Study of Kayak Hull Design on Calm Water Resistance

Francisco Lobato da Rocha Barros

Dissertação do MIEM

Orientador: Doutor Alexandre Miguel Prior Afonso

Co-Orientador: Prof. Mário Augusto Pires Vaz

Faculdade de Engenharia da Universidade do Porto

Mestrado Integrado em Engenharia Mecânica

Fevereiro 2015

Abstract

A Computational Fluid Dynamics (CFD) model to simulate the water and air flow around competition kayak hulls was developed, with the principal aim of allowing accurate drag force predictions. The models used the Volume of Fluid method to determine the location of the free surface, and the k-ω Shear-Stress Transport (SST) model to account for turbulent effects. The open-source CFD software package OpenFOAM was used for this purpose, namely its solver application *interFoam*.

The model was first tested on the Wigley hull, a typical validation case for models of flow around marine vessels. Effects of mesh refinement and alteration were studied, and results for total drag force and hull wave profiles were obtained from the model for several different Froude numbers and compared to experimental measurements.

The CFD model was then applied to two competition sprint kayaks manufactured by Nelo, the Vanquish III and Vanquish IV models. Drag force was calculated for a range of velocities and compared to experimental data and to other numerical results.

The results taken from the Wigley hull simulations had a good adjustment to experimentally obtained curves, and mesh refinement was found to lead to more accurate values, not only for drag forces, but also for hull wave profiles.

The drag forces obtained for the kayak models were generally close to experimental results up to velocities around 3.69 m/s, after which the numerical model underestimated the value of the total drag. Other available numerical results showed similar problems, possibly due to fundamental differences between the numerical and experimental procedures, such as the fact that the numerical models assumed that the only motion of the hull was its forward velocity.

Resumo

Foi desenvolvido um modelo de Dinâmica de Fluidos Computacional (CFD – *Computational Fluid Dynamics*) para a simulação do escoamento de água e ar em torno de cascos de kayaks de competição, com o objetivo principal de permitir prever com exatidão a força de arrasto. Foi usado o método do Volume de Fluidos (VoF – *Volume of Fluid*) para determinar a localização da superfície livre, e o modelo k-ω *Shear-Stress Transport* (SST) para lidar com os efeitos turbulentos. O pacote de software open-source OpenFOAM foi usado para a modelação, nomeadamente a sua aplicação *interFoam*.

O modelo foi inicialmente testado no casco de Wigley, um caso de validação típico para modelos de escoamentos em torno de embarcações. Foram estudados os efeitos do refinamento e da alteração da malha, e obtiveram-se resultados para o arrasto total e para a elevação de onda ao longo do casco para diferentes números de Froude, que foram comparados com medições experimentais.

Seguidamente, o modelo CFD foi aplicado a dois kayaks de sprint de competição fabricados pela Nelo, o Vanquish III e o Vanquish IV. A força de arrasto foi calculada para uma gama de velocidades e comparada a dados experimentais e a outros resultados numéricos.

Os resultados das simulações com o casco de Wigley ajustaram-se bem às curvas obtidas experimentalmente, e verificou-se que o refinamento da malha produz valores com maior exatidão, não só para as forças de arrasto, mas também para a elevação das ondas.

As forças de arrasto obtidas para os kayaks encontraram-se, no geral, próximas dos resultados experimentais até velocidades à volta dos 3,69 m/s. Para velocidades mais altas, o modelo numérico subestimou o valor do arrasto total. Outros resultados numéricos disponíveis demonstraram problemas semelhantes, possivelmente devido a diferenças fundamentais entre os procedimentos numérico e experimental, tais como o facto de os modelos numéricos terem assumido que o único movimento do casco é a sua velocidade de avanço.

Acknowledgements

I would like to thank:

My thesis supervisor, Dr. Alexandre Afonso, for his assistance throughout the course of the project and the transmission of valuable knowledge in the area of Computational Fluid Dynamics.

My co-supervisor, Prof. Mário Vaz, for helping to develop the premise of this study, providing insight and experience from previous related works, and coordinating practical aspects of the project.

Eng. Nuno Viriato Ramos, for his help with the CAD modelling of the studied kayaks and assistance in the preparation of experimental measurements.

Beatriz Gomes, for sharing experimental results that were vital to the development of this project, and the insight into experimental procedures from previously developed projects.

Everyone at LOME who allowed me to develop this project in a pleasant work environment.

My family and friends, for their support throughout the course of the semester.

Table of Contents

List of figures

List of tables

Nomenclature

Variables

A	Wave amplitude
B	Hull breadth
C_f	Frictional drag coefficient
C_p	Pressure drag coefficient
C_r	Residual drag coefficient
C_t	Total drag coefficient
C_v	Viscous drag coefficient
Co	Courant number
D_f	Frictional drag force
D_r	Residual drag force
D_t	Total drag force
D_w	Wave drag force
F_{px}	Pressure forces in the x direction
F_{vx}	Viscous forces in the x direction
Fr	Froude number
g	Gravitational acceleration
H	Hull draught
k	Turbulence kinetic energy per unit mass
L	Hull length
p	Pressure
p_{rgh}	Dynamic pressure
Re	Reynolds number
S	Wetted surface area
S_{ij}	Mean strain rate tensor
t	Time
U	Freestream velocity
$\boldsymbol{U}$	Velocity vector
U_i	Velocity in direction i
$\overline{U_i}$	Mean velocity in direction i
u_i'	Fluctuating velocity in direction i
x	Spatial coordinate
y	Spatial coordinate

z	Spatial coordinate
α	Volume fraction of a fluid
α_1	Water volume fraction
δ_{ij}	Kronecker delta function
δt	Simulation time step
δx	Cell size
ε	Turbulence kinetic energy dissipation rate
ζ	Hull shape function
θ	Direction of wave propagation
η	Wave elevation
μ	Dynamic viscosity
μ_T	Eddy viscosity
ν	Kinematic viscosity
ν_T	Kinematic eddy viscosity
ρ	Density
τ_{ij}	Reynolds stress tensor
ω	Turbulence kinetic energy dissipation frequency

Indices

i, j	Generic direction
m	Scaled model
r	Full-sized model
$1, 2$	Fluids in a multiphase flow

Abbreviations

CFD	Computational Fluid Dynamics
DES	Detached Eddy Simulation
DNS	Direct Numerical Simulation
ICF	International Canoe Federation
ITTC	International Towing Tank Committee
LES	Large Eddy Simulation
PDE	Partial differential equations
RANS	Reynolds-Averaged Navier-Stokes equations

SRI	Ship Research Institute
SST	Shear Stress Transport
UT	University of Tokyo
VoF	Volume of Fluid

1 Introduction

1.1 Context of the study

This study was conducted in the context of the final dissertation project in the Integrated Master's Degree in Mechanical Engineering at the Faculty of Engineering of the University of Porto (FEUP). The work was developed at INEGI's Laboratory of Optics and Experimental Mechanics, where previous projects on the topic have been undertaken, namely the experimental measurement of drag forces and the experimental and numerical structural analysis of a kayak.

The relevance of this study arises from the fact that there are few available numerical and experimental studies concerning modern kayak designs, and comparative analysis of experimental and numerical results is lacking. The conclusions obtained and the models developed in the course of this project are intended to be useful towards the general study and advancement of kayak design.

The development of scientific knowledge on kayak hull design is especially pertinent in this period leading to the 2016 Olympic Games. It is also of local importance, since Portugal has been an internationally prominent country in canoeing in recent years, obtaining its only medal in the 2012 Olympic Games in this sport, and being home to Nelo, the most successful kayak manufacturer in the world at the moment, having produced 25 out of 36 medal-winning kayaks and canoes in the latest Olympics.

1.2 Objectives

The aim of this study is to develop a numerical model of the flow around a kayak hull moving in calm water using the open-source Computational Fluid Dynamics (CFD) software package OpenFOAM. The choice of software is supported on the fact that it is open-source and widely accepted by the CFD community, both in academic and industrial settings.

The flow is modelled as a two-phase, incompressible, isothermal flow, involving water and air, and the hull is assumed to have no translational or rotational motion aside from its forward velocity.

In the first stage of the project, simulations are conducted on a Wigley hull, which is a mathematically defined hull shape that is used as a validation case for numerical studies of the flow around ship hulls. Different mesh refinements are tested and the retrieved data for the drag force on the hull and wave elevation are compared with experimental results.

The second part of the project is the application of the validated numerical model on the flow around two sprint kayak models, the Nelo Vanquish III and the Nelo Vanquish IV, and their comparison to experimental and numerical data.

The developed numerical models should be able to accurately predict drag forces on kayak hulls for a range of velocities compatible with the speeds typically reached by the tested kayaks (about 10 to 20 km/h). The completed work is also intended to be a possible starting step towards the development of an optimisation process guiding the alteration of kayak hull geometry in order to minimise total drag.

1.3 Structure of the dissertation

This introductory chapter presents an insight into the context of this project, and covers general information related to subjects that are relevant to the work developed. A brief description of the sport of canoeing is made, followed by the presentation of fluid mechanics concepts related to the movement of a marine vessel. Official rules for kayak design and the impact of the evolution of kayak models throughout the history of canoeing are then mentioned. Finally, the state of the art is discussed, both for the study of kayaking and for the Wigley hull.

Chapter 2 covers the governing equations of the flow, including general flow equations, multiphase flow modelling, and turbulence modelling.

Chapter 3 is a general description of OpenFOAM, the software used in this project.

Chapter 4 describes the conducted simulations in detail, both for the Wigley hull and the Nelo kayak models.

Chapter 5 presents and compares the results for drag forces and wave hull profiles obtained in the various simulations and analyses differences and similarities to previous numerical and experimental results.

Chapter 6 covers the conclusions reached through the study developed and outlines possible future work that can be made following these conclusions.

1.4 Canoeing

1.4.1 General information on canoeing

Canoeing is a broad term for a number of sporting events involving racing in canoes or kayaks. In the latter case, it is also frequently referred to as kayaking. Canoeing is an Olympic sport in several of its forms, and it is supervised by the International Canoe Federation (ICF).

The main conceptual differences between kayaks and canoes are the position of the paddlers and the type of paddle. On a kayak, the paddlers are seated and use a double-bladed paddle to propel the vessel, paddling alternately on their right and left sides. On

a canoe, the paddlers are kneeling on one leg, with the opposite foot on the floor of the boat, and use a single-bladed paddle on only one of the sides of the vessel [1].

Both kayaks and canoes can have one, two or four paddlers. This is reflected on the nomenclature of canoeing events: the name of each event, alongside the type of race, features the letter K or C, which specifies whether the vessel used is a kayak or a canoe, respectively, and the number 1, 2, or 4, according to the number of paddlers. So, for instance, "K2 500 m Sprint" refers to a 500 m sprint kayak race with two paddlers on each kayak [1].

1.4.2 Types of canoeing disciplines

1.4.2.1 Canoe Sprint

In sprint canoeing, the canoeists or kayakers race in a straight line, in calm water, over a set distance, which can be 200 m, 500 m, 1000 m, or 5000 m [2].

Canoe sprint was the first official canoeing event, and it is still considered the traditional form of canoeing. It has been an Olympic sport since 1936. As of the 2016 Olympic Games, there are 12 Olympic canoe sprint categories [1].

1.4.2.2 Canoe Slalom

Canoe slalom, unlike canoe sprint, takes place in turbulent water (white water), and the race course is not in a straight line, instead having gates that the paddlers must go through in a certain direction (downstream or upstream). Courses are 200 to 400 metres long and can be natural or artificial [3, 4].

The first Olympic canoe slalom event took place in 1972, and it has featured in every Olympic Games since 1992 [3]. There are four canoe slalom events in the Olympics as of the 2016 Games [1].

1.4.2.3 Canoe Marathon

Canoe marathon consists of long distance races in natural bodies of water. Marathon courses often include portages, where paddlers must carry their canoe or kayak on foot between certain sections of the course [5].

There is no fixed length for canoe marathon races, only a minimum limit of 15 km for women and juniors (athletes under 18 years of age), and 20 km for men. Races often consist of several laps around a course, as is the case, for instance, in World and European championships [1, 5]. Canoe marathon is not an Olympic sport.

1.4.2.4 Other disciplines

There are several other canoeing disciplines, some of which will be briefly described in this section.

Wildwater canoeing races take place in rivers or artificial white water courses. The premise of this discipline is for the paddlers to race downstream in fast moving white water. There are sprint categories and long distance categories [1, 6].

Canoe polo, similarly to water polo, is a ball sport played in teams, where each team attempts to score goals against the opposing team, and the team with the most scored goals wins [1, 7].

Dragon Boat is a discipline using very long open canoes for 10 or 20 paddlers using single-bladed paddles. It incorporates various aesthetic elements related to ancient Chinese tradition [1].

Canoe freestyle, which takes place around stationary features of rivers (such as standing waves), is a discipline where each paddler has a certain amount of time to perform special moves, each of which gives them a certain amount of points depending on its difficulty [1].

Canoe ocean racing is a discipline whose races take place in the sea, using a long and narrow type of kayak called a surfski [1].

Finally, Paracanoe refers to a number of canoeing events for sportspeople with physical disabilities. It will make its debut on the Paralympic Games at the 2016 Games in Rio de Janeiro, with kayak and Va'a events. Va'a is a long boat of Polynesian origin, which was found to be particularly well-suited for competition racing for people with disabilities [1].

1.5 Hull hydrodynamics

1.5.1 Drag force on a hull

Drag can be defined as the net force in the direction of the flow that a fluid exerts on an object that is moving through it. For an object immersed in a certain fluid, there are two components of drag: pressure drag (or form drag), which is a result of the pressure distribution along the surface of the object, and friction drag, caused by the shear forces acting on it [8].

Marine vessels move through an interface between two fluids, typically water and air. When a vessel moves, its movement will cause a deformation in the shape of that interface, creating a pattern of surface waves, and the hull of the vessel needs to exert a force to sustain this pattern.

The drag force (or resistance force) on a ship hull can then be considered to be the sum of three components: the pressure drag and friction drag typical of any external flow, and a wave-making drag component [9].

The dimensional analysis of the drag force should then include the total drag force D_t, the length of the hull L, its velocity U, the wetted surface area S, the liquid

medium's density ρ and kinematic viscosity ν, and the gravitational acceleration g, on which the surface wave pattern heavily depends.

These parameters can be expressed in terms of three nondimensional quantities, the total drag coefficient C_t, the Reynolds number Re and the Froude number Fr:

$$C_t = \frac{D_t}{\frac{1}{2}\rho S U^2} \tag{1.1}$$

$$Re = \frac{UL}{\nu} \tag{1.2}$$

$$Fr = \frac{U}{\sqrt{gL}} \tag{1.3}$$

We can then see that the total drag force can be written as a function of the Reynolds and Froude numbers:

$$C_t = C_t(Re, Fr) \tag{1.4}$$

One other important aspect to consider in the description of the drag force on a hull is Froude's hypothesis, which states that the total drag can be expressed as the sum of a Reynolds number-dependent component and a Froude number-dependent component:

$$D_t \cong D_f(Re) + D_r(Fr) \tag{1.5}$$

Here, D_f is the frictional drag force, which depends solely on the Reynolds number and corresponds to the viscous forces, and D_r is the residual drag, which depends exclusively on the Froude number, and includes the drag due to pressure forces, that is, both the wave-making resistance and the form drag, which is considered constant in the context of Froude's hypothesis and is often very small in comparison to the former.

It also follows that the drag coefficient can be separated in the same way as the forces, into a frictional drag coefficient and a residual drag coefficient:

$$C_t \cong C_f(Re) + C_r(Fr) \tag{1.6}$$

Froude's hypothesis becomes extremely important when performing tests on models, as it may become very difficult to scale both the Reynolds number and the Froude number in a scaled model. Scaling simply one of them only requires changing the velocity, while scaling both requires the adjustment of the velocity to match the Froude number, and then the fluid's kinematic viscosity to match the Reynolds number, and it may be difficult to find suitable fluids with a low enough viscosity [9].

1.5.2 Surface waves

As was mentioned in the previous section, the movement of an object on a free surface will create a pattern of surface waves, which are the result of the balance between the kinetic energy transferred to the fluid by the object and the fluid's potential energy resulting from the Earth's gravitational field [9].

A surface wave pattern can be characterised by the vertical wave elevation η. Considering y to be the vertical spatial coordinate,

$$y = \eta(x, z, t) \tag{1.7}$$

The pattern produced by a moving vessel in calm water at a large enough distance from the vessel itself can be described by the Kelvin wave pattern, which assumes a steady state scenario and an infinitely deep body of water. If, in a coordinate system that moves with the vessel, x is the coordinate with the direction of the velocity of the vessel, z is the horizontal coordinate perpendicular to the velocity of the vessel, and θ is the angular direction of wave propagation measured such that $\theta = 0$ in the opposite direction of the x axis, the Kelvin wave pattern is given by:

$$\frac{z}{x} = -\frac{\cos\theta \sin\theta}{1 + \sin^2\theta} \tag{1.8}$$

The plot for the above equation is presented in Figure 1, which shows some important features of this pattern. One of them is the existence of a maximum and minimum at $^{z}/_{x} = \pm 2^{-3/2}$, which corresponds to an angle of approximately 19°28' from the x axis. This means that the wave pattern created in the wake of a vessel will, at a far enough distance from the vessel, lie fully within a 19°28' angle from the direction that the vessel is travelling from. The value of the wave propagation direction that yields these maximum and minimum values is $\theta = \sin^{-1}\frac{1}{\sqrt{3}} \cong 35°16'$, which means that the waves in the limits of the wake propagate at an angle of 35°16' from the x axis.

Another feature that is visible in the plot is the fact that, with the exception of the maximum and minimum and the points where $z = 0$, there are two distinct values of θ for every value of $^{z}/_{x}$. This means that there are two systems of waves that satisfy the condition presented in equation (1.8). These two types of waves are referred to as transverse waves and diverging waves, and their specific visual patterns are presented in Figure 2.

The waves corresponding to the Kelvin pattern are a far-field phenomenon, that is, they occur at a relatively large distance from the vessel. The near-field wave patterns, which depend mostly on the shape and velocity of the vessel, are more prominent in its proximity.

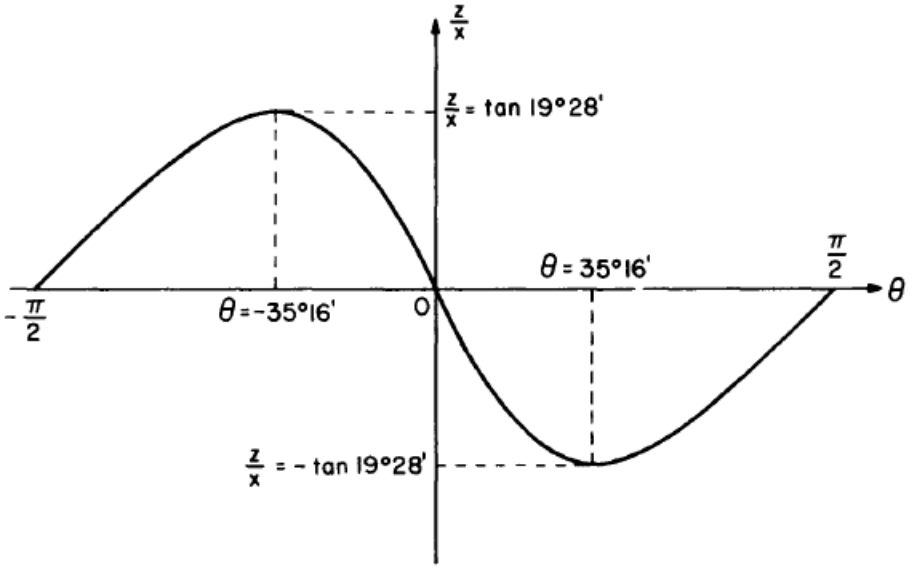

Figure 1 – Plot of the location relative to the vessel of waves propagating in a certain direction, according to the Kelvin wave pattern (from [9])

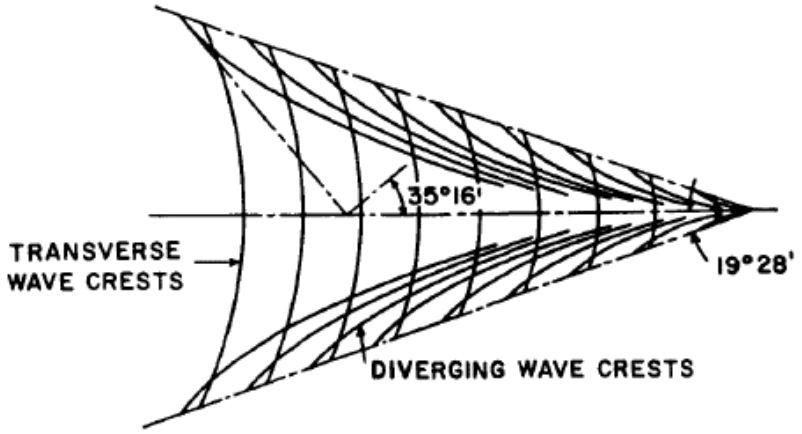

Figure 2 – Schematic representation of the two types of waves in the Kelvin wave pattern (from [9])

1.5.3 Estimation of drag forces

There are several analytical and empirical curves that can serve as an approximation for the frictional drag coefficient on a marine vessel. The semi-empirical Schoenherr line, which provides the value of the friction drag for turbulent flow parallel to a flat plate, is given by the following expression [9]:

$$\frac{0.242}{\sqrt{C_f}} = \log_{10}\left(Re \cdot C_f\right) \tag{1.9}$$

There are also curves based on experimental data for marine vessels that differ slightly from the Schoenherr line. These include the ITTC (International Towing Tank Conference) line and the Hughes line, shown in Figure 3. These curves account for the

fact that there is, in fact, a Reynolds number-dependent component of the form drag, and so the term “frictional drag” as applied to these curves does not refer only to the drag created by shear forces, but to the entirety of the drag components that depend exclusively on the Reynolds number [9].

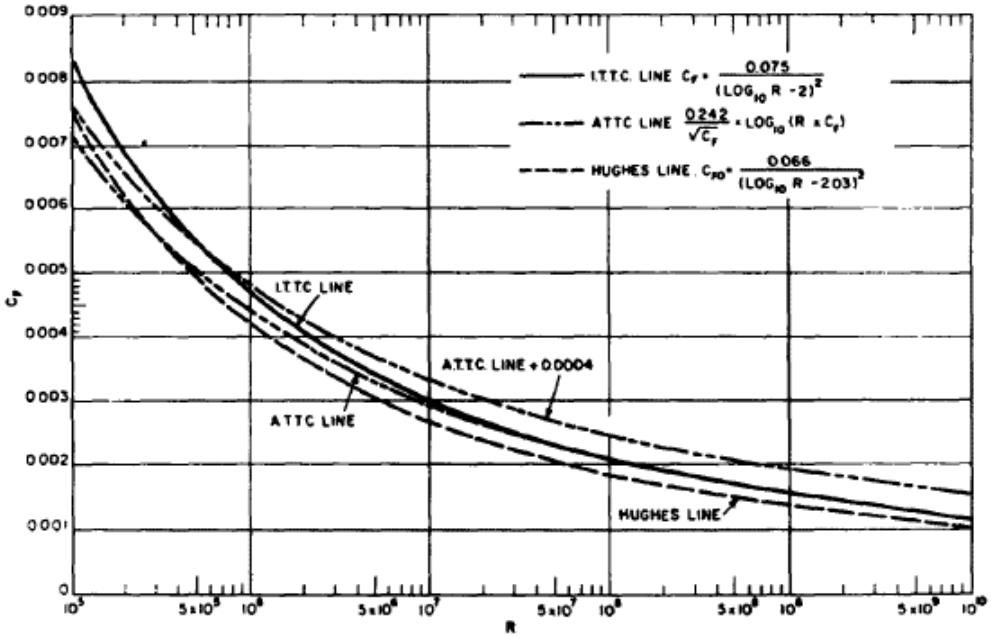

Figure 3 – Frictional drag coefficient vs. Reynolds number according to the Schoenherr line (shown here as “ATTC line”), the ITTC line, and the Hughes line (from [9])

Wave resistance can be estimated experimentally by measuring the amplitude of the wave profile in the wake of the vessel. If $A(\theta)$ is the amplitude of a wave that propagates in a direction θ, the wave drag D_w for a certain flow velocity U can be given by:

$$D_w = \frac{1}{2}\pi\rho U^2 \int_{-\pi/2}^{\pi/2} |A(\theta)|^2 \cos^3\theta \, d\theta \tag{1.10}$$

It is then evident from equation (1.10) that as long as the $A(\theta)$ function is known, the wave resistance can be computed. The estimation of this drag component by experimental measurement of the amplitude function is given the name of wave pattern analysis and the value obtained from that method is often called wave pattern resistance.

In the absence of experimental data, the wave drag can be estimated by resorting to the thin-ship theory of wave resistance, which is built from the assumption that the beam of the hull is much smaller than any other of the characteristic lengths relevant to the problem. This simplification allows for a way to analytically describe the amplitude function with only the knowledge of the shape of the hull, making it possible to use equation (1.10) to determine the wave resistance.

If a hull shape is defined by a function ζ such that $z = \pm\zeta(x, y)$, then the amplitude function, as determined by thin-ship theory, is given by:

$$A(\theta) = \frac{2}{\pi} \cdot \frac{g}{U^2} \sec^3\theta \iint \frac{\partial \zeta}{\partial t} \exp\left[\frac{g}{U^2} \sec^2\theta \, (y - ix\cos\theta)\right] dx \, dy \qquad (1.11)$$

Substituting this value for the amplitude function in equation (1.10) results in Michell's integral:

$$D_w = \frac{4\rho g^2}{\pi U^2} \int_0^{\pi/2} \sec^3\theta \left| \iint \frac{\partial \zeta}{\partial t} \exp\left[\frac{g}{U^2} \sec^2\theta \, (y - ix\cos\theta)\right] dx \, dy \right|^2 d\theta \qquad (1.12)$$

Solving Michell's integral analytically is not generally of interest, since it would only be possible for very particular $\zeta(x, y)$ functions, and consequently very particular hull shapes. In reality, the hull shape cannot usually even be described analytically with simplicity. However, the value of Michell's integral for a certain hull shape can normally be reached using numerical methods.

1.5.4 Motion of a marine vessel

Marine vessels usually have movements other than their forward velocity. They can be caused by waves and other hydrodynamic phenomena, or mechanical aspects relating to the boat itself, such as propulsion mechanisms (which, in the case of a kayak, refers to the movement of the paddler and the paddle).

The motion of a vessel can be translational or rotational in all directions. Using the convention used in previous sections, translational movement in the x, y and z axes, respectively, is known as surge, heave and sway, whereas rotational movement in these same directions is given the name of roll, yaw and pitch (Figure 4). These movements usually have an oscillatory nature [9, 10].

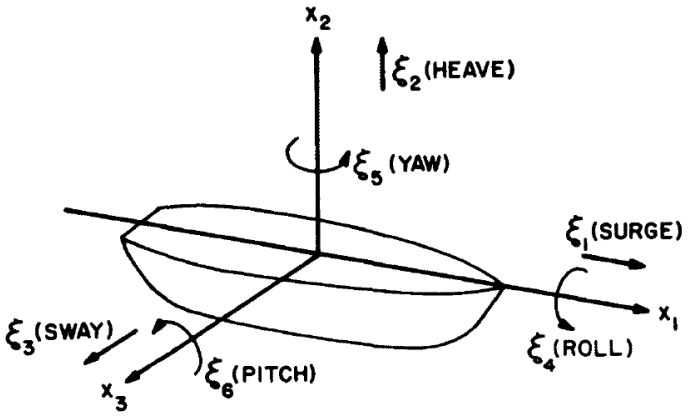

Figure 4 – Translational and rotational motion of a marine vessel in three dimensions (from [9])

Of these six motions, roll and pitch are considered to be of particular importance because of their role in boat stability. The stability of the position of a vessel depends on

the balance of its weight and the buoyant force exerted by the water. When roll and pitch are kept within a certain range of values (which depend on the shape of the hull), the displacement of the centre of mass relatively to the centre of buoyancy causes a moment that makes the vessel return to its original position. However, when the roll or pitch become too high, this moment takes the opposite direction, reaching an unstable condition that makes it turn over [9].

Aside from these oscillatory motions, a moving marine vessel also experiences a permanent displacement relative to its position when stationary. This phenomenon is known as squat, and it is typically seen as the combination of two movements, sinkage and trim.

Sinkage refers to the vertical downward displacement of the vessel, caused by the fact that the vessel accelerates the water below the hull. The added velocity of the water causes a decrease in its average pressure, thus forcing the vessel to move vertically towards the water [11, 12].

Trim is a permanent rotation of the vessel around the horizontal axis perpendicular to its movement: the front of the vessel rises and the back of the vessel drops. This is because the relative movement between the vessel and the water causes a pressure differential between its front and back, which, apart from intervening in pressure drag (as seen in section 1.5.1), is responsible for this rotation [11, 12].

Squat becomes more significant when the vessel is travelling at higher velocities and when it is moving in shallow water. The shape of the vessel and the geometrical configuration of its surroundings, if significant, are also influential on sinkage and trim [11, 12].

1.6 Kayak hull design

1.6.1 Current rules for kayak construction

Competition kayaks must abide by the rules set by the International Canoe Federation for the discipline in which they are used. In this section, the latest versions of the rules for K1 sprint [2] and K1 marathon [5] events will be focused on, since these are the disciplines that the kayak studied in this project is suitable for.

The rules for both of the aforementioned categories impose a maximum length and a minimum weight for kayaks. The maximum length is 5.2 m for both sprint and marathon; the minimum weight is 12 kg for sprint and 8 kg for marathon.

There are few rules regarding kayak construction in canoe marathon. The only requirement that the hull geometry must obey is that none of its sections or longitudinal lines should be concave, if measured vertically or horizontally. The only other geometrical limitation is concerned with the dimensions of rudders, which may not be over 10 mm thick in case they form an extension to the length of the kayak. In addition,

any foreign substance that enhances the performance of the kayak, such as hull lubricants, is forbidden.

The rules for canoe sprint are somewhat more thorough. The ones applicable to kayak construction are enumerated below:

- The kayak must remain buoyant when filled with water;
- The athlete's body must not be attached to the kayak;
- None of the kayak's sections or longitudinal lines should be concave, if measured vertically or horizontally;
- The highest point on the deck must be lower than the highest point of the front edge of the cockpit;
- Foreign substances which may enhance the performance of the kayak are not allowed;
- With the exception of moving seat systems, moving parts that may be used to propel the kayak are not allowed;
- Devices that provide real-time feedback to the paddler are not allowed;
- Kayaks may have one rudder, which must be placed under the hull;
- The kayak must be designed as a sit-in, not a sit-on (i.e. in such a way that the paddler is sitting inside a compartment, as opposed to on top of the kayak).

1.6.2 Hull design and kayak performance

Since canoe sprint was introduced as an Olympic sport in 1936, the race times of gold medallists have, as a general rule, become smaller as the years passed. Several factors are believed to have contributed to this general decrease, and the most important advances in kayak performance are considered to have been associated to three main aspects: hull design, paddle blade design, and factors related to paddling technique (both training methods and changes in kayak design [13].

The effect of the evolution of hull design on racing performance can be confirmed by the fact that, historically, the steepest decreases in the racing times of Olympic winners have often coincided with the introduction of new designs (Figure 5). In addition, the racing times of individual kayakers at the Olympic Games have been known to decrease remarkably after they adopted these designs [13, 14].

Throughout most of the history of kayaking, advances in kayak design were mostly achieved by trial and error and guided by knowledge obtained from data for other marine vessels with different characteristics and purposes to the ones of kayaks. Only since the 1980s has scientific research on kayaks been widely made and used to develop new models, starting with the widespread use of towing tests to determine drag forces, and continuing with the development of Computer Aided Design (CAD) and Computational Fluid Dynamics (CFD) software [13].

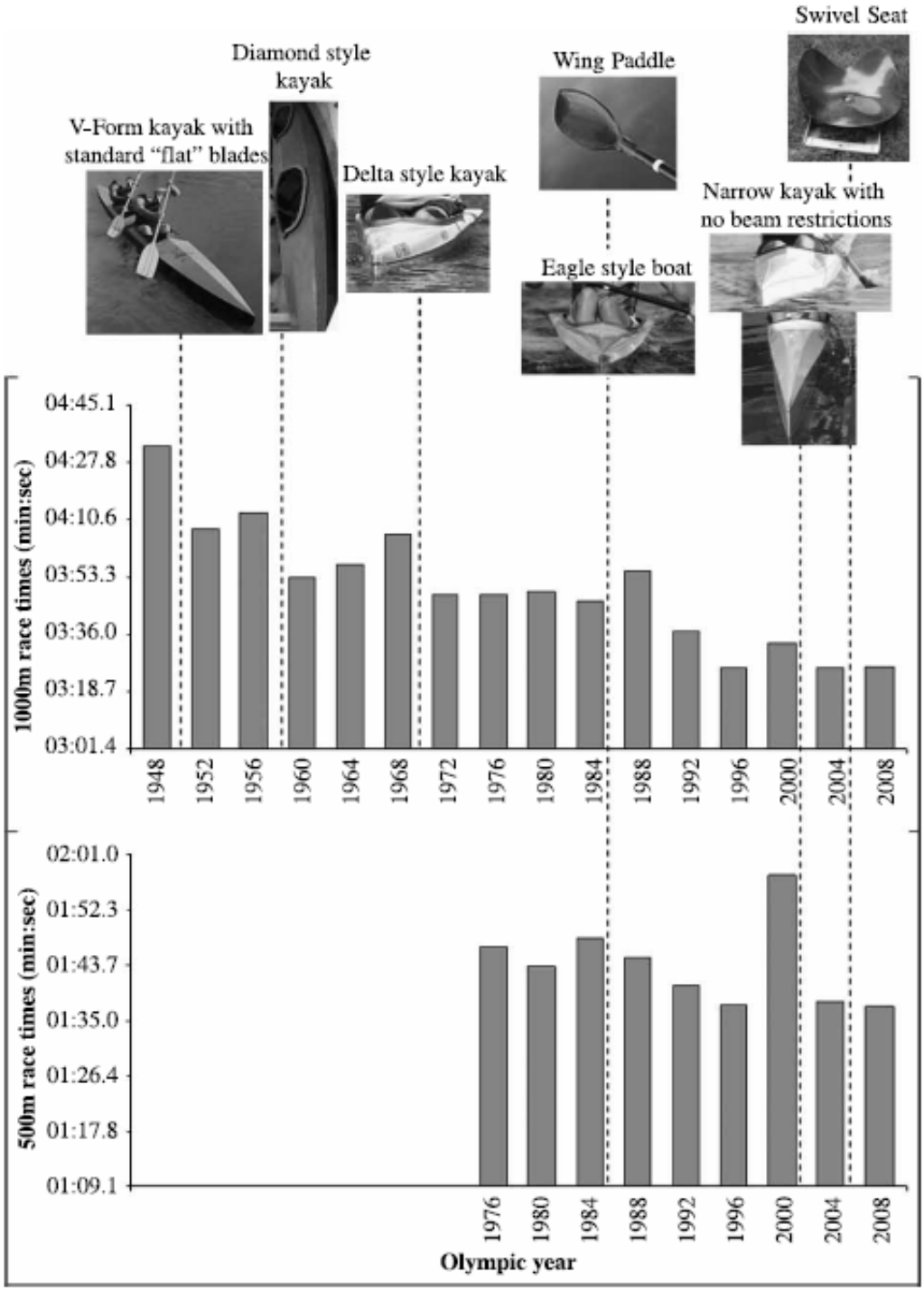

Figure 5 – Evolution of winning times of men's K1 Olympic 1000 m and 500 m races, with key improvements indicated (from [14])

The latest great development in kayak design was brought about by a change in the ICF regulations, which, in November 2000, removed all requirements for minimum beam width. It was known by then that smaller beam-to-length ratios allowed for higher maximum speeds to be reached, and, in the period leading up to this change in the IFC rules, there was a tendency towards designing kayaks that reached their maximum width

high above the waterline, so that the submerged part of the kayak became as narrow as possible while still verifying the minimum beam requirement. Consequently, the lifting of this requirement brought about a general tendency for sprint and marathon kayaks to become as narrow as possible [14].

1.7 State of the art

1.7.1 Kayaks

Some modern K1 sprint kayak designs, including the models used for this project, have been the target of numerical calculations and experimental measurements.

Gomes et al. [15] performed experimental tests on two models by Nelo: an M-sized Vanquish III and an ML-sized Vanquish I. The kayaks were towed by a bimaran motorboat at different velocities and their total drag force was measured. The results are higher than most other values found experimentally or numerically for other modern K1 kayaks, including other Nelo Vanquish models. Upon private contact with the authors of this paper, it was learned that there is a possibility that the kayak was in the wake of the motorboat that towed it, and that the drag was then affected by the incoming waves.

Later, Gomes et al. [16] measured the total resistance of Nelo Vanquish IV kayaks using a towing system placed on land. Three sizes of this model were used (M, ML and L), and each kayak size was tested with three differently weighted paddlers. The weight of the paddler was found to have more influence on drag force than the kayak size used.

Mantha et al. [17] simulated the flow around three L-sized Nelo Vanquish models (I, II and III) using the k-ω model. The results showed that, from the Vanquish I to the Vanquish II, viscous drag was larger, but pressure drag decreased, resulting in a lower total drag, while from the Vanquish II to the Vanquish III, viscous drag returned to lower values, once again decreasing total resistance force.

Tzabiras et al. [18] measured total resistance and trim and numerically determined total resistance and wave elevation using a RANS method and a potential flow method on an M-sized kayak provided by the Pan-Hellenic Kayak and Canoe Trainers Association. The RANS method was found to be accurate and the potential method was found to provide a reasonable qualitative estimative.

The results from the above studies are summarised in Figure 6.

Other aspects of kayaking have been studied scientifically. Baker [19] studied biomechanical effects related to paddler and paddle movement and propulsion force. Laurent et al. [20] performed CFD analyses on the flow around a paddle using RANS methods. Willman [10] presented numerical results with RANS methods, as well as experimental results, on the drag and lift forces on kayak rudders.

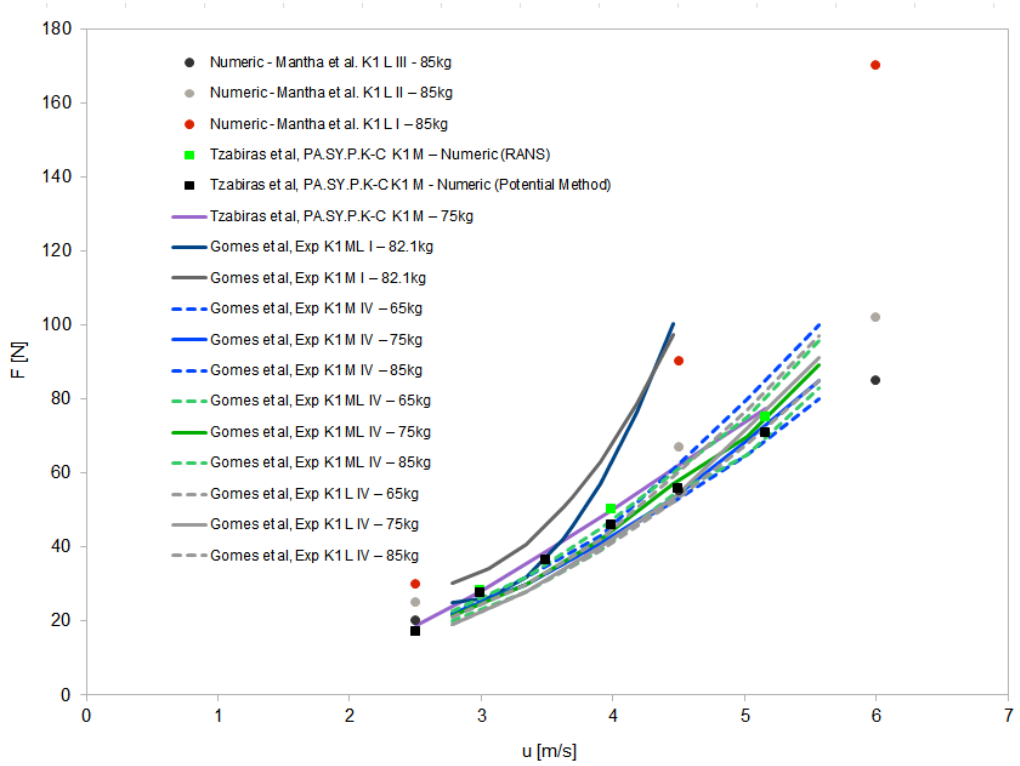

Figure 6 – Summary of experimental and numerical results on K1 sprint kayaks

1.7.2 Wigley hull

The Wigley hull is a hull shape described by a relatively simple mathematical expression whose movement on a free water surface has well known flow properties. As such, it is used as a validation case for numerical simulations involving boat hulls. The shape of a Wigley hull is the surface described by:

$$y = \frac{B}{2}\left[1 - \left(\frac{2x}{L}\right)^2\right]\left[1 - \left(\frac{z}{H}\right)^2\right] \tag{1.13}$$

The geometrical parameters L, B, and H are, respectively, the length, breadth and draught of the hull, i.e., its maximum dimensions in the x, y and z directions. They are usually chosen so as to verify a breadth-to-length ratio (B/L) of 0.1 and a draught-to-length ratio (H/L) of 0.0625 [21].

Many experimental results relating to the free surface flow of a Wigley hull were obtained by initiative of the 16th International Towing Tank Conference (ITTC) Resistance Committee from 1981 to 1983, in an attempt to provide comprehensive experimental data on a number of hull shapes to be used in the validation of numerical methods for hull flow calculations [22]. The data, obtained by member organisations of the ITTC, included results on total resistance, hull wave profile, hull pressure, among other measurements.

In the context of the aforementioned programme, four Japanese organisations (University of Tokyo, Ship Research Institute, Ishikawajima-Harima Heavy Industries Co., Ltd. and Yokohama National University) conducted separate experiments on Wigley hulls, and analysed and published them jointly [23]. Each of the organisations used different values for L, B and H, but all except the Yokohama National University used the B/L and H/L ratios disclosed above. The measured flow characteristics were total resistance, wave pattern resistance, wake survey data, hull wave profile, and hull pressure, for various Froude numbers. Different restrictions on the movement of the hull were also imposed, providing data for free sinkage and trim, fixed position (no sinkage or trim allowed), and fixed trim and free sinkage. Other ITTC member organisations also performed measurements on Wigley hulls, obtaining similar results (Figure 7; Figure 8) [22, 24].

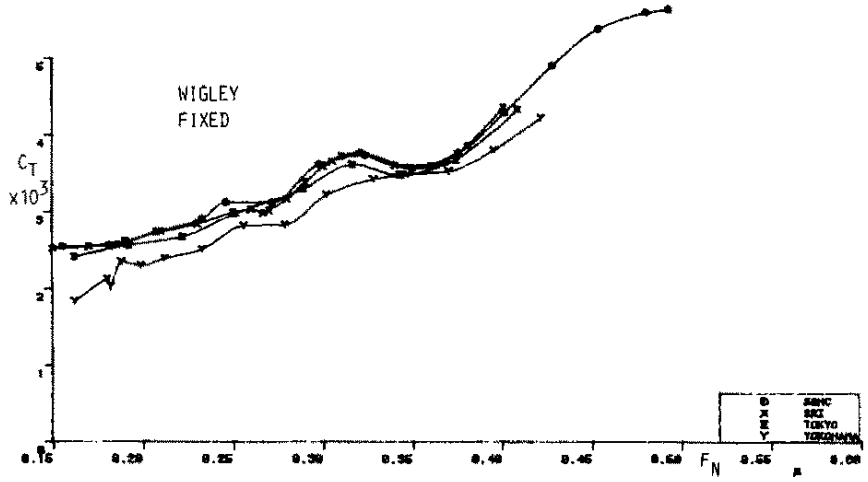

Figure 7 – Experimental results for total resistance on a fixed position Wigley hull in function of Froude number (from [22])

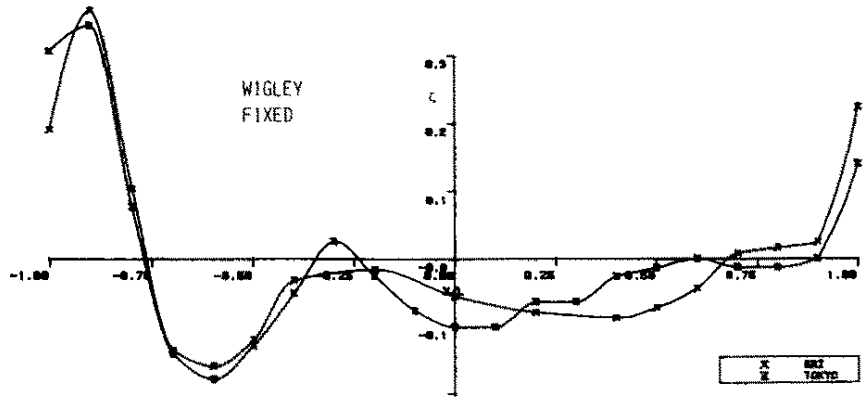

Figure 8 – Experimental results for the hull wave profile for a Wigley hull, for an unknown Froude number (from [22])

Several numerical simulations using different methods have since been made in the range of Froude numbers from 0.25 to 0.408, and the results reached were often close to the ones discussed above.

Perez et al. [25] used the k-ε and k-ω SST models to calculate total resistance and hull wave profile. The k-ε model was found to be more accurate for low Froude numbers and the k-ω SST was found to be more accurate for high Froude numbers. Both methods matched hull wave profile experimental results with equal accuracy, underestimating wave elevation in the front of the hull but being close to experimental values in the remaining length of the hull. Different mesh configurations and domain sizes were also tested.

Pranzitelli et al. [26] calculated resistance coefficients, hull wave profiles and wave elevation using the k-ω SST and Realisable k-ε (a variant of k-ε) models. The former overestimated the value of the total resistance by about 1%, while the latter underestimated it by about 2%. Results for hull wave profiles were nearly coincident with experimental results. Methods based on potential flow (i.e., assumed irrotational and incompressible) were also used, and shown to be much less accurate than the RANS methods mentioned above.

Inok et al. [27] used OpenFOAM software to implement the k-ω SST method. The simulations only included the flow on one of the sides of the ship, assuming the other side as symmetric. Wave elevation along the hull and in the rest of the free surface was evaluated. Results were more accurate for Froude numbers of 0.316 and 0.354. For lower Froude numbers, wave height was calculated to be below experimental values along the middle of the hull's length, while for higher Froude numbers, both the height of points with positive elevation and the depth of points with negative elevation were exaggerated.

Harpal and Patel [28] used the Realisable k-ε model to calculate resistance, hull wave profile, and heave and pitch motions. The error in resistance coefficient calculation was around 1% for most of the range of Froude numbers, becoming smaller for higher velocities, and reaching 0.05% for the highest Froude number simulated.

2 Governing equations

2.1 Continuity equation and Navier-Stokes equations

Any flow that involves only a Newtonian incompressible fluid can be fully described by the Navier-Stokes and continuity equations in the simplified forms presented below [8, pp. 307-308].

Continuity equation:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \tag{2.1}$$

Navier-Stokes equations:

$$\rho\left(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}\right)=-\frac{\partial p}{\partial x}+\rho g_x+\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right) \tag{2.2}$$

$$\rho\left(\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}\right)=-\frac{\partial p}{\partial y}+\rho g_y+\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}\right) \tag{2.3}$$

$$\rho\left(\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}\right)=-\frac{\partial p}{\partial z}+\rho g_z+\mu\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}\right) \tag{2.4}$$

2.2 Multiphase governing equations

The Volume of Fluid (VoF) method is a numerical method used for tracking the location of interfaces between fluids. As such, it can provide the full geometrical configuration of the free surface of the water.

The VoF method tracks the value of a variable α which corresponds to the volume fraction of one of the fluids in each cell. So, if the fluid being tracked is water, the value of α will be equal to 1 in cells completely filled with water and equal to zero in cells that do not contain any water, taking intermediate values for cells intersecting the free surface [29].

Fluid properties such as density and viscosity are calculated as a weighted average of the different fluids' properties according to the volume fraction. So, for a case with only two fluids, 1 and 2, where α is the volume fraction of 1, the density ρ and kinematic viscosity ν will be given by [30]:

$$\rho=\alpha\rho_1+(1-\alpha)\rho_2 \tag{2.5}$$

$$\nu=\alpha\nu_1+(1-\alpha)\nu_2 \tag{2.6}$$

The value of α for each point in the domain can typically be calculated from previous values of α and the velocity field. There are many methods that can be used to calculate the volume fraction of a fluid for a given time instant. The method used in OpenFOAM's *interFoam* solver decomposes the velocity vector field $\boldsymbol{U}$ in the same way as density and viscosity are decomposed, and creates an auxiliary vector field $\boldsymbol{U_r}$ that is used in the calculation of the α field [31]:

$$\boldsymbol{U} = \alpha \boldsymbol{U_1} + (1 - \alpha)\boldsymbol{U_2} \tag{2.7}$$

$$\boldsymbol{U_r} = \boldsymbol{U_1} - \boldsymbol{U_2} \tag{2.8}$$

The volume fraction of phase 1 is then given by:

$$\frac{\partial \alpha}{\partial t} + \boldsymbol{\nabla} \cdot (\alpha \boldsymbol{U}) + \boldsymbol{\nabla} \cdot [\alpha(1 - \alpha)\boldsymbol{U_r}] = 0 \tag{2.9}$$

2.3 Turbulence modelling

2.3.1 The purpose of turbulence modelling

Turbulent flows are typically hard to describe with detail, due to the fact that turbulence is a very complex phenomenon [32]. It is always three-dimensional, it never reaches a steady state, and it has implications on a very wide range of time and length scales. Therefore, fully characterising a turbulent flow requires very high computational power and very accurate numerical methods.

On the other hand, CFD analyses do not usually aim to obtain a complete knowledge of all the flow properties. Some of the more complex features of the flow can be replaced by adequate simplifications that allow for a solution that is manageable while being accurate enough. The implementation of these simplifications is known as turbulence modelling.

The main types of turbulence models, in order of increasing complexity, are Reynolds-averaged Navier-Stokes equations (RANS) models, Detached Eddy Simulation (DES), Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS). The more complex a model is, the more accurate it is, but it will also require higher computational power.

RANS models substitute all turbulent effects by estimative time-averages. LES, on the other hand, only estimates turbulent effects that occur under a certain length scale, while calculating exact solutions for turbulent phenomena occurring at larger scales. DES is a hybrid of RANS and LES; it uses RANS methods near walls, where LES is typically less accurate, and LES in the rest of the domain. DNS, corresponding to the highest possible level of complexity, is the computation of flow properties without resorting to modelling, using the Navier-Stokes equations in their unaltered form to simulate turbulent effects in all length and time scales.

2.3.2 Reynolds averaging

Reynolds averaging is a specific mathematical formulation of a velocity field that allows for convenient modelling of turbulent flow [32]. In Reynolds averaging, each component U_i of the velocity vector $\boldsymbol{U}$ is expressed as a sum of a mean velocity $\overline{U}_i$ and a fluctuating velocity u_i':

$$U_i = \overline{U}_i + u_i' \tag{2.10}$$

The mean velocity is typically obtained from a time average using a convenient time scale that is large enough to cover the effects of turbulence and small enough to accurately reflect variations of velocity that do not occur due to turbulence.

If the continuity equation and the Navier-Stokes equations are rewritten using equation (2.10) and the same time-averaging operation is applied on them, the Reynolds-averaged forms of these equations can be obtained. For an incompressible fluid, they are:

$$\frac{\partial \overline{U}_i}{\partial x_i} = 0 \tag{2.11}$$

$$\frac{\partial}{\partial x_j}\left(\rho \overline{U}_i \overline{U}_j\right) = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left(2\mu S_{ij} - \overline{\rho u_i' u_j'}\right) \tag{2.12}$$

In equation (2.12), the term S_{ij} is the mean strain rate tensor, given by:

$$S_{ij} = \frac{1}{2}\left(\frac{\partial \overline{U}_i}{\partial x_j} + \frac{\partial \overline{U}_j}{\partial x_i}\right) \tag{2.13}$$

The $-\overline{\rho u_i' u_j'}$ terms from the Reynolds-averaged Navier-Stokes equations, known as the Reynolds-stress tensor τ_{ij}, cannot be exactly determined, since turbulent effects are eliminated in the averaging process. The difference between the several methods based on RANS lies essentially in the way they estimate the Reynolds stresses.

Many RANS models use the Boussinesq assumption, which provides an approximation to the Reynolds stresses by introducing the concept of eddy viscosity (μ_T):

$$\tau_{ij} = 2\mu_T S_{ij} - \frac{2}{3}\rho k \delta_{ij} \tag{2.14}$$

In the above equation, δ_{ij} is the Kronecker delta function, which equals 1 if i and j are the equal and 0 if they are different, and k is the kinetic energy per unit mass associated to the turbulent fluctuations of velocity:

$$k = \frac{1}{2}\left(\overline{u'^2} + \overline{v'^2} + \overline{w'^2}\right) \tag{2.15}$$

Equation (2.12) can then, taking into account the Boussinesq assumption, be written as:

$$\frac{\partial}{\partial x_j}\left(\rho \bar{U}_i \bar{U}_j\right) = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left[2(\mu + \mu_T)S_{ij} - \frac{2}{3}\rho k \delta_{ij}\right] \tag{2.16}$$

It can be understood from equation (2.16) that RANS models using the Boussinesq assumption are concerned with estimating μ_T and k, that is, the eddy viscosity and the kinetic energy due to turbulence.

The solutions used by some of these models will be detailed in the next subsection. The models discussed are two-equation models, i.e., they rely on the use of two differential equations to achieve its estimates. One of these equations involves the turbulence kinetic energy, while the other is concerned with an auxiliary variable related to turbulence energy dissipation. Closure coefficients, which are constants used in these equations whose values are empirically obtained, must also be provided.

2.3.3 The k-ε model

The particular form of the k-ε model discussed in this section is the one formulated by Jones and Launder in 1972 [33], known as the Standard k-ε model. As briefly mentioned in the previous subsection, it uses an equation corresponding to the energy balance of the turbulence kinetic energy, k, as presented below:

$$\rho \frac{\partial k}{\partial t} + \rho \bar{U}_J \frac{\partial k}{\partial x_j} = \tau_{ij} \frac{\partial \bar{U}_\iota}{\partial x_j} - \rho \varepsilon + \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_T}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] \tag{2.17}$$

This equation involves the variable ε, which is the rate of turbulence energy dissipation per unit mass. It can be described by the equation:

$$\rho \frac{\partial \varepsilon}{\partial t} + \rho \bar{U}_J \frac{\partial \varepsilon}{\partial x_j} = C_{\varepsilon 1} \frac{\varepsilon}{k} \tau_{ij} \frac{\partial \bar{U}_\iota}{\partial x_j} - C_{\varepsilon 2} \rho \frac{\varepsilon^2}{k} + \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_T}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] \tag{2.18}$$

The eddy viscosity, in the context of this model, is given by:

$$\mu_T = \frac{\rho C_\mu k^2}{\varepsilon} \tag{2.19}$$

The most widely used values for the closure coefficients introduced in equations (2.17), (2.18) and (2.19) are the ones arrived at by Launder and Sharma in 1974 [34], which are the following:

$$C_{\varepsilon 1} = 1.44;\ C_{\varepsilon 2} = 1.92;\ C_\mu = 0.09;\ \sigma_k = 1.0;\ \sigma_\varepsilon = 1.3$$

2.3.4 The k-ω model

In the k-ω model, the dissipation of turbulence kinetic energy is expressed not in terms of a dissipation rate, but in terms of a characteristic frequency of the dissipation process, ω, which is derived from dimensional analysis and can have several different interpretations for its physical meaning. The following equations correspond to the k-ω model presented by Wilcox in 1988 [35].

The k equation is a slight variation of equation (2.17):

$$\rho\frac{\partial k}{\partial t}+\rho\bar{U}_J\frac{\partial k}{\partial x_j}=\tau_{ij}\frac{\partial \bar{U}_\iota}{\partial x_j}-\beta^*\rho k\omega+\frac{\partial}{\partial x_j}\left[(\mu+\sigma^*\mu_T)\frac{\partial k}{\partial x_j}\right] \tag{2.20}$$

The ω equation is given as:

$$\rho\frac{\partial \omega}{\partial t}+\rho\bar{U}_J\frac{\partial \omega}{\partial x_j}=\alpha\frac{\omega}{k}\tau_{ij}\frac{\partial \bar{U}_\iota}{\partial x_j}-\beta\rho\omega^2+\frac{\partial}{\partial x_j}\left[(\mu+\sigma\mu_T)\frac{\partial \omega}{\partial x_j}\right] \tag{2.21}$$

The eddy viscosity is calculated by the expression:

$$\mu_T=\frac{\rho k}{\omega} \tag{2.22}$$

Finally, the values proposed by Wilcox [35] for the closure coefficients introduced in equations (2.20), (2.21) and (2.22) are the following:

$$\alpha=\frac{5}{9};\ \beta=\frac{3}{40};\ \beta^*=\frac{9}{100};\ \sigma=\frac{1}{2};\ \sigma^*=\frac{1}{2}$$

2.3.5 The k-ω Shear-Stress Transport (SST) model

The k-ω SST model, developed by Menter in 1994 [36], intends to blend the k-ε and k-ω models in such a way that it overcomes each of the methods' main weaknesses: the k-ε model is not as accurate as desirable in near-wall zones, while the k-ω model is inaccurate when dealing with high pressure gradients and too sensitive to the freestream values of ω [37, 38].

The k-ω SST model combines the two by using a blending function that allows for the flow near walls to be modelled in a similar manner to the k-ω model, and the flow further away from walls to be modelled more like in the k-ε model.

The k equation is the following:

$$\rho\frac{\partial k}{\partial t}+\rho\bar{U}_J\frac{\partial k}{\partial x_j}=\tau_{ij}\frac{\partial \bar{U}_\iota}{\partial x_j}-\beta^*\rho k\omega+\frac{\partial}{\partial x_j}\left[(\mu+\sigma_k\mu_T)\frac{\partial k}{\partial x_j}\right] \tag{2.23}$$

The ω equation features the aforementioned blending function F_1:

$$\rho\frac{\partial\omega}{\partial t}+\rho\bar{U}_j\frac{\partial\omega}{\partial x_j}=\alpha\frac{\omega}{k}\tau_{ij}\frac{\partial\bar{U}_i}{\partial x_j}-\beta\rho\omega^2+\frac{\partial}{\partial x_j}\left[(\mu+\sigma_\omega\mu_T)\frac{\partial\omega}{\partial x_j}\right]+2(1-F_1)\frac{\rho\sigma_{\omega2}}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial\omega}{\partial x_j} \tag{2.24}$$

The blending function F_1 equals zero away from solid walls and one near walls, allowing the model to behave like a k-ε model or like a k-ω model accordingly. This function is given by:

$$F_1=\tanh\left[\left\{\min\left[\max\left(\frac{\sqrt{k}}{\beta^*\omega y},\frac{500\nu}{y^2\omega}\right),\frac{4\rho\sigma_{\omega2}k}{CD_{k\omega}y^2}\right]\right\}^4\right] \tag{2.25}$$

y is the distance to the nearest wall and $CD_{k\omega}=\max\left(2\frac{\rho\sigma_{\omega2}}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial\omega}{\partial x_j},10^{-10}\right)$.

The expression for the eddy viscosity features a second blending function F_2:

$$\mu_T=\frac{\rho a_1 k}{\max(a_1\omega,\Omega F_2)} \tag{2.26}$$

$$F_2=\tanh\left[\left[\max\left(\frac{2\sqrt{k}}{\beta^*\omega y},\frac{500\nu}{y^2\omega}\right)\right]^2\right] \tag{2.27}$$

Ω is equal to $\sqrt{2W_{ij}W_{ij}}$, where $W_{ij}=\frac{1}{2}\left(\frac{\partial\bar{U}_i}{\partial x_j}-\frac{\partial\bar{U}_j}{\partial x_i}\right)$.

Every closure coefficient φ in the k-ω SST model has two values: one of those is used near walls (φ_1) and the other away from walls (φ_2). The value taken by the closure coefficients is mediated by the F_1function:

$$\varphi=F_1\varphi_1+(1-F_1)\varphi_2 \tag{2.28}$$

The closure coefficients used are:

$$\beta^*=0.09,\alpha_1=\frac{5}{9},\beta_1=\frac{3}{40},\sigma_{k1}=0.85,\sigma_{\omega1}=0.5,\alpha_2=0.44,\beta_2=0.0828,\sigma_{k2}=1,$$
$$\sigma_{\omega2}=0.856,a_1=0.31$$

Some basic alterations to this model that can be implemented include the replacement of Ω by $S=\sqrt{2S_{ij}S_{ij}}$ in the calculation of the eddy viscosity, and the use of an upper limit to the term $\tau_{ij}\frac{\partial\bar{U}_i}{\partial x_j}$ in the k equation.

3 Software description

3.1 Introduction

The numerical simulations conducted for this project were obtained with the software package OpenFOAM, a collection of open-source C++ libraries and applications for CFD analysis with the Finite Volume Method. Its applications include solvers, which solve flow equations for a specific type of flow, as well as utilities for pre-processing (e.g. mesh generation) and post-processing (e.g. force calculation) [39, 40].

The data for an OpenFOAM simulation are not contained in a single file, instead being in the form of several files organised in a directory. Each OpenFOAM case directory initially contains three subdirectories:

- *constant*, whose files contain the fixed parameters of the simulation, including values of fluid properties, settings for turbulence models, settings for mesh generation, and the configuration of the mesh after it is generated;
- *system*, whose files are related to the execution of applications, including settings for the solver used, equation discretisation and solution methods, and mesh refinement settings;
- A subdirectory named after the initial time instant (typically *0*), which contains files named after every field (variable calculated by the solver), each including the initial conditions and boundary conditions applicable to that field.

As a simulation runs, the solver will periodically create subdirectories named after time instants, containing the value of each field for each cell in that instant, as well as the boundary conditions initially defined. In that way, if a simulation is stopped, it can be continued by specifying the latest recorded time instant as the initial instant.

Visualisation of results can be made using ParaView, an auxiliary post-processing program provided with every installation of OpenFOAM, which reads the subdirectories for the various time instants and presents the results visually.

3.2 Mesh generation

OpenFOAM includes two main mesh generation applications, *blockMesh* and *snappyHexMesh* [40].

The *blockMesh* utility allows for the generation of a mesh consisting of blocks, defined by up to eight vertices, that contain hexahedral mesh elements. It is only suited for geometries that can be easily achieved with a series of simple blocks.

Each block is assigned its own local coordinate system, and the user can define how many cells the block will be divided into along each coordinate. The implementation of

mesh grading, which makes the size of cells in a certain direction become progressively smaller or larger along that direction, is also an option.

The *snappyHexMesh* utility is used for the definition of more complex geometries. It presupposes the previous existence of a mesh created with *blockMesh* (known as the background mesh), as well as files in the STL format containing the shape of the geometric features to be implemented. *snappyHexMesh* adapts the existing mesh to the new geometry and refines the region around it (i.e., generates smaller-sized cells in the region). It can also be used to refine other regions of the domain.

The *snappyHexMesh* application operates in three steps. First, it removes the cells located inside the STL surface and splits the cells intersecting it. It can perform the splitting operation several times, as defined by the user. Its second operation is the displacement of points near the STL surface towards the surface itself, making the mesh non-hexahedral in this region. Finally, cells are added on the surface in order to achieve higher accuracy in the flow near walls.

The *snappyHexMeshDict* file, which allows for the control of parameters relating to *snappyHexMesh*, also allows the user to define simple regions within the domain, such as boxes or spheres, that can also suffer a local refinement. The file presents many control options related to each one of the three stages mentioned above, including, most importantly for this project, the definition of cell refinement levels.

The cell level of a region is the number of times the cells in that region are split. When splitting occurs, cells are split halfway through each of their edges. Therefore, since cells are three-dimensional, one cell will be split into eight different cells at a time. Cell level is defined separately for each refined region. When the levels of two adjacent areas differ by more than 1, *snappyHexMesh* also refines a region around the higher level area, and this operation is repeated until every cell in the mesh is only adjacent to cells whose refinement is either one level lower, one level higher, or the same.

3.3 Discretisation and solution of equations

As seen in section 2, flows are governed by a number of partial differential equations (PDE) depending on spatial dimensions and time. In CFD analysis, the continuous nature of space and time is substituted by discrete approximations; the values of variables are only known for specific time instants and, if the Finite Volume Method is used, in the centres of cells in the mesh. Therefore, setting up an OpenFOAM simulation requires the user to arbitrate which mathematical models will be used to approximate the different PDE terms for discrete spatial and temporal domains.

The *fvSchemes* file is the one concerned with PDE discretisation. It requires discretisation schemes to be specified for derivatives with respect to time, gradients, divergences, laplacians, surface-normal gradients, and interpolations for calculations of

values outside cell centres. Specific methods must also be specified for the solution of algebraic equations, in the *fvSolution* file.

3.4 Boundary conditions

In the *blockMeshDict* file, apart from specifying the information needed to generate the background mesh, the user can divide the geometrical limits of the domain into several regions, or patches. Boundary conditions on patches must be imposed in order to close the equation system governing the flow, and the initial values for each field must be provided to start the simulation.

In the subdirectory corresponding to the initial time instant, as has been mentioned before, there is a file corresponding to each field calculated by the solver. In each of these files, an initial condition and a boundary condition must be specified for each patch, as well as an initial value for all the cells within the domain. When the initial values for a field are not uniform, the OpenFOAM application *setFields* can be used. *setFields* writes the value for each individual cell in the field file by following simple instructions given by the user.

The basic boundary condition types are *fixedValue*, which imposes a value on a patch for all time instants, and *zeroGradient*, which sets field values on boundaries to the value of the adjacent cell centre (which is equivalent to setting the patch-normal component of the field gradient to zero).

3.5 Simulation control

The execution of a solver follows parameters that must be specified in a file in the *system* subdirectory called *controlDict*. These control settings include the time instants for which the simulation starts and ends, the time step between consecutive simulated instants, the time intervals at which field values are recorded, and the precision these values are recorded with. *controlDict* also indicates which post-processing utilities are to be used during the run, and the settings for these utilities.

The value of the time step is of particular importance to simulation control since its choice can influence whether or not the field variables converge to an appropriate value. A useful parameter for this purpose is the Courant number (Co) of a cell, calculated from the magnitude of the velocity in that cell $|\boldsymbol{U}|$, the time step δt, and the cell size in the direction of the velocity δx:

$$Co = \frac{\delta t |\boldsymbol{U}|}{\delta x} \tag{3.1}$$

A Courant number of less than 1 throughout the domain is required to achieve numerical stability, and appropriate convergence of the solution is sometimes only achieved with lower limits for this parameter. It is possible to run simulations on an adjustable time step, i.e. instead of a fixed time step being imposed, a new time step is

calculated at each instant in such a way that the maximum Courant number in the mesh does not exceed an imposed maximum value.

4 Simulation setup

4.1 General information

The CFD simulations in this project were run with OpenFOAM version 2.1.1, using the *interFoam* solver, which is suited for two-phase flows with incompressible, immiscible and isothermal fluids, and uses the Volume of Fluid method (see section 2.2) for interface tracing [39]. The turbulence model used was k-ω SST, which has been found to be more accurate for higher Froude numbers [25], and produced good results in the studies conducted by Pranzitelli et al. [26] and Inok et al. [27].

Firstly, to validate the numerical methods, simulations were run for the free surface flow around a Wigley hull (see section 1.7.2), for a range of experimentally tested velocities and for different mesh configurations and refinements. Then, the flow around the Vanquish III L model was simulated for several velocities, mesh refinements, and waterline levels. A small number of simulations for different velocities was also performed for the Vanquish IV M model.

The simulation cases were initially adapted from the Wigley hull tutorial provided as an example case for the *interFoam* solver. In each of the cases, the hull surface was placed on a fixed position in the spatial domain and the initial distribution of water and air was defined. A fixed freestream velocity value in the opposite direction of the forward movement of the hull was imposed on the fluids.

The variables calculated by *interFoam* for every cell centre were:

- The velocity vector $\boldsymbol{U}$;
- The dynamic pressure p_{rgh}, given by $p_{rgh} = p - \rho g h$, where h is the vertical coordinate of the point in question;
- The water volume fraction α_1, used in the VoF method;
- The variables related to the turbulence model:
    - Kinematic eddy viscosity ν_t, given by $\nu_t = \mu_t / \rho$;
    - Turbulence kinetic energy k;
    - Turbulence kinetic energy dissipation frequency ω.

The coordinate system used was consistent with the way the Wigley hull shape is defined in equation (1.13). The x axis points in the direction opposite to the hull's movement, or, in the case of these simulations, where the hull is fixed and the water flows against it, the direction of the freestream velocity. The y axis points to the right side of the hull, and the z axis is a vertical axis pointing upward (Figure 9).

Copies of the files that specify boundary conditions and discretisation and solution schemes are included in Appendix A and Appendix B.

Every simulation ran on a similar mesh with the hull scaled down so that its length was set to 1 m, and other flow characteristics were adjusted accordingly when necessary.

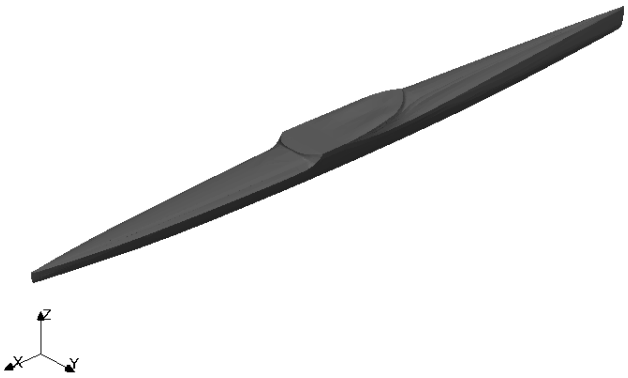

Figure 9 – Coordinate system used for all the simulations conducted, relative to the Vanquish III kayak

4.2 Mesh configuration

The background mesh used for the simulations consists of a single block, and the range of the domain was similar among the various cases:

- $x = -1.5$ m to $x = 2.5$ m;
- $y = -1.5$ m to $y = 1.5$ m;
- $z = -0.5$ m to $z = 0.0399$ m or $z = -0.5$ m to $z = 0.31$ m, depending on the hull model used.

The origin of the coordinate system was placed so that $x = 0$ halfway through the hull's length, $y = 0$ halfway through the hull's breadth, and $z = 0$ at the initial water surface level.

Four patches were defined on the boundary regions of the block:

- The *inlet* patch, upstream from the hull, where the water and air flow into the domain;
- The *outlet* patch, downstream from the hull, where the water and air flow out of the domain;
- The *atmosphere* patch, which is the upper limit of the domain;
- The *sides* patch, which includes the lateral and lower limits of the domain.

The *snappyHexMesh* application was used after the generation of the background mesh, not only to insert the hull geometry into the domain (using an STL model of the hull) and refine the cells near it, but also to define zones where certain cell refinement levels were performed. Three box-shaped zones were defined for refinement (z_{max} refers to the maximum value of the z coordinate in the mesh):

- *box1*, defined by corners (-1.5, -0.5, -0.25) and $(2, 0.5, z_{max})$

- *box2*, defined by corners (-1.5, -0.2, -0.15) and $(1.5, 0.2, z_{max})$
- *box3*, consisting of every cell where $z >$ -0.3

Adjustments in the position of the hull were made by implementing the necessary changes in the STL models, rather than in the meshing applications.

An example of the full geometrical configuration of the domain is presented in Figures 10 to 13, and the position of the patches is shown in Figure 14.

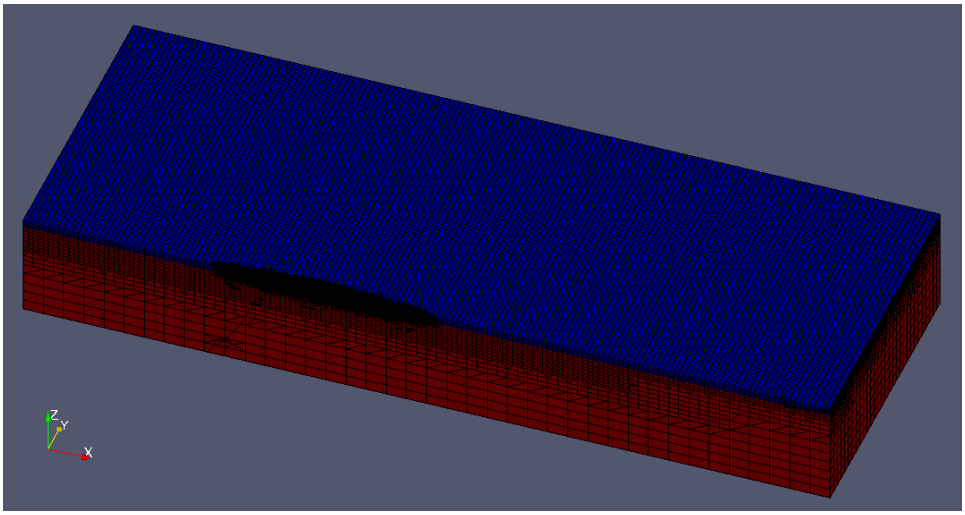

Figure 10 – Mesh used in simulations with cell outlines highlighted (sectioned at the y = 0 plane)

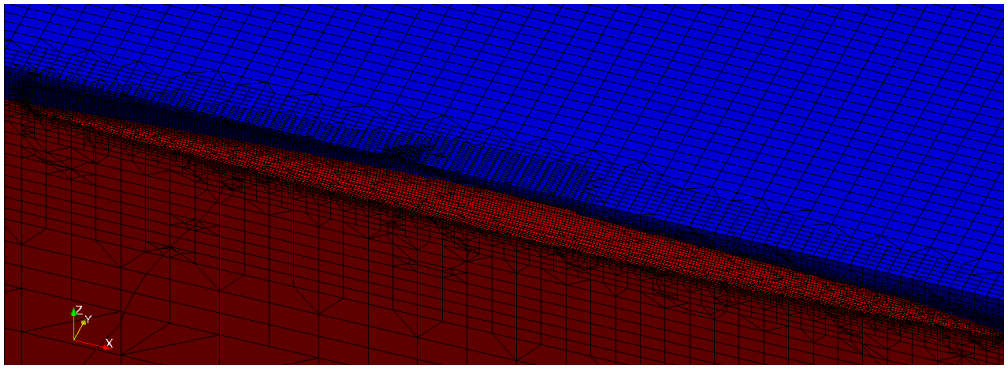

Figure 11 – Mesh near the hull with cell outlines highlighted (sectioned at the y = 0 plane)

Figure 12 – Domain configuration in the y = 0 plane; darker colours represent higher refinement

Figure 13 – Domain configuration in the x = 0 plane; darker colours represent higher refinement

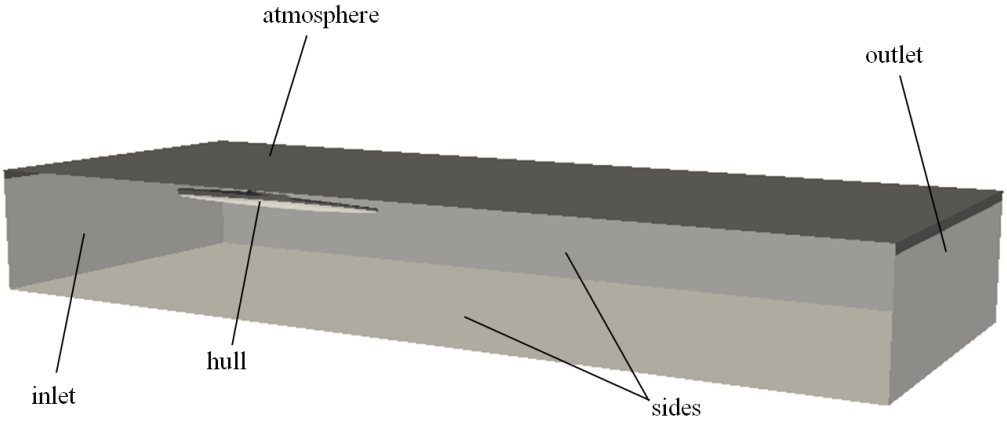

Figure 14 – Patches defined from the boundaries of the domain

4.3 Boundary conditions

The boundary conditions imposed were the same for all simulation cases and are represented in Figure 14. A comprehensive disclosure of the boundary conditions for all fields is included in Appendix A. This section is only intended to provide a general insight on the conditions whose implementation contributed greatly to the way the simulation cases were defined.

The *sides* patch was defined as a *symmetryPlane* patch, which establishes all field values as if the flow is reflected on its surface. This condition was kept from the original Wigley hull example case since it ensures that there is no mass flow across the *sides* patch, thus preventing the water level from decreasing progressively. The domain was made broader in the *y* direction than the one from the Wigley hull example, so that the surface wave pattern near the hull would not be affected by wave reflections on the sides.

The initial distribution of water and air throughout the domain was set using the *setFields* utility to write the values of the α_1 field in the appropriate boundary condition file. The value of α is initially imposed as 0 (i.e. completely empty of water) for the whole domain, and *setFields* sets it to 1 (i.e. completely filled with water) for all cells where $z \leq 0$. Mass flow into the domain through the atmosphere patch was set to carry an α_1 value of 0.

The velocity field in the *inlet* patch was set to the value of the intended freestream velocity in the positive x direction. The same value was imposed on the whole domain for the initial time instant. The velocity vector was set to zero on the hull surface.

The dynamic pressure was set to zero on the *outlet* patch and the *atmosphere* patch, which is equivalent to setting the pressure field in these patches to the value of the hydrostatic pressure.

The boundary conditions for turbulence parameters ν_t, k and ω correspond to either boundary conditions developed specifically for these fields, or to recommended freestream boundary conditions for the turbulence model [38].

4.4 Control parameters

All cases were set to run a sufficient amount of time for the simulation to converge. The default end time used was 60 seconds, although some of the simulations were stopped before the 60 second limit when their high computing time was limitative to the execution of other simulations and the results had already converged.

An adjustable time step was implemented, with a maximum Courant number of 0.3 for all cells, and a maximum time step of 1 second.

A script for calculating forces on the hull surface using OpenFOAM's *forces* function was added to the *controlDict* file. The x, y and z components of the pressure and viscous forces and the pressure and viscous moments were recorded for every time instant simulated.

4.5 Post-processing

The three-dimensional decomposition of viscous and pressure forces and moments is calculated and output by the *forces* function. Given the pressure force in the x direction F_{px} and the viscous force in the x direction F_{vx}, the pressure drag coefficient C_p, the viscous drag coefficient C_v and the total drag coefficient C_t can be calculated for each time instant:

$$C_p = \frac{F_{px}}{\frac{1}{2}\rho S U^2} \tag{4.1}$$

$$C_v = \frac{F_{vx}}{\frac{1}{2}\rho S U^2} \tag{4.2}$$

$$C_t = \frac{F_{px} + F_{vx}}{\frac{1}{2}\rho S U^2} = C_p + C_v \tag{4.3}$$

The values of these coefficients will converge to a certain value, starting to oscillate around that fixed value. The final calculated values for the coefficients were time-averages taken from the range of time instants where the values show this oscillating behaviour.

The wave hull profiles were obtained with the ParaView application, by obtaining the coordinates of points on the hull where $\alpha_1 = 0.5$, and plotting the non-dimensional wave elevation z/L against the non-dimensional hull length coordinate x/L. Two sets of values are obtained, for the right and left sides of the hull.

4.6 Wigley hull

4.6.1 Geometric models

An STL-format model of a Wigley hull (Figure 15) is included in the Wigley hull example case provided with OpenFOAM. Its dimensions (see equation (1.13)) are $L = 1$ m, $B = 0.1$ m, $H = 0.0625$ m.

In initial simulations, difficulties arose due to the fact that wave elevation sometimes crossed the upper limit of the domain in some regions near the hull. Following this, the mesh was extended in its height from 0.0399 to 0.31 metres. The Wigley hull model, being an open surface, had to be closed, so that the meshing utilities would not create cells on its inside, and made higher, so that the waves would not move to the top of the hull.

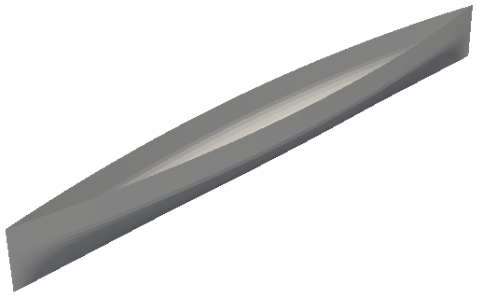

Figure 15 – Wigley hull model

The simplest solution found was to place a rotated duplicate of the hull on top of the existent one, joining them in the same STL file, as seen in Figure 16. Using this model implies the assumption that the contribution of aerodynamic forces on the total force on the hull is negligible, which is reasonable due to the much lower density and viscosity of air when compared to water. This adjusted model was the one used in all Wigley hull simulations.

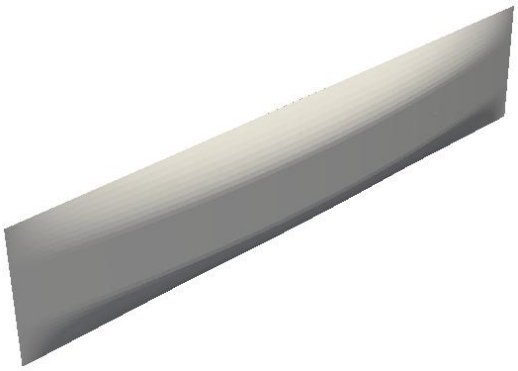

Figure 16 – Wigley hull model adjusted to withstand higher wave elevations

4.6.2 Velocity values

Wigley hull simulations were run for six different freestream velocity values. Four of those values correspond to the Froude numbers for which there were experimental results for fixed position hull wave profiles from the Ship Research Institute [23]. The remaining two values were chosen so as to cover the full range of Froude numbers for which total resistance was experimentally obtained.

The Froude numbers used are presented in Table 1, along with the corresponding velocities for a hull length of 1 m.

4.6.3 Simulation cases

The initial background mesh had 40 cells in the x direction, 60 cells in the y direction and 15 cells in the z direction, and was refined to level 1 in the *box1* region, level 2 in the *box2* and *box3* regions, and level 3 around the hull. All velocities in Table 1 were simulated using this mesh.

Table 1 – Froude numbers used in Wigley hull simulations and corresponding velocities

Froude number	Velocity (m/s)
0.250	0.783
0.267	0.836
0.289	0.905
0.316	0.990
0.350	1.096
0.400	1.253

For the remaining cases, only the velocities corresponding to $Fr = 0.316$ and $Fr = 0.400$ were analysed.

The aforementioned initial mesh was then refined so that the background mesh had 80 cells in the *x* direction, 120 cells in the *y* direction and 30 cells in the *z* direction. The mesh was refined to level 2 in the *box1* region, level 3 in the *box2* and *box3* regions, and level 3 or 4 around the hull, depending on the hull geometry on each particular cell.

The same case was run using only the region of the domain where $y > 0$, imposing a symmetry boundary condition on the $y = 0$ plane. This way, only the flow around the right half of the hull was simulated.

Local mesh refinement around the hull was also implemented on the initial unrefined mesh. Simulations were conducted where the cell level in this region was changed to 4 and 5.

Table 2 systematises the main features of every Wigley hull simulation run.

Table 2 – Summary of the simulations performed on the Wigley hull

Mesh refinement	**Number of cells in background mesh**			**Refinement level**			
	x	*y*	*z*	*box1*	*box2*	*box3*	hull
No refinement	40	60	15	1	2	2	3
Total refinement	80	120	30	2	3	3	3 – 4
Total refinement with symmetry condition	80	60	30	2	3	3	3 – 4
Hull refinement (level 4)	40	60	15	1	2	2	4
Hull refinement (level 5)	40	60	15	1	2	2	5

4.7 Vanquish III kayak

4.7.1 Geometric model

The external shape of the L-sized Vanquish III kayak had already been surveyed for a previous project, so an STL model was available. As was previously mentioned, the model, originally 5.2 m long, was scaled down so that its length became 1 metre.

Figure 17 – Vanquish III kayak model

4.7.2 Waterline level

The location of the waterline on the surface of a hull should be such that the buoyant force balances the weight of the vessel. This happens when the mass of the displaced water volume V_d is the same as the mass of the vessel m:

$$m = \rho V_d \tag{4.4}$$

In this case, the mass of the vessel is the mass of the kayak, assumed to be the maximum possible value of 12 kg [41], added to the mass of the paddler, assumed to be 75 kg, a total of 87 kg. For a water density of 1000 kg/m^3, the displaced water volume would then be 0.087 m^3. Using ParaView to determine the plane below which the volume inside the hull is 0.087 m^3, the waterline position was found to be 122 mm above the bottom of the hull.

However, footage of towing tests and measurements from other kayak models seem to place the waterline above this level. Mantha et al. [17], through an undisclosed method, also arrived at a higher waterline level than the one expected for the weight assumed.

Considering these two contrasting accounts, simulations were conducted for two distinct waterline levels: 122 mm above the hull, as theoretically derived, and 150 mm above the hull, estimated from towing test footage.

Near the end of the time devoted to this study, the opportunity arose to measure the height of the waterline experimentally on a Vanquish III kayak, and for a paddler weight of 78 kg, the waterline was measured to be approximately 123 mm above the lowest point on the hull, which is close to the one found by applying equation (4.4).

4.7.3 Velocity values and model scaling

The range of velocities tested was made to match the range of measured and extrapolated values by Gomes et al. [15], 10 to 20 km/h (or 2.78 to 5.56 m/s). Most other known numerical and experimental results are approximately in this range as well, since the typical speeds of sprint and marathon kayaks are between these values.

Since the kayak is scaled down for the simulations, the velocity values must likewise be scaled down, in order not to change the value of the respective Froude numbers:

$$\frac{U_m}{\sqrt{gL_m}} = \frac{U_r}{\sqrt{gL_r}} \tag{4.5}$$

In the above equation, the subscript m is applied to properties of the model and the subscript r is applied to properties of the real kayak hull. Following equation (4.5), the scaled velocity value can be expressed as a function of the ratio between lengths:

$$U_m = \left(\frac{L_m}{L_r}\right)^{1/2} U_r \tag{4.6}$$

If L_m and L_r are substituted by their values, the scale factor between velocities will be known:

$$U_m = \left(\frac{1}{5.2}\right)^{1/2} U_r = 0.4385 U_r \tag{4.7}$$

The Reynolds number must also be kept constant after scaling:

$$\frac{U_m L_m}{\nu_m} = \frac{U_r L_r}{\nu_r} \tag{4.8}$$

As the ratio between velocity and length cannot be changed without altering the Froude number, the kinematic viscosity of the fluids must be altered:

$$\nu_m = \frac{U_m L_m}{U_r L_r} \nu_r \tag{4.9}$$

Combining equations (4.6) and (4.9), the viscosity ratio can be expressed as a function of the length ratio alone:

$$\nu_m = \left(\frac{L_m}{L_r}\right)^{3/2} \nu_r \tag{4.10}$$

Substituting the appropriate length values yields:

$$\nu_m = \left(\frac{1}{5.2}\right)^{3/2} \nu_r = 0.0843 \nu_r \tag{4.11}$$

The velocities simulated for cases involving the Vanquish III are presented in Table 3. The kinematic viscosity of water was changed from 1×10^{-6} to 8.43×10^{-8} m^2/s, and the kinematic viscosity of air was changed from 1.48×10^{-5} to 1.25×10^{-6} m^2/s.

For the calculation of the drag coefficients, the initial wetted area of the model is needed:

$$S_m = \left(\frac{L_m}{L_r}\right)^2 S_r = \left(\frac{1}{5.2}\right)^2 S_r = 0.0370 S_r \tag{4.12}$$

The values of forces were taken from the obtained drag coefficients using the reverse method from the one presented in equations (4.9), (4.10) and (4.11).

Table 3 – Real and model velocity values used in Vanquish III simulations

Real velocity (m/s)	Model velocity (m/s)	Froude number
2.78	1.219	0.389
3.69	1.618	0.517
4.60	2.017	0.644
5.50	2.412	0.770

4.7.4 Mesh properties

Unlike in the Wigley hull simulations, where several different mesh configurations were used, the only difference between the meshes used for the various kayak simulations was the refinement level near the hull surface.

The meshes used in these simulations had the maximum z coordinate set to 0.0399 m, which provided enough space for the kayak to fit into the domain up to the cockpit. The background mesh contained 40 cells in the x direction, 60 cells in the y direction and 10 cells in the z direction, being refined to level 1 in the *box1* region and to level 2 in the *box2* and *box3* regions.

4.7.5 Simulation cases

Four parameters were changed across the various simulations conducted on the Vanquish III kayak:

- Froude number, by changing the freestream velocity (see Table 3);
- Waterline level (122 mm or 150 mm from the lowest point on the hull);
- Mesh refinement near the hull surface (cell level 3, 4 or 5);
- Reynolds number, by scaling or not scaling the kinematic viscosity of the fluids.

Several combinations of those parameters were simulated. Table 4 clarifies the different conjugations used.

Table 4 – Summary of the simulations performed on the Vanquish III kayak

Hull draught	Viscosity	Hull refinement level	Froude numbers
150 mm	Scaled	3	0.517; 0.644
		4	0.389; 0.517; 0.644; 0.770
		5	0.389; 0.517; 0.644; 0.770
	Not scaled	3	0.389; 0.517; 0.644; 0.770
122 mm	Scaled	4	0.517; 0.644
		5	0.389; 0.517; 0.644; 0.770
	Not scaled	5	0.389; 0.517; 0.644; 0.770

4.8 Vanquish IV kayak

The external shape of an M-sized Nelo Vanquish IV kayak was measured during the course of this project, and the resulting model (Figure 18) was scaled down to meet the 1 m length imposition.

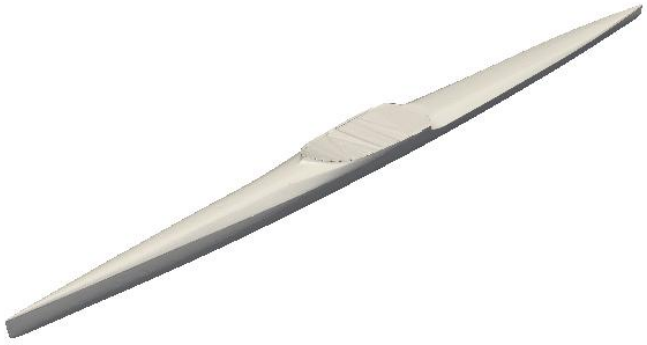

Figure 18 – Vanquish IV kayak model

There was access to measurements regarding the position of the waterline for various paddler weights, and a weight of 85 kg was assumed, corresponding to a draught of 128 mm. The background mesh and refinement regions implemented were the same as the ones used for the Vanquish III kayak.

Due to the very small amount of time remaining before the submission of this dissertation, the only simulations conducted were for hull refinement levels of 4 and 5, with scaled kinematic viscosity, for all four velocities in Table 3.

5 Results and discussion

5.1 Wigley hull

5.1.1 Drag coefficients

As has been mentioned in section 4.5, the steady state values for the drag forces were obtained by observing the dependence of the drag force in time, identifying the instants during which the forces are oscillating around a fixed value, and time-averaging the forces throughout that time range. Examples of the convergence and averaging on Wigley hull cases are shown in Figure 19 and Figure 20.

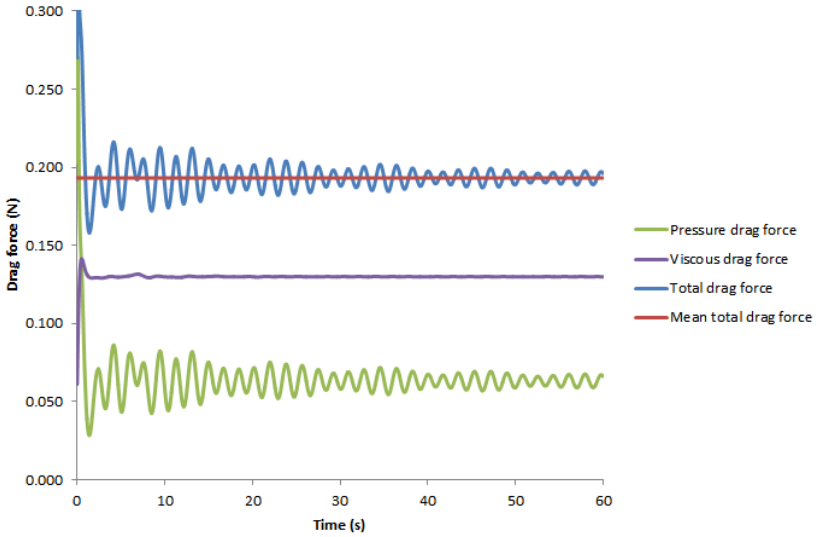

Figure 19 – Evolution of drag force throughout the simulation and indication of the mean force for the Wigley hull with Fr = 0.250

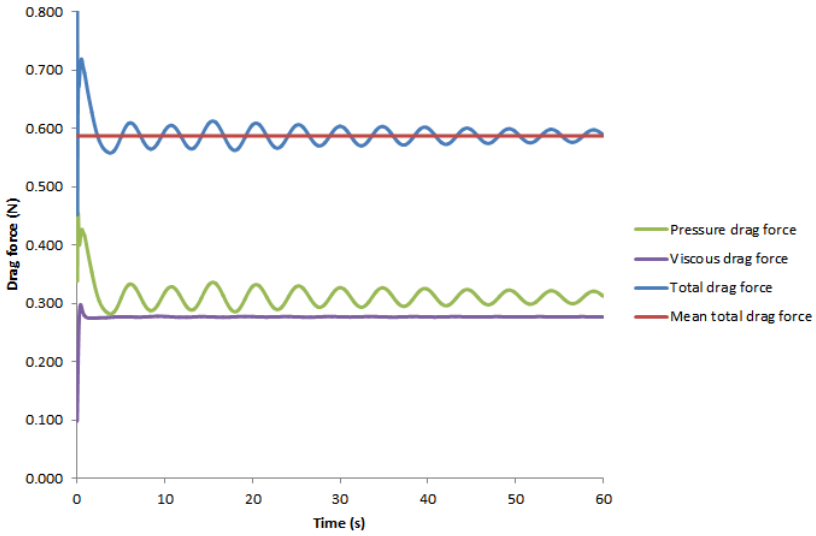

Figure 20 – Evolution of drag force throughout the simulation and indication of the mean force for the Wigley hull with Fr = 0.400

The total drag coefficients obtained for the unrefined mesh and for the cases with full mesh refinement are presented in Figure 21.

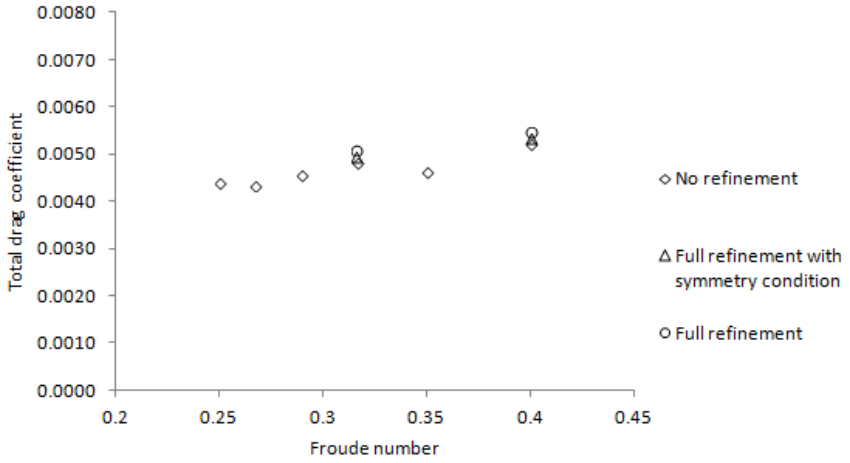

Figure 21 – Total drag coefficients on the Wigley hull for different full mesh refinements

Mesh refinement appears to produce higher values of total drag. The coefficients obtained by only simulating the flow on the right side of the hull and imposing a symmetry plane condition are in between the results from the unrefined mesh and the results from the fully refined mesh using the whole domain. However, the computational time was over four times lower, which may present an advantage.

The comparison between results for the different cell refinement levels near the hull surface is shown in Figure 22. The cell level of 3 corresponds to the initial unrefined mesh.

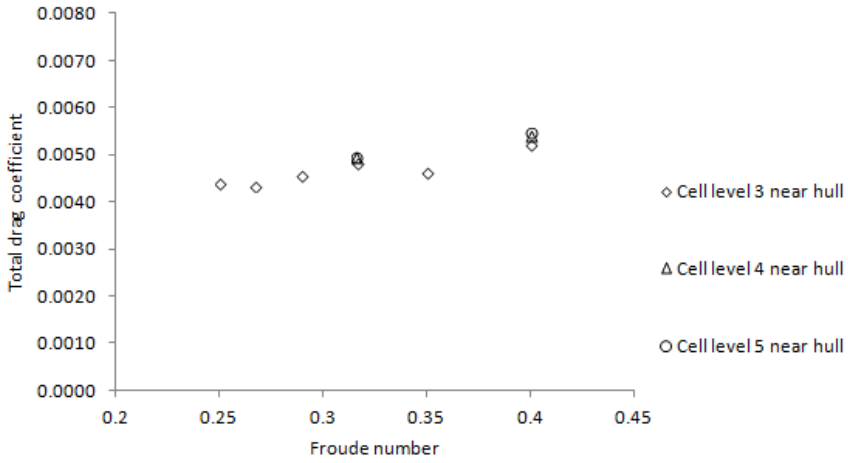

Figure 22 – Total drag coefficients on the Wigley hull for different refinement levels around the hull

The total resistance values obtained also rise with the local increase of mesh resolution around the hull. For a Froude number of 0.316, the results rise visibly less than in when the whole mesh is refined. However, for a Froude number of 0.400, the increment arising from the greater refinement of the hull region approaches the one obtained with full mesh refinement remarkably well. Due to the comparatively greatly

reduced computational time when only the hull region is refined, changing cell level around the hull is an efficient procedure, especially for Froude numbers closer to the velocity range of sprint and marathon kayaks.

5.1.2 Mesh dependence study

For the Froude numbers for which different mesh resolutions were tested, it is possible to estimate the result that would be obtained with a mesh of theoretically infinite resolution using Richardson extrapolation [42].

Richardson extrapolation makes the assumption that, for a certain calculated variable, the error caused by discretisation E can be written as a function of a measure of grid spacing h in the following way:

$$E(h) = Ch^p \tag{5.1}$$

The variable p is the order of grid convergence, and its value can be estimated in the knowledge of the results obtained for three meshes of different refinement level. Since only two different full mesh refinements were tested, the grid convergence can be assumed to be second-order ($p = 2$), which is a common approximation.

The extrapolated value of the desired variable f_0 can be taken from two known values f_1 and f_2 calculated from two differently refined grids with a known refinement ratio r. If f_1 is the value obtained in the less refined grid:

$$f_0 = f_1 + \frac{f_1 - f_2}{r^p - 1} \tag{5.2}$$

The full mesh refinement performed for the Wigley hull had a refinement ratio of 2 in regions not affected by the *snappyHexMesh* refinement, but a refinement ratio of 4 in areas refined by this application. Since a large area around the hull was refined by *snappyHexMesh*, the grid refinement ratio is assumed to be 4 for the purpose of this extrapolation.

The graph in Figure 23 presents the second-order extrapolation results for the total drag force for the Froude numbers of 0.316 and 0.400 by plotting the calculated and extrapolated values against the grid spacing variable h, which is chosen to be the spacing in the x direction for the region with the highest level of refinement.

If the extrapolation is assumed as the correct value, each of the simulation cases will carry the error shown in Table 5.

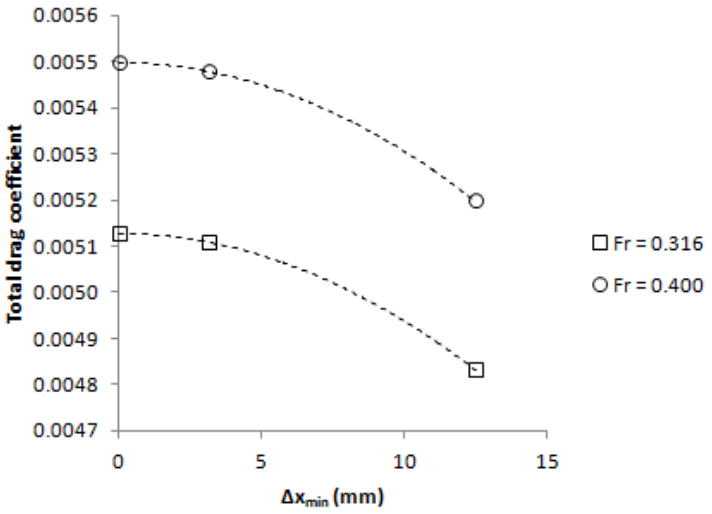

Figure 23 – Total drag dependence on full mesh refinement

Table 5 – Total drag coefficient error for each simulation case, according to Richardson extrapolation

Mesh refinement	**Error ($Fr = 0.316$)**	**Error ($Fr = 0.400$)**
No refinement	5.8%	5.4%
Total refinement	0.36%	0.34%
Total refinement with symmetry condition	3.4%	3.0%
Hull refinement (level 4)	3.7%	1.9%
Hull refinement (level 5)	2.8%	0.35%

5.1.3 Experimental benchmarking

The total drag coefficients obtained for the unrefined mesh and from the Richardson extrapolation are compared with experimental results for a fixed position Wigley hull (from [23]) in Figure 24.

It can be seen that in the unrefined mesh, results are, in general, close to the ones obtained by the Ship Research Institute (SRI), even though they fall below the range of experimental values for larger Froude numbers (especially 0.400).

The extrapolation results are higher, approaching the value from the University of Tokyo (UT) for a Froude number of 0.316, and rising slightly above the SRI value for a Froude number of 0.400.

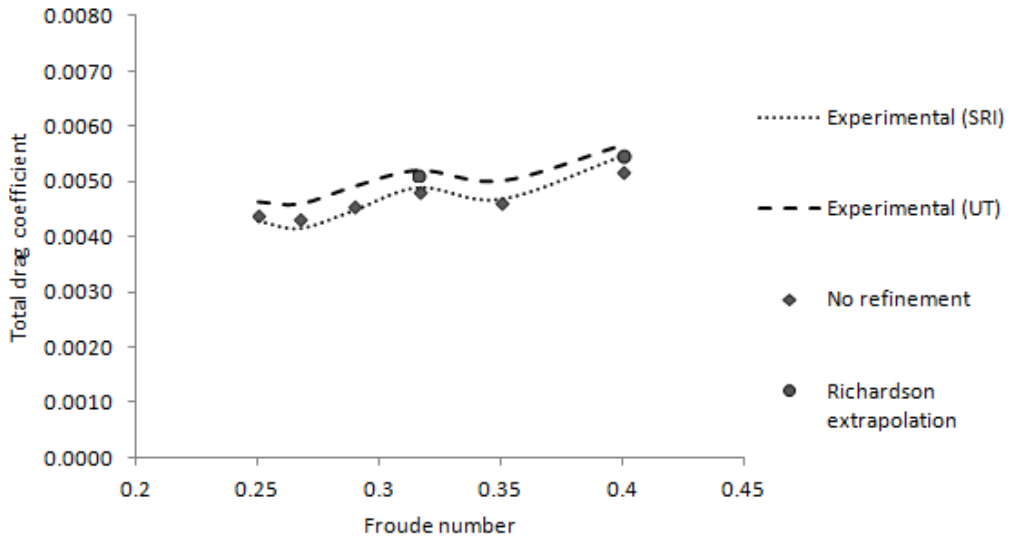

Figure 24 – Total drag coefficients on the Wigley hull for different full mesh refinements

5.1.4 Hull wave profile

The wave elevation data are presented and compared with the SRI and UT experimental results for a fixed position Wigley hull from Figure 25 to Figure 28. Only the four lowest Froude numbers simulated are coincident with the ones for which experimental results are available. Therefore, only Froude numbers up to 0.316 will be compared.

The profiles obtained with the unrefined mesh show visible differences from one side of the hull to the other. Even though, in general, the curves follow the experimental data relatively well, there are many common failures throughout the whole range of Froude numbers.

The magnitude of the wave crest near the bow of the hull is underestimated, a feature also encountered in the hull wave profiles obtained by Harpal and Patel [28], and Perez et al. [25]. The trough following it was also calculated to be deeper than the experimental results show. The simulation results also consistently exhibit a trough at $x/L \cong$ -0.1 and a peak at $x/L \cong 0.1$ that do not match experimental data.

The results from the refined mesh used for $Fr = 0.316$ are, on the other hand, nearly the same between them. The curves from the right and left sides of the hull are almost undistinguishable, and the results using half of the mesh with a symmetry condition differ only slightly from those.

All the peaks found in the results from the unrefined mesh were eliminated or greatly diminished when the mesh was refined. However, small punctual deviations appear throughout the profile, its deepest part is still shown as being deeper than it truly is, and there seems to be some disparity at the back of the hull (although the experimental results show some disparity between themselves as well).

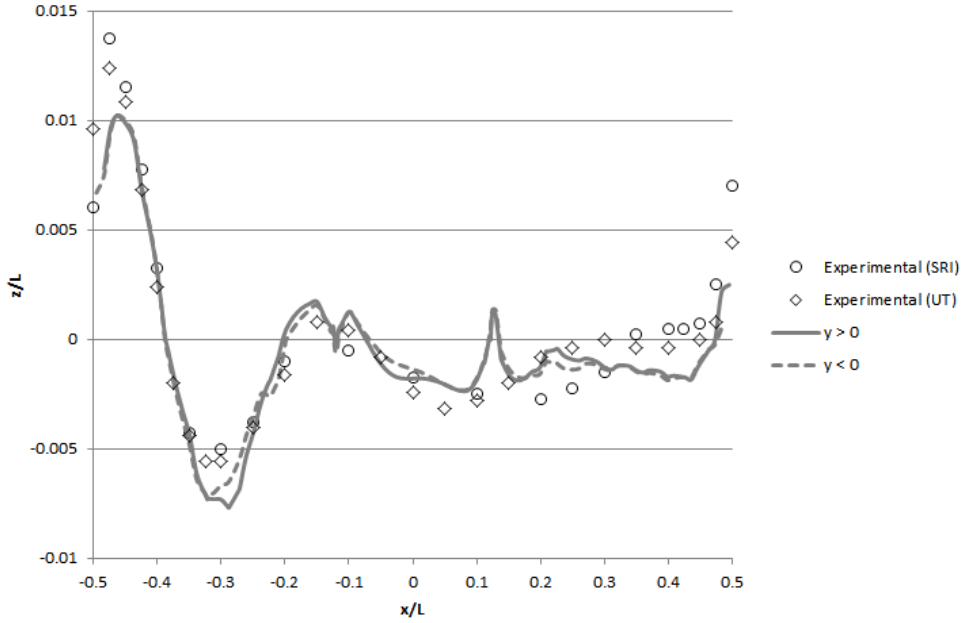

Figure 25 – Wave hull profile on the Wigley hull for Fr = 0.250

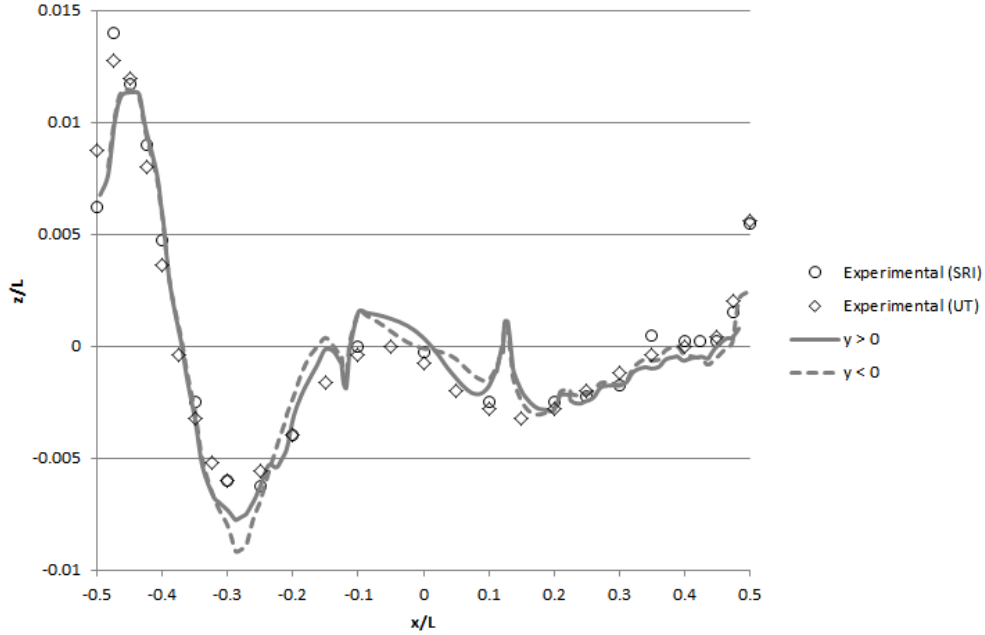

Figure 26 – Wave hull profile on the Wigley hull for Fr = 0.267

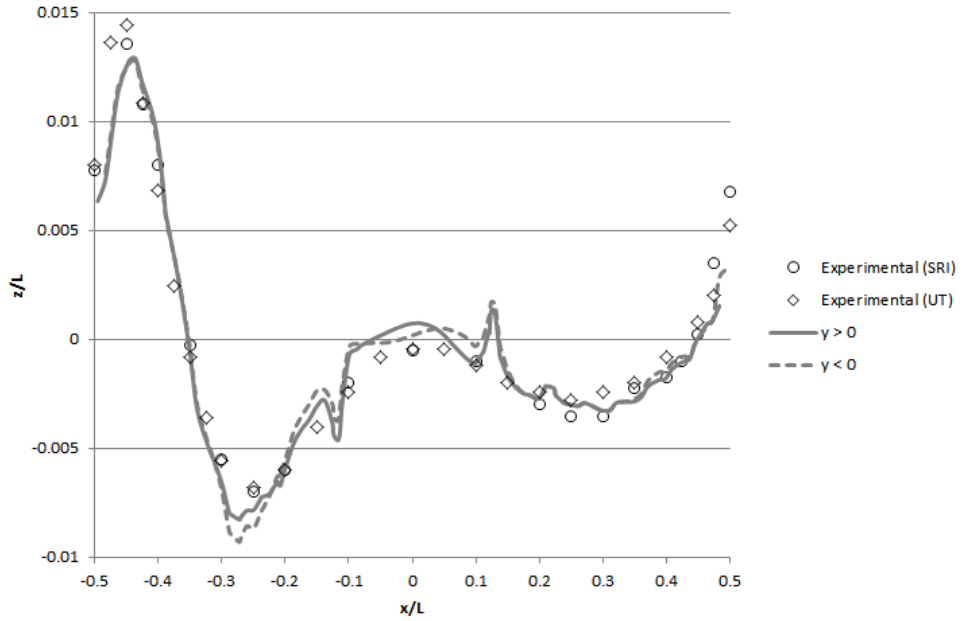

Figure 27 – Wave hull profile on the Wigley hull for Fr = 0.289

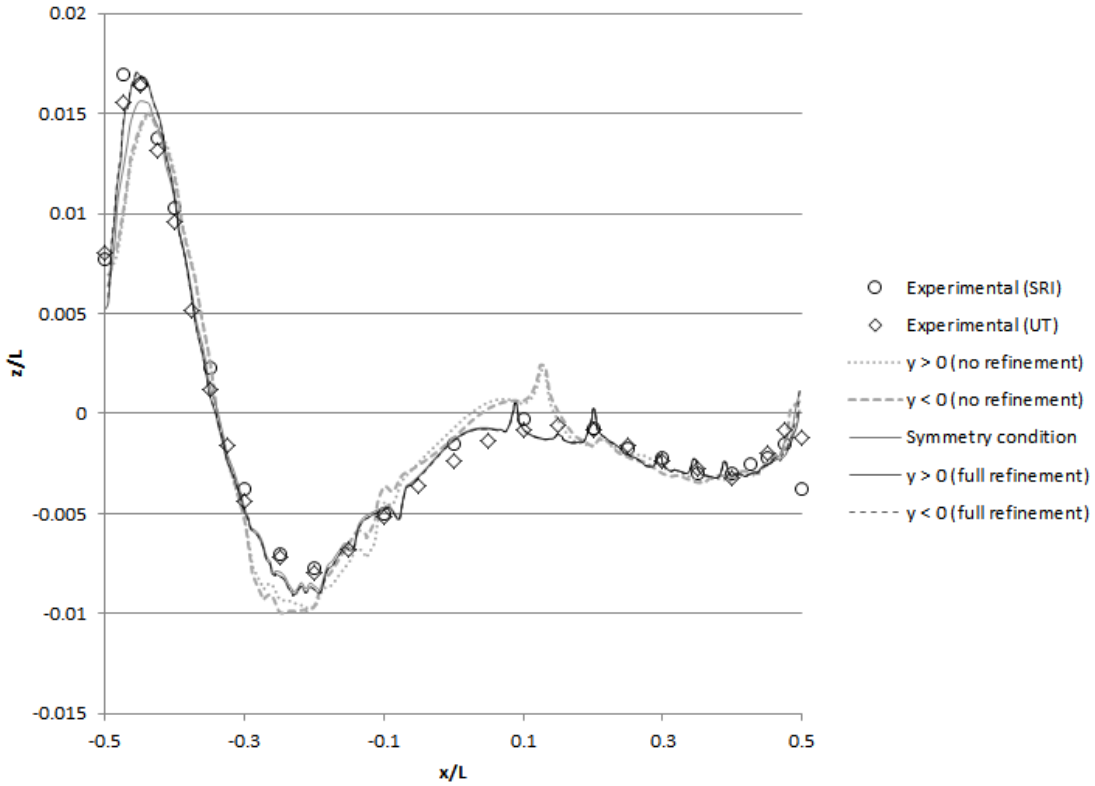

Figure 28 – Wave hull profile on the Wigley hull for Fr = 0.316

5.2 Vanquish III kayak

5.2.1 Drag coefficients

The results for the Vanquish III model with the waterline located 150 mm above the bottom of the hull are shown in Figure 29. They are compared to the results obtained by Mantha et al. [17] for the same model, and the results obtained by Gomes et al. [16] for the Vanquish IV kayak.

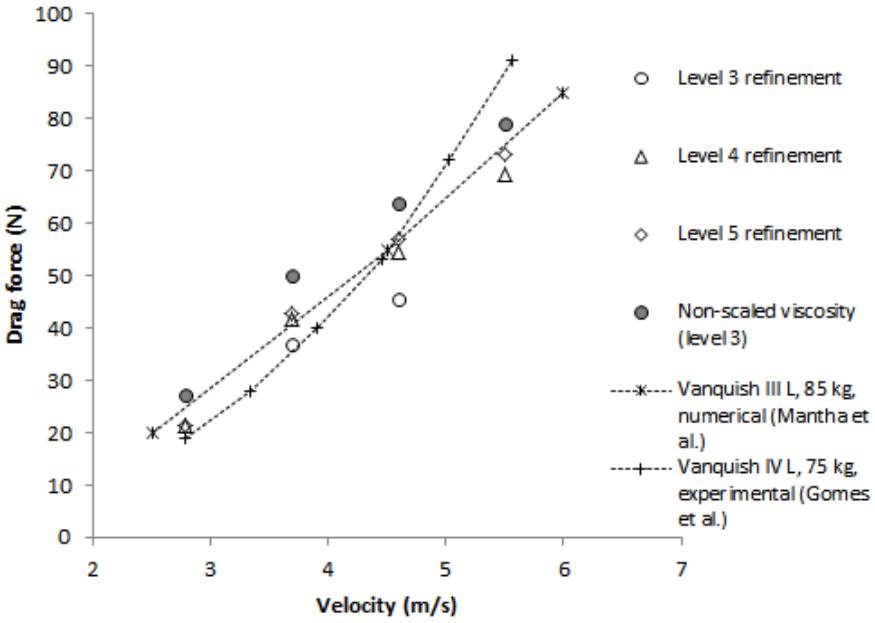

Figure 29 – Total resistance results for the Vanquish III kayak for a 150 mm draught

It can also be seen that, similarly to the results of mesh refinement in the Wigley hull, less refined meshes result in underestimates for values of the total drag force, and that the higher the Froude number is, the more pronounced the differences between different refinements are.

Refinement of the mesh around the hull made the resulting drag forces approach the numerical data obtained by Mantha et al. [17], while still showing some notable differences to experimental results for Froude numbers where previous numerical and experimental results are also not coincident. This may be related to the fact that the numerical simulations, both in this study and the one conducted by Mantha et al. [17], simplify the problem by ignoring the motion of the kayak aside from its forward velocity. On the other hand, finer meshes may also be required for accurate results.

The results obtained by keeping the viscosity values constant when scaling the model are somewhat higher than the values reached when the Reynolds number is kept, which is expected since the magnitude of shear forces rises for larger viscosity values.

The values obtained for the lower waterline location of 122 mm below the hull are shown in Figure 30.

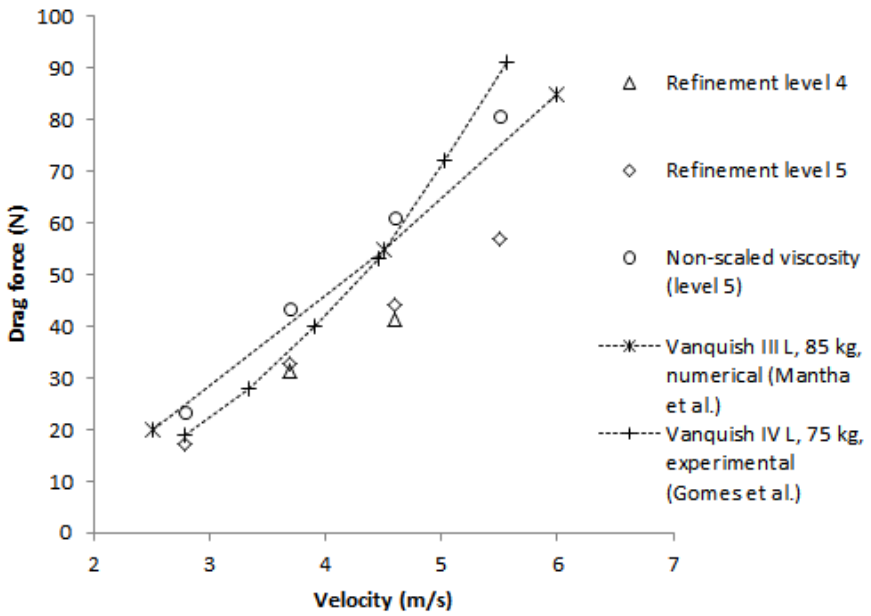

Figure 30 – Total resistance results for the Vanquish III kayak for a 122 mm draught

The results are lower than the ones presented for the higher draught value, especially for higher Froude numbers, which, if experimental evidence did not contradict it, would point towards the initial guess of 150 mm likely being closer to the correct value. There is a good adjustment to experimental results for the two lowest Froude numbers.

These results also confirm that mesh refinement yields higher resistance force values, and that not scaling the viscosity of the fluids results in higher values as well. The variation of the drag force with the Froude number appears to be nearly linear, like in previously seen results, including the ones obtained by Mantha et al [17].

The deviation from experimental (Gomes et al. [16]) and numerical (Mantha et al [17]) results for both values of the hull draught is presented in Table 6, for the mesh with the highest refinement.

Table 6 – Deviation from numerical and experimental results of the drag coefficients for the Vanquish III kayak obtained from the most refined mesh

Velocity (m/s)	**150 mm draught**		**122 mm draught**	
	Numerical (Mantha et al.)	**Experimental (Gomes et al.)**	**Numerical (Mantha et al.)**	**Experimental (Gomes et al.)**
2.78	-13.7%	13.1%	-31.5%	-10.3%
3.69	4.6%	20.3%	-20.2%	-8.2%
4.60	-0.4%	-1.7%	-22.7%	-23.7%
5.50	-2.4%	-17.7%	-24.0%	-35.8%

5.2.2 Wave elevation

The wave elevation in throughout the domain is shown below for a hull draught of 122 mm, a hull refinement level of 5, and scaled viscosity, for Froude numbers of 0.389 (Figure 31) and 0.770 (Figure 32).

In the wave pattern for the lower Froude number, a pattern of several diverging waves can be seen on the sides of the wake, along with one crest and one trough from a transverse wave behind the hull. The wave elevation reaches its highest values near the bow of the hull and immediately behind the hull.

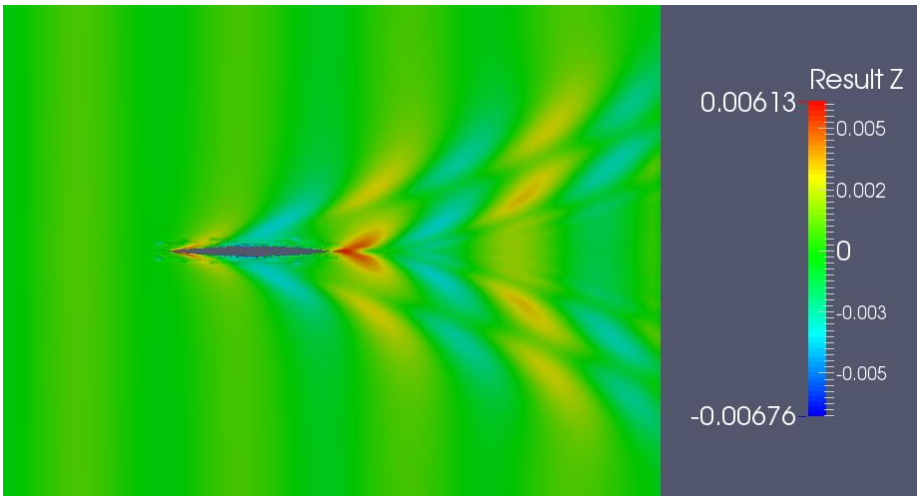

Figure 31 – Wave elevation (displayed as z/L) for the Vanquish III kayak for Fr = 0.389

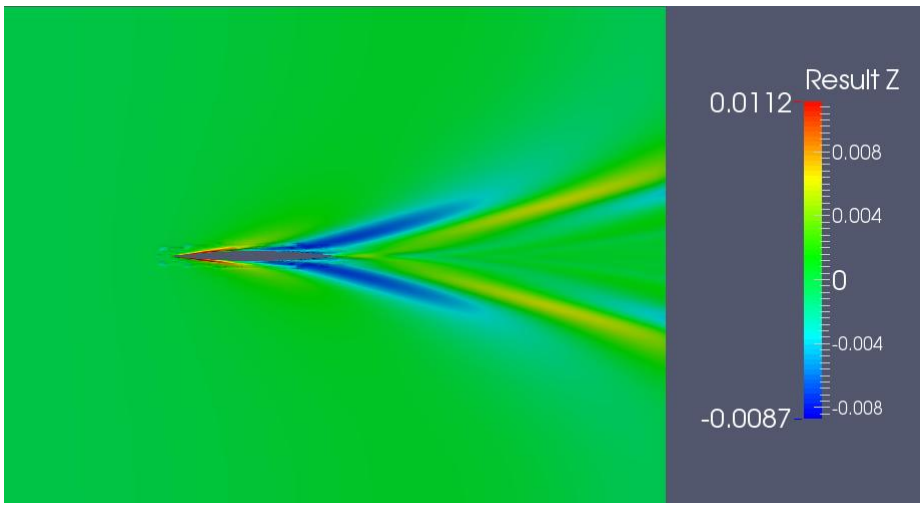

Figure 32 – Wave elevation (displayed as z/L) for the Vanquish III kayak for Fr = 0.770

The higher Froude number reveals a slightly different wave pattern. The diverging waves seen on the sides of the wake are fewer and their length is larger compared to the Froude number of 0.389. Transverse waves are not visible. The magnitude of the wave elevation is also higher, both in the positive direction and in the negative direction. The wake is also narrower for the higher Froude number, even though this is expected to be

only a near-field phenomenon, as the angular width of the wake in the far-field is independent of velocity.

5.3 Vanquish IV kayak

The drag results for the Vanquish IV M kayak are compared in Figure 33 with the experimental results of Gomes et al. [16] for the same kayak with the same displaced water volume, as well as the numeric results of Mantha et al. [17] for the Vanquish III L.

As in the results for the Vanquish III, the evolution of the drag force with the Froude number is approximately linear, and mesh refinement near the hull produces higher values, whose difference to the unrefined results is stronger for higher Froude numbers.

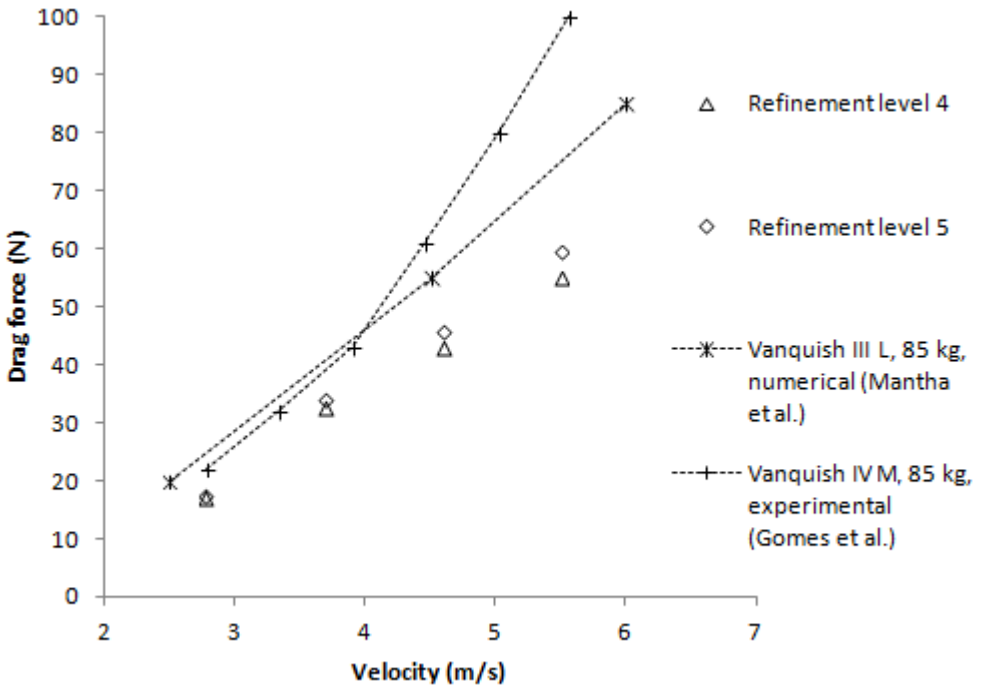

Figure 33 – Total resistance results for the Vanquish IV kayak

The deviations from experimental (Gomes et al. [16]) and numerical (Mantha et al [17]) results for the Vanquish IV are presented in Table 7 for all four velocity values, for the mesh with the highest refinement.

The results obtained are slightly lower than experimental and numerical results. Even though the waterline level is experimentally confirmed, it was obtained for a kayak at rest. Since the kayak will experience sinkage and trim when it moves, it is expected that the submerged volume will be larger and, by virtue of the larger wetted area, drag will increase.

Table 7 – Deviation from numerical and experimental results of the drag coefficients for the Vanquish IV kayak obtained from the most refined mesh

Velocity (m/s)	**Numerical (Mantha et al.)**	**Experimental (Gomes et al.)**
2.78	-30%	-21%
3.69	-17%	-13%
4.60	-19%	-30%
5.50	-21%	-39%

As stated for the Vanquish III model, the error may also stem from the fact that the flow was simulated for a fixed position hull, or from the possibility that the mesh refinement was not sufficient. Simulations with higher mesh refinements, due to their high computation times, could not be performed in the time available for the project.

6 Conclusions and future work

The results obtained for the Wigley hull indicate that the numerical model developed is functional and yields reasonably accurate values for total drag force and hull wave profile in the range of Froude numbers from 0.250 to 0.400.

The drag force values obtained for the Vanquish III model agree with experimental results for a range of Froude numbers from about 0.389 to 0.517, while continuing to agree with other numerical results for higher Froude numbers (up to 0.770) but deviating from experimental data. The Vanquish IV results are only close to experimental data up to a Froude number of 0.517.

On a more general note, it can be observed that in the kayaks, the variation of the total drag with the Froude number is linear as in the numerical results available, instead of the power law roughly followed by experimental values. This may be related to the fact that the simulations imposed a fixed position on the kayak hull, while the three-dimensional movement of the hull may possibly have an impact on the real results.

It is also possible that the mesh refinement was not sufficient (especially in regions other than the vicinity of the hull), or that other parameters of the numerical method were not appropriate. The validation of the method for the Wigley hull case was confined to the Froude numbers for which there were experimental results, which does not cover most of the velocity range simulated for the kayaks.

The results also indicate that equalling the weight of the displaced water volume to the weight of the kayak may not be a valid method of calculating the location of the waterline, and even measured results on a kayak at rest may not correspond to the real waterline, as sinkage and trim are possibly influential enough to alter resistance forces significantly.

Future work to be done in this subject could include, on the numerical side, the use of an adjustable mesh to take the movement of the hull into account in simulations, in order to compensate for possible errors in waterline placement by simulating sinkage and trim, and to investigate whether the rise of total drag for higher Froude numbers is related to the motion of the kayak. This would also presuppose the full CAD modelling of the kayaks, including mass and inertia data.

Different mesh configurations and refinements, as well as discretisation and algebraic solution schemes and boundary conditions could also be tested in order to overcome possible flaws in the numerical model.

Ultimately, an optimisation algorithm could be implemented using the developed numerical model, with the objective of reaching an optimal kayak hull geometry that would minimise total resistance and allow for faster race times.

On the experimental aspect, the collection of data on sinkage and trim for the simulated kayak models would allow for more accurate numerical simulations, as the

sinkage and trim obtained could be applied to the initial position of the kayak in the numerical model.

Experimental results obtained with the kayaks in a fixed position would also help investigate the effect of the movement of the kayak itself on the drag force, and possibly explain the divergence between numerical and experimental results for high Froude numbers.

7 References

[1] International Canoe Federation, “ICF - Planet Canoe,” [Online]. Available: http://www.canoeicf.com/icf. [Accessed 23 December 2014].

[2] International Canoe Federation, *International Canoe Federation Canoe Sprint Rules 2013*, 2012.

[3] Rio 2016™, “Canoe Slalom | Rio 2016,” [Online]. Available: http://www.rio2016.com/en/the-games/olympic/sports/canoe-slalom. [Accessed 22 December 2014].

[4] International Canoe Federation, *International Canoe Slalom Competition Rules 2013*, 2012.

[5] International Canoe Federation, *International Canoe Federation Canoe Marathon Competition Rules 2013*, 2012.

[6] International Canoe Federation, *International Canoe Federation Wildwater Canoeing Competition Rules 2015*, 2014.

[7] International Canoe Federation, *International Canoe Federation Canoe Polo Competition Rules 2013*, 2012.

[8] B. R. Munson, D. F. Young, T. H. Okiishi and W. W. Huebsch, Fundamentals of Fluid Mechanics, Sixth Edition, John Wiley and Sons, 2009.

[9] J. Newman, Marine Hydrodynamics, The MIT Press, 1977.

[10] S. Willman, *Lift and Drag of Kayak Rudders*, University of Oslo, 2011.

[11] E. Lataire, M. Vantorre and G. Delefortrie, “A prediction method for squat in restricted and unrestricted rectangular fairways,” *Ocean Engineering*, no. 55, pp. 71-80, 2012.

[12] M. J. Briggs, M. Vantorre, K. Uliczka and P. Debaillon, “Prediction of Squat for Underkeel Clearance,” in *Handbook of Ocean and Coastal Engineering*, World Scientific Publishing Company, 2010, pp. 723-774.

[13] M. G. Robinson, L. E. Holt and T. W. Pelham, “The Technology of Sprint Racing Canoe and Kayak Hull and Paddle Designs,” *International Sports Journal*, no. Summer 2002, pp. 68-85, 2002.

[14] J. S. Michael, R. Smith and K. B. Rooney, “Determinants of kayak paddling

performance," *Sports Biomechanics,* no. 8, pp. 167-179, 2009.

[15] B. Gomes, N. Ramos, F. Conceição, J. P. Vilas-Boas and M. A. P. Vaz, "Field Assessment of the Kayaks' Total Drag Force," in *15th International Conference on Experimental Mechanics*, Porto, 2012.

[16] B. Gomes, private communication.

[17] V. R. Mantha, A. J. Silva, D. A. Marinho and A. Rouboa, "Numerical Simulation of Two-Phase Flow Around Flatwater Competition Kayak Design-Evolution Models," *Journal of Applied Biomechanics,* no. 29, pp. 270-278, 2013.

[18] G. D. Tzabiras, S. P. Polyzos, K. Sfakianaki, V. Diafas, A. D. Villiotis, K. Chrisikopoulos and S. Kaloupsis, "Experimental and Numerical Study of the Flow Past the Olympic Class K-1 Flat Water Racing Kayak at Steady Speed," *The Sport Journal,* no. 13.4, 2010.

[19] J. Baker, *Biomechanics of Paddling,* AIS Movement Science, Australian Institute of Sport, Canberra, 2012.

[20] A. Laurent, A. Rouard, V. R. Mantha, D. A. Marinho, A. J. Silva and A. I. Rouboa, "The Computational Fluid Dynamics Study of Orientation Effects of Oar Blade," *Journal of Applied Biomechanics,* no. 29, pp. 23-32, 2013.

[21] David W. Taylor Naval Ship Research and Development Center, "Proceedings on the Workshop on Ship Wave-Resistance Computations," in *Workshop on Ship Wave-Resistance Computations,* Bethesda, Maryland, 1979.

[22] International Towing Tank Conference, "Report of the Resistance Committee," in *17th ITTC Committee,* Göteborg, 1984.

[23] H. Kajitani, H. Miyata, M. Ikehata, H. Tanaka, H. Adachi, M. Namimatsu and S. Ogiwara, *The Summary of the Cooperative Experiment on Wigley Parabolic Model in Japan,* 1983.

[24] S. Ju, "Study of Total and Viscous Resistance for the Wigley Parabolic Ship Form," Iowa Institute of Hydraulic Research, Iowa City, 1983.

[25] C. Perez G, M. Tan and P. Wilson, "Validation and verification of hull resistance components using a commercial CFD code," in *11th Numerical Towing Tank Symposium,* Brest, 2008.

[26] A. Pranzitelli, C. de Nicola and S. Miranda, "Steady-state calculations of free surface flow around ship hulls and resistance predictions," *The International Journal of Small Craft Technology,* vol. 154, no. Jul-Dec 2012, 2012.

[27] F. Inok, A. Lavrov and C. Guedes Soares, “Analysis of the free surface turbulent flow around a forward moving Wigley hull with OpenFOAM,” in *Developments in Maritime Transportation and Exploitation of Sea Resources, Vol 1*, 2014, pp. 33-40.

[28] N. Harpal and C. Patel, *Numerical Modeling of Resistance for a Conceptual Seatrain,* CD-adapco Academic Paper Contest, 2011.

[29] C. W. Hirt and B. D. Nichols, “Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries,” *Journal of Computational Physics,* no. 39, pp. 201-225, 1981.

[30] O. Ubbink, *Numerical prediction of two fluid systems with sharp interfaces,* 1997.

[31] E. Berberović, N. P. van Hinsberg, S. Jakirlić, I. Roisman and C. Tropea, “Drop impact onto a liquid layer of finite thickness: Dynamics of the cavity evolution,” *Physical Review,* no. 79, 2009.

[32] D. C. Wilcox, Turbulence Modelling for CFD, La Cañada: DCW Industries, 1994.

[33] W. P. Jones and B. E. Launder, “The Prediction of Laminarization with a Two-Equation Model of Turbulence,” *International Journal of Heat and Mass Transfer,* vol. 15, pp. 301-314, 1972.

[34] B. E. Launder and B. I. Sharma, “Application of the Energy Dissipation Model of Turbulence to the Calculation of Flow Near a Spinning Disc,” *Letters in Heat and Mass Transfer,* vol. 1, no. 2, pp. 131-138, 1974.

[35] D. C. Wilcox, “Re-assessment of the scale-determining equation for advanced turbulence models,” *AIAA Journal,* vol. 26, no. 11, pp. 1299-1310, 1988.

[36] F. R. Menter, “Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications,” *AIAA Journal,* vol. 32, no. 8, pp. 1598-1605, 1994.

[37] F. R. Menter, M. Kuntz and R. Langtry, “Ten Years of Industrial Experience with the SST Turbulence Model,” in *Turbulence, Heat and Mass Transfer 4,* Begell House, Inc., 2003, pp. 625-632.

[38] NASA Langley Research Center, “Menter Shear Stress Transport Model,” 2014. [Online]. Available: http://turbmodels.larc.nasa.gov/sst.html. [Accessed 20 January 2015].

[39] OpenFOAM Foundation, “OpenFOAM® - The Open Source Computational Fluid Dynamics (CFD) Toolbox,” 2014. [Online]. Available: http://openfoam.org/. [Accessed 21 January 2015].

[40] T. Marić, J. Höpken and K. Mooney, The OpenFOAM® Technology Primer, sourceflux UG, 2014.

[41] M.A.R. Kayaks Lda., “NELO,” [Online]. Available: http://www.nelo.eu/. [Accessed 25 January 2015].

[42] NPARC Alliance, “Examining Spatial (Grid) Convergence,” 2008. [Online]. Available: http://www.grc.nasa.gov/WWW/wind/valid/tutorial/spatconv.html. [Accessed 31 January 2015].

Appendix A – Boundary condition files

In this section, a transcription of the files contained in the folder for the initial time instant will be presented. There is a file for every calculated field: velocity, dynamic pressure, water volume fraction, eddy viscosity, turbulence kinetic energy, and turbulence kinetic energy dissipation frequency. File names are presented in brackets after the name of the field.

In the velocity file, the value given for the *internalField* and *inlet* conditions (shown as 2.412 in the file transcribed below) is adjusted according to the freestream velocity intended.

Velocity (U)

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volVectorField;
    location    "0";
    object      U;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dimensions      [0 1 -1 0 0 0 0];
internalField   uniform (2.412 0 0);
boundaryField
{
    inlet
    {
        type            fixedValue;
        value           uniform (2.412 0 0);
    }
```

```
    outlet
    {
        type            zeroGradient;
    }
    sides
    {
        type            symmetryPlane;
    }
    atmosphere
    {
        type            zeroGradient;
    }
    hull_wall
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }
}
// ************************************************************************* //
```

Dynamic pressure (p_rgh)

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      p_rgh;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

```
dimensions      [1 -1 -2 0 0 0 0];

internalField   uniform 0;

boundaryField

{

    inlet

    {

        type            zeroGradient;

    }

    outlet

    {

        type            fixedValue;

        value           uniform 0;

    }

    sides

    {

        type            symmetryPlane;

    }

    atmosphere

    {

        type            fixedValue;

        value           uniform 0;

    }

    hull_wall

    {

        type            zeroGradient;

        value           uniform 0;

    }

}

// ************************************************************************* //
```

Water volume fraction (alpha1)

```
/*--------------------------------*- C++ -*----------------------------------*\

| =========                 |                                                 |

| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |

|  \\    /   O peration     | Version:  2.1.0                                 |

|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
```

```
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "0";
    object      alpha;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dimensions      [0 0 0 0 0 0 0];
internalField   uniform 0;
boundaryField
{
    inlet
    {
        type            calculated;
        value           uniform 0;
    }
    outlet
    {
        type            zeroGradient;
    }
    sides
    {
        type            symmetryPlane;
    }
    atmosphere
    {
        type            inletOutlet;
        inletValue      uniform 0;
        value           uniform 0;
    }
    hull_wall
```

```
    {
        type            zeroGradient;
    }
}
// ************************************************************************* //
```

Eddy viscosity (nut)

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "0";
    object      nut;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dimensions      [0 2 -1 0 0 0 0];
internalField   uniform 5e-07;
boundaryField
{
    inlet
    {
        type            fixedValue;
        value           uniform 5e-07;
    }
    outlet
    {
        type            zeroGradient;
    }
```

```
    sides
    {
        type            symmetryPlane;
    }
    atmosphere
    {
        type            zeroGradient;
    }
    hull_wall
    {
        type            nutkWallFunction;
        value           uniform 0;
    }
}
// ************************************************************************* //
```

Turbulence kinetic energy (k)

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "0";
    object      k;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dimensions      [0 2 -2 0 0 0 0];
internalField   uniform 0.00015;
boundaryField
```

```
{
    inlet
    {
        type            fixedValue;
        value           uniform 0.00015;
    }
    outlet
    {
        type            zeroGradient;
    }
    sides
    {
        type            symmetryPlane;
    }
    atmosphere
    {
        type            inletOutlet;
        inletValue      uniform 0.00015;
        value           uniform 0.00015;
    }
    hull_wall
    {
        type            kqRWallFunction;
        value           uniform 0.00015;
    }
}
// ************************************************************************* //
```

Turbulence kinetic energy dissipation frequency (omega)

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
```

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      omega;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dimensions      [0 0 -1 0 0 0 0];
internalField   uniform 2;
boundaryField
{
    inlet
    {
        type            fixedValue;
        value           uniform 2;
    }
    outlet
    {
        type            zeroGradient;
    }
    sides
    {
        type            symmetryPlane;
    }
    atmosphere
    {
        type            inletOutlet;
        inletValue      uniform 2;
        value           uniform 2;
    }
    hull_wall
    {
        type            omegaWallFunction;
        value           uniform 2;
```

```
    }
}
// ************************************************************************ //
```

Appendix B – Discretisation and solution schemes files

This section includes a transcription of the files that contain information on discretisation schemes and algebraic solution schemes, respectively, *fvSchemes* and *fvSolution*.

fvSchemes

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
ddtSchemes
{
    default         Euler;
}
gradSchemes
{
    default         Gauss linear;
    grad(U)      cellLimited leastSquares 1;
}
divSchemes
{
    div(rho*phi,U)  Gauss linearUpwind grad(U);
    div(phi,alpha)  Gauss vanLeer;
```

```
    div(phirb,alpha) Gauss interfaceCompression;
    div(phi,k)      Gauss linearUpwind grad(U);
    div(phi,omega)  Gauss linearUpwind grad(U);
    div((muEff*dev(T(grad(U))))) Gauss linear;
    div((nuEff*dev(T(grad(U))))) Gauss linear;
}
laplacianSchemes
{
    default         Gauss linear limited 0.5;
}
interpolationSchemes
{
    default         linear;
}
snGradSchemes
{
    default  limited 0.333;
}
fluxRequired
{
    default         no;
    p_rgh;
    pcorr;
    alpha1;
}
// ************************************************************************* //
```

fvSolution

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
```

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
solvers
{
    pcorr
    {
        solver          PCG;
        preconditioner
        {
            preconditioner  GAMG;
            smoother        DICGaussSeidel;
            agglomerator    faceAreaPair;
            mergeLevels     1;
            nCellsInCoarsestLevel 10;
            cacheAgglomeration true;
            tolerance       1e-5;
            relTol          0;
        };
        tolerance       1e-5;
        relTol          0;
    };
    p_rgh
    {
        solver          GAMG;
        smoother        GaussSeidel;
        agglomerator    faceAreaPair;
        mergeLevels     1;
        nCellsInCoarsestLevel 10;
```

```
        cacheAgglomeration true;


        tolerance       1e-6;
        relTol          0.01;
    };
    p_rghFinal
    {
        $p_rgh;
        tolerance       1e-6;
        relTol          0;
    }
    "(U|k|omega).*"
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        nSweeps         1;
        tolerance       1e-7;
        relTol          0.1;
    };
}
PIMPLE
{
    momentumPredictor yes;
    nCorrectors     2;
    nNonOrthogonalCorrectors 0;
    nAlphaCorr      1;
    nAlphaSubCycles 3;
    cAlpha          0.5;
    maxCo           1.;
    maxAlphaCo      1.;
    nAlphaSweepIter 1;
    rDeltaTSmoothingCoeff 0.1;
    rDeltaTDampingCoeff 1;
    maxDeltaT       1;
}
```

```
relaxationFactors
{
    fields
    {
    }
    equations
    {
    }
}
// ************************************************************************* //
```