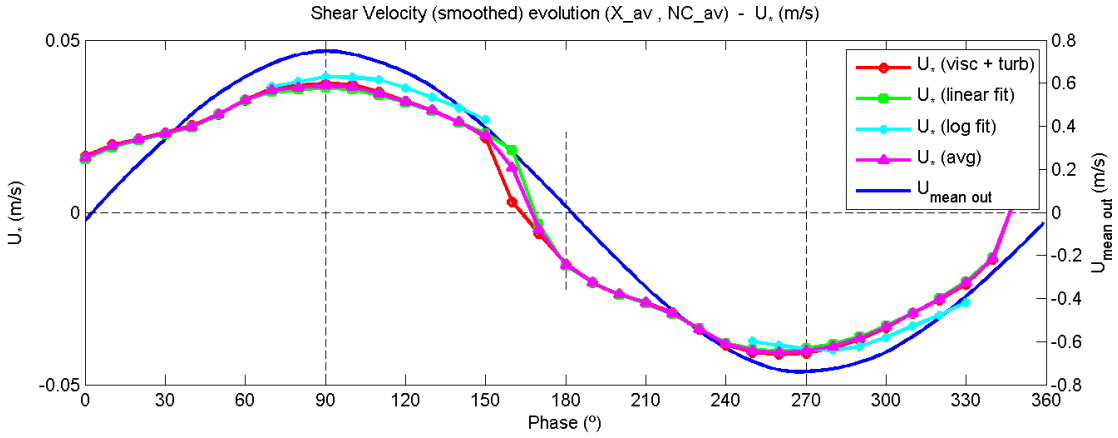
C.10.13 Skewness of velocity fluctuations



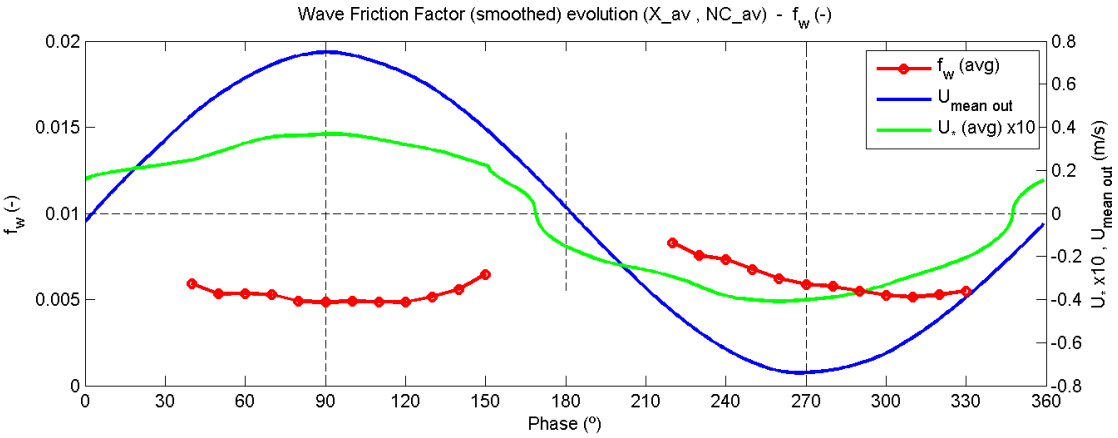
C.10.14 Bed shear stress



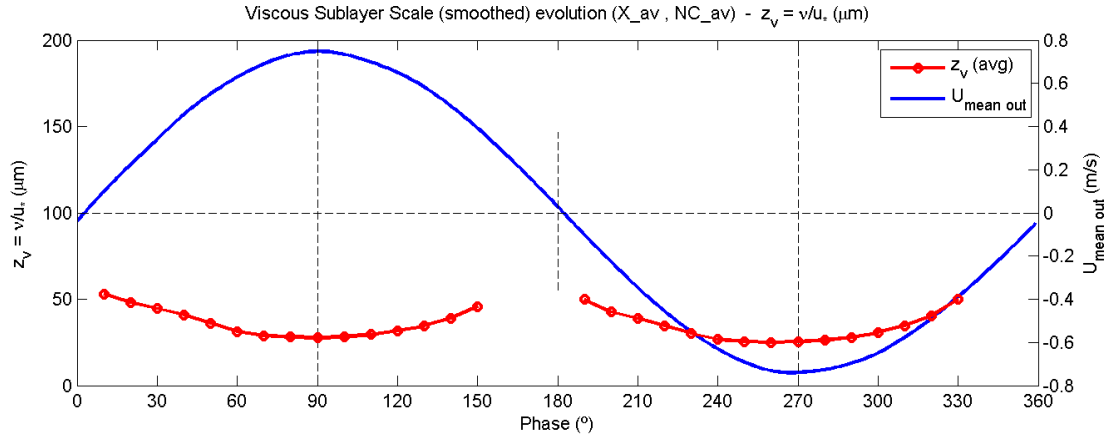
C.10.15 Shear velocity



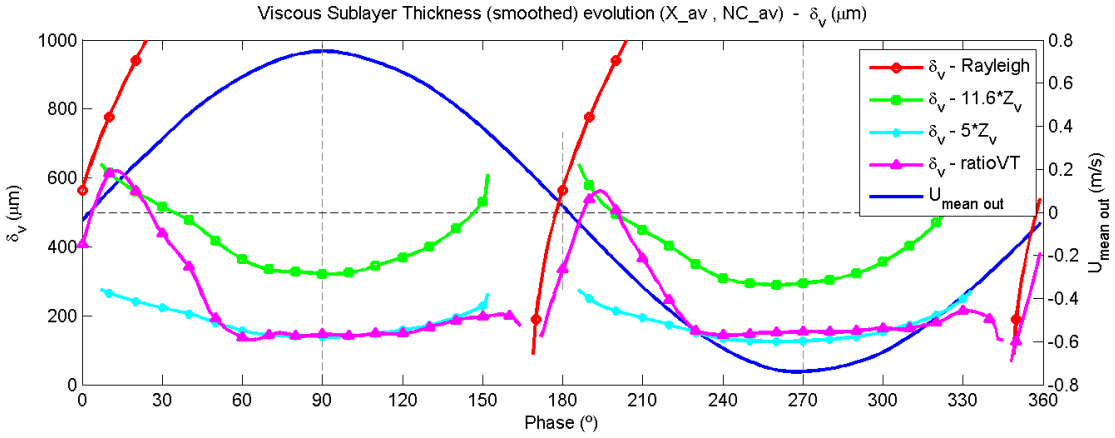
C.10.16 Wave friction factor



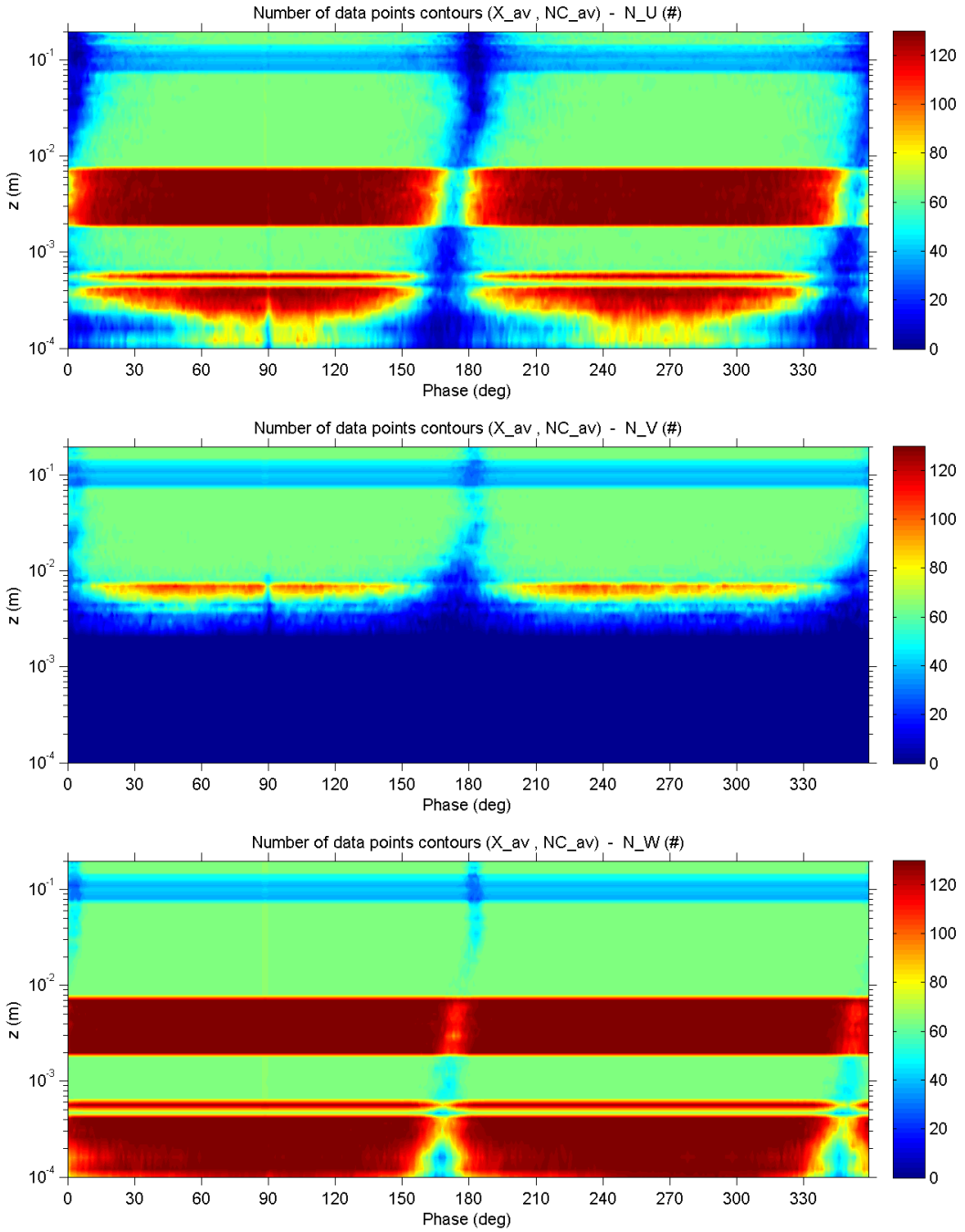
C.10.17 Viscous length scale



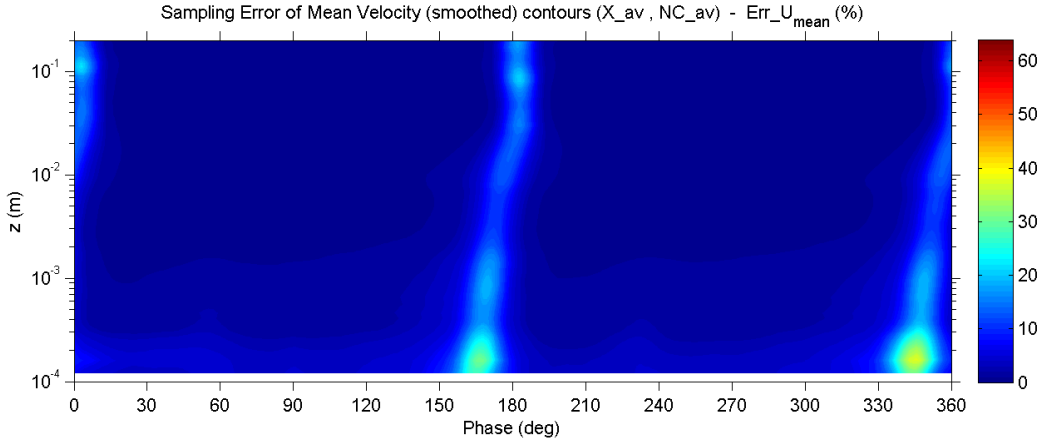
C.10.18 Viscous sublayer thickness



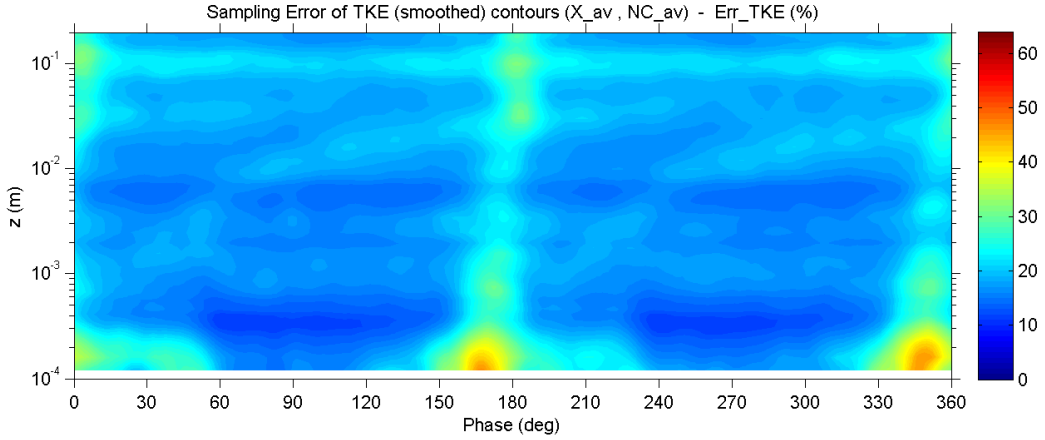
C.10.19 Number of valid data points



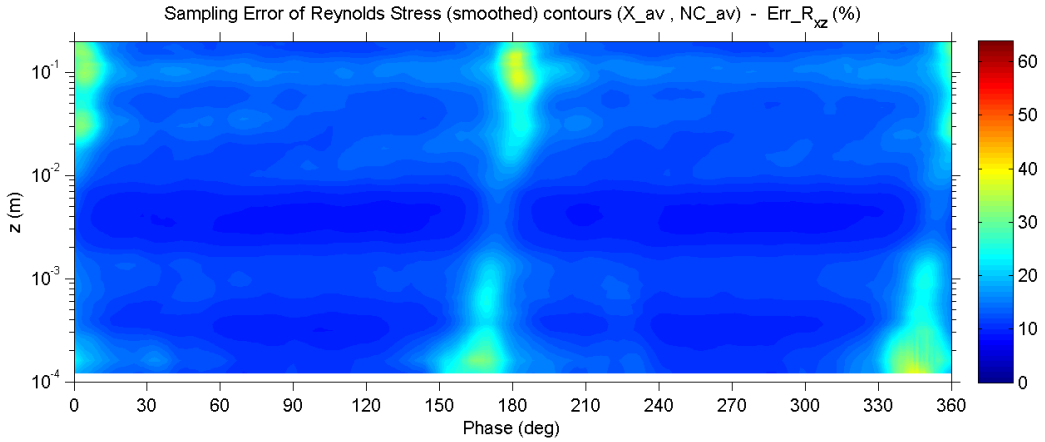
C.10.20 Percent error of mean velocity



C.10.21 Percent error of turbulent kinetic energy



C.10.22 Percent error of Reynolds shear stress



C.10.23 Percent error of RMS velocity fluctuations