ALMA MATER STUDIORUM
UNIVERSITA' DI BOLOGNA

SCHOOL OF ENGINEERING AND ARCHITECTURE
- Forlì Campus -

SECOND CYCLE MASTER'S DEGREE IN
INGEGNERIA AEROSPAZIALE/
AEROSPACE ENGINEERING
Class LM-20

GRADUATION THESIS
In Simulation and Modelling in Fluid Dynamics

Numerical study of wind-wave interfacial phenomena

CANDIDATE:
Lorenzo Silvestri

SUPERVISOR:
Andrea Cimarelli

Academic Year 2016/2017

2^{nd} session

Contents

Chapter 1

Introduction to the thesis

The idea of this project arises from some questions that a lot of researchers ask themselves during the last 60 years:

1. How does the wind transfer and dissipate energy through water waves?
2. What is the role of turbulence in the generation of surface waves by wind?
3. How to build a numerical simulation model to study this process?
4. How the interaction between wind and waves can affect sea-structures like off-shore platforms, sea wind turbines?
5. What about coastal erosion? How this problem is influenced by turbulence on the free surface?
6. How is the climate change increasing the effect of erosion?

These are only few questions about all the possible applications of a study on the free surface between air and water.
The answers to these questions are important in many branches of science ranging from physics and oceanography, to engineering and materials science. In fact free surface flows play a rather important role in civil, coastal and marine engineering and their behaviour describes the interaction between atmosphere and ocean. Anyway this subject is so wide-ranging that different features are studied by different disciplines.
Geophysical fluid dynamic (GFD) is interested in large scale flows where rotation of the reference frame, especially the Coriolis force, and stratification, density gradient inside the fluid, are the main sources of the motion.
Oceanography deals with ocean currents both in the ocean benthic/bottom boundary layer(BBL) and in the upper ocean boundary layer (UBL).

A small part of UBL oceanography gives some attention also to the micro-turbulence in the upper ocean and the mutual interaction between waves and turbulence.
This interaction, called wave-turbulence systems, is studied also in astrophysics because of its importance in planets surface.
Meteorology is involved in the study of free surface flows because of their relevance in air mass movements and convection in clouds.
Hydrology is as good as the other subjects when we deal with river flows.
All these environmental and natural sciences give us different approaches to the basics and the principles that stand behind water waves. Regarding all the applications of the free surface flows and all the particular cases in which they are relevant, we need to enter in the engineering world, especially in Naval, Coastal, Environmental, Hydraulic engineering.
What about Aerospace Engineering? Water waves are generated mainly by three sources:the main two sources are the wind blowing over the free surface of water (wind shear) and the gravity forces due to the influence of moon and sun; the other source is the temperature gradient in water and air due to solar and geothermal heating. Aerospace engineering have been always working with wind and turbulence that is one of the main source of surface waves. That's why it could be interesting to analyse this phenomenon by its point of view. Therefore in this thesis we will try to build a numerical model to analyse the water waves and the free surface between air and water in a common open channel flow.

1.1 Free surface flows

In fluid mechanics, the words "fluid" and "flow" are completely different. In fact the former is the subject, the latter is the action. When a fluid is at rest we cannot talk about flow, while when a fluid is moving than we can say that the fluid flows or that there is a flow. A flow has always a main direction and moves from somewhere to somewhere else.
Let's now consider a part of this moving fluid that we will call domain. Inside the domain we can always distinguish two fundamental sections: the inlet section is the section where the flow is coming from; the outlet section is the section where the flow is going to. The features of the remaining sections of the domain are very important for the behaviour of the fluid and the evolution of the flow. In fact all the flows, or moving fluids, we encounter in this world, can be classified by these features and by the way in which the fluid is filling the domain. We can identify two main categories:

1. **Wall bounded flows**
   The flow is bounded by walls either totally or in one part. Walls can be either moving, like in Couette flows, or fixed, like in channel flows (see figure 1.1.1) or boundary layer flows.

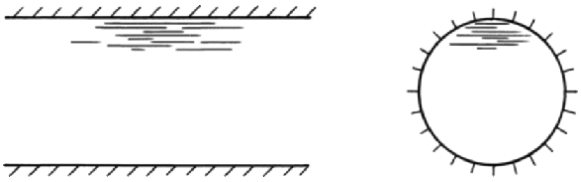


Figure 1.1.1: Wall bounded flow

Anyway, during his course, the fluid is only in contact either with itself or the walls, since that it is completely filling the domain. These flows can be subdivided into other two subcategories:
- *external flows*: in this case the flow is not totally bounded. Examples could be the flow around an aircraft, a ship or a bluff body in general, the boundary layer over a flat plate, inside the ocean or on the earth surface (the atmospheric boundary layer).
- *internal flows*: here the flow is completely bounded from the bottom to the top as in channel flows or pipe flows.
However the fluid is bounded at least in one part from a solid surface, thus a surface where the no-slip condition is applied.

2. **Free shear flows**
In this case the flow is not constrained by any boundary or wall, but it is subjected to the atmospheric pressure all around it. This surface is free from any shear stress that could be generated by the presence of a wall, instead turbulence arise from mean velocity differences. Typical examples of this flows are: jet, wake and mixing layers flows.

Free surface flows does not appear in this usual classification. The reason has to be found in the definition of this type of flows that do not depend on the external boundaries, but depend on an its own internal feature.
A free surface flow can be generally defined as a fluid, either moving or at rest, where a free surface is present. A free surface, that is the common internal feature of all the existing free surface flows, is the surface that forms between two fluids of different densities, under the action of a gravity field. This surface can continuously deform, both in space and time, if stresses are acting along it.
The most common example of a free surface is the air and water interface. We can observe it every day from a glass of water to a pool, from a river to the sea.
From its definition, we can understand why a free surface flow cannot be classified as a wall bounded flow or a free shear flow. In fact it could be both of them: a jet of water inside an air domain, that could be a waterfall in nature (see figure 1.1.2a) or a drain in technology (see figure 1.1.2b), is a free shear flow, but also a free surface flow since that a free surface is present.

(a) *Waterfall* (b) *Drain*

Figure 1.1.2: Free surface flows and free shear flows

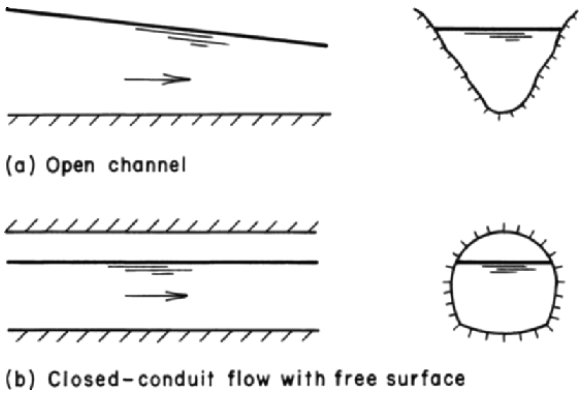
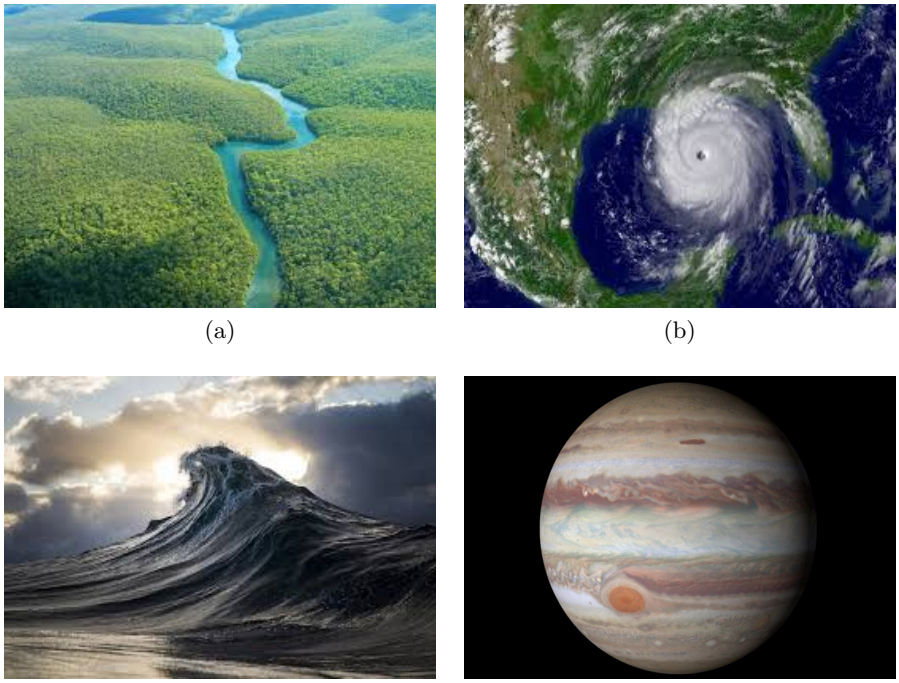


Figure 1.1.3: Free surface flows

For the same reason a flow in a channel could be a free surface flow if water is not completely filling the domain and a free surface is generated, as shown in figure 1.1.3. Another example could be the oceanic boundary layer where a wall is present at the bottom, but at the top there is a free surface between water and the atmosphere. In these cases the flow is both a wall bounded and a free surface flow.

From these examples and definitions, another interesting question could arise: does a free surface form only where a density difference is present between two fluids? In fact, a clear interface can be also present between two regions of the same fluid, where different flow regimes are existing. The interface between the boundary layer region and the free stream region in a boundary layer flow, or between the thin shear layer of a air jet flow and the quiescent air all around it, can be examples of a free surface generated by different flow characteristics velocities and lengths, and not from different densities. This question will be left open and in this work a free surface is considered to be generated only by different densities.



(a) (b)

(c) (d)

Figure 1.1.4: Examples of free surface flows

From all the examples made until this point, it should be clear that free surface flows are ubiquitous in nature and technology; other examples of free surface flows in nature can be the Earth atmosphere clouds movements, the stratified gas flow in Jupiter's atmosphere, rivers, water waves and currents, flows around boats.

Among all the free surface flows that have been presented in this subsection, we will concern only about the generation and propagation of water waves in a channel and their mutual interaction with turbulence. This interaction is the main responsible for the formation of vorticity structures, that are very important in the sediment dynamics about the sea bed and in the coastal erosion process. In other words, studying the wave dynamics we can deeply understand all the problems related with coastal engineering and sea environment.
Observation and modelling of the wave dynamics are also of interest for the forecasting of the sea state, which is essential for navigation and meteorology. Moreover the air-sea exchange of momentum, heat and mass is fundamental for a lot of natural process in the Earth, but also in some planet's satellites. As already introduced from the sixth question in section 1, the climate change and the raising of the sea level makes more relevant the problems related with navigation and erosion.

Due to their large range of application in nature and engineering, it is very difficult to define a general theory for free surface flows. Each type of free surface has to be described individually. That is why in the following section, some basic concepts and nomenclature relative to water waves are introduced.

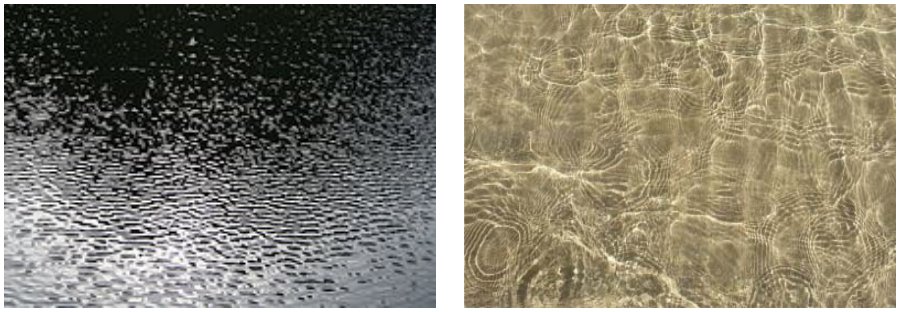
1.2 Water waves

The free surface of a water container, under only the action of uniform normal stresses along the entire area, remains flat. If we apply in some point a tangential or a non-uniform normal stress, the surface will be perturbed from its original state and this perturbation will propagate with a wavy form. Let's consider three different examples: a glass of water, a lake and the sea. If we touch the flat surface of a glass of water with a finger, we will see how a circular wave starts propagating all around the point in which the force has been applied. After few seconds the free surface will come back to its original state, if no other disturbances are applied. As a second example let's consider a small calm lake. The smaller is, the larger will be the probability to see his surface completely flat. If at a certain time a boat pass through the lake, we will notice a wave propagating from the sides of the boat to the lake edge. Finally let's observe the sea surface near the coast. It is never flat: there are always some waves of different lengths and time scales acting on it, depending on the force and so on the perturbation that is generating those waves. From the previous examples we can conclude that a wave is generated by a perturbation acting along a surface region. Secondly, we can state that different sources of energy can generate different type of waves. Concerning the ocean motion and considering only natural forcing, we can say that water waves are generated by three main processes (see reference [1]):

1. *The gravitational force* induced by the moon and the sun. This force creates tides, that are waves characterized by a long period wave motion varying between 12 and 24 hours. Their length scale is of the order of the basin scale while the velocity scale ranges between 1 and 100 m/s.

2. *The wind blowing along* the water surface. This wind stress is responsible for the most of ocean currents in the upper layer. The length, time and velocity scale of the wind waves is strictly dependent both by the strength and other features of the wind like fetch, duration and direction and by the water depth.

3. *The local temperature gradients* between air and sea generate heat fluxes, evaporation, and precipitation that is a thermodynamical forcing capable of generating additional currents.



(a) (b)

Figure 1.2.1: Ripples formed on a lake (left figure) and on the sea surface (right figure)

These are the three main mechanisms that can generate water waves inside the ocean. However, other secondary events as seismic activities, volcanic eruption, local meteorological phenomena and storms can generate respectively tsunamies, seiches and storm surges.
In this project we will only deal with wind waves, thus those waves in which the disturbing force is the wind stress. Since that they are generated and propagated on the free surface of water, instead being formed in depth, it is sometimes referred to them as surface waves.
As we mentioned, wind waves have a large range of frequencies, but we can classify them in two main categories, depending on the restoring force acting to propagate the disturbance:

1. **Capillary waves or ripples**, are those waves in which the main restoring force is the surface tension. They have a high frequency and a period that is less than 0.1 seconds. Typically, ripples appear suddenly on calm water where and when wind start blowing on it, as shown in figure 1.2.1; anyway if the wind stop blowing, they will disappear. Capillary waves are the very first stage of wind wave generation process. The wave height can range from 0 to 0.2 meters. The intensity of the wind that can generate these ripples is between 0.3 and 1.5 m/s (level 1 in the Beaufort scale).

2. **Gravity waves**, are those waves in which the main restoring force is the gravity, while surface tension plays only a secondary role. They have a lower frequency than capillary waves, and a period ranging between 0.1 seconds and 5 minutes. This is a very large interval of frequencies and that is why they are further subdivided into ultra-gravity waves, ordinary gravity waves and infra-gravity waves, as illustrated

Classification	Period band	Generating forces	Restoring forces
Capillary waves	<0.1 s	Wind	Surface tension
Ultragravity waves	0.1–1 s	Wind	Surface tension and gravity
Gravity waves	1–20 s	Wind	Gravity
Infragravity waves	20 s to 5 min	Wind and atmospheric pressure gradients	Gravity
Long-period waves	5 min to 12 h	Atmospheric pressure gradients and earthquake	Gravity
Ordinary tidal waves	12–24 h	Gravitational attraction	Gravity and Coriolis force
Transtidal waves	>24 h	Storms and gravitational attraction	Gravity and Coriolis force

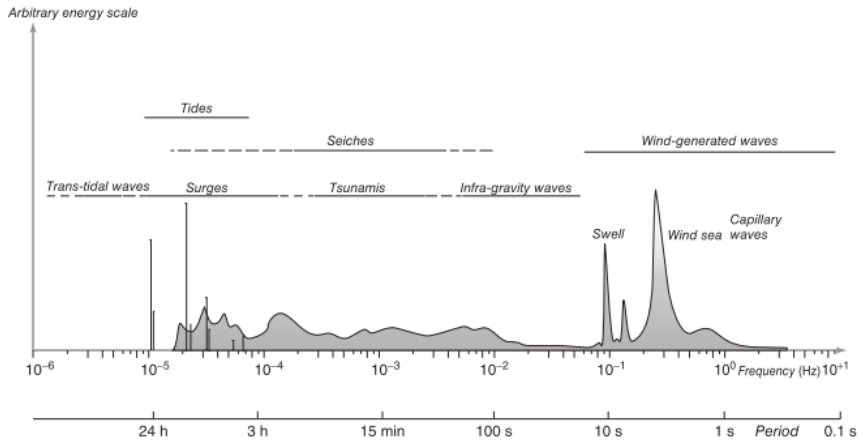
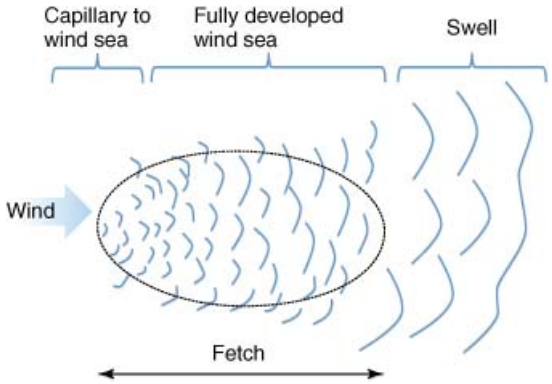


Figure 1.2.2: The classification and the spectrum of ocean waves

in 1.2.2. They are generated from wind which intensity vary from 1.5 m/s with a wave height of 0.2-0.5 m, up to around 25 m/s with a wave height of about 12 m.

Ordinary gravity waves are the common waves that we usually see. They are the subsequent sea stage after the ripples generated by wind blowing. Actually we should differentiate gravity waves in two different states: seas and swells (or long waves). Seas, or wind seas, are irregular waves generated immediately after the ripples state and, thus, under the direct effect of local wind as illustrated in figure 1.2.3b. The resulting wave field is an irregular pattern, due to the interaction of a large number of components with different phases, directions of propagation and wave period (see figure 1.2.3a). When seas propagate over a depth that is much larger than the wavelength (when the water depth can be considered as infinite), longer waves travel faster than shorter ones. While shorter waves energy easily disperse, long waves rapidly move outside the generating area and become known as swells (figure 1.2.3c). Swells have a typical wavelength higher than 260 m, instead seas wavelengths can arrive up to 220 m. On the other hand, swell wave height is typically lower than the seas one and that is the reason why the energy dispersion in swells is much weaker than in seas and they can travel for very long distance before disappearing. As an example, a long wave with period larger than 13 s (wavelength larger than 260 m) and small amplitude loses about half of its energy over a distance of about 20,000 km. It is not



(a) *Wind wave formation process*

(b) *Sea state*

(c) *Swell state*

Figure 1.2.3: The different phases of waves: ripples, seas and swells.

uncommon, therefore, that a swell that generates in the Antarctic Ocean travels all the way to the Alaska with very little energy dissipation.

However, as all the gravity waves, swells increase in amplitude and decrease in wavelength when approaching a coastal environment. As they move towards the shore, they brake, rapidly loosing energy. Wave breaking phenomena have been extensively studied in the past years, but we will not deal with them in this work.
Apart from the wave sources (the disturbing forces), the restoring forces, the wavelength, the height and the period, there are other important features to be mentioned. Firstly, we could distinguish waves by the water depth over which they propagates: deep water waves and shallow water waves. The first are defined as that waves for which the depth is much larger than the wave length and, to give a reference value, the ratio $h/\lambda > 1.2$, where h is the mean water depth and λ is the wavelength (see figure 1.2.4). Under these conditions the wave features are not affected by the sea bottom. Capillary waves are a typical example of deep water waves.

The latter are defined as that waves for which the ratio $h/\lambda < 1/20$ and thus the wavelength is much bigger than the mean water depth. In this case

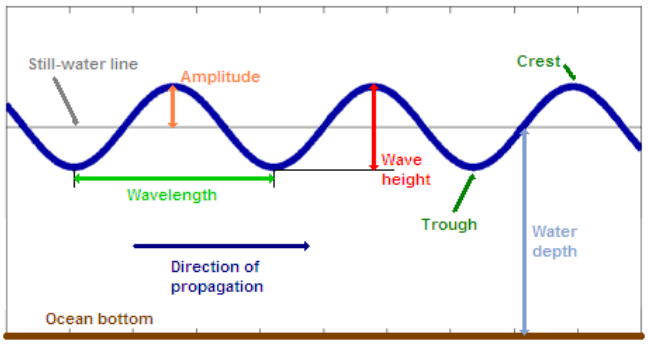


Figure 1.2.4: Important parameters for describing a wave

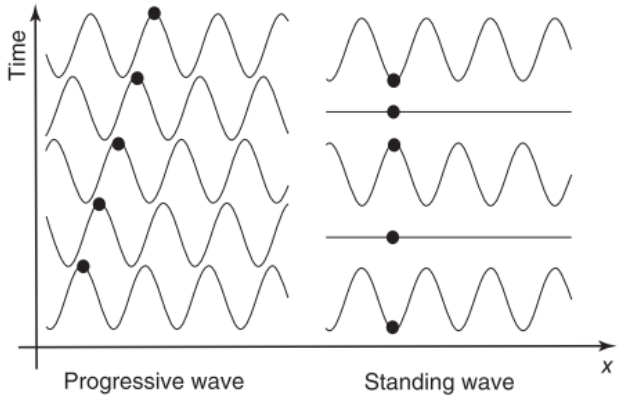


Figure 1.2.5: Standing and progressive waves

the surface oscillations are strictly influenced by the sea floor and we talks about shallow water waves. Tides can be always considered as an example of this wave type, both in open ocean and nearby shore, due to their large wavelength. All the other waves for which h/λ is in between this two extreme values, are called intermediate water waves and their behaviour is only partly affected by the sea bottom. Anyway, it is clear that a deep water wave generated in the open ocean, can transform into a shallow water wave once approaching a coastal region.

In order to complete the description of water waves, we should also introduce a classification regarding the way in which a wave moves and travels through the water. We can make distinction between a standing wave and a progressive wave as illustrated in figure 1.2.5. Most of the waves we encounter on the ocean surface are progressive waves: the wave front and thus the profile of the oscillations, moves through the fluid and the energy is transferred from one particle to another one. In this way, the energy is

moving away from his source, even though no mass transfer is happening, and it will never return. On the other hand, if the wave is reflected back and superimpose to the incident wave, we will face a standing wave. In this latter case there is no horizontal propagation of energy, but each particle has its own characteristic vibration. Typically this vertical motion of the water surface happen on closed bay, basin or harbour where waves are reflected back.

It could be very useful to insert and summarize all these considerations about wind waves and the factors affecting them inside a dimensional analysis of wind-wave interaction. The goal of this analysis is to find some basic parameters that are governing this physical process: in this way we can expect similar results, whenever similar parameters are found. The works of Toba [2] and Johnson [3] have been taken as a reference for this analysis.
A lot of parameters plays a not negligible role in the wind wave interaction. In order to be as clear as possible we will group these parameters in three categories:

1. **Wind parameters**: these are all the parameters describing the wind that is blowing over the free surface. We can write:

$$\text{wind} = f(u_\tau; \delta; t_w; F_w; \rho_a; \mu_a; \Phi_a; \Theta_a, \text{water surface parameters}) \tag{1.2.1}$$

   where: $u_\tau = \sqrt{\tau_{fs}/\rho_a}$ is the interface friction velocity that is measuring the shear stress (τ_{fs}) along the free surface due to the turbulent wind; δ is the boundary layer thickness (the outer length scale) of the wind; t_w, F_w are the wind duration and fetch, where the second one is the length of water over which the wind has been blowing during the time t_w; ρ_a, μ_a are respectively the density and dynamic viscosity of air; Φ_a is the wind direction; Θ_a is the wind temperature; finally the wind is strictly dependent also on the water waves and thus on their parameters, that are described in the following point.

2. **Water surface parameters**: all the parameters describing the wave motion, shape and the water domain. We can write:

$$\text{water surface} = f(c_p, k_p, h, H_s, \rho_w, \mu_w, \Phi_w, \Theta_w) \tag{1.2.2}$$

   where: $c_p = w_p/k_p$ is the phase velocity (or phase speed) of the wave relative to the peak (indicated by the subscript "p") of the wave frequency spectrum and w_p is the peak angular frequency; k_p is the peak wavelength; h is the mean water depth; H_s is the significant wave height that is usually defined as four times the standard deviation of the surface wave elevation; all the other parameters are the same described in the wind part but they are relative to water.

3. **Physical process constants**: they are relative to the restoring forces of waves. Two main parameters are important for wind-waves: g, the gravity acceleration, and σ, the surface tension coefficient.

Therefore any other parameter that we want to use for describing wind-wave interaction, that we will indicate with P, will be a function of all the previous parameters:

$$P = f(u_\tau, \delta, t_w, F_w, \rho_a, \mu_a, \Phi_a, \Theta_a, c_p, k_p, h, H_s, \rho_w, \mu_w, \Phi_w, \Theta_w, \sigma, g) \quad (1.2.3)$$

Before proceeding with the Buckingham theorem and starting to finding all the dimensionless variables needed to describe this process, we should introduce some simplifications based on what we want to investigate in this physical process and on what we can neglect without loosing fundamental informations:

- we are investigating a localized phenomena that is not dependent on temporal variables like the time duration of wind or spatial ones like the wind fetch. We will assume a stationary wind blowing in the same direction of the waves. Through this simplification we can skip the terms t_w, F_w, Φ_a, Φ_w inside equation (1.2.3);
- we assume the process to be isothermal and thus without considering the effect of temperature field on both air and water;
- we take advantage of the dispersion relation coming from linear theory in order to relate the phase speed (and so also the angular frequency) to the wave number: $w_p^2 = (gk_p + \frac{\sigma}{\rho_w}k_p^3)\tanh(k_p h)$. In this way we can consider only the phase speed in order to describe the wave shape;
- we will consider that a changing of the mean water depth h is affecting only the phase speed c_p of the wave and thus we will detect his effect by indirectly observing a change in c_p;
- it has been confirmed by many experiments and observations, that the significant wave height H_s is a function of other parameters as c_p and u_τ: the oldest relation is called the Toba's relationship and relates the wave steepness (the ratio between the wave height and the wave length) with the wave age (a parameter that will come out from the dimensionless analysis);
- the ratios ρ_a/ρ_w and μ_a/μ_w are assumed to be constant number, therefore we can choose either air parameters or water parameters to describe the flow. Air parameters will be chosen in this treatment and instead of using density and dynamic viscosity, we will group them into a unique parameter that is the kinematic viscosity $\nu_a = \mu_a/\rho_a$.

This choice is comprehensible since that we are dealing with turbulent flows and the dynamic viscosity is no more directly important as for laminar flows.

- if we are interested only in the air-water interaction, without considering any other type of fluid, we can assume the surface tension coefficient to be always constant. Through this assumption we are saying that the change of any variable of the process is never due to the change of the surface tension coefficient σ, since that we are keeping it constant. The same consideration can be made about the gravity acceleration g, since that we are interesting only in the earth environment.

Introducing all the previous simplifications in (1.2.3) we finally obtain:

$$P = f(u_\tau, \delta, \nu_a, c_p) \tag{1.2.4}$$

Let consider that we want to study an important parameter regarding water waves: the sea surface roughness y_0. Sometimes, in the literature, it is referred to it as z_0, but we will call it y_0 since that for us the y direction will always represent the sea-bottom normal direction. This parameter is usually defined by considering the overlap logarithmic law of the wind behaviour close to the free surface (as it will be defined later in section 3.5). Substituting y_0 inside (1.2.4):

$$f(y_0, u_\tau, \delta, \nu_a, c_p) = 0 \tag{1.2.5}$$

Applying the Buckingham theorem (or Π theorem), we can describe this process by three dimensionless variables: the fundamental dimensions involved in the process are length L and time T, while the total number of variables is five, hence their difference give us the number of dimensionless groups Π_i needed in order to describe the wind wave interaction. Taking as repeaters, or basis of our dimensionless groups, u_τ and ν_a we obtain the following variables:

$$\Pi_1 = z_0 u_\tau^a \nu_a^b \longrightarrow y_0^+ = y_0 \frac{u_\tau}{\nu_a} : \text{ Dimensionless roughness length} \tag{1.2.6}$$

$$\Pi_2 = \delta u_\tau^a \nu_a^b \longrightarrow Re_\tau = \frac{u_\tau \delta}{\nu_a} : \text{ Frictional Reynolds number} \tag{1.2.7}$$

$$\Pi_3 = c_p u_\tau^a \nu_a^b \longrightarrow \beta = \frac{c_p}{u_\tau} : \text{ Wave age} \tag{1.2.8}$$

From this dimensional analysis we can derive that $y_0^+ = f(Re_\tau, \beta)$. Sometimes y_0^+ is also called the roughness Reynolds number and indicated as Re_r. Some words need to be spent also about the wave age parameter β. The name of this variable is giving us some clues: from the age of a person

we can usually understand a lot of things, and his personality, his work, his family and other features will depend on his age. Following this analogy, a lot of wave informations are contained in the phase speed c_p: frequency, wavelength, mean water depth, fetch and time duration of the wind. That is the reason why its adimensional form is called "wave age".
In order to conclude this analysis we can say that any other parameter that we want to find to describe the wind-wave process, under the assumptions made before, can be expressed as a function of the friction Reynolds number and the wave age; a lot of experimental relationships has been found between these fundamental parameters and they are still applied for waves current forecasting. Clearly, the dimensional analysis suggested here is not the only possible one: many other analysis are present in the literature and other important parameters (as for example the Charnock parameter $z_0 g/u_\tau^2$) are derived; however in this text it has been reported the most suitable one for our purposes.

In this section it has been summarized the main concept behind water waves. It is clear that these are only few and essential informations about this topic. Water waves, as the most part of geophysical fluid dynamic problems, are a very large subject: different sources produce a very large amount of different waves with completely different length, time and velocity scales. In this thesis, among all this sources and scales, shown in figure 1.2.6, we will deal only with microturbulence generated by wind on the upper ocean surface. As a consequence we will study only small length scales of the order of few centimeters and time scales around few seconds.
Why do we limit our study only on these small scales?
Essentially for two reasons: firstly, the energy transfer from turbulent wind to water happens at small scales and this transfer is the main responsible for the wave initiation as it is explained in Chapter 2; secondly, to analyse a larger range of scales without loosing accuracy of the results, we need a higher computational power than the one that has been used in this work.

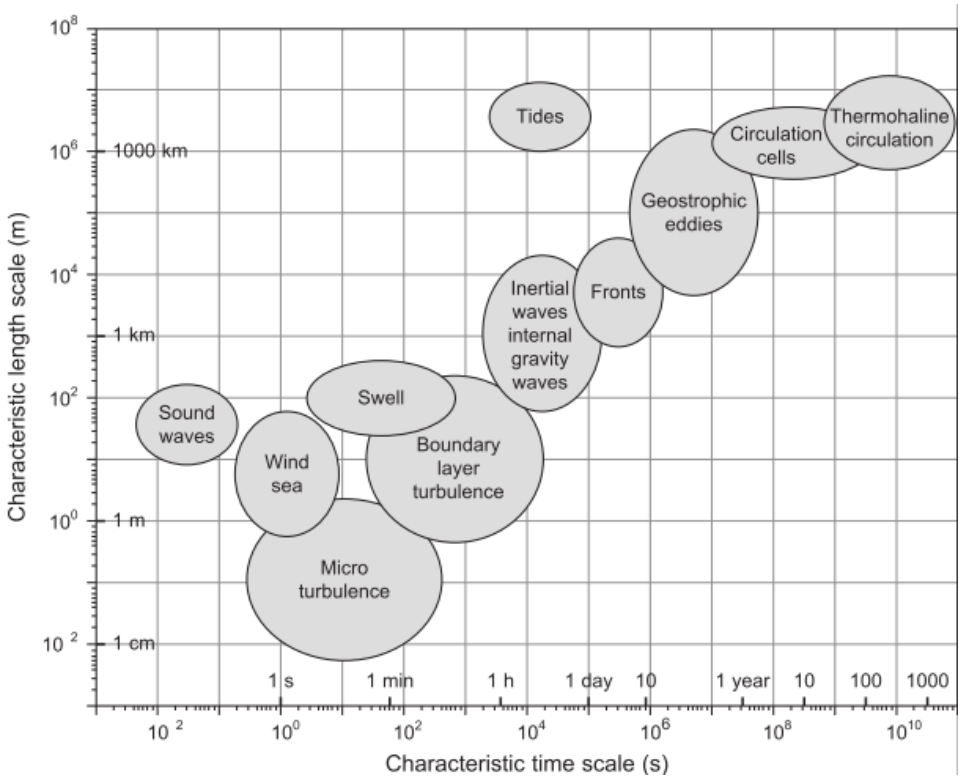


Figure 1.2.6: Characteristic time scales and length scales of water waves

Chapter 2

State of the art

The wind wave generation and interaction has been extensively studied from the theoretical point of view since the 1880. Experiments to collect data and confirms theory started later, approximately in the 1930, when the technology provided the right instrumentation to make some measurements on waves and winds. Lastly, together with the computational power development, numerical simulation method, from 1990 using eddy viscosity models up 2000 using DNS, has been used to investigate this process.
In 1887 Kelvin and Helmholtz [4] developed the irrotational theory of waves induced by the parallel movement of two fluids with different densities $(\rho, \rho^{'})$ and different velocities $(U, U^{'})$. Assuming both fluid to be two-dimensional, incompressible, irrotational, of unlimited depth, and considering small disturbances around the steady state of the boundary between the two fluids, a solution can be found for the wave height growth rate, linearizing the equations. The basic results coming from this theory, also called linear wave theory, can be found in any oceanographic or geophysical book. The final results coming from this assumptions is the following:

$$\frac{w}{k} = \frac{\rho U + \rho' U'}{\rho + \rho'} \pm \sqrt{\frac{g}{k}\frac{\rho - \rho'}{\rho + \rho'} - \frac{\rho\rho'}{(\rho + \rho')^2}(U - U')^2} \tag{2.0.1}$$

In (2.0.1) we consider the free surface wave height function(see figure 2.0.1) as $\eta = ae^{i(wt-kx)}$, therefore w is the angular frequency of the wave and k is the wave number. The form of this function and the value of the constant a comes from the linear wave theory. To deeply understand the meaning of (2.0.1) we need to reshape it considering the phase velocity of the wave $c = w/k$. We can think at this velocity as the velocity of the crest of the wave. Rewriting (2.0.1) we obtain

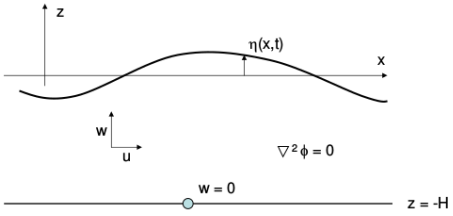


Figure 2.0.1: Free surface wave height definition: $\eta(x,t)$ is the distance of the free surface from the mean water depth, that is the height for which water is at rest.

$$c = \bar{U} \pm \bar{c}; \tag{2.0.2}$$

$$\bar{U} = \frac{\rho U + \rho' U'}{\rho + \rho'}; \tag{2.0.3}$$

$$\bar{c}^2 = {c_0}^2 - \frac{\rho\rho'}{(\rho+\rho')^2}(U - U')^2; \tag{2.0.4}$$

$$c_0 = \frac{g}{k}\frac{1-\rho/\rho'}{1+\rho/\rho'}; \tag{2.0.5}$$

where $\bar{U}$ and $\bar{c}$ are the averaged value of velocity and phase velocity, while c_0 is the phase velocity in correspondence of no current in both fluids ($U = U' = 0$). The instability of a wave can be observed from (2.0.4): when the term involving $U - U'$, that is the relative velocity of the wind with respect to the water, is bigger than c_0, the phase velocity c becomes imaginary and the free surface height η progressively increase in amplitude leading to an unstable process. Assuming that we are dealing with air ($\rho = \rho_a = 1\ kg/m^3$) and water ($\rho' = \rho_w = 1000\ kg/m^3$) we can say from (2.0.5) that $c_0 = g/k$, neglecting the contribution of the density ratio $\rho_a/\rho w = 10^{-3}$. Therefore the instability condition will be:

$$U - U' = U_a - U_w = U_r > c_0 \frac{1+\rho_a/\rho_w}{\sqrt{\rho_a/\rho_w}} \approx 30\frac{g}{k} \tag{2.0.6}$$

The main problem of this formulation, apart from all the assumptions made to reach the final result, is that the predicted critical velocity of the wind U_r, that is the minimum velocity capable of generating an unstable wave, does not match the reality. Concerning this, let's consider the smallest existing wave in the ocean spectrum that is a ripple or a capillarity wave. Usually a ripple have a phase velocity of 0.2 m/s. This would means that the critical velocity would be $U_r = 30 * 0.2 = 6$ m/s. So the ripples will be

(a) (b)

Figure 2.0.2: Examples of Kelvin-Helmholtz instability on clouds and Saturn

stable until the relative velocity of the wind will reach 6 m/s.

This concept of critical wind speed has been discussed a lot during the 1990's, not only regarding the correct numerical value or the mathematical formulation, but also regarding his effective existence. The main problem was a fundamental question arising by considering the water completely at rest. What is the mechanism through which wind generates and amplifies water waves? If neither gravity field nor wind is acting on the free surface, waves do not form and the free surface remains flat. This means that the density difference between water and air is not enough to generate waves. If the wind start blowing, under the predictions of irrotational theory, no ripples will form until the critical wind velocity is reached. This is completely in disagreement with observations, since that velocities of 1 m/s have been found to be sufficient to raise waves.
The first effort to answer these questions was given by Jeffrey [5] in 1925 with his sheltering theory. Jeffrey abandoned the irrotational hypothesis on the air and realized that the interaction between wind and wave cause the wind itself to move irregularly. In fact, considering an already existing wave, the wind, blowing over the water, may be unable to follow the deformed free surface and it is sheltered by waves. Therefore the pressure of the air, in analogy with a flow over a cylinder, will be higher in front of the crest, and lower on the opposite side, where a recirculation zone (the "sheltered zone", see figure 2.0.3), is created.

This fluctuating pressure distribution is the main responsible for the amplification of the wave. The force applied to the free surface takes into account only the normal stresses (pressure fluctuations) and can be evaluated by:

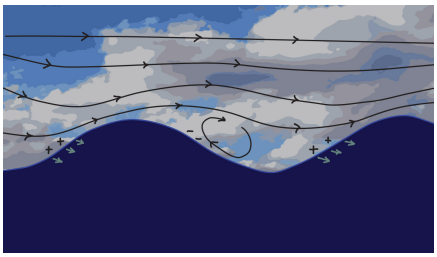


Figure 2.0.3: Sheltering mechanism illustration

$$F = s\rho_a U_a^2 \frac{d\eta}{dx} \tag{2.0.7}$$

where s is the sheltering constant and U_a the wind velocity over the crests. The formula (2.0.7) is called "the sheltering hypothesis" and comes from the analogy of this problem with the one of the thrust produced by a current over an inclined lamina in the direction of the flow.
There were two problems in this theory: the former was the assumption of no tangential stresses, thus no air friction acting along the free surface; the latter was that the observed value of the constant s, made by solid wave models, was smaller than what predicted by the sheltering model and the resulting force caused by the wind in the sheltered zone, was not enough to let waves grow.
Sverdrup and Munk [6] in 1947 included the tangential stress in the momentum transfer between wind and sea, stating that a net increase of energy of the wave results from the friction forces. Anyway, no turbulence was considered in those models and still the process of formation of waves from calm sea, was not explained.
The role of turbulence in the theory of wave generation appeared in 1947 with the pioneering contributions of Phillips [7] and Miles [8]. Nowadays, their theory about water wave formation is named the Miles-Philips mechanism, that is the union of the two mechanisms described by the two scientists and it is recognized to be a fundamental theory for wind-waves.
Phillips in his paper [7] formulated the mathematical problem as follow:

> Given that at an initial instant a turbulent wind commences to blow across an infinite sheet of deep (inviscid) water originally at rest, generating a distribution of fluctuating pressure on the surface which is a stationary random function of position and time, the aim is to study the properties of the surface displacement at subsequent times.

In Phillips' mind was more plausible that the initiation and development of a wave was a consequence of the fluctuations of normal pressure upon the surface, which are a typical feature of a turbulent wind, that is the one we

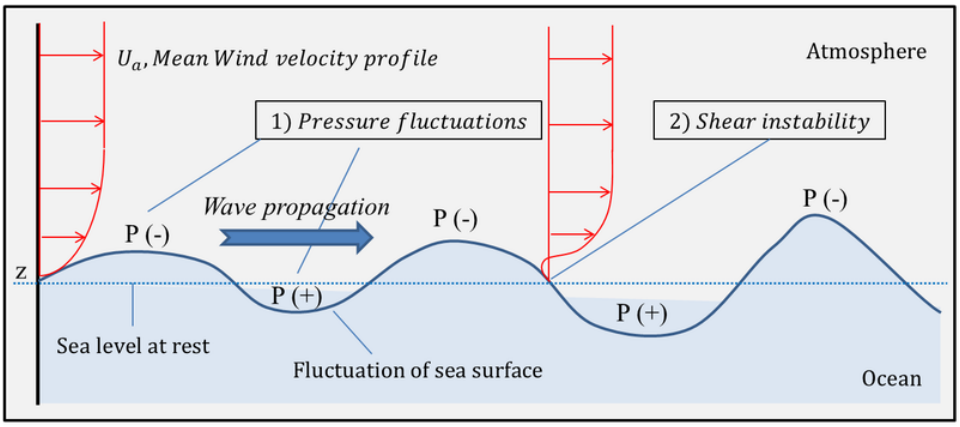


Figure 2.0.4: An illustration of the Phillips-Miles mechanism: the Phillips one is a resonance phenomenon due to pressure fluctuations, while the Miles one is due to shear instability of the wind profile

are used to see in reality. In fact wind blowing over ground or water never consists of steady and parallel streams of air, but rather an "irregular series of "puffs" and "lulls" carrying eddies and swirls distributed in a disordered manner". This random stresses distribution over the surface contains components with a large range of wave numbers, each of them corresponding to different turbulent eddies of different length scale impinging on the water. The surface response to these stresses is not uniform and, among all the frequencies from which is disturbed, it will be exited more from the ones coinciding with the possible vibrating modes of the free surface. At this point a resonance mechanism between the eddies and the water surface begin and surface waves start growing and their amplitude starts increasing. In Phillips' resonance mechanism the tangential stresses are neglected together with the non linear interaction between wave and wind, that is assumed to be negligible in the first stage of the wave growth. The main result coming from this theory, using the Fourier analysis of the stresses distribution, is the linear relation between the early growth of the surface height and the time:

$$\bar{\eta^2} = \frac{\bar{p'^2} t}{2\sqrt{2\rho_w^2 U_c g}} \tag{2.0.8}$$

where $\bar{\eta^2}$ is the mean square surface height, $\bar{p'^2}$ the mean square turbulent pressure on the water surface, t is the time, U_c the convection speed of the surface pressure fluctuations, and ρ_w the water density.

At the same moment, Miles purposed another type of resonance mech-

anism that form the basis of wave formations and that was based on very different considerations and assumptions. In contrast to Phillips, he assumes the air to be inviscid, incompressible and to have a mean shear profile $U(y)$ in absence of wave motion, while the surface is still calm. Fluctuations on pressure and velocity due to turbulence are neglected, while the ones related with wave motion are considered. Because of the above assumptions, the instability of the air is studied starting from the inviscid Orr-Sommerfeld equation (also called Rayleigh equation) that is the starting point of the theory of stability of laminar flows and it is also the center of Miles' theory. On the other hand water motion is assumed to be two-dimensional, inviscid, incompressible and irrotational; moreover the slope of the wave surface is considered sufficiently small to make possible the linearization of the equations. The mean motion of the water, that may be generated by the mean air flow, is assumed to be null.
The Miles' model, instead of Phillips, is a coupled model, since that the excitation (air flow) is dependent on the response (wave motion). This coupling between air and water leads to instability of the wind shear profile that occurs when there is an inflectional point in the mean velocity ($d^2U/dy^2 = 0$) and the shear stress is maximum in that point (so that dU/dy is maximum). This condition is obtained by solving the eigenvalue problem of the Rayleigh equation. Therefore a wave motion of the type $\eta = ae^{ik(x-ct)}$ will be stable or unstable according to the curvature of the wind profile at the height where the wind speed is equal to the wave speed (the so called "Miles' critical height" h_{cr}): if $U''(h_{cr}) < 0$ then it is unstable, otherwise it is stable. The energy transfer form the wind to the water happen in this critical layer over the free surface where the instability occurs.
Another important result of the Miles' theory was that the growth rate of the wave appears to depend exponentially on the time and not linearly as described by Phillips in (2.0.8). This result was in better agreement with the experiments carried out in those years, even if we can imagine that was not so easy to measure both the free surface elevation and the wind induced fluctuations at the same time, resulting in not so accurate and precise measurements. Anyway the Miles' model is limited by the neglect of turbulent interaction between surface wave and wind profile, that is playing a fundamental role in the neighbourhood of the free surface.
Nevertheless, Phillips and Miles mechanisms are considered the most important theories of wind wave generation. Sometimes they are considered as consequential processes: the former mechanism is related to the very first stage of wave growth and the latter is considered to happen later. But usually it is regarded to as simultaneous mechanisms.
How does wind acting on water give rise to waves? Thanks to Miles and Phillips contributions, this question has been partly answered. In fact there are still a lot of details to explain in this theory.
In the following years from the 1960 to the 1990 much efforts have been

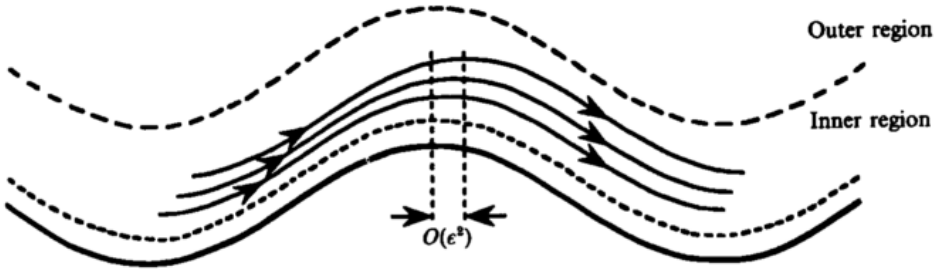


Figure 2.0.5: Non-separated sheltering growth mechanism. The boundary layer is disturbed by the wave motions and becomes thicker on the side after the wave crest (leeside), due to friction forces between the wind and water. This leads to an asymmetric pressure distribution over the free surface.

done to confirm those theories and to overcome their shortcomings through experiment and numerical simulations. The first attempts to introduce turbulence was by means of mixing length models: the turbulent Reynolds stresses coming from the Reynolds averaged Navier-Stokes equation are expressed as a function of the mean velocity by introducing a turbulent eddy viscosity ν_t:

$$- < u_i' u_j' >= \nu_t (\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}) \tag{2.0.9}$$

where ν_t, that represent the rate of interaction of small eddies, is modelled as the product of a mixing velocity v and a mixing length l: $\nu_t = vl$. These parameters are usually represented either by simple linear model, as the Prandtl's mixing length where l is directly proportional to the wall distance, or by more complete model that takes into account the properties of the flow field as the $k - \epsilon$ model, where the eddy viscosity is related to the kinetic energy k and the dissipation energy rate ϵ.
Anyway mixing length models are not suitable for wave generation process since that the eddies in the outer layer, far from the surface, are too slow to transfer a significant amount of momentum on a small time scale as the one of early growth waves. In some sense the eddies of the mean air flow have no enough time for transporting momentum in a wave period. That's why a different approach for describing turbulence, including also smaller eddies acting near the surface, is needed.
The most important efforts to improve the theory behind this phenomenon have been done by Belcher and Hunt in 1993 [9] and in 1998 [10] when they reviewed all the main theories about turbulence over waves. Belcher and Hunt in 1993 introduced the so called non-separated sheltering model, modifying the one of Jeffreys [5].

As shown in figure 2.0.5, this mechanism assumed that, below certain

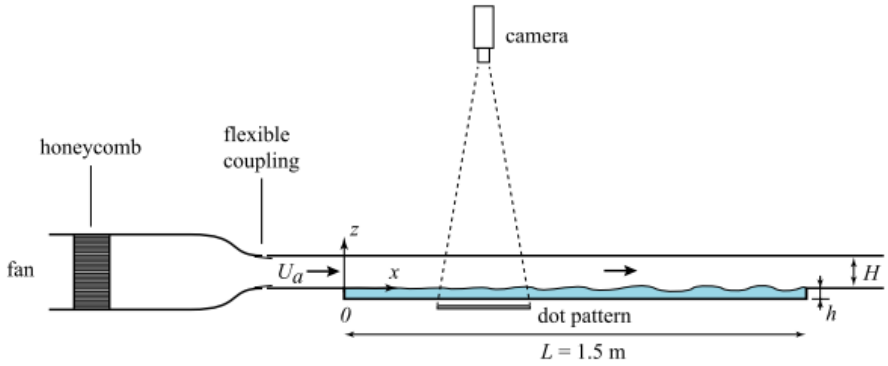


Figure 2.0.6: Experimental setup of FS-SS method used by Paquier and Moisy

value of the wave slope, the wind flow remains attached to the wave surface. The asymmetric pressure distribution is due to the thickening of the boundary layer in the leeside of the wave, instead of the form pressure distribution due to separation. In a boundary layer the pressure is impressed by the outer flow: since that the streamlines of the outer flow are less curved by the thickening of the boundary layer in the leeside part of the wave, then a lower pressure will be imposed to the inner flow with respect to the windward direction and the result will be a force on the wave parallel to the wind direction.
The last recent theoretical model developed on the basis of Phillips theory and rapid distortion models[1], has been presented by Texeira and Beltcher [11] in 2006.
On the experimental side, the improvement of optical methods have made possible the access to surface deformation with very high resolution. We recall here the most recent contributions in this field made by A.Paquier, F.Moisy and M.Rabaud in 2015 [12] using the Free-surface Synthetic Schlieren (FS-SS) technique and Veron and Melville in 2001 [13] by using a Colour Imaging Slope Gauge system and an Infra-red camera.

On the numerical side, with increases in computer power, a DNS approach to this type of problem is now possible. Important and recent works about simulation of turbulent flows over waves have been made by Sullivan, Moeng, Tsai in [14] and [15] starting from the 2000.

Most of the direct simulations made to study the wave formation process are made on stationary waves or wavy boundaries as shown in figure 2.0.7, thus without considering the water surface at rest, but starting directly from

[1]Rapid distortion models are those ways of simulating turbulence when it is distorted by a curvy surface. They are widely used to study problems like flows over hills or waves in geophysical fluid dynamics.

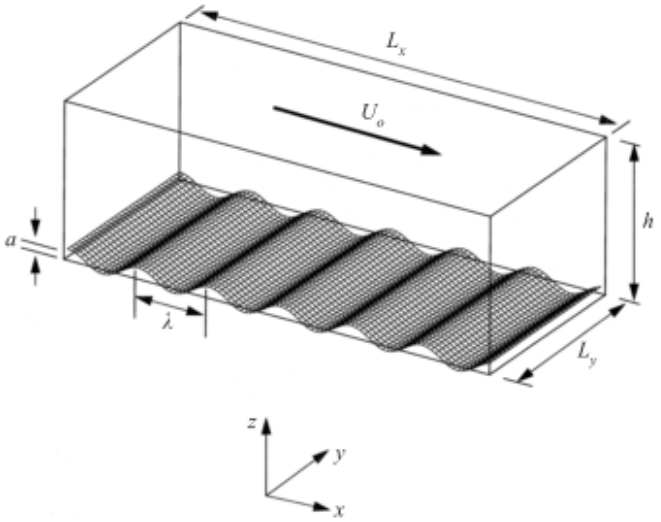
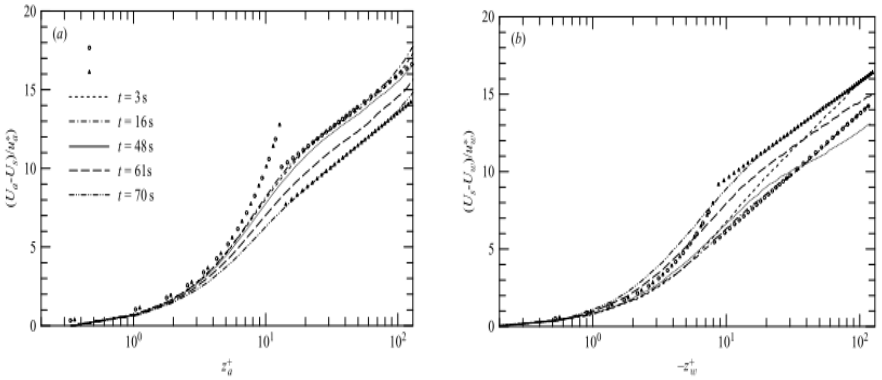


Figure 2.0.7: Numerical simulation domain used by Sullivan

a solid stationary or moving wave. This simplification of the problem was due to the lack of suitable models to simulate water, air and their interface in turbulent conditions. Nowadays this is possible thanks to the development of very accurate interface-capturing approach as the Volume of Fluid method, firstly thought and developed by Hirt and Nichols [16] in 1981 and now implemented also in commercial CFD programs as OpenFOAM.
Recently, numerical studies have been conducted on turbulence effects in two-phase flow, but just few of them deal with the wind wave generation processes as the work of Lin, Moeng and Tsai [15] in 2008. In this work a Direct Numerical Simulation of two turbulent flows (air and water) interacting across a deformable interface under the action of a wind is performed. The wind is simulated by an upper moving wall at 3 m/s as in a Couette Flow. The turbulent Reynolds number in the air measured at the upper wall is $Re_\tau = u_\tau h/2\nu$ is about 115, where h is the height of the channel, u_τ the friction velocity at the wall in the air and ν is the air kinematic viscosity. The mean streamwise velocity profiles obtained through this DNS are shown in figure 2.0.8. This work will be taken as our main reference to compare and validate the results obtained in this thesis.

The main outcomes from this DNS simulation are listed below:

- At the first seconds, the early stage, the wave growth behaviour is linear with time, thus according to Phillips mechanism that remains the main factor for the initiation and the support of wave motion. However, the measured growth rates are 1-2 times larger than what



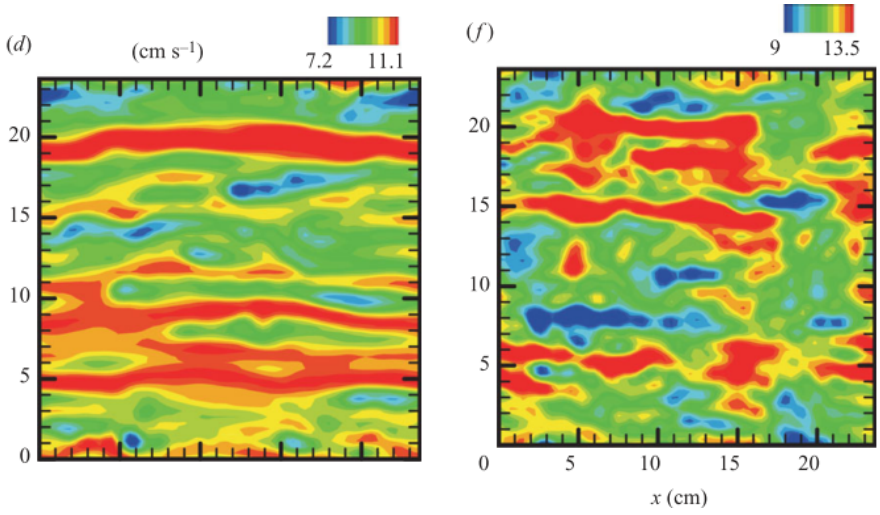
(a) (b)

Figure 2.0.8: Mean streamwise velocity profile obtained by Lin, Moeng and Tsai in [15]. Different time instants of the wave growth process are shown.

predicted by the theory of Phillips in 1957. The flow field in this linear stage has a streaky structure (see figure 2.0.9a), typical of shear driven turbulent flows over a flat surface.

- At the later stage the wave growth behaviour is exponential with time, consistently to what predicted by the Beltcher and Hunt non-separated sheltering mechanism in 1993. In this phase the form stress, the stress due to different pressure distribution along the free surface caused by the wave slope (sheltering mechanism), is the main actor in transferring energy from wind to waves. The streaky structure of the flow is interrupted by wave motions (see figure 2.0.9b) that strongly affect the flow field in the thin viscous sublayer over the surface.

Despite all the improvements made both on the experimental side, by the introduction of new optical methods to measure field variables, and on the numerical side, by the implementation of accurate multiphase numerical model and the increasing of computational power, the interaction between wave and wind is not completely explained yet. The mechanisms that generate surface waves are still an open issue, especially from the theoretical side: the mathematical difficulties in analysing and modelling a turbulent flow over a complex moving surface are still very difficult obstacles to overcome.



(a) (b)

Figure 2.0.9: Snapshots of instantaneous streamwise velocity along the channel free surface at respectively $t = 2.6$ seconds and $t = 66$ seconds

Chapter 3

Governing Equations and Numerical simulation

In this chapter we will analyse the mathematics of the model of a two phase flow and its free surface. All the assumptions made to solve our problem are faced in the first section where the fundamental equations governing an incompressible fluid are explained. The work of professor L.Davidson [17] and the books of S.Pope [18] and Schlichting [19], together with the notes of the courses in Applied Aerodynamics of professor A.Talamelli and in Simulation and modelling in fluid dynamics of professor A.Cimarelli, have been used to write down this section.
The second and the third sections cover the interface capturing method used to model the free surface, that is the Volume of fluid method, and the way it is implemented in OpenFOAM within the InterFoam solver, respectively. To understand correctly this implementation, some elements of finite volume method and discretization of the equations has been introduced in the third section, following the works of Jasak [20], Ubbink [21] and Rusche [22]. Regarding the VOF method and interFoam solver we referred to Hirt and Nichols [16] and Lopes [23].
The simulation that has been done during this work to understand the interaction between turbulence and water waves is finally described in the last section. The main references used for the pre-processing are the OpenFOAM users guide [24], the programmers guide [25] and the very useful extended code guide [26].

3.1 Fundamental assumptions and equations

Water and air are the most common fluid in this world and we know exactly what are the laws that govern them. If we consider either water or air, separately, we can easily describe their behaviour through the Navier-Stokes equations, that were formulated around the 1830. The main assumptions

behind these equations are the following:

- the fluid is a continuum and can be described without taking into account the motion of each molecule, but defining macroscopic properties (as density, pressure and bulk velocity) at infinitesimal volume elements (fluid particle) without committing large errors.
- the "principle of the local state"(or "localization lemma") is valid in the sense that at each point in the flow field the same equations of state hold as in a system at rest. In a mathematical view we could say that if $f(x, y, z, t)$ is a continuously differentiable function inside the volume v, than $\int_v f(x, y, z, t) = 0$ if $f(x, y, z, t) = 0$.
- the fluid is newtonian and thus the stress tensor T_{ij} is linearly proportional to the deformation tensor. Moreover the it is isotropic and symmetric.
- the fluid is a Stokes fluid and thus the Stokes hypothesis is valid: the transfer of molecular energy (thermodynamic pressure) due to the bulk viscosity (rotation, vibration and intermolecular attraction) is negligible compared to the transfer due to mechanical pressure (translation of molecules).

The fluid flow can be described by means of the three conservation laws: mass, momentum and energy. The Navier-Stokes equations are the union of the three equations describing these laws. In this thesis a further assumption has been done: the effect of temperature is not considered and the fluid is treated as isothermal. Therefore we will not deal with the internal energy conservation law, but only with mass and momentum conservation.
Taking into account all the assumptions made before, the mass conservation law, also known as the continuity equation, and the momentum balance, written in index notation, are:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0 \qquad \text{Mass conservation} \tag{3.1.1}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_j u_i) = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + f_i \qquad \text{Momentum conservation} \tag{3.1.2}$$

where in the equation (3.1.2) on the LHS the first term is the local time variation of momentum (ρu_i); the second term is the transport of momentum due to the velocity field (convective rate of change). On the RHS there are all the sources of momentum: the surface forces, the first and second term, and the body forces, the last term. The first term is the pressure gradient acting on the fluid, where p is the mechanical pressure (that coincide

with the thermodynamic one under the stokes hypothesis), that is defined as third part of the trace of the stress tensor. The second term, instead, is the deviatoric part of the stress tensor, thus the tangential stresses acting on the material volume of fluid. This term represent the diffusion of momentum through viscosity. Under the assumptions of Newtonian and Stokes fluid we can write τ_{ij} as:

$$\tau_{ij} = 2\mu S_{ij} - \frac{2}{3}\mu\frac{\partial u_k}{\partial x_k}\delta_{ij} \qquad where \quad S_{ij} = \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}) \tag{3.1.3}$$

The body forces, indicated in equation (3.1.2) as f_i are essentially represented by the gravity force by writing $f_i = \rho g_i$.
Finally for a multiphase flow we should also include the surface tension force f_σ as a source of momentum. This will be done later in section 3.3.
Regarding the mass conservation, it is convenient to decompose the second term and rewrite it as:

$$\frac{\partial \rho}{\partial t} + u_j\frac{\partial \rho}{\partial x_i} + \rho\frac{\partial u_i}{\partial x_i} = 0 \tag{3.1.4}$$

where: the first term is the time rate of change of density; the second is the convective rate of change due to the velocity field; the last term is taking into account the velocity of the fluid particle itself (this term arises from the application of the Reynolds transport theorem).
Equations (3.1.4) and (3.1.2) are the most general form of the Navier-Stokes equations and they are well defined once the momentum per unit volume ρu_i is a continuous function all over the domain. This is always verified for a domain in which only a fluid is present, instead the situation changes if we have a multiphase flow. In fact, in our study, we have to consider a domain where two immiscible fluids, water and air, are present. Firstly we can state that air and water, in our case, can be consider as incompressible fluids: the density of water ($\rho_w = 1000\ kg/m^3$) remains always the same all over the time and space and so does the density of air ($\rho_a = 1\ kg/m^3$). In other words, the mass of water and air can be changed inside the domain only by a flow across the boundaries. Moreover, the velocity function $u_i(x, y, z, t)$ is, by definition, continuous through all the domain and also at the free surface; this means that across the free surface, while the density is abruptly changing, the velocity remains continuous. These comments about the density and velocity can be used to further simplify the (3.1.4), that has been introduced for this purpose: the first and second term cancel out since that density, for each fluid, is constant in time and space ($\frac{D\rho}{Dt} = 0$, the material or total derivative of density is null). Therefore for each incompressible fluid the mass conservation can be rewritten as:

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{3.1.5}$$

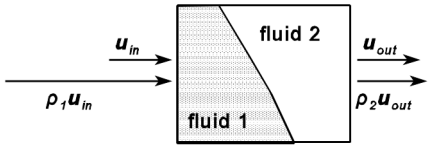


Figure 3.1.1: Continuity of the velocity and discontinuity of the momentum in two phase flows

The equation (3.1.5) will be called the phase continuity equation, since that it regards the continuity of the velocity function for each phase.
What about the momentum? Does it remain continuous through all the domain? How does it behave across the free surface?
To explain this concept, let's consider the situation in figure 3.1.1.

The fluid 1 with higher density, that could be water in our case, is entering in the domain with a velocity u_{in} and pushing the fluid 2 with lower density, that could be air in our case, out of the domain with a velocity u_{out}. The velocity of the fluid that is entering is equal to the one that is leaving ($u_{in} = u_{out}$), but the momentum that is flowing inside the domain will be much bigger than the one that is flowing outside ($\rho_1 u_{in} >> \rho_2 u_{out}$). This is due to the high density ratio between the two immiscible fluids.
From this example we can derive that the momentum function ρu_i is not continuous in the domain due to the density discontinuity at the free surface:

$$\rho = \begin{cases} \rho_w & \text{if } (x_i, t) \text{ is inside water} \\ \rho_a & \text{if } (x_i, t) \text{ is inside air} \end{cases} \tag{3.1.6}$$

Moreover, also the dynamic viscosity will be a step function like density:

$$\mu = \begin{cases} \mu_w = 10^{-3} \text{ Pa s} & \text{if } (x_i, t) \text{ is inside water} \\ \mu_a = 1.48 * 10^{-5} \text{ Pa s} & \text{if } (x_i, t) \text{ is inside air} \end{cases} \tag{3.1.7}$$

As a consequence of this discontinuity, we cannot use the equation (3.1.2) since that the momentum function ρu_i should be continuous and differentiable over the whole flow domain. Therefore, in order to rewrite the momentum balance in suitable way for multiphase flows, we need to introduce a new method for considering the free surface and making the momentum function continuous and differentiable. This method is the so called Volume of Fluid method (V.O.F.) and will be described in section 3.2.
In this section we have faced the main mathematical problems related to free surface flows and their description. We have found a correct form for the continuity equation, the phase continuity equation (3.1.5), while we still miss two important ingredients for a proper description of the momentum balance: the volume of fluid and the surface tension force.

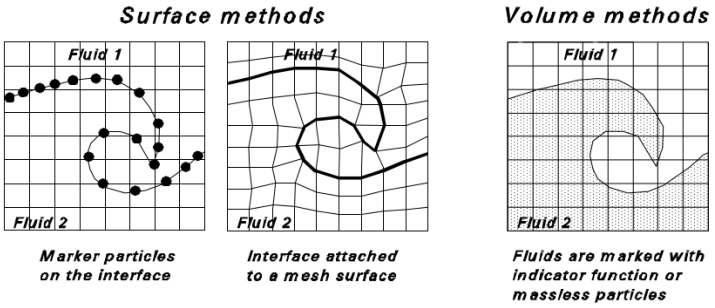


Figure 3.2.1: Surface fitting and surface capturing techniques

3.2 The VOF method

The discontinuity of the momentum across the free surface, that has some analogy with the study of compressible flows, is the main problem that led the researchers to develop suitable methods to analyse and numerically compute free surface flows.
The existing methods for the computation of free surface flows can be grouped into two main categories:

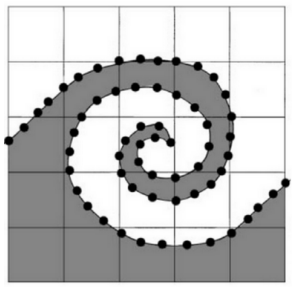
1. **Surface methods**: in these methods the free surface is tracked explicitly either by marking it with special tracer particles (marker particles), or by forcing the mesh to remain attached and move with it. When a surface method is used, then the correspondent numerical technique is called "surface fitting";

2. **Volume methods**: in this second category the free surface and its particles are tracked by some massless particles or by a indicator function. When a surface method is used, then the correspondent numerical technique is called "surface capturing".

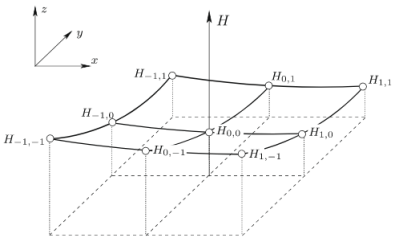
A schematic representation of these 2 categories is shown in figure 3.2.1.

In this thesis we will just mention and briefly explain all the existing methods that have been developed in the past years from the 1969 up to the 2000, while we will linger over the volume of fluid method.
Regarding the surface methods (surface fitting), as already explained, the free surface is tracked with marker points located at the interface, and then some interpolation method is used in order to find the surface between those points. We can distinguish three main ways of marking the free surface:

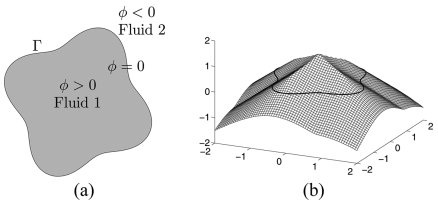
- **Interface particles**: a set of massless particles, as shown in figure 3.2.2a, is spread all over the free surface and each particles is convected by the fluid velocity in its location in a Lagrangian manner. This method is not so accurate when the particles distance becomes large



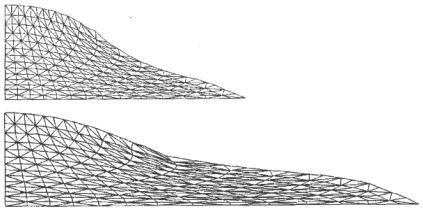
(a) *Interface particles method*



(b) *Height functions method*



(c) *Level set function method*



(d) *Surface fitted method*

Figure 3.2.2: Examples of surface methods applications

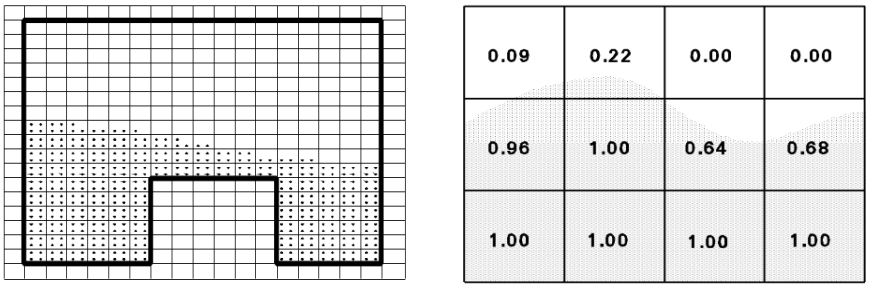
and it requires a large memory to storage the position of all marker particles.

- **Height functions**: the free boundary is identified by defining its distance from a reference line (the free surface height η) as a function of position along that reference line (as shown in figure 3.2.2b). In case of a free fluid surface, the surface height evolution in time and space $\eta(x,t)$ is governed by the following kinematic boundary condition:

$$\frac{\partial \eta}{\partial t} + u\frac{\partial \eta}{\partial x} = v \tag{3.2.1}$$

where (u, v) are the component of the velocity field along (x, y). Essentially through the condition (3.2.1) (that is the same boundary condition used in the linear wave theory), we are saying that the surface must move with the fluid. This kinematic boundary condition is suitable only for two dimensional problems and has the severe limitation that does not work for all those shapes, like bubbles,drops or breaking waves, having multiple y values in correspondence of a single x.

- **Level set methods**: a continuous scalar function $\phi(x, y, z, t)$, called the level set function is introduced all over the fluid domain (see figure 3.2.2c). The value of the level set function at each point is defined as



(a) *Marker particles method* (b) *Volume fractions method*

Figure 3.2.3: Examples of volume methods applications

the minimum distance between that point and the interface . Therefore, the interface is defined to be where the function has a value of zero. In order to distinguish between the two fluids, on each side of the interface a negative sign is attached to the scalar function for one of the fluids. In this way the fluid have a scalar property, the level set function, that is propagating solving a scalar convection equation. Anyway some numerical problems could arise when free surface regions ($\phi = 0$) intersect each others, as for example two merging bubbles.

- **Surface fitted methods**: these methods comprehend all the techniques where the mesh is being attached to the interface. These methods are limited to interfaces which are not subjected to large deformations, because these lead to significant distortion of the mesh as illustrated in figure 3.2.2d.

On the other hand, the second main category of interface capturing methods, the volume methods, can be subdivided in two different techniques:

- **Marker particles**: it is very similar to the interface particles, but, since that we are treating volume methods instead of surface methods, the marker particles are spread over all fluid occupied region, thus on both side of the interface as shown in figure 3.2.3a. Each particles is specified to move with the fluid velocity in its location in a Lagrangian manner. The free surface is defined by looking at those mesh cell containing markers, but having at least a neighbouring cell without any particle. This method can treat very complex phenomena as wave breaking and can be used also for three dimensional computations. Anyway, it requires a large computational power and memory to store and follow the motion of a large number of particles.

- **Volume fractions**: this methods comprehend a lot of scheems, among which the Volume of fluid (VOF) method is the one that will be used

in this study and it was introduced by Hirt and Nichols in 1981 [16]. However the common feature of volume fraction methods is that an indicator scalar function $\alpha(x_i, t)$ is used to represent the fractional volume of a cell occupied by the fluid as written in the following:

$$\alpha = \frac{V_{fluid}}{V_{cell}} \qquad \alpha = \begin{cases} 1 & \text{if } (V_{fluid} = V_{cell}) \\ 0 & \text{if } (V_{fluid} = 0) \\ 0 < \alpha < 1 & \text{if } (V_{fluid} < V_{cell}) \end{cases} \tag{3.2.2}$$

As can be seen from figure 3.2.3b, the cell with α values between 0 and 1 must then contain the free surface. The main advantage of this method is the less required computational power with respect to marker particles method, since that in order to capture the interface we need only to storage one variable, the indicator function, instead of a large number of particles.
Obviously the description of the method is not complete since that we have still to define two aspects: the first is the algorithm for accurately computing the evolution of the indicator function (called the advection technique); the latter is the way in which the free surface is reconstructed once the value of α of each cell is known (the so called reconstruction technique). Many different advection and reconstruction techniques have been developed in order to have an accurate tracking of the free surface, but we will focus only in the one used by the VOF method developed by Hirt and Nichols.

The VOF method is perhaps the most widely used and implemented method for analysing free surface flows. In fact it is also present in the OpenFOAM CFD software where it was implemented by Ubbink in 1997, within the InterFOAM solver for multiphase flows. In this method the advection of the indicator function is governed by the usual transport equation used for describing a substance being transported by a bulk motion:

$$\frac{\partial \alpha}{\partial t} + \frac{\partial}{\partial x_j}(u_j \alpha) = 0 \tag{3.2.3}$$

In incompressible flows, saying conservation of mass is equivalent to saying conservation of volume and therefore, equation (3.2.3) is expressing the conservation of the volume fraction. In a lagrangian view of the problem this would mean that the volume fraction will remain constant in each cell. The way in which the flux of α from a cell (donor) to its neighbour (acceptor) is evaluated (in other words, the scheme used to discretize the advection equation), together with the reconstruction technique used to redefine the interface, is the so called "donor-acceptor scheme" and it is well explained both in the original article by Hirt and Nichols [16] and by Ubbink [21]. We

will deal with all the numerical problems related with this computation in the next subsection, but for now we will jump directly to the application of the VOF method to the momentum equation.
As explained in section 3.1, the VOF method has to be introduced in order to make the momentum function ρu_i continuous across the interface and model the two immiscible fluids as an effective continuous one. In order to do so, we cannot use the definition of α as a step function as shown in equation (3.2.2), instead we need to define a transition region of finite thickness δ, located between the 2 fluids, in which α is smoothly varying from 0 to 1. Obviously this region does not exist in reality, but it is a very good approximation of the real free surface behaviour. Perhaps, at this moment, some questions could arise: what do we mean with smoothly varying indicator function? What is the degree of smoothness that is required from our set of equations? The most stringent requirement on the smoothness of the indicator function is not given by the density (and consequently viscosity) function in (3.1.2), but from the equation used to evaluate the interface curvature in the model of the surface tension force, that is given in the next section. From this model the indicator function must be twice differentiable. As a consequence we can define this function as a twice differentiable function that fulfil the following requirements

$$\alpha(x_i,t) = \begin{cases} 1 & \text{if } (x_i,t) \text{ is inside fluid 1 (water)} \\ 0 & \text{if } (x_i,t) \text{ is inside fluid 2 (air)} \\ 0 < \alpha_\delta < 1 & \text{if } (x_i,t) \text{ is inside the transition region } \delta \end{cases} \tag{3.2.4}$$

and his propagation in time is given by equation (3.2.3).
Therefore now we can finally use a single momentum equation for the two fluids system (water and air), defining new continuous functions for density and viscosity as following:

$$\begin{cases} \rho = \rho_w \alpha + \rho_a (1-\alpha) \\ \mu = \mu_w \alpha + \mu_a (1-\alpha) \end{cases} \tag{3.2.5}$$

Through equations (3.2.3), (3.2.4) and (3.2.5) we have completed the interface capture model. However, we need to discuss some critical issues of these equations and introduce some corrections to solve them. This will be done in the next subsection.

3.2.1 VOF critical issues and relative solutions

The main issues of volume methods, as the VOF method, have to be found in the discretization of the advection equation (3.2.3). When we approach

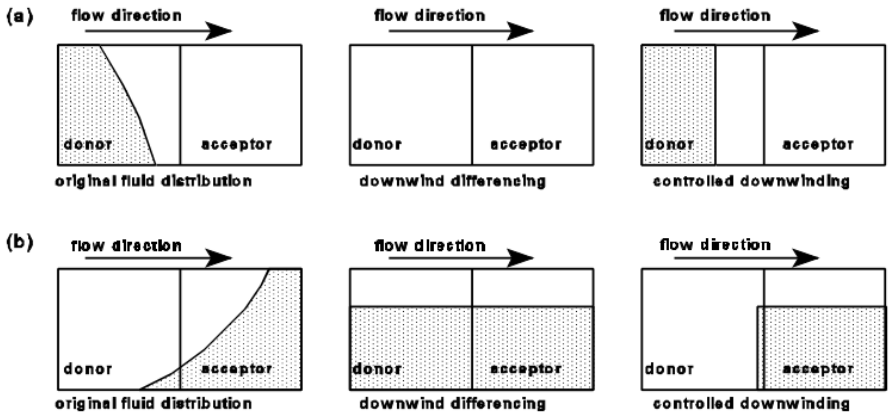
the computation of this equation by normal differencing schemes (as central differencing, downwinding and upwinding), we face two main problems:

1. **the boundedness of** α: a normal differencing scheme may give rise to non-physical volume fraction values larger than unity or smaller than zero when it is applied to the equation. In other words, the phase fraction is not conserved during its transport from one cell to another, meaning that some fluid mass has been generated from nowhere.

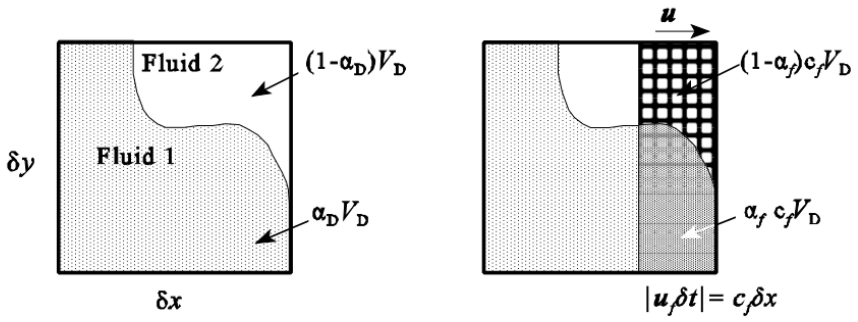
2. **smearing of sharp interface**: normal differencing schemes applied to (3.2.3), introduce too much numerical diffusion and smear the step profile of the interface over several cells. In this way sharp interface cannot be detected.

In the VOF method the first problem is solved by introducing a controlled downwinding inside the donor-acceptor scheme. However it still present the second problem of distortion of the interface. For solving this problem an artificial compression term is added to the advection equation. This solution is the one implemented in OpenFOAM and was found by Weller in 2002 [27]. The strong coupling between the boundedness of VOF scheme and the artificial compression term allows the InterFOAM solver to have a very accurate interface resolution.
First of all, to explain the controlled downwinding scheme, we need to introduce the donor-acceptor scheme. Let's imagine that the fluid is being transported from one cell to the neighbouring only along the x direction. In this case the direction of the velocity vector u_i is defining also the direction of the volume fraction flux across the face between the two neighbouring cells. We will then call the cell that is loosing a certain amount of volume fraction the "donor" cell, while the "acceptor" cell will be the one gaining that amount of volume of fluid. Therefore the acceptor cell and donor cell will depend on the direction of the flow as illustrated in figure 3.2.4a.

In order to clearly understand the downwinding scheme and his controlled version we need to introduce some basic parameters of the donor cell as shown in figure 3.2.4b: V_D is the volume of the donor cell and α_D is the value of the phase fraction in the donor cell. As a consequence $\alpha_D V_D$ will be the amount of fluid 1 (water) that is present in the donor cell, while $(1-\alpha_D)V_D$ will be the amount of fluid 2 (air) that is there. Clearly the velocity u is the velocity with which the volume fraction is being transported across the cell face f. Finally the volume fraction of the acceptor cell is written as α_A.
The goal of the differentiation scheme is to find a correct way for defining α_f, that is the volume fraction of the donor cell to be convected to the acceptor cell in the time interval δt.
The normal downwinding scheme use only the acceptor cell volume of fluid $(\alpha_A V_A)$ to predict α_f, implying that the donor cell has to donate the same



(a) *Donor-acceptor scheme*



(b) *Donor cell parameters*

Figure 3.2.4: The donor-acceptor scheme

fluid as presently contained in the acceptor cell ($\alpha_f = \alpha_A$), ignoring the presence of the other fluid in the donor cell: no matter if the acceptor cell is requesting more fluid than what is actually present in the donor cell. The exceeding amount will be created from nowhere, leading to non-physical results and not fulfilling the conservation of the fluid mass, as explained previously.
Therefore, a controlled downwinding scheme is introduced to fulfil the following requirements:

- **Boundedness**: no fluid mass can be either generated or destroyed, but can only be transported ($0 < \alpha < 1$);
- **Availability**: the amount of fluid flux (either fluid 1 or 2) required by the acceptor cell cannot be bigger than the amount of fluid available in the donor cell.

The availability requirement can be formulated in mathematical terms by defining the so called "interface Courant number" as $C_f = |u_f \delta t|/\delta x$, and expressing the amount of water transported as $\alpha_f C_f V_D$ (see figure 3.2.4b). So the availability requirements, one for each fluid, read:

$$\text{water transported} < \text{water available} \Rightarrow \alpha_f C_f V_D < \alpha_D V_D \tag{3.2.6}$$

$$\text{air transported} < \text{air available} \Rightarrow (1-\alpha_f) C_f V_D < (1-\alpha_D) V_D \tag{3.2.7}$$

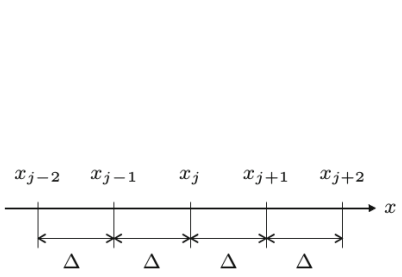
Simplifying and putting together equations (3.2.7) and (3.2.6) we find:

$$\frac{\alpha_D}{C_f} - \frac{1-C_f}{C_f} < \alpha_f < \frac{\alpha_D}{C_f} \tag{3.2.8}$$

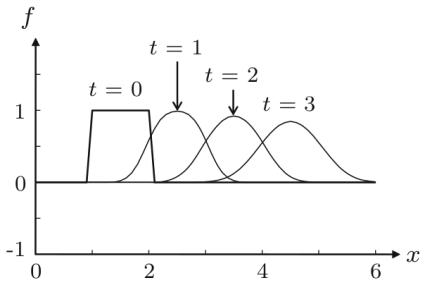
The final decision on how much is α_f is made by using the following "MIN-MAX" criteria:

$$\alpha_f = MIN\left\{ MAX\left\{ \frac{\alpha_D}{C_f} - \frac{1-C_f}{C_f}, \alpha_A \right\}, \frac{\alpha_D}{C_f} \right\} \tag{3.2.9}$$

This condition expressed in (3.2.9) is the way in which the controlled downwinding ensure boundedness and availability of the volume fraction. The MIN feature prevents the fluxing of more fluid from the donor cell than it has to give, while the MAX feature accounts for an additional fluid flux, whether the volume requested by the acceptor exceed the available amount. Controlled downwinding means that the donor cell will firstly donate all the available fluid required by the acceptor cell and then will start to donate the other fluid. It can be explained by looking at the examples shown in figure 3.2.4a. This is the way in which the VOF method has solved the problem of boundedness of α.
As we have previously mentioned, another problem arises from the application of a downwinding scheme: the numerical diffusion. Let's consider for a



(a) *Stencil of a uniform grid*



(b) *Smearing effect of numerical diffusion*

Figure 3.2.5: The numerical diffusion error

moment that the transport of α is only along the x direction with a constant velocity u, so is governed by the following transport (or advection) equation:

$$\frac{\partial \alpha}{\partial t} + u\frac{\partial \alpha}{\partial x} = 0 \tag{3.2.10}$$

Equation (3.2.10) does not contain any diffusion term, meaning that the function α will be transported through the domain with a velocity u and without changing his shape. We introduce the downwinding scheme to differentiate the transport term along a uniform grid as shown in figure 3.2.5a:

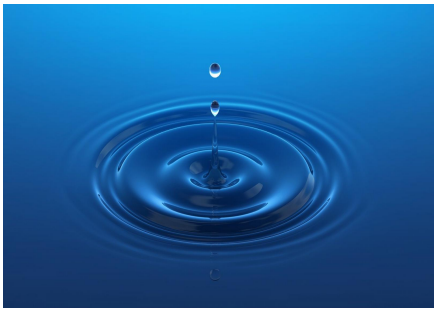
$$\frac{\partial \alpha}{\partial t}|_j + u\frac{\alpha_{j+1} - \alpha_j}{\Delta} = 0 \tag{3.2.11}$$

If know we substitute the Taylor expansion of α_{j+1} inside the equation (3.2.11) we end up with the following one (evaluated for the j point of the grid):

$$\frac{\partial \alpha}{\partial t} + u\frac{\partial \alpha}{\partial x} + u\frac{\Delta}{2}\frac{\partial^2 \alpha}{\partial x^2} + u\mathrm{O}(\Delta^2) = 0 \tag{3.2.12}$$

where we can notice that the truncation error due to the downwind scheme (the third and fourth terms of the LHS) becomes a numerical diffusion term in the transport equation (numerical in the sense that is not physical). The effect of this term is shown in figure 3.2.5b where f could be interpreted as our volume fraction function : the sharp shape of α is not maintained over the time and it diffuses over the other cells (smearing effect). In order to oppose the effect of diffusion and compress the interface back to the original sharp configuration, Weller inserted a compressive terms inside the governing equation of the phase fraction:

$$\frac{\partial \alpha}{\partial t} + \frac{\partial}{\partial x_j}(\overline{u_j}\alpha) + \underbrace{\frac{\partial}{\partial x_j}(u_{r_j}\alpha(1-\alpha))}_{\text{compressive term}} = 0 \tag{3.2.13}$$



(a) *Water capillarity waves*

(b) *Water striders*

Figure 3.3.1: Natural phenomena where the surface tension force is acting

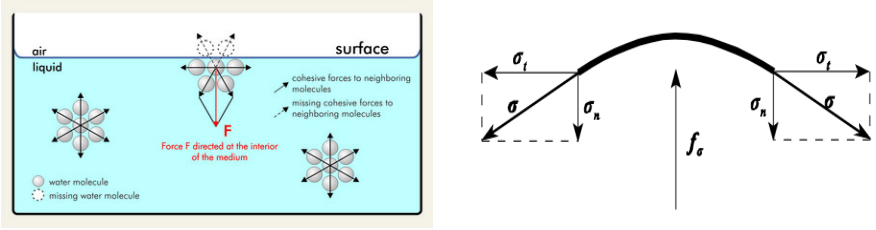
In equation (3.2.13) we have defined two new velocity fields:

$$\begin{cases} u_{r_j} = u_{w_j} - u_{a_j} & \text{compression velocity} \\ \overline{u_j} = \alpha u_{w_j} + (1-\alpha) u_{a_j} & \text{weighted averaged velocity} \end{cases} \tag{3.2.14}$$

where with u_{w_j} and u_{a_j} are respectively the velocity of the fluid 1 (water) and the fluid 2 (air). The basic idea behind this model is that the contributions of air and water velocities to the evolution of the free surface are proportional to their relative phase fraction. This is the reason why a weighted averaged velocity $\bar{u_j}$ is considered to act in the vicinity of the interface. Clearly, when we are outside the transitional region where $0 < \alpha < 1$, the equation (3.2.13) comes back to its original form (see equation (3.2.2)) and the compressive term is no more acting; this aspect can be demonstrated by substituting definitions (3.2.14) inside the original equation (3.2.2) and considering the weighted averaged velocity instead of the usual velocity field. Therefore the compressive term is acting only in the free surface zone and is helping us to obtain a sharper interface.
Equation (3.2.13) was the remaining equation to close our interface-capturing model before introducing the surface tension force model and finally close the entire system of equations.

3.3 The surface tension

Surface tension is very important in many natural phenomena involving free surface flows. For example in capillary and ultragravity waves, as explained in section 1.2, surface tension is the main restoring force. As another example, a lot of light insects are able to walk or run on the water surface thanks to this force (see figure 3.3.1) Therefore we cannot neglect it, especially for our thesis objectives.



(a) *Microscopic view:cohesive forces* (b) *Surface tension balance*

Figure 3.3.2: Illustration on how the surface tension acts

To explain how this force works, we need to look at the microscopic features of a free surface as illustrated in 3.3.2a. Cohesive forces, the attractive intermolecular forces that exist between the molecules of a fluid, are the responsible for the surface tension: a molecule of water that is found to be inside the fluid, is pulled from the surroundings by the same force in all the directions, resulting in a net null force; instead, a molecule that is located along the free surface, will be pulled inward since that the adhesive forces (between molecules of different fluid as water and air) are much lower than cohesive forces (between molecules of the same fluid). This inward force all over the free surface causes the interface to behave as an elastic membrane, where the molecules tend to remain attached each others and the free surface area occupied tend to be minimized.

When a force is applied to this liquid elastic membrane, as the wall adhesion or the weight of an object, the area of the interface start to increase. Than the inward force opposes any increase in the free surface area. It is clear that the surface tension force will be proportional to the amount of interface area created and so to the energy needed for this increase. That is why we define the surface tension coefficient σ as the amount of work necessary to create a unit area of free surface. The magnitude of this coefficient is strictly dependent on the nature of the two fluids that are in contact. In our case, for water and air at room temperature, we will consider $\sigma = 0.07$ N/m or J/m^2.
It can be easily observed every day that when cohesive forces are stronger than adhesives forces, the liquid form a convex meniscus(as mercury in a glass container), while when adhesive forces are stronger, the surface of the liquid curves up (as water in a glass): the free surface is not flat, but has a curvature, even if the two fluids in contact are in equilibrium and the free surface is not accelerating. In this case the surface tension force will have not only a tangential component but also a normal one as shown in figure 3.3.2b. If the 2 fluids are in equilibrium, since that the tangential forces σ_t cancel out each others, there should exist a normal force f_σ (caused by a

pressure jump) over the interface that balance the resultant of the normal components σ_n. Otherwise, if this force is not able to counteract this interface pressure jump, the free surface will start to accelerate. From these observations we can derive some basic features of the surface tension force:

1. it is strictly related to the pressure jump across the free surface $f_\sigma = \partial P/\partial x_i$;
2. the pressure jump is a function of the surface tension coefficient σ and the curvature of the free surface κ; thus we can write: $\Delta P = \sigma\kappa$;
3. the surface tension force is acting only at the free surface.

Translating this three statements into a mathematical formula we can write:

$$f_\sigma = \int_{s(t)} \sigma\kappa_{fs} n_{i_{fs}} \delta(x_i - x_{i_{fs}}) dS \tag{3.3.1}$$

where κ_{fs}, n_{ifs} and $x_{i_{fs}}$ indicates respectively the mean curvature, the normal vector and the position vector at the free surface. Instead the function:

$$\delta(x_i - x_{i_{fs}}) = \begin{cases} 1 & \text{if } (x_i = x_{i_{fs}}) \text{ the point is on the free surface} \\ 0 & \text{if } (x_i \neq x_{i_{fs}}) \text{ the point is not on the free surface} \end{cases} \tag{3.3.2}$$

is translating in mathematical terms the feature number 3.
Now we can finally add to the equation (3.1.2) the source of momentum due to surface tension force:

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_j u_i) = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho g_i + f_\sigma \quad \text{Momentum conservation} \tag{3.3.3}$$

We should notice that the model used for the surface tension force presents two main problems: the first is its discontinuity through the flow domain due to the term described in (3.3.2), so that some difficulties could arise in evaluating that integral; the latter is that it is not explicit in the sense that f_σ depends on the interface shape and position that is unknown, because it is not tracked explicitly in the volume of fluid method.
This two problems are solved by introducing the continuous surface force model (CSF) of the surface tension force. In this model f_σ is considered as a continuous volume force, such as gravity, acting only in the transition region, thus in the region where the indicator function α is smoothly varying between 0 and 1 (see figure 3.3.3).

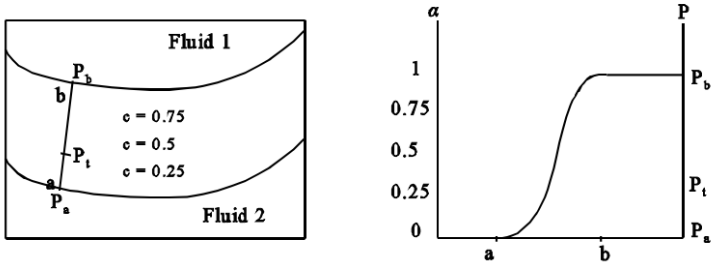


Figure 3.3.3: The transitional area: α and p pressure jump are smoothly varying.

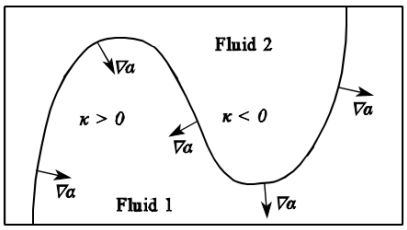


Figure 3.3.4: Representation of normal and curvature at the interface in the CSF model

In this model we read:

$$f_\sigma = \int_{s(t)} \sigma \kappa_{fs} n_{i_{fs}} \delta(x_i - x_{i_{fs}}) dS \approx \sigma \kappa n_i \tag{3.3.4}$$

$$n_i = \frac{\partial \alpha}{\partial x_i} \tag{3.3.5}$$

$$\kappa = -\frac{\partial}{\partial x_i}\left(\frac{n_i}{|n_i|}\right) \tag{3.3.6}$$

By using equation (3.3.5) we have defined the normal vector to the layers in the transitional region as the gradient of the indicator function. The curvature of each layer, instead, is defined in (3.3.6) by the divergence of the unit normal vector. Since that α has been defined as twice differentiable and in the solution procedure it is evaluated before solving the momentum equation, then all these quantities are continuous through all the domain and the problems of the initial force model is solved. Last, a schematic representation of the relations between indicator function, normals and curvature is shown in figure 3.3.4.

3.4 The interFoam solver

InterFoam, as written in the OpenFOAM user guide [24],is a solver for 2 incompressible, isothermal immiscible fluids using a VOF (volume of fluid) phase-fraction based interface capturing approach. In this section we will see both how and where the equations of our mathematical model are written and the way in which they are solved. Before doing so, we have first to define our model: with respect to all the equations that we have presented up to now, to complete our system and to write down it in the exact form in which is implemented in the solver, we need to make two further modifications.
The first one is to simplify the stress tensor τ_{ij} described in equation (3.1.3), by applying the continuity equation and obtaining:

$$\tau_{ij} = 2\mu S_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \tag{3.4.1}$$

Then we can rearrange the gradient of this term as following:

$$\frac{\partial \tau_{ij}}{\partial x_j} = \frac{\partial}{\partial x_j} \left(\mu \frac{\partial u_i}{\partial x_j} \right) + \frac{\partial u_i}{\partial x_j} \frac{\partial \mu}{\partial x_j} \tag{3.4.2}$$

The second modification concerns the definition of pressure. In equation (3.3.3), the first term of the RHS is the pressure gradient where p is the static pressure. This kind of pressure will never appear in the InterFoam solver because it will complicate the setting of boundary conditions: since that the static pressure contains inside the term due to the hydrostatic component $\rho g_i x_i$, all the boundary conditions on pressure should be function of the position of the cell. Instead if we define a new variable $p_{rgh} = p - \rho g_i x_i$, called the static contact pressure, and we substitute it inside the momentum equation, we can easily define uniform boundary conditions on p_{rgh} all over the domain, without considering whether where is a fluid rather than another. To do not make confusion between all the pressure definitions existing in the fluid dynamic world, in table 3.4.1 has been reported some basic notations used for all this pressures.

The static pressure gradient can be rewritten as:

$$\frac{\partial p}{\partial x_i} = \frac{\partial}{\partial x_i}(p_{rgh} + \rho g_i x_i) = \frac{\partial p_{rgh}}{\partial x_i} + \rho g_i + g_i x_i \frac{\partial \rho}{\partial x_i} \tag{3.4.3}$$

Taking into account all the previous considerations coming from section 3.1, 3.2, the CSF model of surface tension in 3.3 and the last two simplifications made in this section, we can finally write down all the equations representing the mathematical model of a free surface flow between water and air using a VOF interface capturing method:

Pressure	Description
Static	Actual pressure of the fluid which is associated not with his motion but with its state. It is defined as the mean of the three normal stresses acting over three fluid element surfaces orthogonal to each other, in the point at rest with respect to the fluid. For a real fluid in motion this pressure is just a convention, while it can be defined only for an ideal fluid at rest. From the British Standards Institutions it is defined as "The pressure at a point on a body moving with the fluid".
p	$p = -\frac{1}{3}\sum_{i=1}^{3} T_{ii}$ where T_{ii} is the stress tensor of the fluid element
Hydrostatic	The part of the static pressure that is due entirely to the body forces, such as gravity, and not considering any inertial or viscous force.
p_{hyd}	$p_{hyd} = \rho g h$ where h is the height of the fluid column above the fluid element.
Total	The maximum pressure measured when the fluid is brought at rest through an isoentropic and adiabatic process. At a stagnation point, where the fluid velocity is null, without considering the gravity contribution, the total pressure coincides with the stagnation pressure (static pressure at a stagnation point). This always holds for incompressible flows.
p_{tot}	$p_{tot} = p + p_{dyn}$
Dynamic	The difference between the total pressure and the static pressure. When the Bernoulli's law hold then: $p_{dyn} = \frac{\rho U^2}{2}$
p_{dyn}	$p_{dyn} = p_{tot} - p$
Static contact	The static pressure due only to contact forces or surface forces, avoiding the contribution of gravity
p_{rgh}	$p_{rgh} = p - p_{hyd}$

Table 3.4.1: Different type of pressure

Fundamental equations:

Continuity:

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{3.4.4}$$

Momentum:

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_j u_i) = \\ = -\frac{\partial p_{rgh}}{\partial x_i} - g_i x_i \frac{\partial \rho}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\mu \frac{\partial u_i}{\partial x_j}\right) + \frac{\partial u_i}{\partial x_j}\frac{\partial \mu}{\partial x_j} - \sigma \frac{\partial}{\partial x_i}\left(\frac{n_i}{|n_i|}\right)\frac{\partial \alpha}{\partial x_i} \tag{3.4.5}$$

Advection of α:

$$\frac{\partial \alpha}{\partial t} + \frac{\partial}{\partial x_j}(\overline{u_j}\alpha) + \frac{\partial}{\partial x_j}(u_{r_j}\alpha(1-\alpha)) = 0 \tag{3.4.6}$$

Constitutive equations:

Physical properties:

$$\rho = \rho_w \alpha + \rho_a(1-\alpha) \tag{3.4.7}$$

$$\mu = \mu_w \alpha + \mu_a(1-\alpha) \tag{3.4.8}$$

Definition of α:

$$\alpha(x_i,t) = \frac{V_{fluid}}{V_{cell}} = \begin{cases} 1 & \text{if } (x_i,t) \text{ is inside fluid 1 (water)} \\ 0 & \text{if } (x_i,t) \text{ is inside fluid 2 (air)} \\ 0 < \alpha_\delta < 1 & \text{if } (x_i,t) \text{ is inside the transition region } \delta \end{cases} \tag{3.4.9}$$

Compressive terms:

$$u_{r_j} = u_{w_j} - u_{a_j} \tag{3.4.10}$$

$$\overline{u_j} = \alpha u_{w_j} + (1-\alpha)u_{a_j} \tag{3.4.11}$$

In our case of study the free surface is between water and air, thus the constant $\sigma = 0.07\ Kg/s^2$. The y-axis is oriented along the channel height and therefore $g_i = (0, 9.81, 0)\ m/s^2$.
All the fundamental equations ((3.4.4), (3.4.5), (3.4.6)) and the constitutive equations ((3.4.7), (3.4.8), (3.4.9), (3.4.10), (3.4.11)) are discretized and solved by the InterFoam solver. Obviously this is not an easy task because they present the same problems of Navier Stokes equations (among all the problems we could underline the non linearity and the coupling between velocity and pressure), and, additionally, they are complicated by the presence of the phase fraction.
In the following, as explained at the beginning of this section, we will describe how they are written and how they are solved. This two aspects are

very important to have clear in mind how our simulation is working, and, if it does not, to detect eventual errors. In fact, control and modify equations and procedures is one of the main advantages of using the OpenFOAM software: all the algorithms are accessible and modifiable allowing the user to adapt a solver to any particular necessity.

3.4.1 Equations code and solver procedure

Once OpenFOAM is installed in your calculator, you can find all the applications in the installation folder named "app"; all the solvers can be found in this folder and they are written in C++ language. The InterFOAM solver is found to be among the multiphase solvers and we can understand how it works only by looking the main file "interFoam.C". In the following we will show this file and comment it line by line.

Code 3.1: Main solver file interFoam.C

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 2011-2016 OpenFOAM Foundation
     \\/     M anipulation  |
-------------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software: you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM.  If not, see <http://www.gnu.org/licenses/>.

Application
    interFoam

Description
    Solver for 2 incompressible, isothermal immiscible fluids using a VOF
    (volume of fluid) phase-fraction based interface capturing approach.

    The momentum and other fluid properties are of the "mixture" and a single
    momentum equation is solved.

    Turbulence modelling is generic, i.e. laminar, RAS or LES may be selected.

    For a two-fluid approach see twoPhaseEulerFoam.

\*---------------------------------------------------------------------------*/

#include "fvCFD.H"
```

```
41  #include "CMULES.H"
42  #include "EulerDdtScheme.H"
43  #include "localEulerDdtScheme.H"
44  #include "CrankNicolsonDdtScheme.H"
45  #include "subCycle.H"
46  #include "immiscibleIncompressibleTwoPhaseMixture.H"
47  #include "turbulentTransportModel.H"
48  #include "pimpleControl.H"
49  #include "fvOptions.H"
50  #include "CorrectPhi.H"
51  #include "localEulerDdtScheme.H"
52  #include "fvcSmooth.H"
53
54  // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
55
56  int main(int argc, char *argv[])
57  {
58      #include "postProcess.H"
59
60      #include "setRootCase.H"
61      #include "createTime.H"
62      #include "createMesh.H"
63      #include "createControl.H"
64      #include "createTimeControls.H"
65      #include "createRDeltaT.H"
66      #include "initContinuityErrs.H"
67      #include "createFields.H"
68      #include "createFvOptions.H"
69      #include "correctPhi.H"
70
71      turbulence->validate();
72
73      if (!LTS)
74      {
75          #include "readTimeControls.H"
76          #include "CourantNo.H"
77          #include "setInitialDeltaT.H"
78      }
79
80      // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
81
82      Info<< "\nStarting␣time␣loop\n" << endl;
83
84      while (runTime.run())
85      {
86          #include "readTimeControls.H"
87
88          if (LTS)
89          {
90              #include "setRDeltaT.H"
91          }
92          else
93          {
94              #include "CourantNo.H"
95              #include "alphaCourantNo.H"
96              #include "setDeltaT.H"
97          }
98
99          runTime++;
100
101         Info<< "Time␣=␣" << runTime.timeName() << nl << endl;
102
```

```
103         // --- Pressure-velocity PIMPLE corrector loop
104         while (pimple.loop())
105         {
106             #include "alphaControls.H"
107             #include "alphaEqnSubCycle.H"
108 
109             mixture.correct();
110 
111             #include "UEqn.H"
112 
113             // --- Pressure corrector loop
114             while (pimple.correct())
115             {
116                 #include "pEqn.H"
117             }
118 
119             if (pimple.turbCorr())
120             {
121                 turbulence->correct();
122             }
123         }
124 
125         runTime.write();
126 
127         Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
128             << "  ClockTime = " << runTime.elapsedClockTime() << " s"
129             << nl << endl;
130     }
131 
132     Info<< "End\n" << endl;
133 
134     return 0;
135 }
136 
137 
138 // ************************************************************************* //
```

- **Lines 1-38**: in all the OpenFOAM solvers, the first lines are dedicated to the description of the solver.

- **Lines 40-52**: in this first introductory part all the useful files.H are included in the solver. All the basic solvers usually share this first part because there are very common tools inside: all the finite volume method machinery for discretizing the equations is included in "fvCFD.H"; "CrankNicolsonDdtScheme.H" and "EulerDdtScheme.H" are telling us the time differentiation scheme that are supported by interFoam; "turbulentTransportModel.H" is including all the turbulence models as laminar (that in OpenFoam means DNS), RAS (Reynolds Avaraged simulation), LES (Large eddies simulation) or DES (Detached eddy simulation); "fvOptions.H" is allowing us to insert additional source terms inside the momentum equation and it will be used in our simulations to insert a prescribed pressure gradient to emulate the wind blowing; "pimpleControl.H" calls the solver PimpleFoam that is a merged PISO-SIMPLE algorithm for solving the

velocity-pressure coupling in the fundamental equations. Regarding the PIMPLE algorithm, it could allow us to use larger time steps (not fulfilling the requirement of maximum unitary Courant number) also in transient and turbulent flows, taking advantages of the SIMPLE algorithm relaxation factors. Anyway we will not go deep inside this solver, since that, during our simulations, the requirement on the maximum Courant number will be always fulfilled and, therefore, using PimpleFoam will be exactly the same as using PisoFoam.

- **Lines 56-69**: here the main function of the solver is started and all its specific "files.H" are called: "createTime.H" and "createMesh.H" are discretizing the domain both in time and space; "createFields.H" is a very important file that initialize and create all the fields needed for the calculation, reading them from the disk. This last file is shown in code 3.2 and commented to explain all the passages.

Code 3.2: Create Fields

```
1  Info<< "Reading␣field␣p_rgh\n" << endl;
2  // the file p_rgh is read from the disk
3  // and initialized as a scalar field
4  volScalarField p_rgh
5  (
6      IOobject
7      (
8          "p_rgh",
9          runTime.timeName(),
10         mesh,
11         IOobject::MUST_READ,
12         IOobject::AUTO_WRITE
13     ),
14     mesh
15 );
16
17 Info<< "Reading␣field␣U\n" << endl;
18 // the file U is read from the disk
19 // and initialized as a vector field
20 volVectorField U
21 (
22     IOobject
23     (
24         "U",
25         runTime.timeName(),
26         mesh,
27         IOobject::MUST_READ,
28         IOobject::AUTO_WRITE
29     ),
30     mesh
31 );
32
33 #include "createPhi.H"
34 /* this file is creating the face flux of velocity (Phi)
35  interpolating the vector field through all the cells */
36
37 /* The following TwoPhaseMixture commands are reading
38 the transport properties and creating the new density,
39 viscosity and velocity functions */
```

```
40  Info<< "Reading transportProperties\n" << endl;
41  immiscibleIncompressibleTwoPhaseMixture mixture(U, phi);
42
43
44  volScalarField& alpha1(mixture.alpha1());
45  volScalarField& alpha2(mixture.alpha2());
46
47  const dimensionedScalar& rho1 = mixture.rho1();
48  const dimensionedScalar& rho2 = mixture.rho2();
49
50
51  // Need to store rho for ddt(rho, U)
52  volScalarField rho
53  (
54      IOobject
55      (
56          "rho",
57          runTime.timeName(),
58          mesh,
59          IOobject::READ_IF_PRESENT
60      ),
61      alpha1*rho1 + alpha2*rho2
62  );
63  rho.oldTime();
64
65
66  // Mass flux
67  surfaceScalarField rhoPhi
68  (
69      IOobject
70      (
71          "rhoPhi",
72          runTime.timeName(),
73          mesh,
74          IOobject::NO_READ,
75          IOobject::NO_WRITE
76      ),
77      fvc::interpolate(rho)*phi
78  );
79
80
81  // Construct incompressible turbulence model
82  autoPtr<incompressible::turbulenceModel> turbulence
83  (
84      incompressible::turbulenceModel::New(U, phi, mixture)
85  );
86
87  /* the gravitational constant is read and the new field "gh"
88   is formed for the definition of the total pressure p */
89  #include "readGravitationalAcceleration.H"
90  #include "readhRef.H"
91  #include "gh.H"
92
93
94  volScalarField p //total pressure including the hydrostatic component
95  (
96      IOobject
97      (
98          "p",
99          runTime.timeName(),
100         mesh,
101         IOobject::NO_READ,
```

```
102             IOobject::AUTO_WRITE
103         ),
104         p_rgh + rho*gh
105     );
106
107     /* Pressure is always defined as relative to some reference pressure
108      cell and value. They are initialized as written in the pimple.dict
109      in the system folder. It is very important to put the refCell
110      far from regions where the pressure is really unsteady or where
111      cyclic boundary conditions are applied. */
112     label pRefCell = 0;
113     scalar pRefValue = 0.0;
114     setRefCell
115     (
116         p,
117         p_rgh,
118         pimple.dict(),
119         pRefCell,
120         pRefValue
121     );
122
123     if (p_rgh.needReference())
124     {
125         p += dimensionedScalar
126         (
127             "p",
128             p.dimensions(),
129             pRefValue - getRefCellValue(p, pRefCell)
130         );
131         p_rgh = p - rho*gh;
132     }
133
134     mesh.setFluxRequired(p_rgh.name());
135     mesh.setFluxRequired(alpha1.name());
136
137     // MULES flux from previous time-step
138     surfaceScalarField alphaPhi
139     (
140         IOobject
141         (
142             "alphaPhi",
143             runTime.timeName(),
144             mesh,
145             IOobject::READ_IF_PRESENT,
146             IOobject::AUTO_WRITE
147         ),
148         phi*fvc::interpolate(alpha1)
149     );
150
151     // MULES Correction
152     tmp<surfaceScalarField> talphaPhiCorr0;
153
154     #include "createMRF.H"
```

After the code 3.2 has finished to run, we come back to the main code InterFoam.C.

- **Lines 73-97**: all the time controls are loaded, especially the function "CourantNo.H" that will calculate the time step needed in order to fulfil in each cell the following condition : $Co = |U|\delta t/\delta x < 1$. In this

way the stability of the solution is ensured in each cell.

- **Lines 99-101**: the time is incremented and displayed on the screen.
- **Lines 103-107**: finally we enter inside the most important part of the solver where the equations are written and solved. This part is called the "pressure velocity PIMPLE corrector loop". At the beginning of this loop all the quantities relative to the phase fraction are calculated and the advection equation 3.4.6 is solved by calling it in the file "alphaEqnSubCycle.H". This file is shown in code 3.3 and its function is essentially to compute the advection equation, inside the same time interval δt, as many times as specified by the number of subcycles ("nAlphaSubCycles").

Code 3.3: Alpha subcycles

```
1  if (nAlphaSubCycles > 1)
2  {
3      dimensionedScalar totalDeltaT = runTime.deltaT();
4      surfaceScalarField rhoPhiSum
5      (
6          IOobject
7          (
8              "rhoPhiSum",
9              runTime.timeName(),
10             mesh
11         ),
12         mesh,
13         dimensionedScalar("0", rhoPhi.dimensions(), 0)
14     );
15
16     tmp<volScalarField> trSubDeltaT;
17
18     if (LTS)
19     {
20         trSubDeltaT =
21             fv::localEulerDdt::localRSubDeltaT(mesh, nAlphaSubCycles);
22     }
23
24     for
25     (
26         subCycle<volScalarField> alphaSubCycle(alpha1, nAlphaSubCycles);
27         !(++alphaSubCycle).end();
28     )
29     {
30         #include "alphaEqn.H"
31         rhoPhiSum += (runTime.deltaT()/totalDeltaT)*rhoPhi;
32     }
33
34     rhoPhi = rhoPhiSum;
35 }
36 else
37 {
38     #include "alphaEqn.H"
39 }
40
41 rho == alpha1*rho1 + alpha2*rho2;
```

For each subcycle the density and the velocity flux are reinitialized and reinserted in the file "alphaEqn.H" where we can see all the equations that we discussed in section 3.2. This file is shown and commented in code 3.4.

Code 3.4: Alpha Equation solver

```
{
    word alphaScheme("div(phi,alpha)");
    word alpharScheme("div(phirb,alpha)");

    tmp<fv::ddtScheme<scalar>> ddtAlpha
    (
        fv::ddtScheme<scalar>::New
        (
            mesh,
            mesh.ddtScheme("ddt(alpha)")
        )
    );

    // Set the off-centering coefficient according to ddt scheme
    scalar ocCoeff = 0;
    if
    (
        isType<fv::EulerDdtScheme<scalar>>(ddtAlpha())
     || isType<fv::localEulerDdtScheme<scalar>>(ddtAlpha())
    )
    {
        ocCoeff = 0;
    }
    else if (isType<fv::CrankNicolsonDdtScheme<scalar>>(ddtAlpha()))
    {
        if (nAlphaSubCycles > 1)
        {
            FatalErrorInFunction
                << "Sub-cycling is not supported "
                   "with the CrankNicolson ddt scheme"
                << exit(FatalError);
        }

        ocCoeff =
        refCast<const fv::CrankNicolsonDdtScheme<scalar>>(ddtAlpha())
        .ocCoeff();
    }
    else
    {
        FatalErrorInFunction
        << "Only Euler and CrankNicolson ddt schemes are supported"
        << exit(FatalError);
    }

    scalar cnCoeff = 1.0/(1.0 + ocCoeff);

    // Standard face-flux compression coefficient
    surfaceScalarField phic(mixture.cAlpha()*mag(phi/mesh.magSf()));

    // Add the optional isotropic compression contribution
    if (icAlpha > 0)
    {
        phic *= (1.0 - icAlpha);
        phic += (mixture.cAlpha()*icAlpha)*fvc::interpolate(mag(U));
```

```
55      }
56
57      surfaceScalarField::Boundary& phicBf =
58          phic.boundaryFieldRef();
59
60      // Do not compress interface at non-coupled boundary faces
61      // (inlets, outlets etc.)
62      forAll(phic.boundaryField(), patchi)
63      {
64          fvsPatchScalarField& phicp = phicBf[patchi];
65
66          if (!phicp.coupled())
67          {
68              phicp == 0;
69          }
70      }
71
72      tmp<surfaceScalarField> phiCN(phi);
73
74      // Calculate the Crank-Nicolson off-centred volumetric flux
75      if (ocCoeff > 0)
76      {
77          phiCN = cnCoeff*phi + (1.0 - cnCoeff)*phi.oldTime();
78      }
79
80      if (MULESCorr)
81      {
82          fvScalarMatrix alpha1Eqn
83          (
84              (
85                  LTS
86               ? fv::localEulerDdtScheme<scalar>(mesh).fvmDdt(alpha1)
87               : fv::EulerDdtScheme<scalar>(mesh).fvmDdt(alpha1)
88              )
89            + fv::gaussConvectionScheme<scalar>
90              (
91                  mesh,
92                  phiCN,
93                  upwind<scalar>(mesh, phiCN)
94              ).fvmDiv(phiCN, alpha1)
95          );
96
97          alpha1Eqn.solve();
98
99          Info<< "Phase-1 volume fraction = "
100             << alpha1.weightedAverage(mesh.Vsc()).value()
101             << "  Min(" << alpha1.name() << ") = " << min(alpha1).value()
102             << "  Max(" << alpha1.name() << ") = " << max(alpha1).value()
103             << endl;
104
105         tmp<surfaceScalarField> talphaPhiUD(alpha1Eqn.flux());
106         alphaPhi = talphaPhiUD();
107
108         if (alphaApplyPrevCorr && talphaPhiCorr0.valid())
109         {
110         Info<< "Applying the previous iteration compression flux" << endl;
111         MULES::correct(alpha1, alphaPhi, talphaPhiCorr0.ref(), 1, 0);
112
113             alphaPhi += talphaPhiCorr0();
114         }
115
116         // Cache the upwind-flux
```

```
117         talphaPhiCorr0 = talphaPhiUD;
118
119         alpha2 = 1.0 - alpha1;
120
121         mixture.correct();
122     }
123
124
125     for (int aCorr=0; aCorr<nAlphaCorr; aCorr++)
126     {
127         surfaceScalarField phir(phic*mixture.nHatf());
128
129         tmp<surfaceScalarField> talphaPhiUn
130         (
131             fvc::flux
132             (
133                 phi,
134                 alpha1,
135                 alphaScheme
136             )
137           + fvc::flux
138             (
139                -fvc::flux(-phir, alpha2, alpharScheme),
140                 alpha1,
141                 alpharScheme
142             )
143         );
144
145         // Calculate the Crank-Nicolson off-centred alpha flux
146         if (ocCoeff > 0)
147         {
148             talphaPhiUn =
149                 cnCoeff*talphaPhiUn + (1.0 - cnCoeff)*alphaPhi.oldTime();
150         }
151
152         if (MULESCorr)
153         {
154             tmp<surfaceScalarField> talphaPhiCorr(talphaPhiUn() - alphaPhi);
155             volScalarField alpha10("alpha10", alpha1);
156
157             MULES::correct(alpha1, talphaPhiUn(), talphaPhiCorr.ref(), 1, 0);
158
159             // Under-relax the correction for all but the 1st corrector
160             if (aCorr == 0)
161             {
162                 alphaPhi += talphaPhiCorr();
163             }
164             else
165             {
166              alpha1 = 0.5*alpha1 + 0.5*alpha10;
167              alphaPhi += 0.5*talphaPhiCorr();
168             }
169         }
170         else
171         {
172          alphaPhi = talphaPhiUn;
173
174          MULES::explicitSolve(alpha1, phiCN, alphaPhi, 1, 0);
175         }
176
177         alpha2 = 1.0 - alpha1;
178
```

```
179             mixture.correct();
180         }
181
182         if (alphaApplyPrevCorr && MULESCorr)
183         {
184             talphaPhiCorr0 = alphaPhi - talphaPhiCorr0;
185         }
186
187         if
188         (
189             word(mesh.ddtScheme("ddt(rho,U)"))
190          == fv::EulerDdtScheme<vector>::typeName
191         )
192         {
193             rhoPhi = alphaPhi*(rho1 - rho2) + phiCN*rho2;
194         }
195         else
196         {
197             if (ocCoeff > 0)
198             {
199           // Calculate the end-of-time-step alpha flux
200           alphaPhi = (alphaPhi - (1.0 - cnCoeff)*alphaPhi.oldTime())/cnCoeff;
201             }
202
203             // Calculate the end-of-time-step mass flux
204             rhoPhi = alphaPhi*(rho1 - rho2) + phi*rho2;
205         }
206
207         Info<< "Phase-1 volume fraction = "
208             << alpha1.weightedAverage(mesh.Vsc()).value()
209             << "  Min(" << alpha1.name() << ") = " << min(alpha1).value()
210             << "  Max(" << alpha1.name() << ") = " << max(alpha1).value()
211             << endl;
212 }
```

- **Line 109**: all the mixture variables such as ρ, ν, μ are corrected to be inserted in the momentum equation.

- **Line 111**: this phase of the PIMPLE solver is called "momentum predictor" and it is introduced by the file "UEqn.H", where the momentum equation is implemented and solved. This name is due to the fact that the velocity solution obtained by solving this equation is not the exact one, but it is just a prediction and an approximation. In fact, to solve the equation, the pressure at the previous time step is used and, therefore, the velocity field obtained is not fulfilling the continuity equation (in the sense that is not divergence free, or is not solenoidal), but only satisfying the momentum one. In other words, in incompressible flows where continuity is satisfied, the pressure gradient is the term that is making the velocity field solenoidal. Since that pressure cannot be calculated at the same time of velocity due to their coupling in the momentum equation, we need to split their calculation into three parts: first we solve the momentum equation considering the pressure at the previous time step (momentum predictor); second we use the predicted velocity to find the amount of pressure needed to

make that velocity a solenoidal field (pressure solution) in the current time step; third we add the gradient of this pressure to the predicted velocity to find the right velocity field (velocity correction). After the third step is reached, the loop is restarted and the new velocity field is evaluated by using the old pressure field and so on. The momentum predictor equations are shown and commented in code 3.5.

Code 3.5: Momentum predictor "UEqn.H"

```
1  MRF.correctBoundaryVelocity(U);
2  /* The LHS of the momentum equation is shown here: */
3      fvVectorMatrix UEqn
4      (
5          fvm::ddt(rho, U) + fvm::div(rhoPhi, U)
6        + MRF.DDt(rho, U)
7        + turbulence->divDevRhoReff(rho, U)
8       ==
9          fvOptions(rho, U)
10     );
11 /* fvOptions allow us to insert new sources of
12    momentum as for example a pressure gradient
13    simulating the wind blowing */
14     UEqn.relax();
15
16     fvOptions.constrain(UEqn);
17 /* the momentum equation is completed by inserting
18    the surface tension force, and the pressure
19    gradient */
20     if (pimple.momentumPredictor())
21     {
22         solve
23         (
24             UEqn
25          ==
26             fvc::reconstruct
27             (
28                 (
29                     mixture.surfaceTensionForce()
30                   - ghf*fvc::snGrad(rho)
31                   - fvc::snGrad(p_rgh)
32                 ) * mesh.magSf()
33             )
34         );
35
36         fvOptions.correct(U);
37     }
```

- **Lines 113-123**: The second step of PIMPLE (pressure solution) is started and the pressure field is calculated solving the pressure equation as implemented in the file "pEqn.H". The relative code is shown and commented in code 3.6. The pressure corrector loop is run as many times as specified by the voice "nCorrectors" in the system directory of our case of study. After the pressure is evaluated by solving a Laplace equation, the velocity field is updated and corrected as shown in the final part of the code.

Code 3.6: Pressure equation "pEqn.H"

```
{
    volScalarField rAU("rAU", 1.0/UEqn.A());
    surfaceScalarField rAUf("rAUf", fvc::interpolate(rAU));

    volVectorField HbyA(constrainHbyA(rAU*UEqn.H(), U, p_rgh));

     surfaceScalarField phiHbyA
     (
         "phiHbyA",
         fvc::flux(HbyA)
       + fvc::interpolate(rho*rAU)*fvc::ddtCorr(U, phi)
     );
     MRF.makeRelative(phiHbyA);
     adjustPhi(phiHbyA, U, p_rgh);

     surfaceScalarField phig
     (
         (
             mixture.surfaceTensionForce()
           - ghf*fvc::snGrad(rho)
         )*rAUf*mesh.magSf()
     );

     phiHbyA += phig;

     // Update the pressure BCs to ensure flux consistency
     constrainPressure(p_rgh, U, phiHbyA, rAUf, MRF);

/* The pressure equation has the form of a  laplace
   equation and is shown in the following */

     while (pimple.correctNonOrthogonal())
     {
         fvScalarMatrix p_rghEqn
         (
             fvm::laplacian(rAUf, p_rgh) == fvc::div(phiHbyA)
         );

 p_rghEqn.setReference(pRefCell, getRefCellValue(p_rgh, pRefCell));

 p_rghEqn.solve(mesh.solver(p_rgh.select(pimple.finalInnerIter())));

/* This is the beginning of the velocity correction step */

         if (pimple.finalNonOrthogonalIter())
         {
             phi = phiHbyA - p_rghEqn.flux();

             p_rgh.relax();

      U = HbyA + rAU*fvc::reconstruct((phig - p_rghEqn.flux())/rAUf);
      U.correctBoundaryConditions();
      fvOptions.correct(U);
         }
     }
/* Continuity errors are shown on the screen through the
   following function */
     #include "continuityErrs.H"

     p == p_rgh + rho*gh;

```

```
62      if (p_rgh.needReference())
63      {
64          p += dimensionedScalar
65          (
66              "p",
67              p.dimensions(),
68              pRefValue - getRefCellValue(p, pRefCell)
69          );
70          p_rgh = p - rho*gh;
71      }
72  }
```

- **Lines 125-130**: once the Pimple loop is finished, the execution time and the clock time of the simulation are printed and a new loop is started.

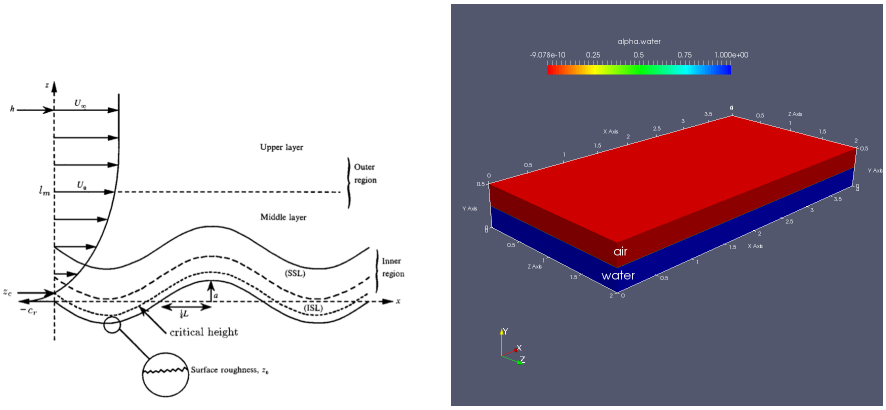
Since that the main files involved in our simulation have been explained, we can now briefly summarize the procedure followed by the interFoam solver to find the right solution:

1. All the variables are initialized through the domain and the time step is adjusted by considering the requirement on the Courant number;
2. The transport equation for the indicator function (equation (3.4.6)) is solved and all the physical properties are corrected (equations (3.4.7) and (3.4.8));
3. The PIMPLE loop starts and the momentum equation (equation (3.4.5)) is solved by consider the pressure at the previous time step;
4. The new pressure solution is founded by the pressure corrector loop and the velocity field is updated. The loop is finally restarted;

3.5 Simulation details

In this section we will deal with the practical part of this thesis. One simulation has been successfully performed and a numerical model for studying the wind-wave interfacial phenomena has been developed. These phenomena, as mentioned in section 2, has never been studied before using the VOF method as implemented in the OpenFOAM suite.
All the relative simulation parameters are explained and described in the following:

- **The physical system and his domain**: once we start a simulation it is always of fundamental importance to know what we are trying to model, what is the physical process behind this model, and what we are trying to see through our simulation. All these things can be easily understood, looking at the thesis title : “Numerical study of



(a) *Schematic representation of the simulation*

(b) *The domain of the system*

Figure 3.5.1: The physical system and his domain

wind-wave interfacial phenomena". Wind and wave, so air and water respectively, are the main actors of the simulation: a turbulent wind is blowing over the water surface in order to study the generation process of waves and the air-water interaction along the free surface, as illustrated in figure 3.5.1a.

The wind intensity at the outer edge of the boundary layer U_∞ is about 0.6 m/s and so the Reynolds number relative to the air boundary layer thickness is $Re_\delta = U_\infty \delta/\nu \approx 10000$. The mean direction of the wind is along the x axis (in the positive sense), that is identifying the length of our channel. The z and y direction are defining the channel width and height respectively, as shown in figure 3.5.1b. The dimensions of our domain will be reported as adimensional numbers related to the height of the water column h_w:

$$h^+ = \frac{h}{h_w} = 2; \qquad l^+ = \frac{l}{h_w} = 16; \qquad w^+ = \frac{w}{h_w} = 8;$$

h = channel height $\quad$ l = channel lenght $\quad$ w = channel width

Actually in our case h^+, l^+, w^+ would have the same values if we had defined them with respect to the boundary layer thickness δ, since that it is equal to h_w.

- **The external forcing of the system**: the way in which our flow is driven can be different from one simulation to another: it could be driven by a moving wall, by a pressure gradient, by the gravity field and so on. There are many external ways of forcing a system to move, and each of them will cause a different flow configuration. In the simulation the wind is driven by a pressure gradient, that is imposed

to the system adding a term to the momentum equation, as being explained in subsection 3.4.1. Actually, initializing the simulation, we do not directly define a pressure gradient $\partial p_{rgh}/\partial x_i$ that will act all over the time, instead it is evaluated by fixing the average velocity $\bar{U}$ in our channel, where $\bar{U}$ is defined as:

$$\bar{U} = \frac{\int_V U dV}{\int_V dV} \tag{3.5.1}$$

Substituting the initial velocity profile of our channel (explained later in this section) in (3.5.1), and considering that it is dependent only on the vertical direction y, we will get:

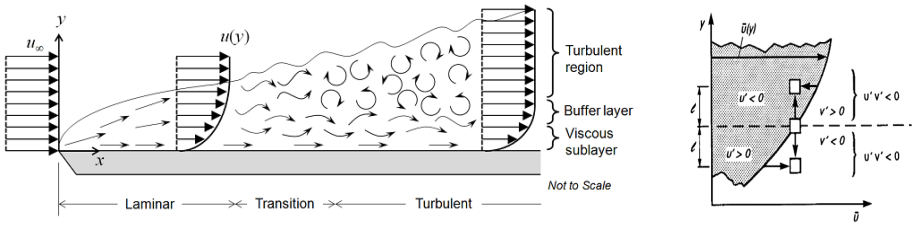
$$\bar{U} = \frac{\int_0^h U(y) dy}{h} = 0.295 \text{ m/s} \tag{3.5.2}$$

Therefore, during each time step of the simulation, the velocity field will be corrected in order to have an average velocity equal to the one obtained from equation (3.5.2). This correction will be made refreshing the pressure gradient each time step and finding its increment needed for fulfilling the requirement on $\bar{U}$. Clearly, this incremented pressure gradient will be then applied to the momentum equation (3.4.5) and weighted with the density of each fluid. The solution of this updated momentum equation will give us the final velocity field. Following this method we can reproduce the effect of a strong wind blowing over a slow moving water free surface.

- **The similarity parameters**: in order to compare two simulations we need them to be similar. Usually, an adimensional number, function of different variables involved in the problem, is found to be the key parameter of the simulation: if that parameter changes, than the simulation changes; instead if it does not change, the simulation will produce the same results. In case of common incompressible flows, this adimensional number will be either the Reynolds number or the friction Reynolds number (Re_τ). The latter is used to compare wall bounded flows, either external or internal, and it is defined as:

$$Re_\tau = \frac{u_\tau \delta}{\nu} \tag{3.5.3}$$

where $u_\tau = \sqrt{\tau_w/\rho}$ is the friction velocity that is the characteristic velocity for turbulent flows at a given wall shear stress, $\tau_w = \mu \frac{dU}{dy}\rfloor_{wall}$ is the wall shear stress (obviously this formula is valid if the y axis is indicating the wall normal direction), δ is the characteristic length scale of our flow. This length scale could be either the height of the



(a) *Boundary layer over a flat plate: from laminar to turbulent flow*

(b) *The turbulent exchange of momentum*

Figure 3.5.2: The momentum transport in turbulent flow

channel for a channel flow, the half of the channel height for a couette flow or the boundary layer thickness for a boundary layer flow. In other words, the characteristic length scale is telling us what is the biggest flow structure (as a vortex for example) that we could encounter in our flow. We can apply all these considerations to the study of a free surface flow, by defining an interface friction Reynolds number:

$$Re_{\tau_{fs}} = \frac{u_{\tau_{fs}}\delta}{\nu} \tag{3.5.4}$$

where the shear stress is calculated at the free surface (indicated by the index "fs"), instead of being evaluated at the wall. The meaning of the turbulent Reynolds number has to be found in the way in which a turbulent flow transport the momentum (the component parallel to the wall), across the wall normal direction. It is worth to spend some words about the concepts that stand behind this adimensional number. Let's consider a flow over a plate, that could represent the free surface in our case, as shown in figure 3.5.2a.

Until the flow is laminar and each particle moves parallel to the wall, the exchange of momentum between the different parallel "fluid layers" is due only to viscosity: close to the wall the particles are slowed down due to the viscous tangential force, but they will keep going in the same parallel directions. No velocity fluctuations along any directions are present, but only a mean velocity gradient due to the no slip condition at the wall. If in some point of the flow the Reynolds number (Re_δ) reaches a certain value for which the inertial forces ($U_\infty\delta$) becomes too big ($Re_\delta >> 1$) compared to the viscous ones (ν), than the fluid starts to move in other directions with respect the main flow current. The equilibrium between the inertial and viscous forces is broken: the viscous tangential stresses are no more able to transport all the amount of momentum now contained in the flow and, as a consequence, the fluid finds a new way to transport this energy excess. These new means of

transport of momentum are the turbulent fluctuations that suddenly appear in all the directions (in figure 3.5.2b only the fluctuating part in the y direction is represented). Therefore, in this turbulent regime, the momentum will be transferred by two mechanisms instead of a single one: firstly by the turbulent fluctuations (u', v', w') and their relative turbulent stresses τ_t (the so called Reynolds stress tensor) and secondly by the viscous stresses τ_v. In figure 3.5.2b is shown how the vertical fluctuations of velocity (v') are acting to transport the momentum along the x direction. In this case the relative stresses developed are: $\tau_v = -\rho < u'v' >$. Now the question is: what is the order of these stresses and what type of stress (turbulent or viscous) is contributing more to the momentum transfer?
The turbulent Reynolds number has been introduced in order to answer these questions. In fact it represents the ratio between the amount of momentum transferred by the fluctuations and by the viscous stresses:

$$Re_{\tau_{fs}} = \frac{u_{\tau_{fs}}\delta}{\nu} = \frac{\text{turbulent momentum transport}}{\text{molecular viscous momentum transport}} \tag{3.5.5}$$

The wall friction velocity u_τ is representing the characteristic velocity of the turbulent fluctuations, meaning that $u', v', w' \approx O(u_\tau)$. For example in our simulation, when the turbulence becomes fully developed, we measured a wall friction velocity $u_\tau \approx 0.02$ m/s in the air boundary layer; with respect to the previous considerations we could say that the velocity fluctuations around the mean flow in our problem will be of the order of 0.02 m/s and the Reynolds stresses $< u'v' > \approx u_\tau^2$.
As $Re_\tau \longrightarrow \infty$ the viscous transport becomes negligible with respect to the turbulent one. In the fully developed state of the simulation we reached a mean $Re_\tau \approx 333$, hence the turbulent transfer appears to be about 300 times larger than the viscous one. Anyway, this last statement is not valid for all the region of the flow, especially it is not true close to the wall where viscous transport is as important as the turbulent transport. This thin layer is consequently called the thin wall sublayer (or inner layer) and it is shown in figure 3.5.2a.
On the other hand, the outer region of the flow where we can neglect the molecular viscous transport, is called the core layer or the outer region. It is clear that in this two layers, separated by a mixing region called the overlap region, the order of magnitude of the length scales of the flow structures, hence the thickness of these layers, is different: on one hand the outer region has a thickness of the order of δ (the channel height as an example) as explained before; on the other hand the sublayer thickness can be determined by the two characteristic parameters u_τ, ν and it will be called the viscous length.

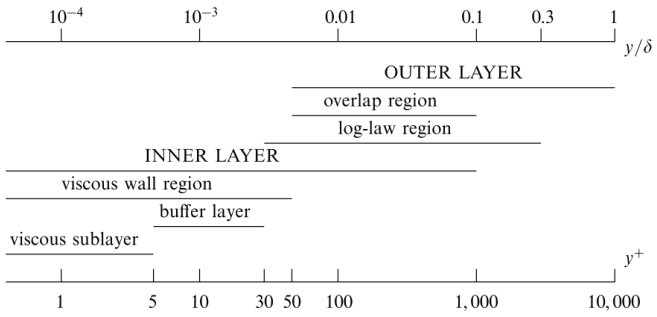


Figure 3.5.3: The different regions of a turbulent flow

$$l_\nu = \nu/u_\tau = \delta/Re_\tau = \text{viscous length}$$

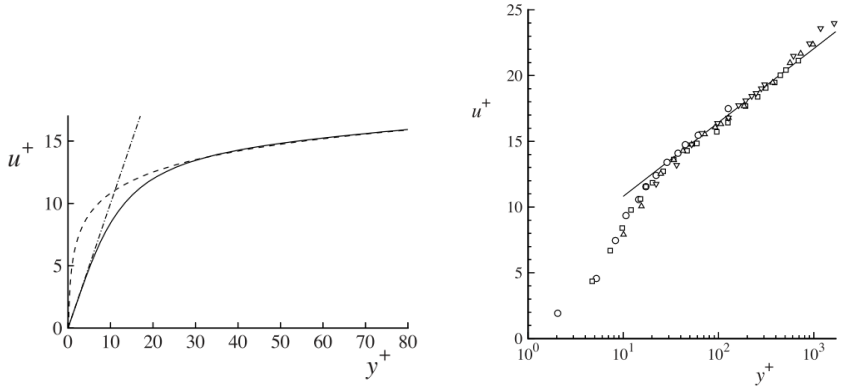
As $Re_\tau \longrightarrow \infty$ the thickness of the inner layer tends to be smaller and smaller than the characteristic thickness of the turbulent flow δ, and all the processes happening inside this layer will be independent on δ. Therefore, if Re_τ is high enough, then all the viscous layers of turbulent flows with finite wall shear stress, will behave in the same manner: the velocity distribution will always follow a law, that, for this reason, is called the "universal law of the wall". To write down this law, we need first to introduce two additional adimensional variables that will be used through all the next section:

$$y^+ = \frac{\nu}{u_\tau} \text{ characteristic stretched wall normal coordinate} \quad (3.5.6)$$

$$U^+ = \frac{U - U_{fs}}{u_\tau} \text{ characteristic velocity} \quad (3.5.7)$$

In (3.5.7) U is indicating the mean velocity and U_{fs} the free surface velocity. Specifying a certain range of y^+ we can differentiate some regions of the inner layer as shown in figure 3.5.3 and we can define the universal laws in each of the inner layer regions (figures 3.5.4a and 3.5.4b):

$$\begin{cases} 0 < y^+ < 5 \text{ pure viscous sublayer } u^+ = y^+ \text{ viscous sublayer law} \\ 5 < y^+ < 50 \text{ buffer layer: matching sublayer and overlap} \\ y^+ > 50 \text{ overlap region } u^+ = \frac{1}{k} log(y^+) + C^+ \text{ logarithmic law} \end{cases} \quad (3.5.8)$$



(a) *Universal laws in linear scale*

(b) *Logarithmic overlap law in logarithmic scale*

Figure 3.5.4: The universal laws of the wall

where $k = 0.41$ is the Karman constant that is almost equal for every wall turbulent flow, while C^+ is a constant depending on the wall roughness. Regarding water waves and their parameters, it is useful to reshape the logarithmic law in order to define the sea surface roughness parameter y_0. Defining this parameter as

$$y_0 = \frac{\nu_a}{u_\tau} \exp^{-kC^+} \tag{3.5.9}$$

we can rewrite the log-law as it is usually reported in oceanographic and geophysics texts:

$$u^+ = \frac{1}{k} log(\frac{y}{y_0}) = \frac{1}{k} log(\frac{y^+}{y_0^+}) \tag{3.5.10}$$

where in (3.5.10) we have introduced the dimensionless sea roughness length $y_0^+ = y_0 u_\tau / \nu_a$. Summarizing all the previous considerations we can say that inside the inner layer, where both turbulent and viscous transport acts to transfer momentum, the velocity and the length is scaled respectively by the friction velocity u_τ and the viscous length l_τ, we find a pure thin viscous sublayer where $u^+ = y^+$ and a region that is connecting the inner layer with the outer one following the logarithmic law.
All the important parameters cited up to now, relatively to the fully developed simulation, are listed in table 3.5.1.

All these parameters will be recalled in the following and we will omit the subscript "fs" since that they will be always evaluated at the free surface.

Interface	Friction Re number	Friction velocity [m/s]	Viscous length [m]
Definition	$Re_{\tau_{fs}} = \frac{u_{\tau_{fs}}\delta}{\nu}$	$u_{\tau_{fs}} = \sqrt{\frac{\tau_{fs}}{\rho}}$	$l_{\nu_{fs}} = \frac{\nu}{u_{\tau_{fs}}}$
Simulation	333	0.0197	$7.5 * 10^{-}4$

Table 3.5.1: Air boundary layer parameters of the simulation

Streamwise	Wall-normal	Spanwise
$N_x = 256$	$N_y = 258$	$N_z = 256$
$\Delta x^+ = 20.83$	$\Delta y^+ = 2.58$	$\Delta z^+ = 10.42$
$\frac{\Delta x}{\Delta y} \approx 8$		$\frac{\Delta z}{\Delta y} \approx 4$

Table 3.5.2: List of dimensionless grid properties

We have introduced all these notions about turbulent wall flows because in the next chapter we will try to match this theory with our physical model, even if we have a free surface instead of a simple wall. The Re_τ parameter is a unique similarity parameter for studying turbulent flow behaviour close to the wall, once the flow is incompressible; in other words, if two flows show the same Re_τ, than their behaviour at the wall will be similar. Anyway for our simulation this is not valid because the interface can deform, and this deformation is not only dependent on Re_τ. This means that if we build two simulations with the same Re_τ at the interface, it is not sure that the free surface behaviour will be similar. We could say that the water waves formed by the interaction of air and water depends not only on what is happening above the free surface, but also under it.
Following the dimensional analysis reported in section 1.2, we should define the wave age number $\beta = c_p/u_\tau$ of our simulation in fully developed regime, by calculating the phase speed at the peak of the wave frequency spectrum. Once β and Re_τ are defined, then any other parameter is a function of them.

- **The adimensional grid properties**: the number of points (N_x, N_y, N_z) that we use to discretize our domain is determining the accuracy of our simulation. In order to say if this number of points is enough to reach a certain accuracy we have to define adimensional parameters for the resolution $(\Delta x^+, \Delta y^+, \Delta z^+)$. These numbers are obtained by scaling the resolution with the viscous length and they are reported in table 3.5.2.

- **The turbulence model**: this point is strictly related to the closure

problem of turbulence, to the grid used in the simulation and to the computational power available. As a flow configuration becomes turbulent, we know that there will be a large range of scales of motion. In order to follow and capture all these scales we need either a very high resolution and computational power or we can solve only some scales and model all the others. In the first case we will be doing a direct numerical simulation (DNS), while in the second we will use one of the turbulence models available up to now, as for example large eddy simulation (LES) or reynolds avarage simulation (RAS). In this simulation, but we are not using any turbulence model we are trying to perform a direct numerical simulation. Actually some grid requirements has to be fulfilled in order to perform a DNS, otherwise we will be doing an implicit LES model and cutting all the scales that are smaller than our resolution (in this case the grid is acting as an implicit filter). The resolution that has been used in the simulation can be considered the best compromise between the computational power available and the requirements requested by a DNS simulation.

- **The initial conditions**: initial conditions are necessary to start solving the numerical model. If we don't know what is the initial configuration of our system, we will not find the configuration at the following time step. Sometimes the initial conditions could also be non-physical, like a strange perturbation applied to the velocity field in order to trigger turbulence, just to reach our desired final conditions in a faster way.
  In our simulation the initial velocity field is a continuous function with a superimpose disturbance along the free surface in order to have a faster development of turbulence. Therefore this velocity field can be rewritten as a sum of a continuous function (U_i) and a Gaussian disturbance (U_d). U_i is derived by imposing some constraints to the velocity profile:

  - the function U_i is a linear function of the wall normal direction y and it is uniform through all the other directions. The velocity profile in both water and air is linear with a different angular coefficient:

$$U_i(y) = \begin{cases} U_i^w(y) = Ay + D & 0 \leq y \leq h/2 \\ U_i^a(y) = By + C & h/2 \leq y \leq h \end{cases} \tag{3.5.11}$$

  - the velocity is maximum with a value of $U_i(h) = 1$ m/s at the upper boundary, while it is null at the lower boundary $U_i(0) = 0$ in order to fulfil the no-slip condition at the sea bottom; moreover the two velocity profiles of air and water need to match at the free surface $U_i^w(h/2) = U_i^a(h/2)$;

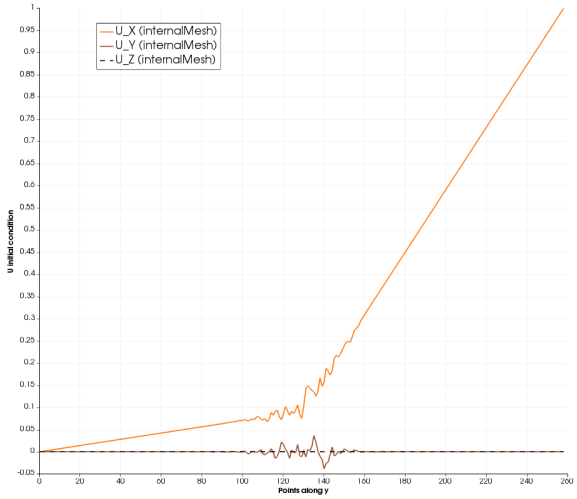


Figure 3.5.5: Initial velocity conditions

– the velocity profile in both water and air is linear with a different angular coefficient. The ratio between these angular coefficients is fixed to be approximately the ratio between the kinematic viscosity of water and air: $A/B \approx \nu_w/\nu_a$;

Applying all these costraints we find the constant A,B,C,D and our initial velocity profile. Then a random Gaussian disturbance along the streamwise and wall normal direction is superimposed:

$$U_d(x,y) = A(x,y)e^{-\beta((x,y)-\mu)^2} \tag{3.5.12}$$

where $A(x,y)$ is a random amplitude function that can have a maximum and minimum value of about the 10% of the maximum velocity value (therefore $-0.1 < A(x,y) < 0.1$); $\beta = 1/(2\sigma^2)$ where we fix the standard deviation to be $\sigma = 0.025$ in order to have the 95% of the disturbances concentrated in a restricted interval around the mean value; μ is the mean value around which the disturbance is applied and it is representing the free surface initial position ($\mu = h/2$). Summing up these 2 functions $U_d(x,y) + U_i(y)$ we obtain the total initial velocity field shown in figure 3.5.5 in all his components. Clearly the spanwise component has been left without any initial profile in order to let the initial conditions to fulfil also the continuity equation 3.4.4.

- **The boundary conditions**: this point is crucial for each simulation since that the solution of the differential equations implemented in the numerical model are strictly connected to their boundary conditions, that is the value that a variable or a quantity related to that variable (as for example its gradient) assumes at the boundary of the domain.

Variable	Patches			
	Top	Bottom	Sides	Inlet/Outlet
U	Symmetry Plane	No Slip	Cyclic	Cyclic
p_{rgh}	Symmetry Plane	Zero Gradient	Cyclic	Cyclic
α	Symmetry Plane	Zero Gradient	Cyclic	Cyclic

Table 3.5.3: Boundary conditions of the simulation

Therefore, translating the physical system into the numerical one with some boundary conditions applied to, is a very delicate task.
All the boundary conditions used are summarized in table 3.5.3. The cyclic conditions on the inlet, outlet and on the sides of the channel allow us to extend the process studied on a virtually bigger domain: instead of having just a single channel, applying the cyclic conditions is like having the same channel repeated backward and forward, on the left and on the right. In fact all the field at the boundary are coupled: if a certain velocity is exiting from the right side, than it will enter from the left exactly as it was flowing out; if a certain pressure field is present at the outlet of the channel, it will be reflected at the inlet and so on. This cyclic condition is different from the symmetry condition applied on the top of the channel: in fact here all the components, instead of being copied, are reflected like in a mirror. This has been imposed in order to cancel out all the wall-normal velocity components and to reproduce the streamwise one, that will be just copied from the one resulting from the pressure driven flow. In this way we are able to simulate the wind boundary layer profile: at the top of the channel the gradient of velocity will be null as the wall normal velocity fluctuations. Finally at the bottom part of the channel a no slip condition is ensured for velocity (Dirichelet explicit condition $U_{i_{bottom}} = 0$), while a zero gradient (Neumann condition) is imposed for pressure and fraction of water.

- **The time controls**: the time step with which the simulation is advancing, how it is evaluated to ensure stability, the numerical scheme used for time integration and the total running time can be very different from on simulation to another. The time step in the simulation was not fixed but it was continuosly adjusted in order to fulfil the requirement on the maximum Courant number evaluated through all the domain in each time step: its value has been oscillating from $\Delta t = 0.007s$ to $\Delta t = 0.014s$. The simulation reached 200 seconds after about 449 hours, that are more or less 19 days.

- **The numerical schemes**: for each operator present inside the nu-

Term	Numerical scheme
First time derivative	Euler Implicit: Implicit and first order accurate in time but bounded for stability.
Divergence	Gauss Linear: the concept is transforming the integral over a control volume of the divergence into an integral over the surface of the cell using the Gauss theorem. Then for each face of the cell the face field can be evaluated using a variety of schemes like the Central differencing, the Upwind differencing or the Blended differencing. These schemes are second order accurate and unbounded for stability.
Gradient	Gauss Linear
Laplacian	Gauss Linear corrected

Table 3.5.4: Numerical schemes

merical model (divergence, gradient, laplacian), we need to choose a proper discretization scheme that ensure us the accuracy of the solution obtained. All the numerical schemes used in this simulation have been reported in table 3.5.4.

Chapter 4

Post-processing and results

In this chapter some results coming from the simulation are reported. The way of importing and post-processing data is worth to be mentioned in this brief introduction: the data coming from the OpenFoam suite, are directly analysed with the open source visualization software ParaView. Through this software we are able to visualize all the flow topology and variables in a three-dimensional view. In order to compute all the statistics regarding the wind boundary layer and the free surface, the programme Matlab has been used. The data used for the Matlab post-processing are imported from ParaView, since that converting OpenFoam data in a suitable format for Matlab is not an easy task.
After importing the data in the post-process software, it is always a good rule to recall what were the goals and the reasons behind our simulation. In this chapter we are going to take into account the first two questions posed in the first page of this work:

1. How the wind transfer and dissipate energy through water waves?

2. What is the role of turbulence in the generation of surface waves by wind?

We will carry out this task by dividing this chapter in four sections, where different flow features are analysed:

- **Flow topology**: in this section all the different scales and structures that compose the wind boundary layer and the free surface are shown and commented, trying to establish a relation between all the different variables and their distribution.

- **Thermalization**: here the evolution of the process is studied, trying to understand when the system reaches its equilibrium and so turbulence can be considered in his fully developed state.

- **Wind boundary layer statistics**: after the thermalization the time will not be considered any more in our study. The process can be considered as statistically stationary and we will compute all the relevant statistics regarding the wind boundary layer.

- **Free surface statistics**: in this last section we are going to analyse and comment all the statistics related to the free surface height.

4.1 Flow topology

The flow topology can be analysed by looking at the distribution of the flow variables along different directions. The variables are chosen depending on what we want to describe: if we want to see how the velocity field is varying thorough all the domain, we will show the velocity fluctuations (u', v', w') over different planes; if we are asking ourselves how pressure is varying, then we need to visualize the pressure fluctuations (p'); if we want to see turbulent structures, then we need to calculate some variable that allow us to localize those structures.

4.1.1 Pressure and velocity fluctuations

We will start studying the flow topology by looking at the pressure fluctuations p', since they play a significant role in wind-wave generation as explained is Chapter 2.

In figure 4.1.1 the pressure fluctuations evolution along xy and xz planes is reported respectively at 5,10,30 and 50 seconds after the wind start blowing over the calm surface. If we look at the plane xz, after 5 seconds the fluctuations are randomly distributed and their amplitude seems to be bigger with respect to the following time steps; as the time goes on they appear to be more organized and less intense. Probably pressure fluctuations are changing their shape because of the wave formation and the so called non separated sheltering effect, as explained in Chapter 2. Looking at the xy plane in figure 4.1.1 we can easily recognize this effect by the presence of an alternate pressure distribution along the free surface due to the different surface curvature: negative and positive pressure spots are indicated respectively by dark red and yellow colours. When the flow becomes fully developed, after 50 seconds, the sheltering effect becomes even more clear as shown in figure 4.1.2.
There is another important feature that is shared by all the figures regarding the pressure distribution along xy or zy planes: long orange and red band of pressure starting from the free surface and going to the outer boundary

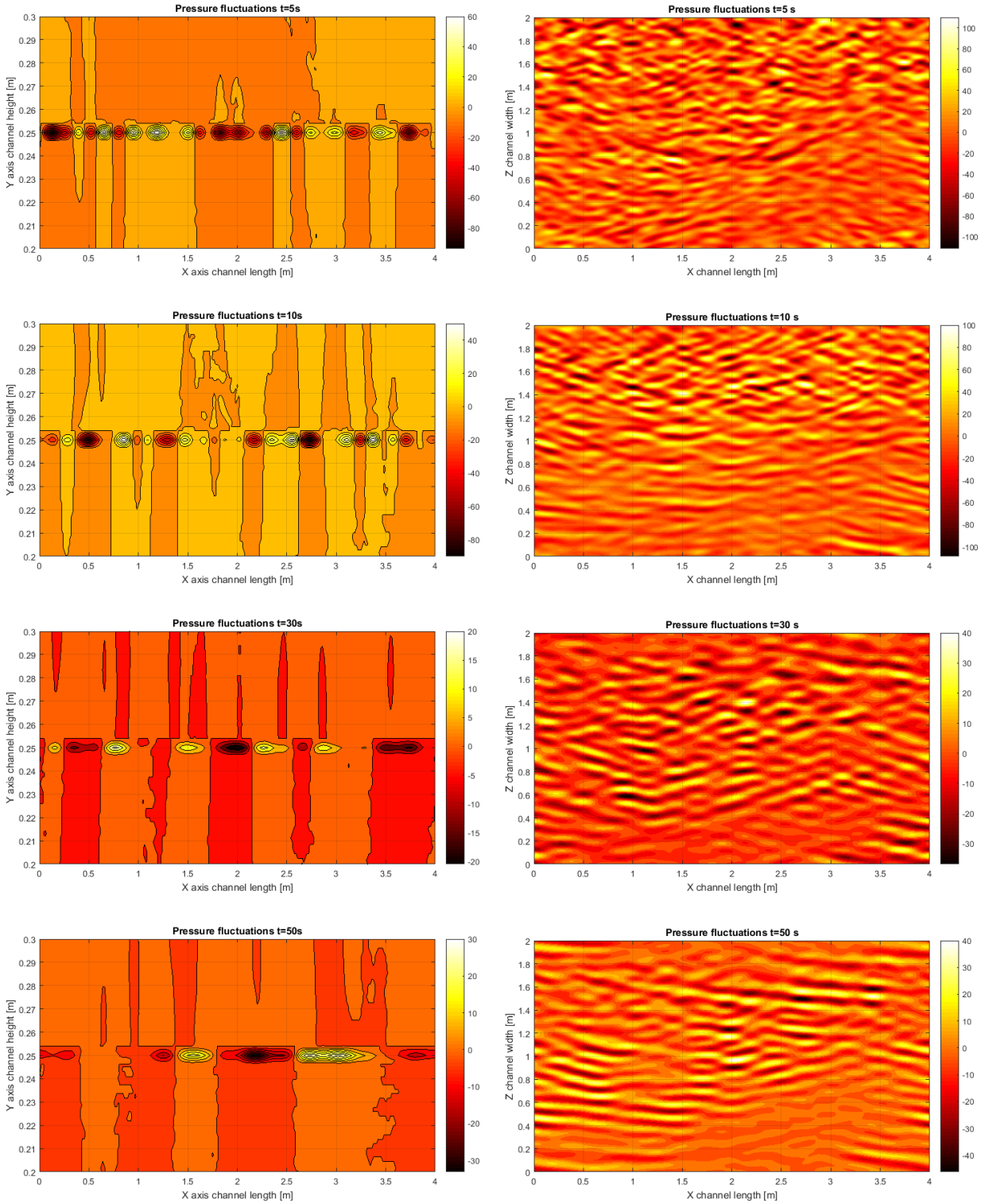


Figure 4.1.1: Slices of pressure fluctuations evolution in time: on the left xy slices at $z = 1.50$ m;on the right xz slices at $y = 0.25$ m.

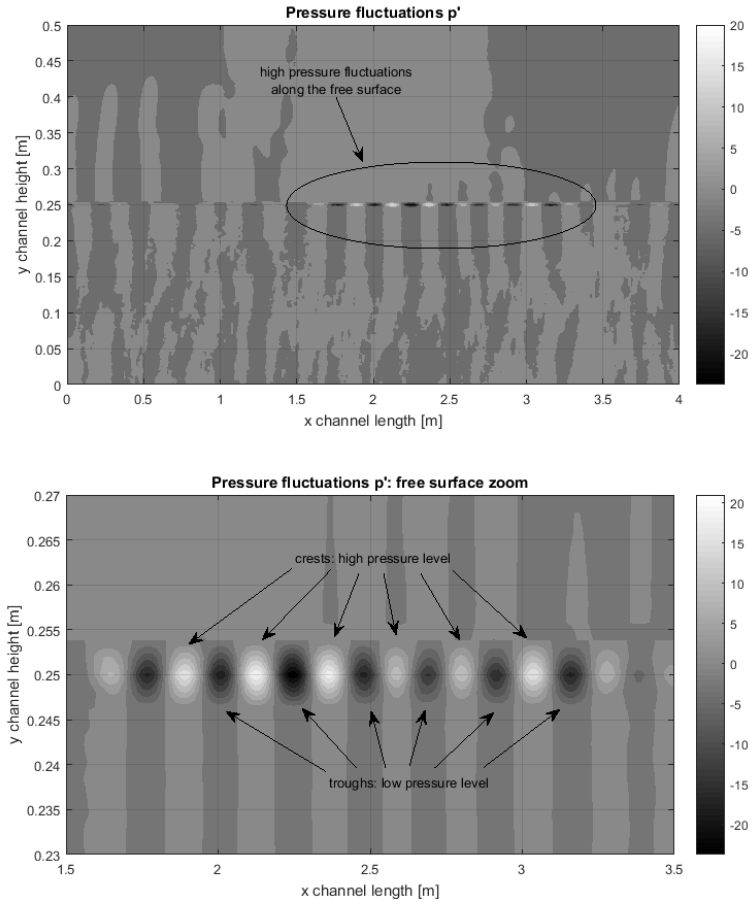


Figure 4.1.2: Fully developed state: pressure fluctuations along the xy plane: on the bottom a zoom on the free surface is shown in order to enlarge the pressure variation.

layer, are shifted with respect to the one that are present inside the water layer. These shifting is due to the wind streamline curvature.

Finally these type of pressure fluctuations are founded to be also along the spanwise direction of the channel as shown in figure 4.1.3.

The velocity fluctuations (u', v', w') in the fully developed state are shown in figure 4.1.4, 4.1.6 and 4.1.5 along different planes.

The structures of the streamwise velocity fluctuations are elongated along the x direction as shown in figure 4.1.4, while they are narrow along the spanwise one. This is not valid for the wall-normal velocity v': it is always elongated along the y direction, like it has to connect the free surface with the outer part of the wind boundary layer. In fact this component is the main responsible for the turbulent transport from the mean flow to the

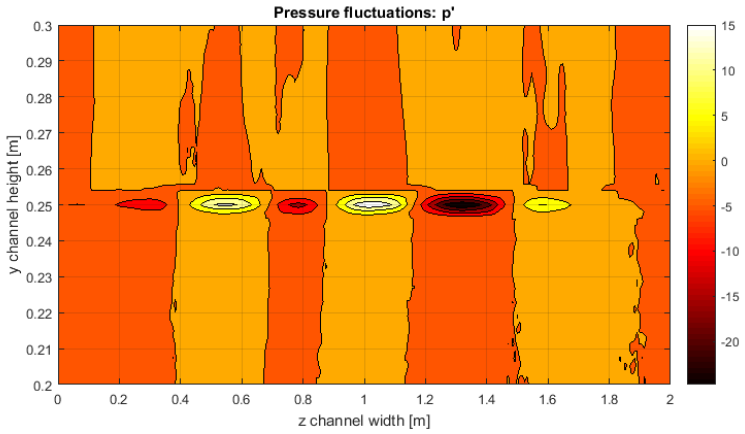


Figure 4.1.3: Fully developed state: pressure fluctuations along the zy plane. A zoom on the free surface zone.

free surface. However the pattern of these fluctuations alternate both along x and z direction: positive and negative structures are connecting the free surface to the upper layer and vice versa. Wind and waves are mutually interacting each other: water is not only absorbing energy from the wind, but it is also giving it back. The wave motion is pushing the streamlines of the wind boundary layer to change direction and, comparing the distribution of v' with p' along yz plane, we can see how positive pressure distribution on the free surface corresponds to positive vertical fluctuations structures. The importance of this component for the wind-wave process will be highlighted later looking at the skewness and the flatness of the boundary layer. On the other hand u' is mainly varying along the wall normal direction: over the free surface there are small and concentrated layers of negative fluctuations (dark red), while in the outer part the positive fluctuations dominate.
In figure 4.1.6 we can see the behaviour of the fluctuations along the free surface and especially the wave pattern looking at the streamwise component: u' will be positive along the crests and negative along the troughs.

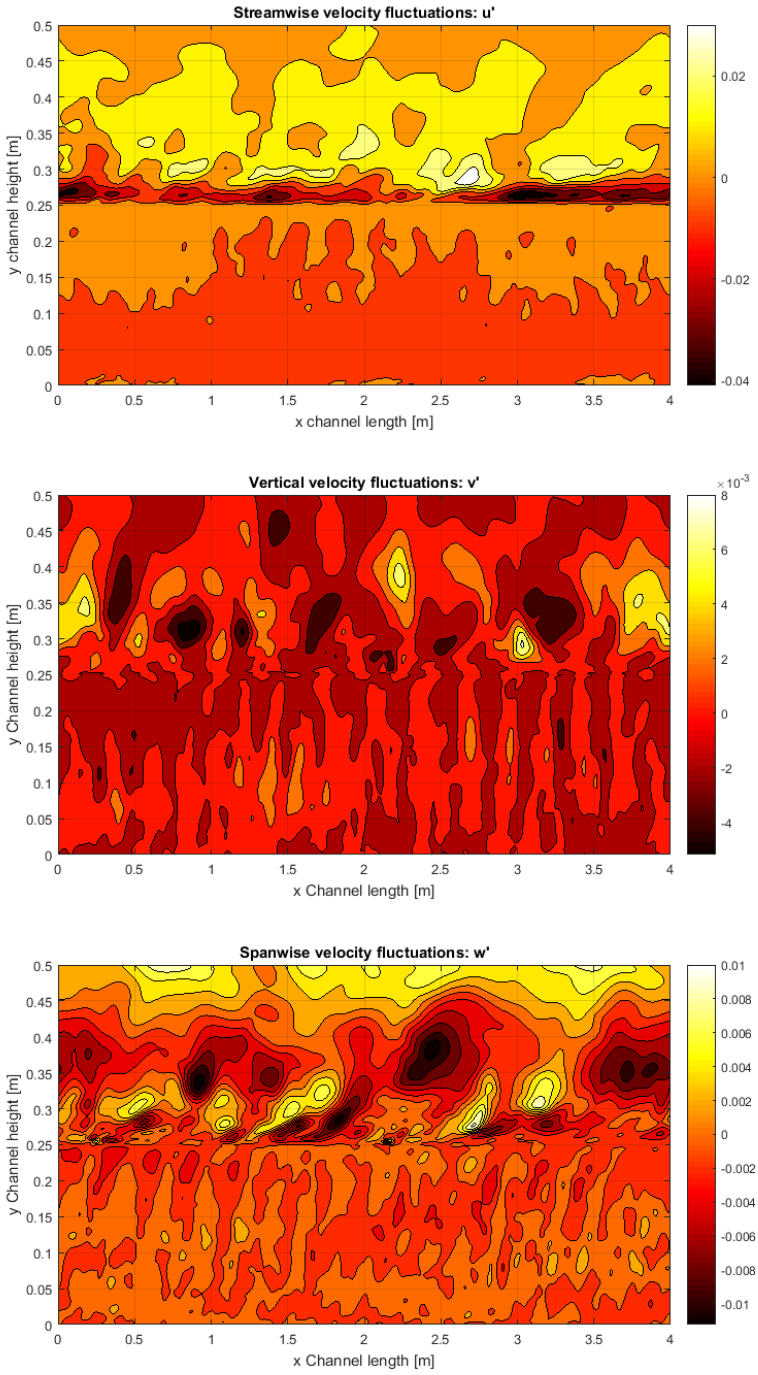


Figure 4.1.4: Slices of velocity fluctuations:xy slices at $z = 1.50$ m. From the top to the bottom: streamwise, wall normal and spanwise fluctuations.

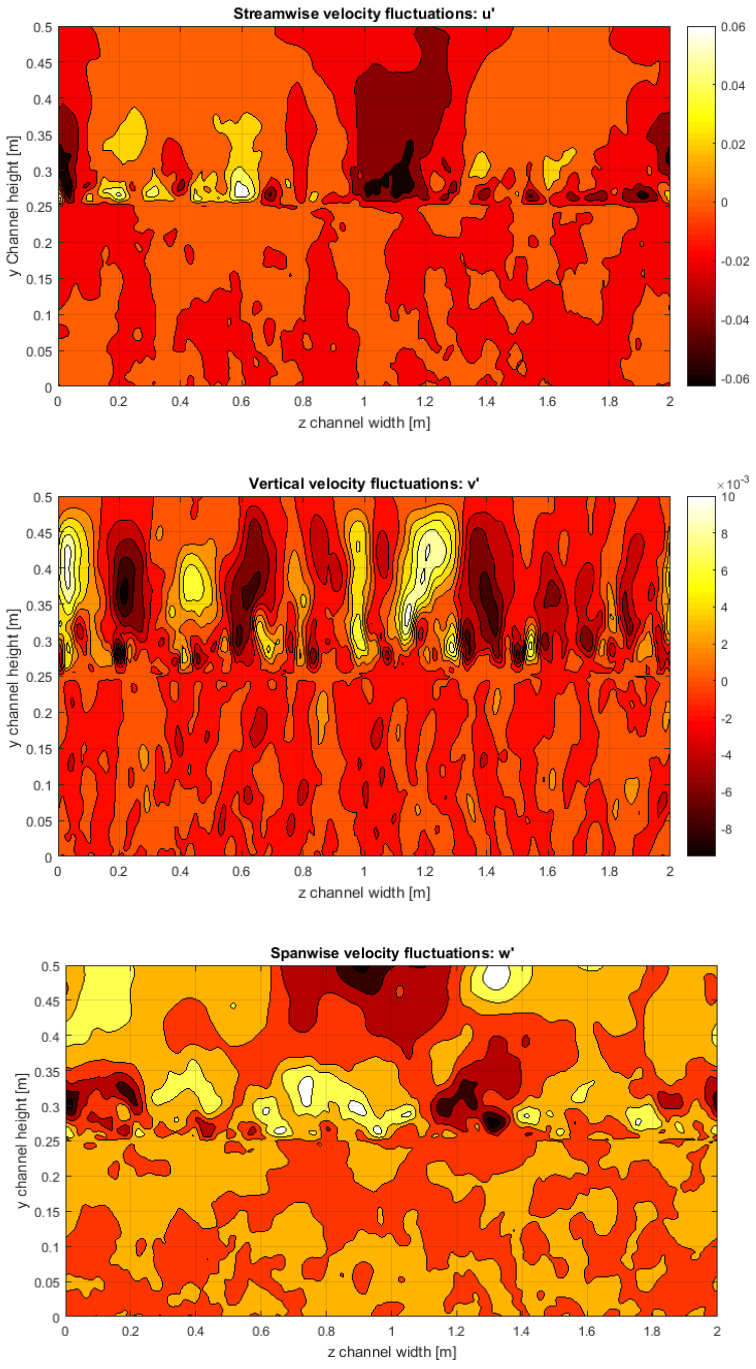


Figure 4.1.5: Slices of velocity fluctuations: yz slices at $x = 3$ m. From the top to the bottom: streamwise, wall normal and spanwise fluctuations.

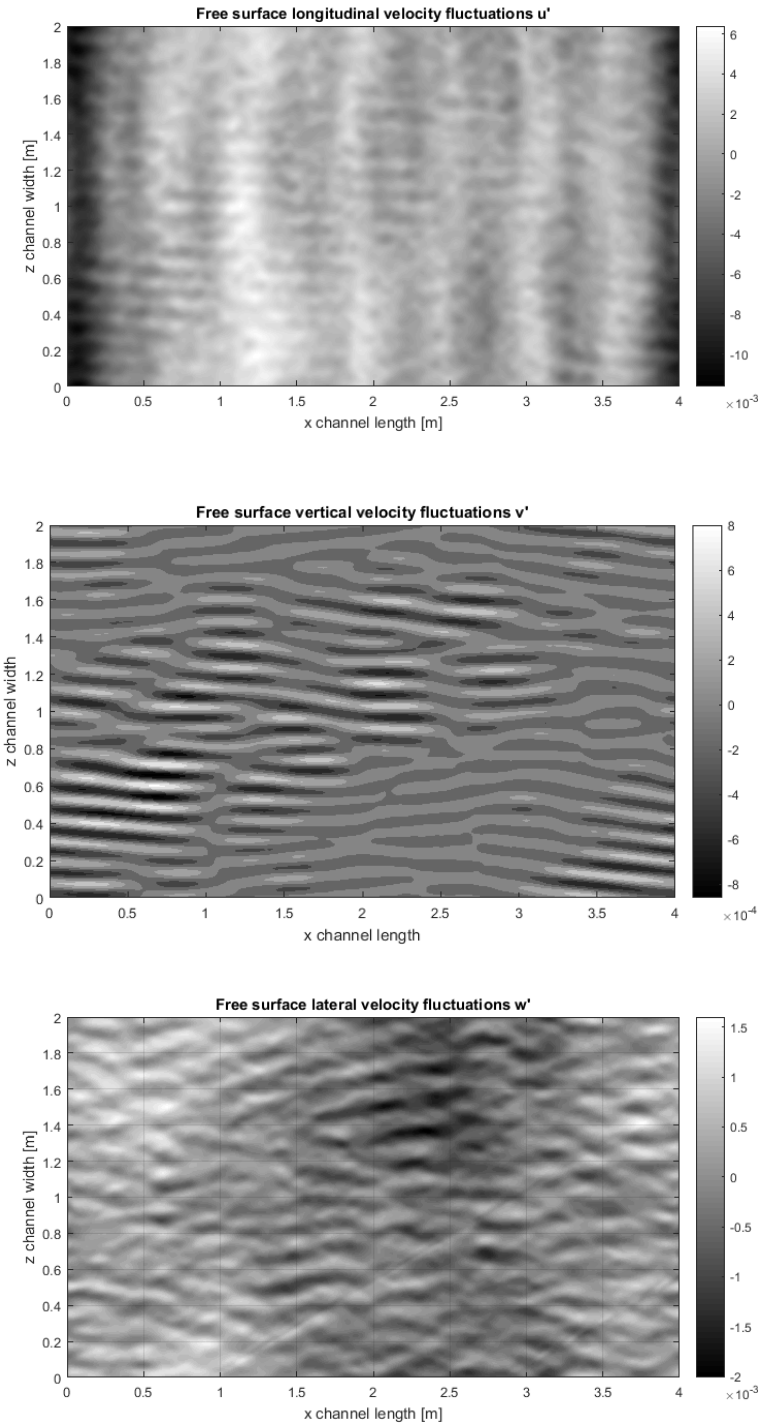


Figure 4.1.6: Slices of velocity fluctuations: on the top xz slices at $y = 0.50$ m of streamwise fluctuation;on the bottom the same slices for wall normal (left) and spanwise (right) fluctuations.

4.1.2 Turbulent structures

There are many different ways to localize and visualize the vortical (turbulent) structures of the flow. For example we could evaluate the enstrophy of the flow, that is measuring the rotational intensity of the flow, and look for high value of this variable in order to detect vortex structures. Another way could be to create isosurfaces of some small and negative value of the pressure fluctuation p', since that vortical motions are usually located in low pressure regions. Anyway in this section we are going to use another widely used method in CFD that is called the "λ_2 criterion". According to this method, the vortex structures are individuated by looking at isosurfaces of a certain negative value of this scalar variable called λ_2. This parameter comes from the decomposition of the velocity gradient tensor into 2 parts: a symmetric part S_{ij}, also called the strain-rate tensor, and an antisymmetric part Ω_{ij}, called the vorticity tensor.

$$\frac{\partial u_i}{\partial x_j} = S_{ij} + \Omega_{ij} \tag{4.1.1}$$

λ_2 is exactly the second eigenvalue of the tensor $[S_{ij}^2 + \Omega_{ij}^2]$. In fact the solution of the eigenvalue problem for that tensor in all the points of the velocity field, will give as result three eigenvalues for each point, such that $\lambda_1 < \lambda_2 < \lambda_3$. It has been found that a point in the velocity field in which at least 2 eigenvalues are negative, can be then the core of a vortex. Therefore in order to fulfil that requirement we need to search for a negative value of λ_2. It is clear that the lower is the value (more negative), the lower will be the number of vortex structures visible in our flow. A usual reference number for starting to research vortex structures is $\lambda_2 \approx -0.2$. Finally we can conclude saying that a vortex tube, or a vortex structure, is a connected region of points where λ_2 is negative.
This analysis has been made for our simulation and a three dimensional view of the channel of water is shown in figure 4.1.7: at the very first instants the vortex structures start to develop from the free surface going into the wind, while under the free surface, in the deep water, no structures are present. As the time goes on, the number of structures increases and their shape becomes elongated along the x-direction. Also the structures on the free surface, coloured by light blue, are streaky and narrow. After the flow becomes fully developed and the waves are formed, the structures on the free surface seems to disappear, but they are just hidding behind the turbulent waves. In fact these structures are more regular, short and aligned with the wave pattern as shown in figure 4.1.8, where the free surface has been isolated from the rest part of the air boundary layer.

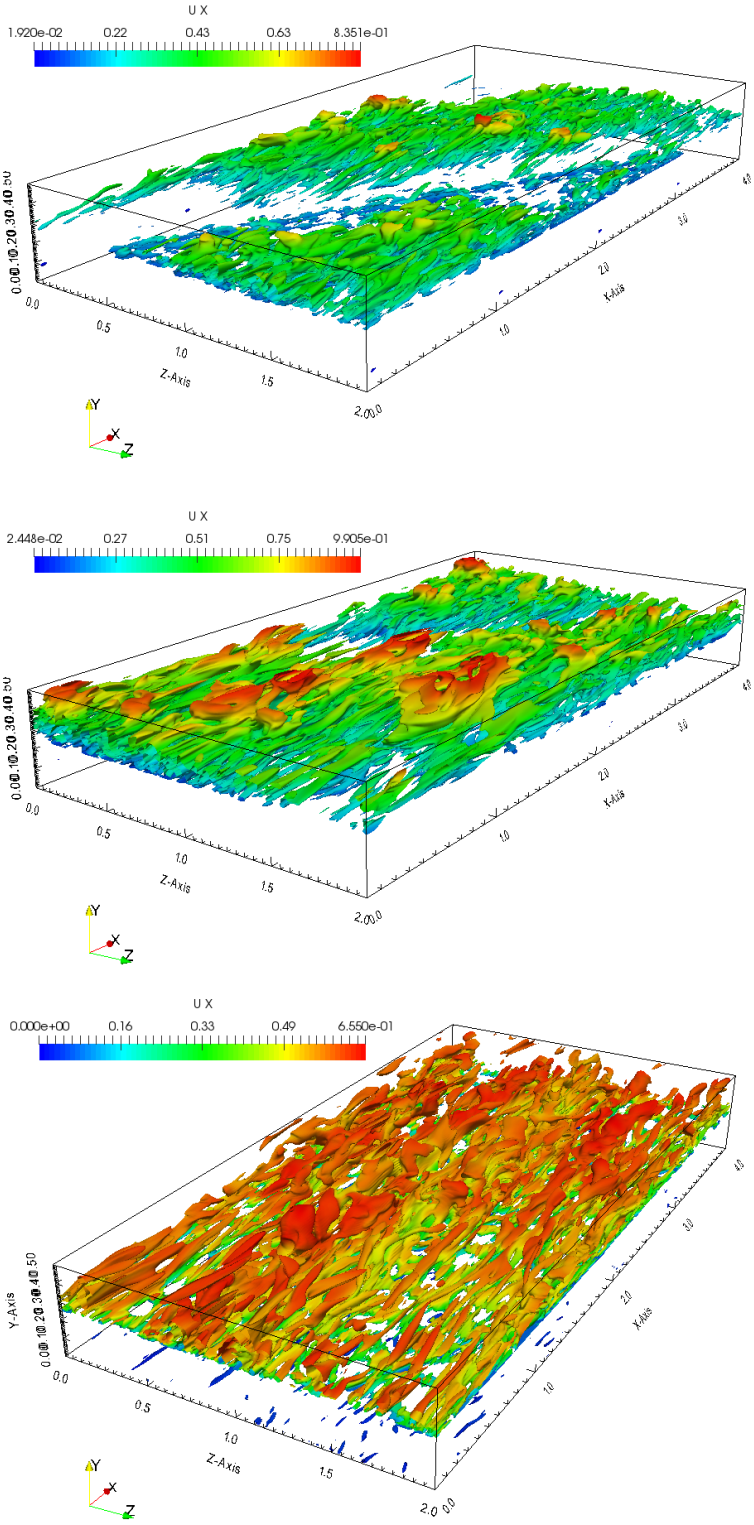


Figure 4.1.7: 3D visualization of vortex structures: isosurface (coloured by the streamwise velocity U_x) of $\lambda_2 = -0.2$ respectively at times 5,10 and 175 seconds (from the top to the bottom)

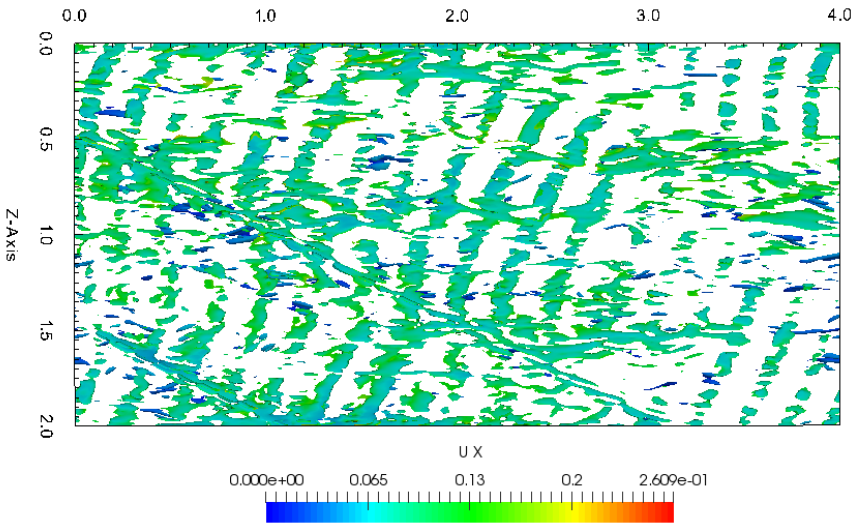


Figure 4.1.8: 3D visualization of vortex structures: isosurface (coloured by the streamwise velocity U_x) of $\lambda_2 = -0.2$ at time 175 seconds along the free surface. The dark blue structures behind the free surface are the turbulent structures that starts developing inside the water layer.

4.1.3 Free surface pattern

The free surface pattern evolution is shown in figure 4.1.9, where we plot the free surface elevation (in millimetres) and the pressure distribution. The free surface wave elevation has been calculated firstly by making an isosurface of the fraction of water α, and secondly subtracting each value of the isosurface by an estimated mean water depth (it has been considered as the half channel height). Instead, the value chosen for α is 0.5, that is in the middle of the transition region between air and water. Starting from 10 seconds the free surface height is about 1/5 of a millimiter up to 150 seconds when it reaches almost 1/2 mm. Looking at the fully developed state we could also estimate the typical wavelength of these waves that appears to be in the range between 5 and 10 cm, that are classified as capillarity-gravity waves.

On the right part of figure 4.1.9 we can see how the pressure distribution is following the wave pattern. Actually there is a big difference in the arrangement of waves and pressure before and after 50 seconds when the flow become fully developed: before this threshold regular waves are not present, but only small ripples forming a grid structure are observed; after 50 seconds this ripples rearrange themselves in a wavy and regular pattern and the same happen for pressure. We could call the first phase as the "raining phase", where the turbulent wind is acting as the rain falling down

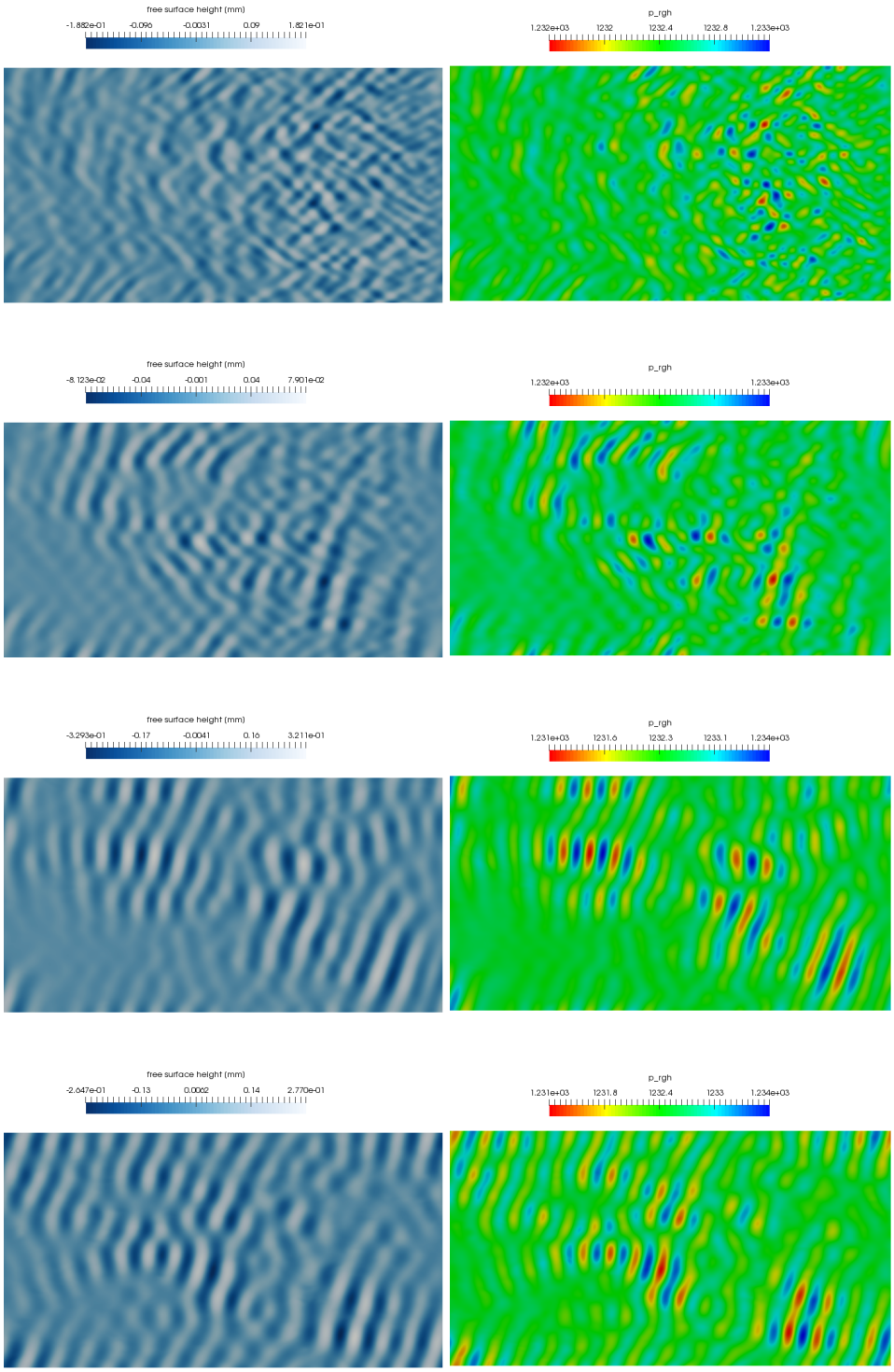


Figure 4.1.9: Wave height and pressure distribution along the free surface (plane xz): time instants 10, 50, 100, 150 seconds from the top to the bottom.

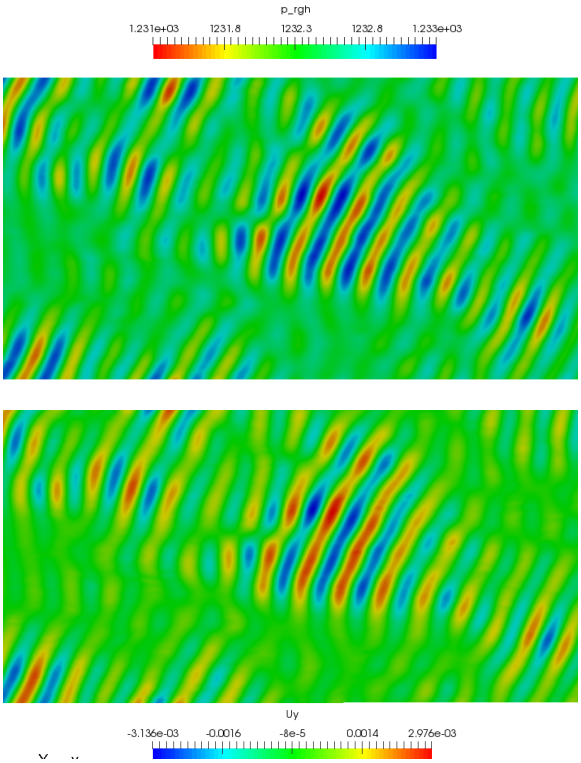


Figure 4.1.10: Comparison between vertical velocity U_y and pressure distribution p_{rgh} along th free surface

on the free surface, and the second phase as the "wavy phase" in the fully developed state of the turbulent wind.

Comparing the pressure distribution and the vertical velocity U_y in figure 4.1.10 we can easily see how they are strongly coupled.

Finally in figure 4.1.11 is shown the streamlines path of tracers moving along the free surface. The curvature of these streamlines is enhanced colouring them either by the pressure distribution or by the wall normal velocity. In this figure the shifting of pressure and vertical velocity is also evident.

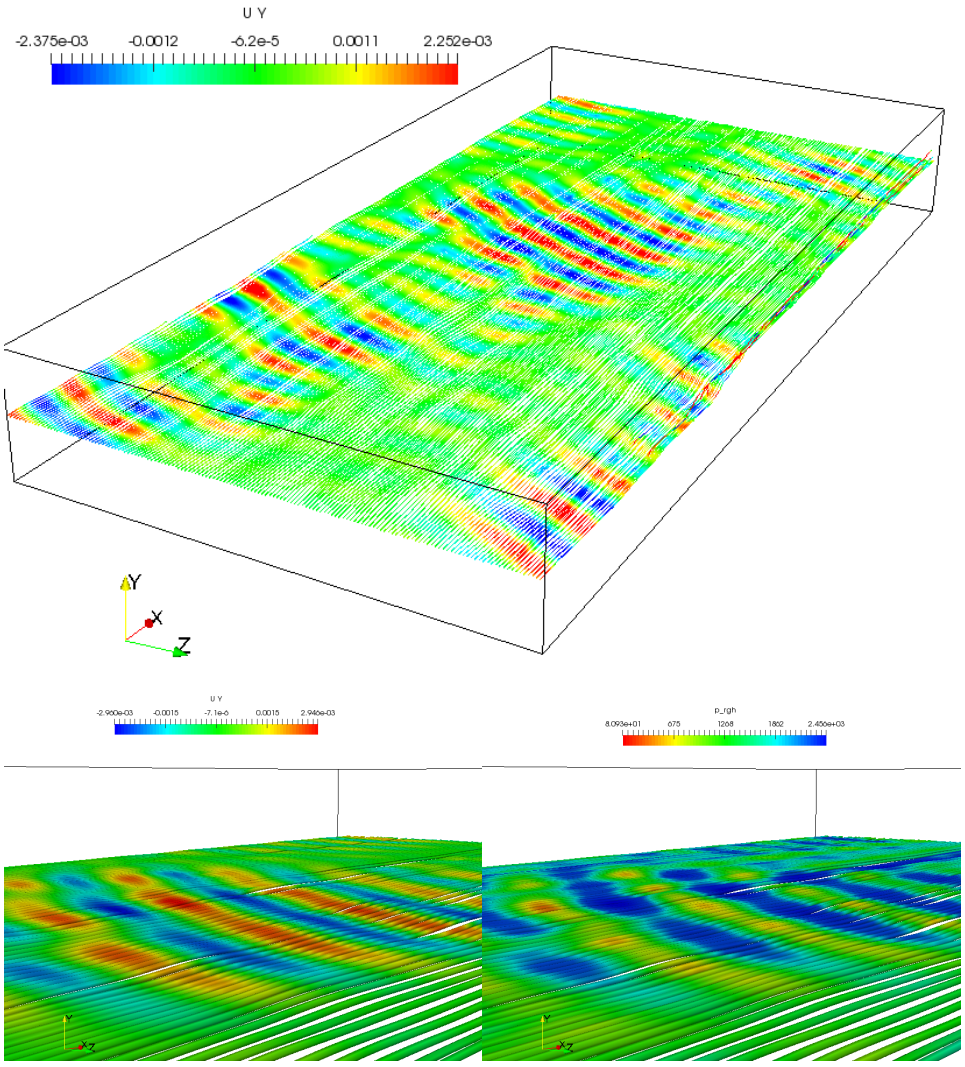


Figure 4.1.11: On the top: streamlines path on the free surface coloured by the vertical velocity. On the bottom left: zoom on the streamlines path coloured by U_y. On the bottom right: zoom on the streamlines path coloured by p_{rgh}.

4.2 Thermalization

The meaning of thermalization has to be found in figure 4.2.1: as the time increase the spatial averaged velocity profile collapse in one narrow band of profiles. This means that the mean flow after 60 seconds has reached his equilibrium and it will be no more dependent on time. The flow is said to be statistically stationary (that is completely different from stationary) since that all his statistics (like the mean or the variance) are steady in time. This state of the flow is called the fully developed state.

In order to deeply understand this concept let's consider for a moment the entire velocity field $U_i(x, y, z, t)$ in our water channel and its three components: the streamwise $U_x(x, y, z, t)$, the wall normal $U_y(x, y, z, t)$ and the spanwise $U_z(x, y, z, t)$. Among all the points of the water channel, we choose a certain point $P_1(x_1, y_1, z_1)$ and we start to measure the velocity signal in all its three components over all the time instants: the streamwise velocity $U_x(x_1, y_1, z_1, t)$ will oscillates around a mean value different from 0 and varying in time up to 60 seconds when the mean starts to stabilize, while the fluctuations are still there (that is why we talk about fully turbulent flow); the spanwise and wall normal components ($U_z(x1, y1, z1, t)$ and $U_y(x1, y1, z1, t)$) will oscillate around a null mean value, since that the mean water current and the mean wind flow are moving only along the streamwise component (due to the pressure gradient that we have imposed along the x-direction). Therefore, from this measurement, we could say that after 60 seconds, the flow in the point P_1 is not steady, but it is varying around a steady mean value. That is the reason why we can refer to this flow as statistically stationary.
However the figure 4.2.1, like all the following figures regarding the boundary layer statistics, are not representing the mean flow evolution in one point of the domain, since that this mean value is not homogeneous through all the three domain directions. In fact if we let varying the coordinate of the point $P_1(x_1, y_1, z_1)$ we will see different behaviour of the velocity field, depending on which direction we are moving: if we move the point along the x or z direction ($P_2(x_2, y_1, z_1)$ or $P_3(x_1, y_1, z_3)$), no difference will be found between the velocity field of each point P_1, P_2 and P_3; on the other hand if we move the point along the wall normal direction ($P_4(x_1, y_4, z_1)$) we will see strong differences between the velocity field of points P_1 and P_4. This is the reason why our flow can be considered as homogeneous along the streamwise and spanwise direction, but not along the wall normal direction. Obviously, the homogeneity of the flow along those directions is due to the cyclic boundary conditions that we have imposed there; along the wall normal direction, due to the no slip condition at the sea bottom, the presence of the interface, the symmetry condition at the top, and the pressure gradient along the streamwise direction, the flow cannot be considered homogeneous and it will show strong gradients of the mean velocity profile.

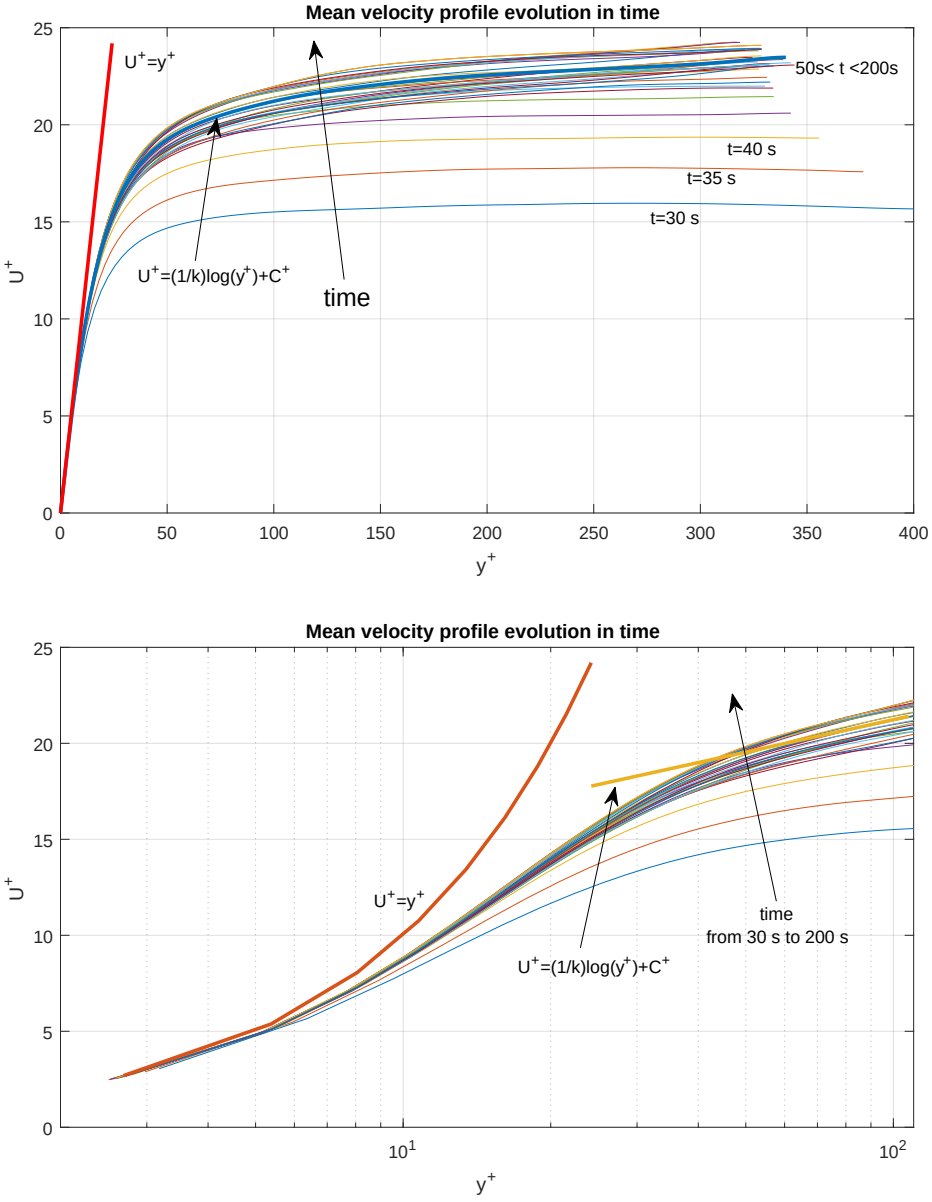


Figure 4.2.1: Mean velocity field evolution in time: $U^+ = U/u_\tau$ and $y^+ = yu_\tau/\nu$. On the top: linear scale plot. On the bottom:logarithmic scale plot.

Therefore, the figure 4.2.1 and all the following figure regarding the wind boundary layer statistics, is representing the velocity field spatially averaged along the homogeneous directions as expressed by the equation (4.2.1) (in the continuous form), and (4.2.2) (in the discrete form), where L_x, L_z are the length and width of the channel, while N_x, N_z are the number of grid elements of the domain.

$$U_i(y,t) = \frac{1}{L_x L_z} \int_0^{L_x} \int_0^{L_z} U_i(x,y,z,t) dx dz \tag{4.2.1}$$

$$U_k(y,t) = \frac{1}{N_x N_z} \sum_{i=0}^{N_x} \sum_{j=0}^{N_y} U_k(x_i, y, z_j, t) \tag{4.2.2}$$

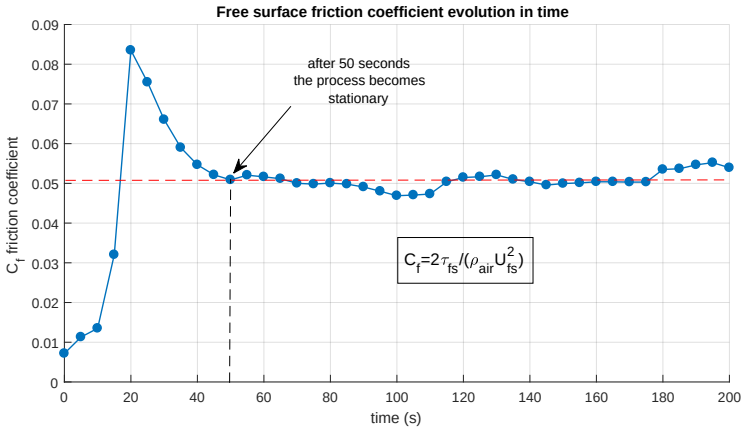
Once the velocity field has been spatially averaged by applying formula (4.2.2), we can easily evaluate the time evolution of some parameter that is better indicating us the stationarity of the process. Usually these parameters are evaluated at the interface as shown in figure 4.2.2.

Both the friction coefficient (see figure 4.2.2a), the friction Re number (figure 4.2.2b) and the viscous length (figure 4.2.2c) illustrate how the process becomes stationary starting from 60 seconds. After this value the mean Re_τ is about 330.

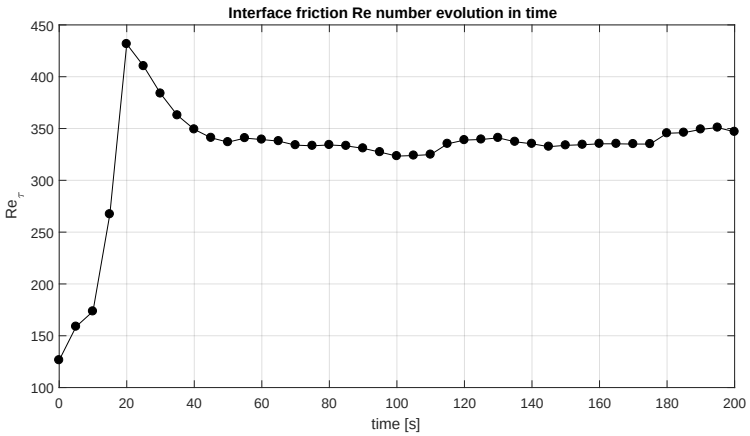
In the Veron and Melville experiment [13], the free surface velocity evolution was studied in order to understand the different phases of the wind-wave generation process: at the first stage the free surface accelerates uniformly up to reach a maximum velocity value that in our case is around 20 cm/s; the second stage is a transition region ($20s < t < 60s$) where the interface starts to decelerate, transferring the energy received from the wind into the water and starting the wavy motions; the last stage is the fully developed state where the surface velocity decreases up to reaching a stable value around 15 cm/s and the wind and wave energy transfer appears to be in equilibrium. The data coming from the experiment cited previously, and the data obtained from our simulation are compared in figure 4.2.3.

These data seems to match exactly in the time window shown (up to 120 seconds): however, since that the wind velocity of the experiment is three times larger than the one used in our simulation, the final free surface velocity in our case will be much smaller and after 120 seconds it will starts to decrease.

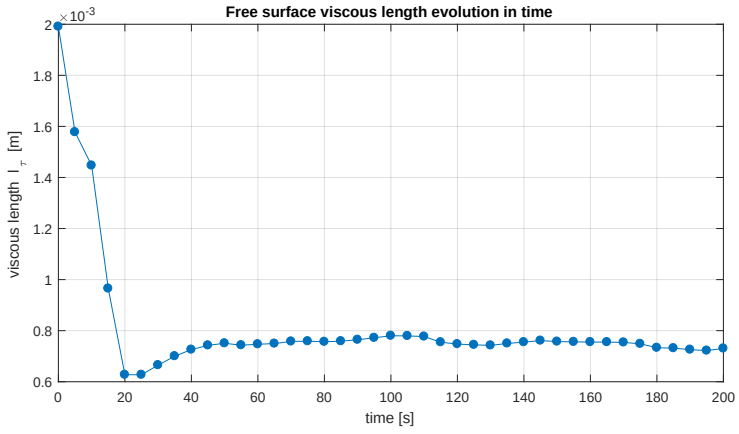
From figure 4.2.3 we can clearly conclude that the free surface velocity is a very interesting parameter in order to visualize the stages of the wind-wave generation process: the interface is that part of the domain connecting wind and water and is indicating us how much energy is transferred from one fluid to the other and vice versa. We could state that if U_{fs} is statistically stationary, then both air and water are exchanging the same amount of energy. Instead if the mean free surface velocity is still decreasing in time, as in our simulation, probably water is not in his fully developed turbulent



(a) *Friction coefficient evolution in time: τ_{fs} is indicating the interface shear stress.*



(b) *Friction interface Re number Re_τ evolution in time*



(c) *Viscous length l_τ evolution in time*

Figure 4.2.2: Parameters used to study the thermalization of the process

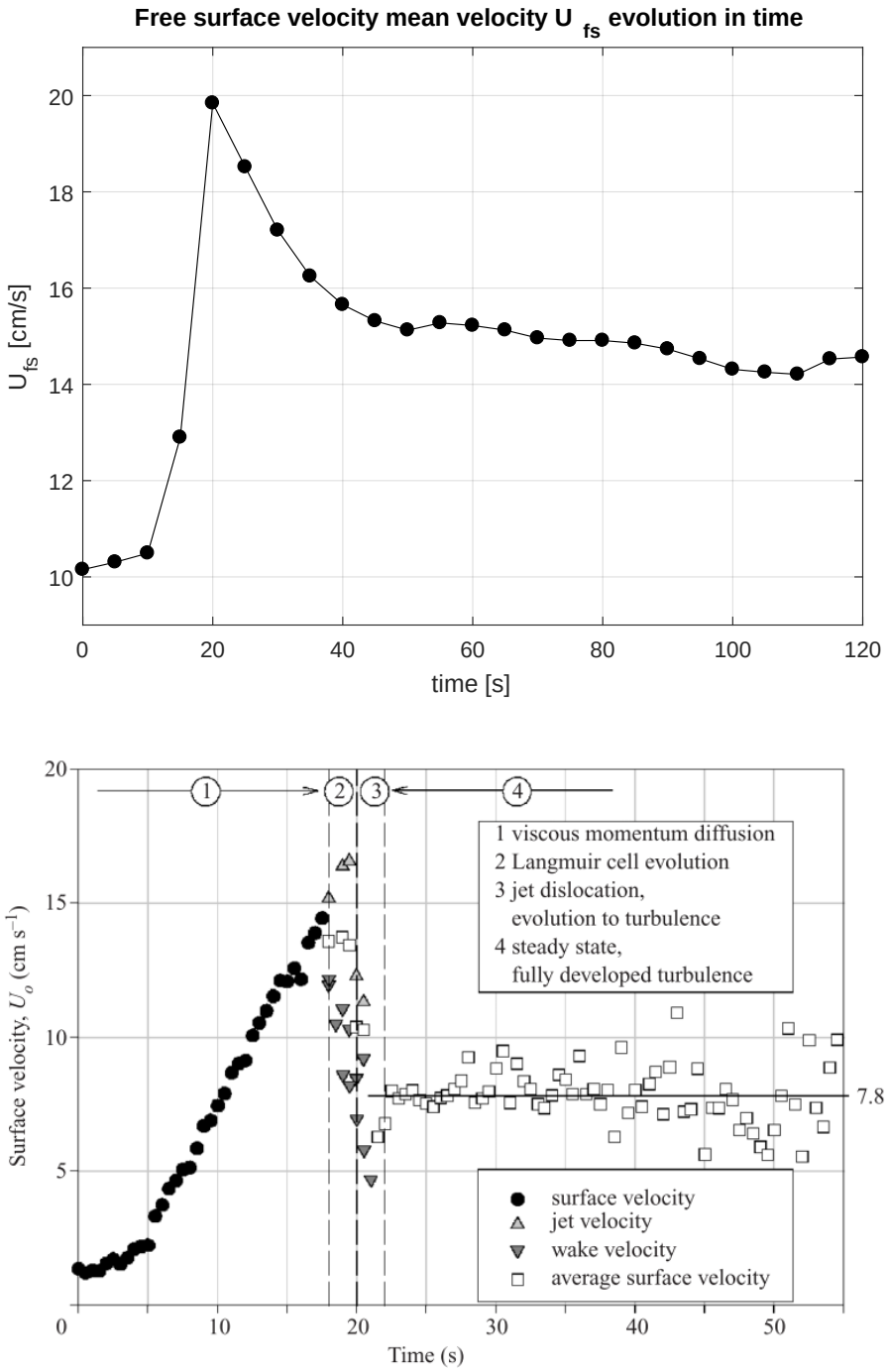


Figure 4.2.3: Comparison between experimental data coming from our simulation (left) and from Veron and Meliville [13] (right). In the experiment the wind speed was of 5 ms^{-1} and the fetch of 10.72 m

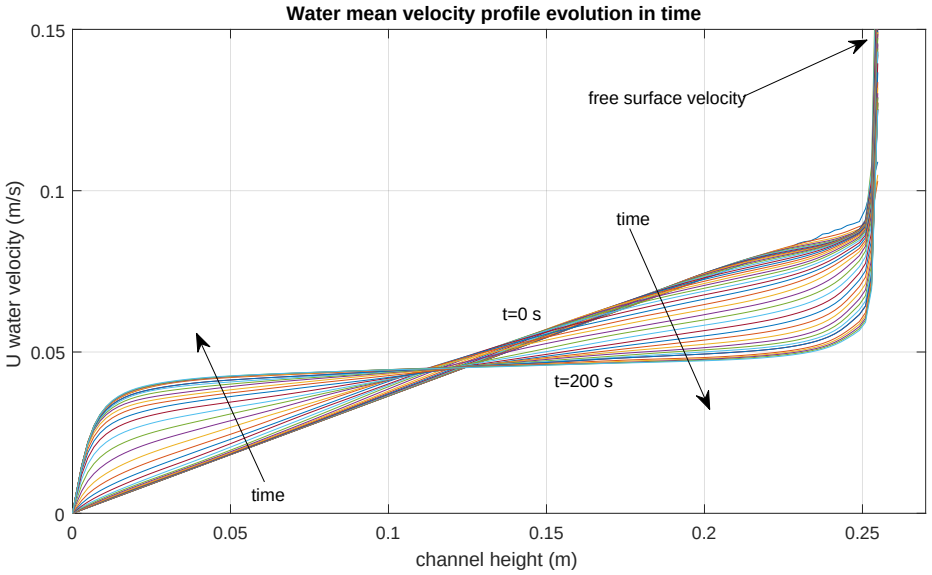


Figure 4.2.4: Mean velocity profile evolution in the water layer

state yet, instead is using the energy provided by wind in order to increase his turbulent state. We could say that waves have not reached their final state yet. In figure 4.2.4 is illustrated the ongoing evolution of the mean water velocity.

Since now we have addressed the stationarity of the flow, we can introduce the concept of time averaging: all the statistics that we will build in the following sections are evaluated starting from the time average of the velocity field. The time window chosen for this calculation is going from $t_0 = 100$ seconds to $t_f = 200\ s$, for a total number of samples equal to $N = 20$. Therefore in the following section the velocity field will be always considered as the time and spatially averaged velocity field. Since that our process in this time window and in the wind boundary layer is statistically stationary, the time average coincide with the ensemble one and we can write:

$$U_i(x,y,z) =< U_i(x,y,z) >= \frac{1}{N}\sum_{n=1}^{N} U_i(x,y,z,t_n) \tag{4.2.3}$$

In this case study the time evolution of our process, will be exactly as repeating it 20 times starting from the same initial conditions.

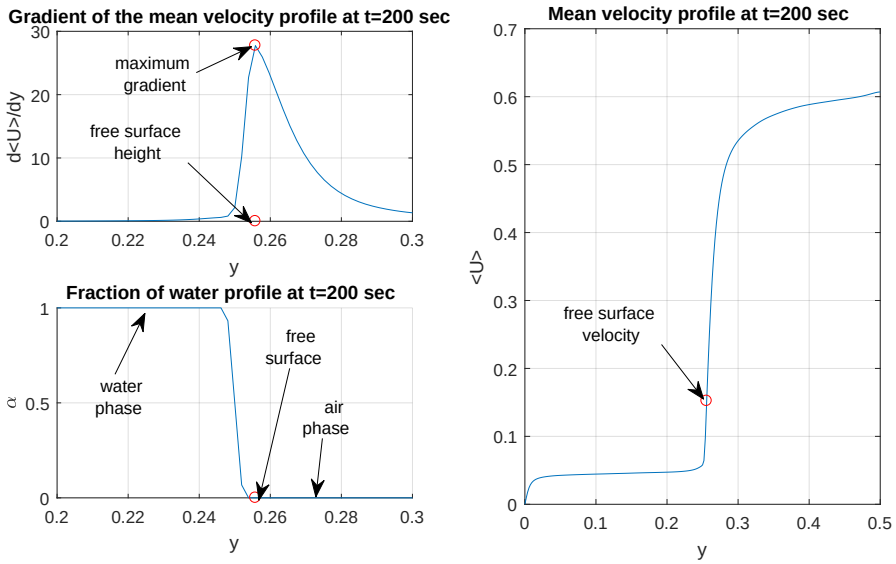


Figure 4.3.1: Comparison between plot of mean velocity profile, volume of fraction distribution, and mean velocity gradient at time 200 seconds, in order to find the correct location of the free surface.

4.3 Wind boundary layer statistics

Taking into account all the previous consideration about spatial, ensemble and time average, we can now show all the statistical quantity that help us in analysing turbulence. Before doing so, we need to find an appropriate scaling for our velocity and wall-normal coordinate, that in our case, as explained in section 3.5, are the interface friction velocity and the viscous length. Obviously in order to calculate the friction velocity, and consequently the viscous length, we need to localize the free surface. This is not an easy task: while studying wall turbulence, it is simple to know exactly where the wall is located since that its wall normal position is constant in time, either for a fixed wall or a moving one; instead for a wavy moving surface this operation becomes more complicate and we need to define a method to establish the exact location of the interface along the wall normal direction. The method used in this work is based on the velocity gradient profile as shown in figure 4.3.1: the free surface will be located in correspondence of the maximum gradient of mean velocity. This criterion is reasonable if we follow the analogy between wall turbulence and wavy wall turbulence.

Following this criterion we can scale our mean velocity and wall normal coordinate in order to get the usual laws of wall turbulence, as explained in the previous chapter.

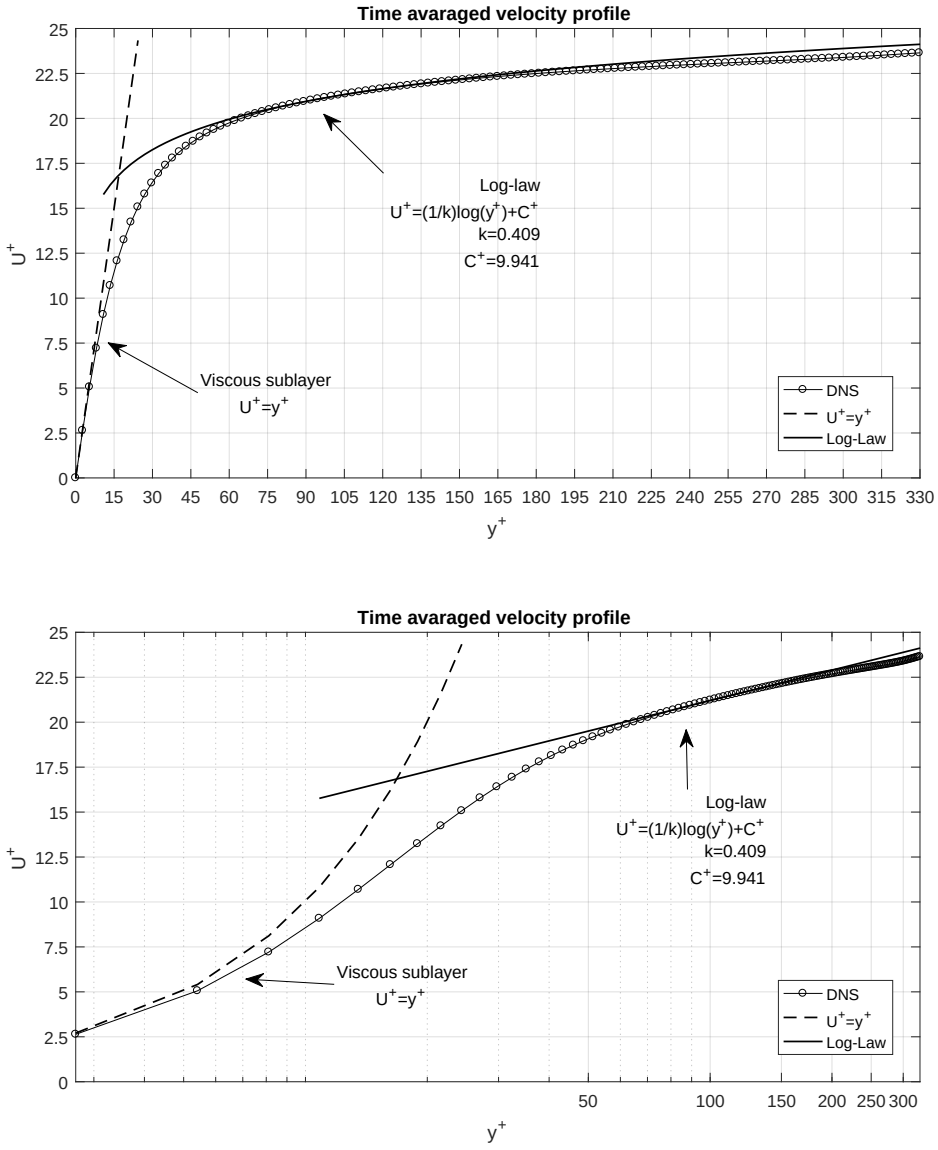


Figure 4.3.2: Boundary layer mean velocity field: on the top the linear scale plot and on the bottom the logarithmic scale plot.

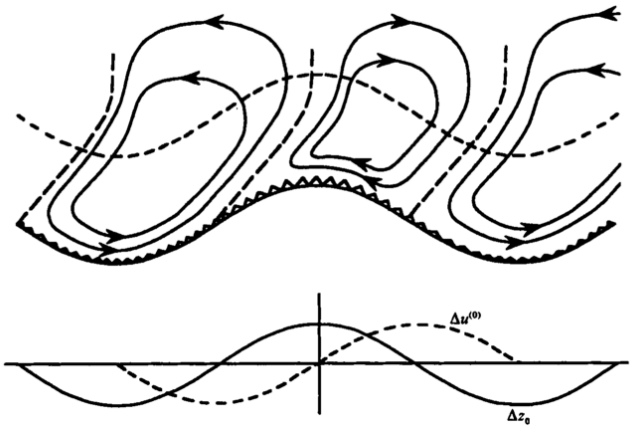


Figure 4.3.3: Effect of ripples generated by turbulent wind along a dominant wave.

In figure 4.3.2 it is shown the mean velocity profile both in linear and logarithmic scale. The inner law is respected for the usual range of $y^+ < 5$, while interpolating the data for $y^+ > 70$ we can find the log-law constants k and C^+. The von Karman constant is the usual one, while the second one is a bit different from the usual values for wall turbulence, since that it depends on the wall roughness. In the simulation this constant appears to be approximately $C^+ \approx 10$. Usually in wall turbulence a value of $C^+ \approx 8$ is defined for a fully rough wall, therefore it is reasonable that for a wavy wall, where ripples are increasing the roughness, this value is bigger. The adimensional sea surface roughness y_0^+, following equation (3.5.9), will be then:

$$y_0^+ = e^{-kC^+} \approx 0.0171 \text{ where } C^+ \approx 9.941 \text{ and } k \approx 0.409 \tag{4.3.1}$$

We recall that this adimensional parameter can also be thought as a roughness Reynolds number $Re_r = y_0^+ = y_0 u_\tau / \nu$. Oceanographers usually consider this parameter as a measure of the influence of ripples along the dominant wave as shown in figure 4.3.3: if $Re_r << 1$, as in our case, the ripples formed along the wave have a negligible effect on the mean air flow in the outer layer, with respect to the one due to the dominant wave (like the non separated sheltering or the miles mechanism); on the other hand if $Re_r > 1$ than the variation of the free surface roughness along the dominant wave, shown by the graphs on the bottom of figure 4.3.3, can cause a further contribution to the asymmetric pressure distribution of the mean flow, that is contributing to the wave growth process. Therefore in the first case, oceanographers will call the water surface aerodynamically smooth, even if a dominant wave is present.

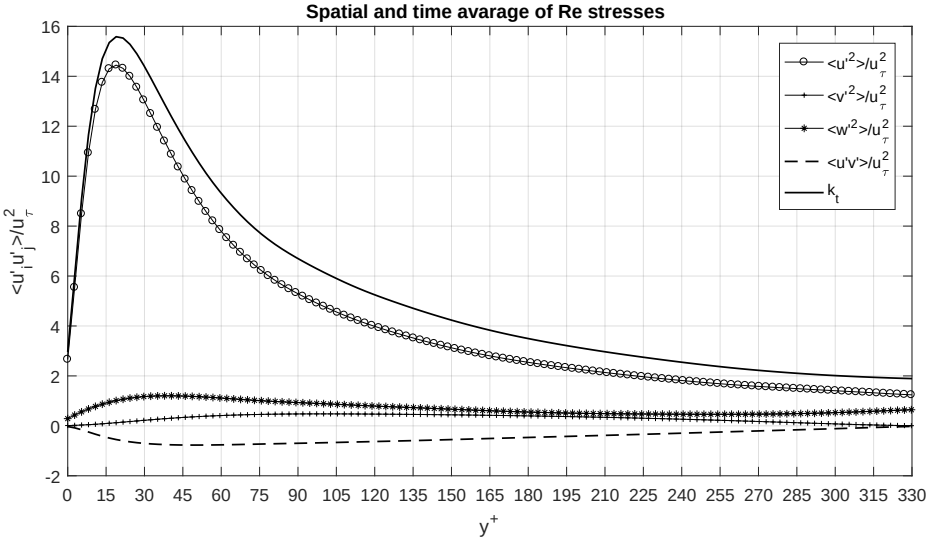


Figure 4.3.4: Boundary layer fluctuating field

However in our simulation, instead of having small ripples over a large dominant wave, we will have even smaller ripples growing over large dominant ripples, and this will be clarified in the next section through the free surface statistics. Therefore in our case we cannot talk about an aerodynamically smooth interface even if the Re_r is much smaller than one, since that this concept is relative to the dominant wave height. If we concentrate only our analysis in the proximity of the wall, in the area that is usually called micro-turbulence, we cannot neglect the effect of those ripples. Anyway as we said in the wave dimensional analysis in the first chapter, the sea surface roughness will be always dependent on two parameters: the wave age and the friction Reynolds number. The dominant wave height is taken into account inside the first parameter, since that we can relate it with the phase speed c_p (that corresponds to the phase velocity of the dominant wave).
Coming back to the effect of the sea surface roughness on the mean velocity profile, we can conclude that the ripples formed on the water surface are affecting the universal law of the wall and can be considered as an increase roughness.
Regarding the second order statistics, that is the variance of the velocity field (scaled by the squared of the friction velocity) representing the fluctuating field, the results are shown in figure 4.3.4.

The normal Reynolds stresses, $< u'^2 >, < v'^2 >, < w'^2 >$ can be considered as the net flux of momentum of the streamwise, wall normal and spanwise velocity components, transported by the correspondent velocity fluctuation u', v', w'. As we expected, close to the wall, the major compo-

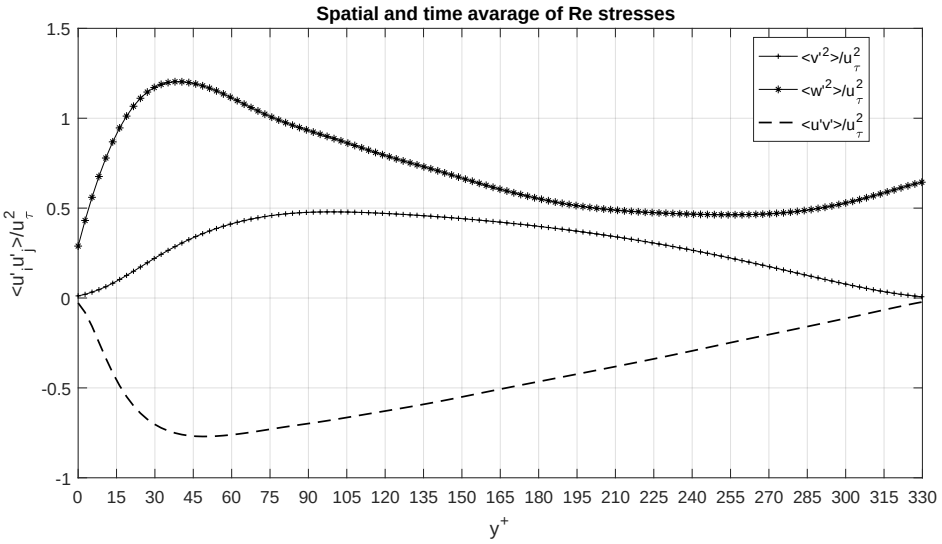


Figure 4.3.5: A zoom on the Re stresses

nent is $< u'^2 >$ that is the one along the main flow direction, while the wall normal component will be damped by the presence of the wall. The maximum flux of momentum due to the streamwise fluctuations is reached approximately at $y^+ \approx 20$, that is corresponding to the buffer region of the mean velocity profile (between the inner region and the logarithmic one). In 4.3.5, is shown respectively a zoom on the shear Reynolds stresses $< u'v' >$ and normal Reynolds stresses $< v'^2 >, < w'^2 >$, where we can notice the effect of the impermeability of the upper wall (due to our boundary conditions): the wall normal fluctuations are damped and their energy is redistributed among the other components, resulting in positive streamwise and spanwise fluctuations along the upper wall. If we had compared figure 4.3.4 to some other results obtained by wall turbulence at the same $Re_\tau = 330$, we could have seen some differences: firstly the relative magnitude of the streamwise fluctuations with respect to the spanwise and vertical ones, is much higher in the free surface case than in a normal wall; secondly, the absolute magnitude of $< u'^2 > /u_\tau^2$ appears to be almost two times larger than the usual one.
Other important informations about the magnitude and the sign of the velocity fluctuations can be derived by looking at the higher order statistics of the flow as the kurtosis and the skewness (also called flatness). The skewness is the third moment of the normalized probability density function of the velocity field and it has been found in each point of the domain, normalized by the cube of the variance and then spatially averaged along x and z directions, following equation (4.3.2). On the other hand Kurtosis is the

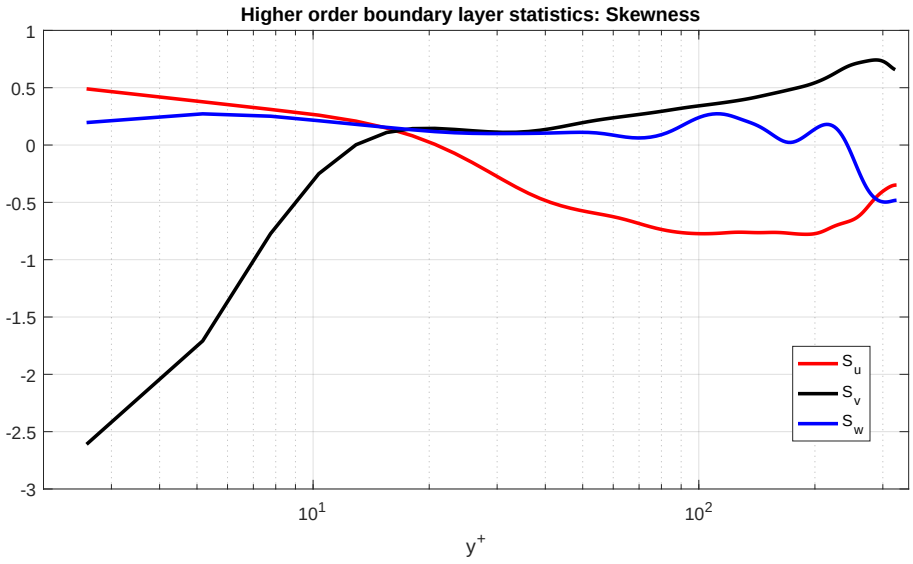


Figure 4.3.6: Higher order boundary layer statistics: skewness.

fourth moment, and it is found analogously by following equation (4.3.3), where $\bar{U}_i$ is indicating the mean velocity field.

$$S_{u_i}(y) = \frac{\frac{1}{N_x N_z}\sum_{i=0}^{N_x}\sum_{j=0}^{N_y}(u_i(x_i,y,z_j) - \bar{U}_i(y))^3}{(\frac{1}{N_x N_z}\sum_{i=0}^{N_x}\sum_{j=0}^{N_y} u_i(x_i,y,z_j) - \bar{U}_i(y))^{3/2}} \tag{4.3.2}$$

$$K_{u_i}(y) = \frac{\frac{1}{N_x N_z}\sum_{i=0}^{N_x}\sum_{j=0}^{N_y}(u_i(x_i,y,z_j) - \bar{U}_i(y))^4}{(\frac{1}{N_x N_z}\sum_{i=0}^{N_x}\sum_{j=0}^{N_y} u_i(x_i,y,z_j) - \bar{U}_i(y))^2} \tag{4.3.3}$$

From equations (4.3.2) and (4.3.3) we obtain three component for each moment, since that they are describing different fluctuating velocity field. All these components are shown in figures 4.3.6 and 4.3.7.

On the first figure we can see the obtained skewness for all the components S_u, S_v, S_w. Since that skewness is evaluated by taking the third power of the fluctuating velocity field, it indicates whether large positive or negative fluctuations dominate: where $S > 0$ then positive large fluctuations dominate, while where $S < 0$ than the dominating fluctuations are the negative ones. On figure 4.3.7 are shown all the kurtosis components K_u, K_v, K_w. Since that kurtosis is calculated by taking the fourth power of the fluctuating velocity field, it will indicates us the compactness of the probability density function: the bigger the kurtosis, the larger the probability to have large fluctuations (either negative or positive); the smaller the kurtosis, the smaller the magnitude of these fluctuations.

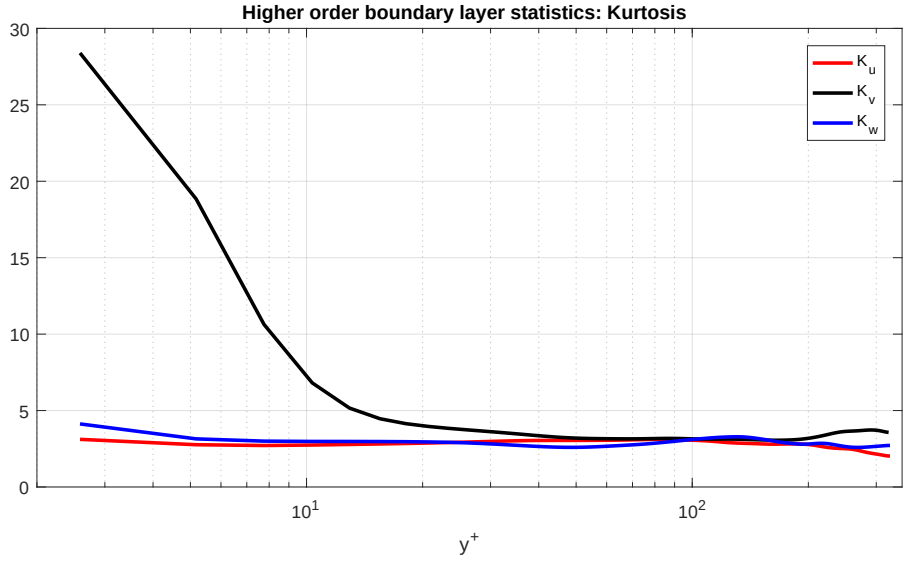


Figure 4.3.7: Higher order boundary layer statistics: kurtosis.

In our case, looking at those figures, the wall normal component of the skewness S_v and kurtosis K_v are the most interesting ones, since they present the largest variation along the y-direction. Taking into account the previous considerations about the physical meaning of these statistical tools, we can say that close to the free surface we have dominant large negative fluctuations v' , with a high probability to have large deviations from the mean flow (that in the wall normal direction is null). This is underlying the importance of the wall-normal fluctuations in the process of wind-wave generation.

4.4 Free surface statistics

We conclude this chapter with a brief analysis of the free surface statistics and we will evaluate them for the time averaged free surface height, indicated in the following as $h(x, z)$. Extracting this quantity from our simulation is not an easy task as it could seems: if the wall normal resolution of the grid is smaller than the free surface elevation $h'(x, z) = h(x, z) - \bar{h}$ (where $\bar{h}$ is the mean water depth or also the mean surface height), than it is very difficult to detect this fluctuations of the surface height. From figure 4.1.9 we have seen that the maximum amplitude of our waves is comparable with our wall normal resolution $h'_{max}/\Delta y \approx O(1)$, anyway it will be much better to have an higher ratio. The field $h(x, z)$ has been extracted by making a contour of the velocity field by fixing the value of the volume of fraction as $\alpha = 0.5$ with the help of Paraview, and then it has been reorganized in the usual disposition. The time averaged mean free surface height is resulted

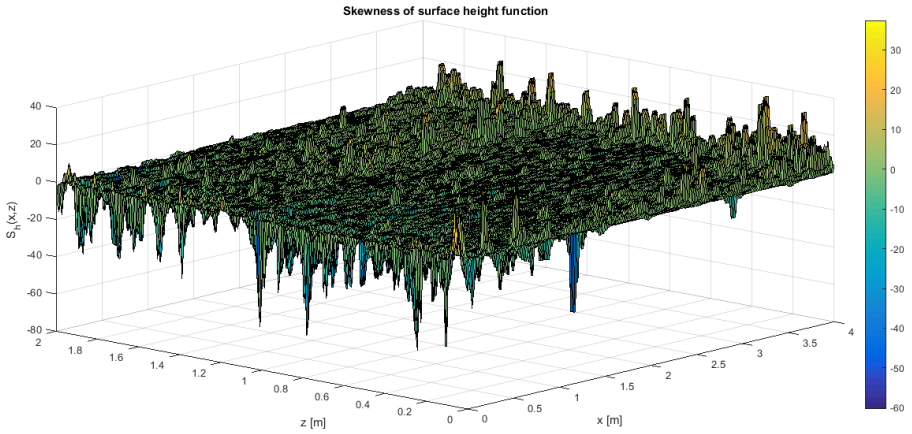


Figure 4.4.1: Higher order free surface statistics: skewness of the free surface elevation

to be approximately $\bar{h} \approx 0.25\ m$. The fluctuating field and all the other higher order statistics have been evaluated as explained previously for the wind boundary layer statistics.
Before spatially averaging, let's have a look to a two dimensional view of the higher order statistics. The skewness is shown in figure 4.4.1: in this figure we can see how negative and positive fluctuations are balanced both along x and z, as expected for a wavy surface. We can also observe that negative fluctuations appears to be predominant in the firs half length of the channel, while they are substituted by positive fluctuations in the second-half of the channel.

This behaviour could indicates us that ripples, that are all the other small negative and positive fluctuations present everywhere in the channel, are growing over a larger dominant wave that have a wavelength of the order of the channel length.

The kurtosis, normalized by the square of the variance, is illustrated in figure 4.4.2: an initial zone of larger fluctuations is followed by a zone of relatively smaller fluctuations, that is followed another time by a zone of relatively larger fluctuations. This is typical of a wave since that it is continuously oscillating around the mean value, getting closer in some points, while going further in some other points. Introducing the spatial average we will obtain just numbers, that are representative of the free surface behaviour: the standard deviation of the free surface elevation is found to be $h_{rms} \approx 0.0014$, around a mean value of $\bar{h} \approx 0.25\ m$; the normalized skewness is very close to zero (as expected) and it is $S_h \approx -0.2$; the normalized kurtosis results to be $K_h \approx 3.09$.

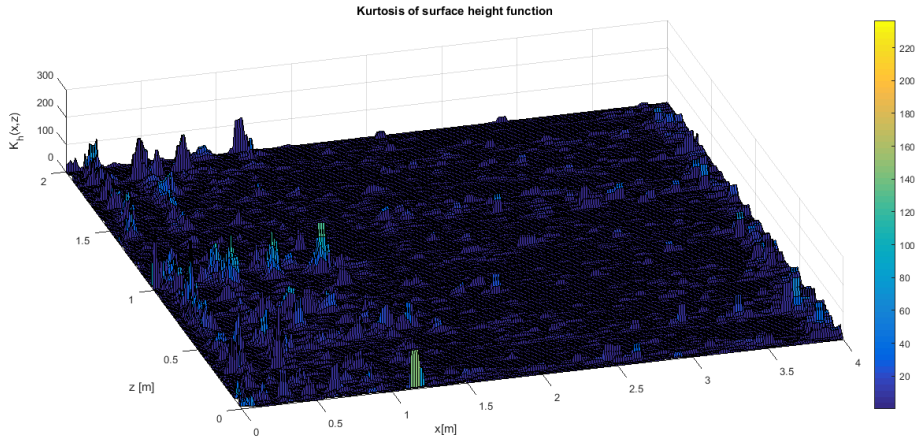


Figure 4.4.2: Higher order free surface statistics: kurtosis of the free surface elevation

Finally we evaluate the wavenumber spectrum of the free surface elevation function $h'(x,z)$, as shown in figure 4.4.3. First of all we recall the physical concept that is behind the spectral analysis. The angular wave number $k = 2\pi/\lambda$, or the linear wave number $k = 1/\lambda$, is representing the spatial frequency of a wave, that is the number of wavelength that are contained in a particular spatial period. Let's consider our channel length L_x and our grid resolution $\Delta x = L_x/N_x$. The smallest wave length that we can detect will be $\lambda_{x_{min}} = 2\Delta x$, since that we need at least two element of the grid to detect a small wave. Therefore the correspondent wave number will be the maximum one $k_{x_{max}} = 1/(2\Delta x) \approx 32$, since that 32 waves of wavelength $\lambda_{x_{min}}$ will be repeating along the channel length. Following this line, the smallest wave number, that corresponds to our resolution wave number, will be the one correspondent to the maximum wavelength contained in our channel length. This wave length will correspond to the channel length itself and therefore our wave number resolution will be $dk_x = 1/(2L_x) = 0.125$ m^{-1}. The resolution along z will be instead $dk_z = 1/(2L_z) = 0.25$ m^{-1} and the maximum wave number will be $k_{z_{max}} = 1/(2\Delta z) = 64$.

In order to understand which wavenumbers are more frequent, meaning that they are repeating more times inside the domain and thus containing most part of the total energy of the waves, we need to transform our surface elevation function $h'(x,z)$ of the physical space into the correspondent function $\hat{h}(k_x,k_z)$ in the wavenumber (spatial frequency) domain. The mathematical tool that we used in order to make this transformation is the two-dimensional Discrete Fourier Transform (2D DFT), and the algorithm with which it is implemented is called the Fast Fourier Transform (FFT).

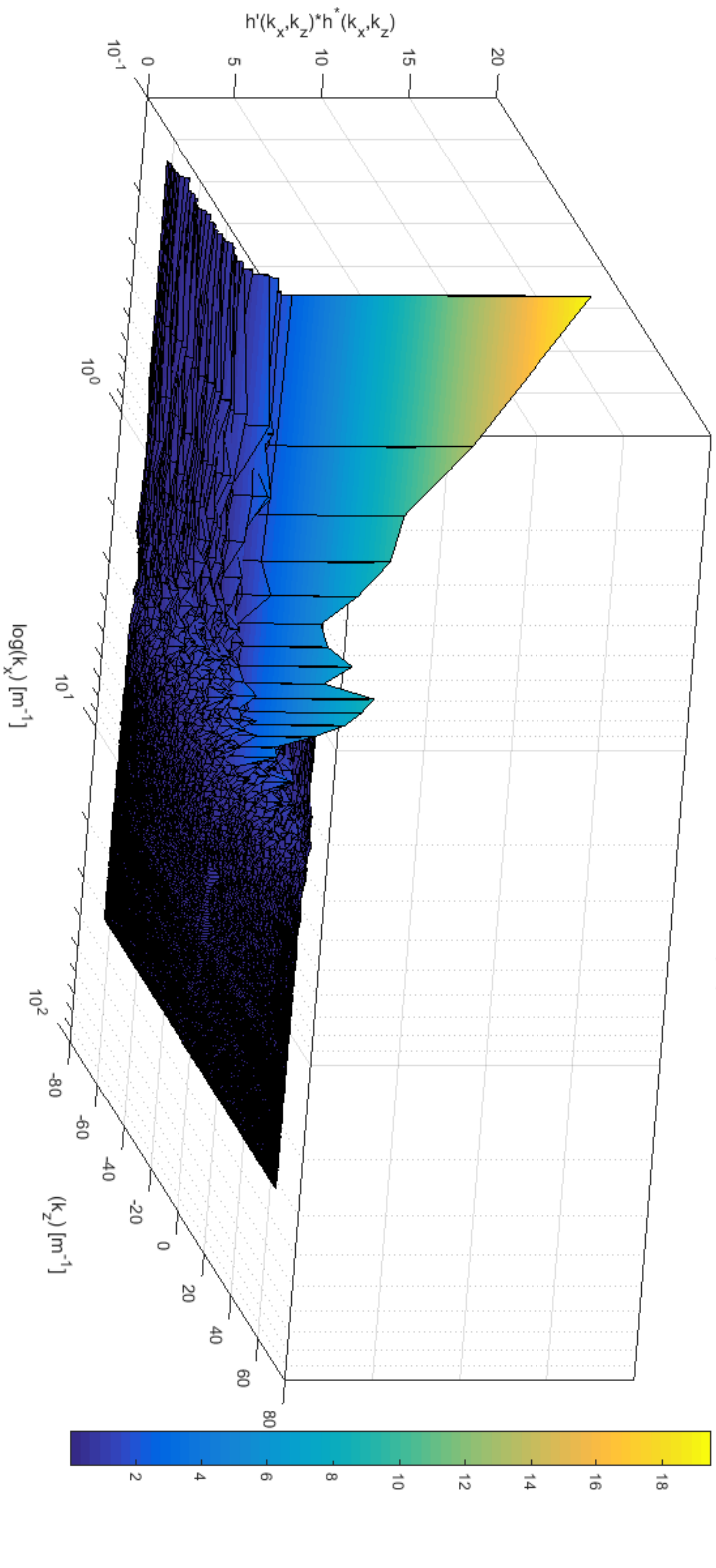


Figure 4.4.3: Wavenumber spectrum of the free surface elevation: from the mean current ($k_x = 0$) to the smallest ripple ($k_x = 32$)

The mathematical formula, applied to our channel, reads:

$$\hat{h}(k_x, k_z) = \sum_{j=0}^{N_x-1} \sum_{k=0}^{N_z-1} h(j,k) e^{-2\pi i(\frac{jk_x}{N_x} + \frac{jk_z}{N_z})} \tag{4.4.1}$$

where i is the complex unity. Applying formula (4.4.1) and plotting the magnitude of the function obtained versus k_x and k_z we finally obtain the figure 4.4.3. From this figure we can see how the mean current (for $k_x \approx 0$) is containing the most part of the energy. As the wave number increase than we find a lower value of energy. Actually at higher wave numbers the energy is not equally distributed: we can notice two peaks of energy around $k = (k_x, k_z) \approx (20, 0)\ m^{-1}$. This means that the dominant "ripples" length along the streamwise direction will be $\lambda_x \approx 5\ cm$. On the other hand along the spanwise direction we do not find these equivalent peaks, meaning that no important waves are growing and moving along the z-direction. This result is reasonable since that these wavelengths are typical of capillary-gravity waves, that are waves where both gravity and surface tension force play an important role. Actually the relationship between these two forces is usually measured by the dimensionless Bond number that in our case, considering $k_x = 20\ m^{-1}$ and $\sigma = 0.07\ kg/s^2$ will be:

$$Bo = \frac{\sigma k^2}{\rho g} \approx 0.003 \tag{4.4.2}$$

This value is showing us that this dominant ripple can be considered as a gravity wave, since that $Bo << 1$. When $Bo \approx 1$ then surface tension and gravity are both important, and in our case the wave number correspondent to this situation would be:

$$k_{gc} = \sqrt{\frac{\rho g}{\sigma}} \approx 370\ m^{-1} \tag{4.4.3}$$

Therefore in our channel the ripple formed are mainly dominated by the gravity force and thus we can apply the classical linear dispersion relationship, coming from linear wave theory, in order to get the streamwise frequency of our dominant wave and consequently the phase speed, as shown in equation (4.4.4).

$$c_p = \frac{w_p}{k_p} \approx 0.7\ m/s \tag{4.4.4}$$

$$w_p^2 = (gk_p + \frac{\sigma}{\rho_w} k_p^3) \tanh(k_p h) \approx \underbrace{gk_p \tanh(k_p h) \approx gk_p}_{\text{deep water gravity waves}} \approx 196\ rad/s^2 \tag{4.4.5}$$

In equation (4.4.5) the deep water gravity assumption has been applied since that $h/\lambda \approx 5$ and therefore $\tanh(k_p h) \approx 1$. Using the phase velocity

obtained we can finally calculate the wave age β, the second dimensionless similarity parameter that is characterizing our simulation together with the friction Re number Re_τ:

$$\beta = \frac{c_p}{u_\tau} \approx 35 \tag{4.4.6}$$

Usually, from oceanographic references, if $\beta > 20$, than the waves can be considered in the fully developed wind sea regime as shown in figure 1.2.3a. Finally we have obtained the two fundamental parameters that are describing our wind-sea state that are Re_τ and β, and a lot of different formulas have been purposed in order to find the relationship between the sea surface roughness and these dimensionless parameters. However, in our case we can only say that the sea surface roughness, defined as $y_0^+ = y_0 u_\tau / \nu$, relative to the fully developed state has found to be around $y_0^+ \approx 0.02$ (see (4.3.1)), and this value has to be dependent on the evaluated $Re_\tau \approx 333$ and wave age $\beta \approx 35$.

Chapter 5

Conclusions

The idea of this thesis arises from some of the questions we posed in the first chapter: in the following we will recall them as reference points for this conclusion.

- **How to build a numerical simulation model to study the wind wave generation process?**
  A DNS simulation has been successfully performed in order to study the wind-wave interaction, especially focusing the attention on the interfacial phenomena. The numerical model is founded to reproduce gravity waves generation under the action of a turbulent wind. The VOF (volume of fraction) method is used in order to capture and model the free surface, while the wind is generated by imposing an external pressure gradient. This last choice, in contrast with the usual ways of simulating wind by a moving wall, turned out to be very effective from both the computational and the physical point of view.
  The simulation can be described by two fundamental parameters: the friction Reynolds number $Re_\tau \approx 330$ and the wave age $\beta \approx 35$. The wave age has been calculated by a spectral analysis of the free surface: through this analysis the wavelength of the main ripple is found to be about $\lambda \approx 5\ cm$, while its phase speed results to be $c_p \approx 0.7\ m/s$. This wavelength and phase speed seems reasonable by comparing it with experiments and observations.
  Studying the thermalization of the process, we have characterized the wind and the wave state: the turbulent wind reaches his fully developed state after 60 seconds; water waves, on the other hand, are not fully developed yet. The free surface velocity U_{fs} turned out to be a very important parameter for the study of the different stages of wave growth.

- **What is the role of turbulence in the generation of surface waves by wind?**
  Turbulence cannot be neglected neither in the early stage of wave growth nor when water is fully developed. The very first ripples generated on the water surface can be considered as the mark left by turbulent pressure fluctuations. These fluctuations are always present along the free surface, and the wave state is strictly dependent on them.
  Wind and waves are always interacting each others: wind is influencing water as same as water does with wind. This is the reason why the properties of the turbulent wind have been studied: the turbulent boundary layer is significantly modified by waves in a way that resembles rough turbulence. The spatially and time averaged mean velocity profile is affected by the roughness of waves as much as the fluctuating field. In this study a sea surface roughness of $y_0^+ \approx 0.017$ is founded by observing the behaviour of the wind close to the free surface. This similarity between a rough wall and waves, can be relevant for modelling flows over wavy walls or predicting wind and wave currents, especially in the Geophysical fluid dynamic field.

- **How does the wind transfer and dissipate energy through water waves?**
  The free surface is that area that is connecting wind and water, where the energy transfer between these two fluids takes place. Therefore the properties of the interface have been analysed in order to understand this transfer mechanism. A lot of efforts have also been done in summarizing and reviewing all the theories about these phenomena, in order to have a better understanding of the results. The Phillips mechanism (turbulent pressure fluctuations along the free surface) and the sheltering mechanism (positive and negative pressure fluctuations respectively in front and on the leeside of the wave due to the boundary layer separation) have been clearly observed.

However, these three answers are still not complete and this work can be considered just as a starting point for the numerical study of wind-wave generation process. In the following we purpose some suggestions for improving the numerical model and planning further analysis:

- the mesh grid resolution, especially along the free surface normal direction Δy, should be increased in order to have a ratio $\Delta y/h_{rms}$ lower than one (where h_{rms} is the root mean squared of free surface elevation), implying a thinner transition region of the volume of fraction and thus a sharper interface. In our simulation this ratio was about

$\Delta y/h_{rms} \approx 1.3$. However, this is not the only measure that could provide a better resolution of the free surface; in fact another solution could be to increase the value h_{rms} by decreasing the water level with respect to the total boundary layer thickness ($\delta/h_w < 1$). Moreover, decreasing the ratio $\delta/h_w < 1$ while maintaining the same Re_τ, should allow us to study also the outer flow characteristics, that is the flow outside of the boundary layer. In fact, in the simulation performed in this thesis, the turbulent boundary layer was completely filling all the air part over the free surface.

- the way of exporting data from the OpenFOAM suite to the post-processing tools should be improved in order to have faster and more accurate results: exporting the data through intermediate graphic tools (as ParaView), is not as effective as directly accessing them.

- the effect of other important scalar variables, as temperature and humidity, in the momentum exchange between wind and water, should be investigated in further simulations by adding the transport equations for these quantities to the numerical model purposed in this thesis.

- the theoretical predictions of Phillips and Miles on the wave amplitude growth, should be analysed by looking at the very first transient of the process. In order to study these first stages, it is suggested to sample data with a lower time step (less than $\Delta t \approx 5\ s$ used in this simulation). Due to our time rate sampling, the study of the first instants of wave growth has not been examined in depth, while the attention has been focused in the fully developed state of wind and wave. To verify whether these states are reached, the advice is to control the friction Re_τ for wind, while the free surface velocity U_{fs} for water.

The Aerospace Engineering point of view, in particular the fluid dynamic branch, can be an important resource in studying and modelling turbulence for predicting wind and water waves evolution, thus helping the main disciplines that usually deal with those problems, like Oceanography and Geophysical fluid dynamic.

List of Figures

List of Tables

Bibliography

[1] Alessandro Toffoli and Elzbieta M Bitner-Gregersen. Types of ocean surface waves, wave classification. *Encyclopedia of Maritime and Offshore Engineering*.

[2] Yoshiaki Toba and Momoki Koga. A parameter describing overall conditions of wave breaking, whitecapping, sea-spray production and wind stress. *Oceanic whitecaps*, pages 37–47, 1986.

[3] HK Johnson, J Højstrup, HJ Vested, and Søren Ejling Larsen. On the dependence of sea surface roughness on wind waves. *Journal of Physical Oceanography*, 28(9):1702–1716, 1998.

[4] Horace Lamb. *Hydrodynamics*. Cambridge university press, 1932.

[5] Harold Jeffreys. On the formation of water waves by wind. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 107(742):189–206, 1925.

[6] Harald Ulrik Sverdrup and Walter Heinrich Munk. Wind, sea, and swell: theory of relations for forecasting. 1947.

[7] Owen M Phillips. On the generation of waves by turbulent wind. *Journal of fluid mechanics*, 2(5):417–445, 1957.

[8] John W Miles. On the generation of surface waves by shear flows. *Journal of Fluid Mechanics*, 3(2):185–204, 1957.

[9] SE Belcher and JCR Hunt. Turbulent shear flow over slowly moving waves. *Journal of Fluid Mechanics*, 251:109–148, 1993.

[10] SE Belcher and JCR Hunt. Turbulent flow over hills and waves. *Annual Review of Fluid Mechanics*, 30(1):507–538, 1998.

[11] MAC Teixeira and SE Belcher. On the initiation of surface waves by turbulent shear flow. *Dynamics of atmospheres and oceans*, 41(1):1–27, 2006.

[12] Anna Paquier, Frederic Moisy, and Marc Rabaud. Surface deformations and wave generation by wind blowing over a viscous liquid. *Physics of Fluids*, 27(12):122103, 2015.

[13] Fabrice Veron and W Kendall Melville. Experiments on the stability and transition of wind-driven water surfaces. *Journal of Fluid Mechanics*, 446:25–65, 2001.

[14] Peter P Sullivan, JAMES C McWILLIAMS, and Chin-Hoh Moeng. Simulation of turbulent flow over idealized water waves. *Journal of Fluid Mechanics*, 404:47–85, 2000.

[15] Mei-Ying Lin, Chin-Hoh Moeng, Wu-Ting Tsai, Peter P Sullivan, and Stephen E Belcher. Direct numerical simulation of wind-wave generation processes. *Journal of Fluid Mechanics*, 616:1–30, 2008.

[16] Cyril W Hirt and Billy D Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of computational physics*, 39(1):201–225, 1981.

[17] Lars Davidson. Fluid mechanics, turbulent flow and turbulence modeling. *Chalmers University of Technology, Goteborg, Sweden (Nov 2011)*, 2011.

[18] Stephen B Pope. Turbulent flows, 2001.

[19] Hermann Schlichting and Klaus Gersten. *Boundary-Layer Theory*. Springer, 2016.

[20] Hrvoje Jasak. *Error analysis and estimation for finite volume method with applications to fluid flow*. PhD thesis, Imperial College London (University of London), 1996.

[21] Onno Ubbink. *Numerical prediction of two fluid systems with sharp interfaces*. PhD thesis, University of London PhD Thesis, 1997.

[22] Henrik Rusche. *Computational fluid dynamics of dispersed two-phase flows at high phase fractions*. PhD thesis, Imperial College London (University of London), 2003.

[23] Pedro Lopes. *Free-surface flow interface and air-entrainment modelling using OpenFOAM*. PhD thesis, University of Coimbra PhD Thesis, 2013.

[24] CFD Open. Openfoam user guide. *OpenFOAM Foundation*, 2(1), 2011.

[25] OpenFOAM User Guide. Programmers guide. *JDT Core., retrieved from on Apr*, 27(3), 2011.

[26] Extended code guide. `http://www.openfoam.com/documentation/cpp-guide/html/`.

[27] HG Weller. Derivation, modelling and solution of the conditionally averaged two-phase flow equations. *Nabla Ltd, No Technical Report TR/HGW*, 2, 2002.