Mesh Validation and Resistance Prediction of the JBC Bulker Design using CFD Method

Cheng XU

Master Thesis

presented in partial fulfillment
of the requirements for the double degree:
"Advanced Master in Naval Architecture" conferred by University of Liege
"Master of Sciences in Applied Mechanics, specialization in Hydrodynamics,
Energetics and Propulsion" conferred by Ecole Centrale de Nantes

developed at University of Rostock
in the framework of the

"EMSHIP"
Erasmus Mundus Master Course
in "Integrated Advanced Ship Design"

Ref. 159652-1-2009-1-BE-ERA MUNDUS-EMMC

Supervisor: Prof. Robert Bronsart, University of Rostock

Reviewer: Prof. Pierre Ferrant, Ecole Centrale de Nantes

Rostock, January 2016

ABSTRACT

In the preliminary stage of ship design, resistance prediction is a primary challenge for designers. The objective of the thesis is to predict the total resistance and its components of JBC bulker with the RANS method. As a mature approach, CFD is an effective tool to estimate resistance. The well investigated hull form KVLCC2 is used to study the accuracy of the CFD calculation and to validate the resistances and the flow field around the hull by comparing it with existing experimental data.

Simulations are performed using the open source software package OpenFOAM. A steady state RANS solver is used to solve the viscous flow around the hull and two-equation K-Omega SST model is selected for the turbulence modeling. The vessels are implemented in a fixed towing condition where the effect of trim and sinkage is neglected. The commercial mesher HEXPRESS is used to generate unstructured hexahedral meshes of the fluid domain. Local grid refinements are carried out by setting the multi-block meshes and the free surface is captured via VOF method. Additional refinement mesh is given in the boundary areas of the hull and the wall function is implemented to simulate the turbulence characteristics.

A suitable value of turbulence intensity is obtained after a set of tests in order to get the right turbulence parameters. In addition, the effect of grids with similar configuration is investigated and 5-level gradual refinement meshes are studied. The resistance components and the wave elevations in three different wave cuts are compared with the experimental results, which show a good agreement. The most proper mesh configuration with good accuracy is selected to convert to the JBC bulker.

At the end, the calculated resistances and three different transverse wave cuts of the bare hull JBC bulker in calm water condition are presented.

CONTENTS

LISTS OF FIGURE

LISTS OF TABLE

NOMENCLATURE

Symbols	SI Units
CFD	*Computational Fluid Dynamics*
DNS	*Direct Numerical Simulation*
EEDI	*Energy Efficiency Design Index*
EFD	*Experimental Fluid Dynamics*
GAMG	*Geometric Algebraic Multi Grid*
JBC	*Japan Bulk Carrier*
LES	*Large Eddy Simulation*
LTS	*Local Time Stepping*
PISO	*Pressure-Implicit with Splitting of Operators*
RANSE	*Reynolds-Averaged Navier-Stokes Equation*
RSM	*Reynolds Stress Models*
SIMPLE	*Semi-Implicit Method for Pressure-Linked Equations*
SST	*Shear Stress Transport*
VOF	*Volume of Fluid*
VTK	Visualization Toolkit

α/alpha1	Volume Fraction Function
δ_{99}	Thickness of the Boundary Layer $[m]$
μ	Dynamic Viscosity $[N\,s/m^2]$
ν	Kinematic Viscosity $[m^2/s]$
ω	Specific Dissipation Rate $[1/s]$
ρ	Fluid Density $[kg/m^3]$
C_f	Frictional Resistance Coefficient
f_x	External Force Components in x axis $[N]$
f_y	External Force Components in y axis $[N]$
f_z	External Force Components in z axis $[N]$
I	Turbulence Intensity
k	Turbulent Kinetic Energy$[m^2/s^2]$
P	Fluid Pressure $[N/m^2]$
R_F	Frictional Resistance $[N]$
R_R	Residuary Resistance $[N]$
R_{PV}	Viscous Pressure Resistance $[N]$
R_T	Total Resistance $[N]$
R_V	Total Viscous Resistance $[N]$
R_W	Wave Resistance $[N]$
S	Wetted Surface Area $[m^2]$
U	Ship Speed / Incoming Flow Speed $[m/s]$
u	velocity component in x axis $[m/s]$
v	velocity component in y axis $[m/s]$
w	velocity component in z axis $[m/s]$

DECLARATION OF AUTHORSHIP

I declare that this thesis and the work presented in it are my own and have been generated by me as the result of my own original research.

Where I have consulted the published work of others, this is always clearly attributed.

Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.

I have acknowledged all main sources of help.

Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

This thesis contains no material that has been submitted previously, in whole or in part, for the award of any other academic degree or diploma.

I cede copyright of the thesis in favor of the University of Rostock

Date: *Signature*

1. INTRODUCTION

Resistance prediction of a vessel is a crucial task during the entire design stages for naval architects. Successful design of new ship can help to achieve the expected speed, which is called contract speed. If the final speed does not meet the requirements from the ship owner, there will be different levels of loss to the shipyard, depending on the speed gap in the end. In the process of design, resistance prediction should be specially taken into account to get more accurate values and be updated simultaneously for later analysis. On the other hand, as the resistance is strongly related to the requirement of engine power, the new design regarding environment issues must adhere to the rules and regulations, such as Energy Efficiency Design Index (EEDI) from International Maritime Organization (IMO). Saving energy is always a key objective that people would take into considerations while performing any kind of human activity. It would be of great importance to develop a fuel-efficient ship hull form in the preliminary design stage. In other words, the optimization of the ship performance is a significant issue in the design process. The effective power or the resistance of a vessel is the primary consideration for the ship owners.

1.1. CFD

The prediction of ship hull resistance plays an importance role in ship design and optimization. In general, there are three typical methods to predict the resistance of a vessel. The first one is the empirical formulas, which is the simplest and fastest way. This method is mainly applied in the early design stage in order to get the main dimensions or the coefficients of the new ships. In most of the cases, the results can be used as a reference and coarse data. Moreover, some of the empirical formulas were obtained from the outdated ship information many years before, which are not that suitable for the new types of ship nowadays. Traditionally, the investigation of ship resistance was conducted by the second method, experimental towing tank test. The result of this approach is reliable, but it is quite time-consuming and costly, sometimes it is even difficult to record the important experimental data at certain positions or certain time periods. For experimental tests, the Reynolds number similarity and Froude number similarity cannot be fulfilled in the same time, which will also causes the errors due to the scaled down model.

Thanks to the high-speed developments of computer technology and numerical analysis, in the last two decades, Computational Fluid Dynamics (CFD) has attracted more attentions

because of its high calculation speed. CFD approach has made a tremendous contribution for scientists to explore, investigate and understand many scientific and technological fields and models, including particle tracing model, chemical combustion process and fluid flow simulations (Shen *et la*, 2014). Currently it is also possible to simulate the whole processes of some phenomena or physical principles in a pure virtual laboratory based on high developed computer technology and mature numerical methods.

On the other hand, in some cases of traditional experiments, it is lack of data information due to the expensive setup cost and limitation of experiment conditions like the limited measurement technology. However, for CFD method, take the flow simulation as an example. It is able to provide insight into the flow characteristics and all the data at any time step or period can be obtained from the simulation results easily. Another advantage of using CFD approach can be seen in the scaled up or down model tests, the properties of the material can be set any values in the code file while sometimes it is impossible to find the material of the specific properties, such as the gas density at a certain pressure. In addition, it is important and necessary to get the features of fluid flow in time to better understand and solve the resistance problems efficiently. Neither can be done by empirical formulas nor experimental tests.

In conclusion, there are a number of advantages for CFD simulation, which are listed below:

- CFD modeling is always faster than physical modeling, especially for the complicated geometries.
- CFD simulations can run with the computer automatically after setting the case, which is generally 20-40% less than a comparable physical model effort (Linfield and Mudry, 2008).
- CFD models are able to modify the model scales easier and it is possible to make any model scales.
- With the advanced post-processing tool, it is easy to obtain, view and measure the calculated data at any time step or position for CFD method.
- CFD modeling is much cheaper than the physical modeling, especially for a series of cases in different conditions.

The CFD computational simulation has been widely used to evaluate the performance of a new vessel in the preliminary design stage, such as the prediction of the resistance by solving Navier-Stokes Equations. In this thesis, the simulation is carried out by using open-source

package OpenFOAM (Open source Field Operation And Manipulation). The nature of OpenFOAM is a CFD toolbox based on object-oriented program, so its code can be easily reused and improved by the users themselves. Moreover, for other commercial solvers, such as Fluent and CFX, they are much more like black boxes, which are hard to go inside to understand and modify the original codes to satisfy the users' case requirements. It is more flexible for open source software package to set the initializations and geometries.

Although the simulation can run faster in the large cluster, the fact is that it is still impossible to get the analytical solution of Navier-Stokes Equations, only in simple 2D flow cases. Numerical CFD methods are required with the high demands of computer resources, especially in the case of turbulence model or complicated geometries. The main characteristic of CFD method is the temporal and spatial discretizations and iterative calculations, thanks to the high speed CPU running. With the progresses of the computer technology and numerical approaches, in the near future, CFD will become more mature tools to solve more physical problems effectively, including the wave flows.

1.2. **JBC & KVLCC2**

The main objective of the thesis is to obtain the total resistance of JBC (JAPAN Bulk Carrier) bulker as well as its components, such as frictional resistance, pressure viscous resistance and wave-making resistance. JBC is a kind of cape-size bulker with single-screwed propeller, which was designed by National Maritime Research Institute, Ship Building Research Center of Japan and Yokohama National University. It is a new test hull in the 2015 Workshop on CFD in Hydrodynamics. Only few investigations regarding CFD prediction of resistance and propulsion have been done by (Korkmaz, 2015). In order to predict the resistance of JBC bulker with a CFD method, a suitable mesh configuration has to be found. This means, there are two important points that need to be considered for CFD numerical simulation: the accuracy of the calculated results and the computing time. Therefore, in this thesis a reference ship with similar hull form called Crude Oil Carrier KVLCC2 is selected for mesh study before running the simulation for JBC bulker.

KVLCC2 is a popular and mature ship hull studied by many CFD researchers and many information data can be found as well as the existing towing tank data, such as the resistance coefficients, wave pattern and the wave elevations.

The influences of different turbulence models, the realizable $k - \epsilon$ model, SST $k - \omega$ model and Spalart Allmaras models are employed to study the accuracy of CFD calculations of flow field and velocity distributions over a hill (Furbo *et al*, 2009). In terms of mesh types, H-O grids are employed to study the double-body flow around KVLCC2 with up to around 6 million cells (L, Eca *et al*, 2010) and a more refined grid resolution using DES method with around 300M mesh cells is done to predict the velocity distributions at the propeller planes (Xing *et al*. 2010). In addition, seakeeping analysis like pitch angle and heave motion of KVLCC2 are investigated by (Manzke and Rung, 2010) with viscous / inviscid-coupling approach. (Sadat-Hosseini *et al*, 2010) compared the numerical calculation for heave, pitch and added resistance in both free and fixed conditions and found no significant differences. The wave pattern and resistance at the design velocity were measured by (W.J. Kim, 2001) in calm water condition and these results are recommended to check the validation of the CFD calculations. The calculated results obtained from OpenFOAM solver in this thesis are also compared with the measured data from W.J. Kim.

The thesis starts with the overview of resistance prediction using CFD method and the introduction of JBC and KVLCC2 ships. Chapter 2 gives the general assumptions for CFD simulation and most important theories used in this thesis, including the resistance decomposition, governing equations, free surface discretization and the turbulence model. Then, all the boundary conditions and the initial files will be introduced in Chapter 3. In this part, the computational domain will be specified and the whole meshes will be generated. Chapter 4 gives the improvements of the mesh refinement and the obtained results are analyzed based on post-processing tool paraFoam. The grid convergence will be investigated in Chapter 5, where the predicted resistance components and the wave elevations will be compared with the experimental results. The best-behaved mesh configuration will be used to JBC bulker to get the final converged resistance prediction in Chapter 6. At the end, Chapter 7 shows the discussion and conclusion of the entire project.

2. THEORY

2.1. Assumptions

The navigation of a vessel, by nature, is a kind of reflection and diffraction problem and its principle is similar to the airplane flying in the air, the only difference is the density of the mediums. In order to better study the project, the simulation model is implemented based on the following assumptions:

1. The flow is 3D turbulent but steady;
2. The temperature is not taken into account, which means the value of water viscosity and the densities of both air and water are constant. Any heat transfer will be neglected.
3. All the simulation cases are performed in bare hull condition without propeller or other appendages and the calculated results are compared with experimental data in the same operational condition.

2.2. Resistance Components

For modern ship design, the requirement of the engine power is one of the most important aspects that should be taken into consideration for the ship designers. During the balance between the engine selection and design speed confirmation, it is necessary to estimate the ship hull resistance once the hull form has been decided and this will help the new designed ship to meet its expected operational requirements. Analyzing different resistance components contributes greatly to the understanding of the CFD simulation. In general, the composition of the total resistance (Eq2. 1-2) is shown in Figure 1.
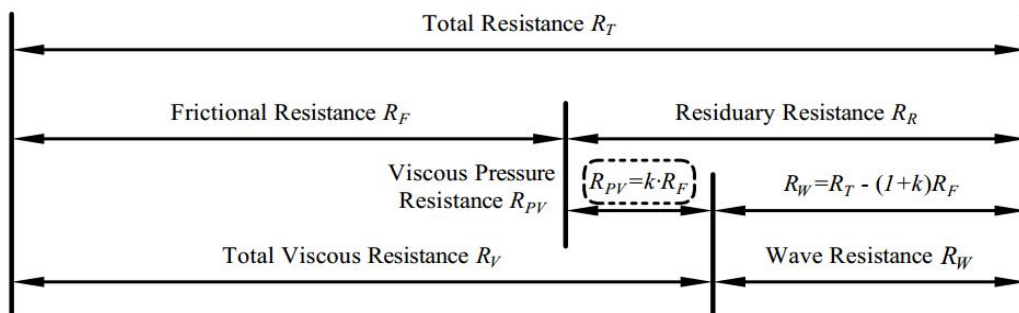
Figure 1. The Decompositions of Total Resistance

$$R_T = R_F + R_R \tag{1}$$

where, R_T is total resistance, R_F is frictional resistance and R_R is residuary resistance. Both of the two components, R_F and R_R, are obtained by using LTSInterFOAM solver in this thesis.

On the other hand, the total resistance can also be decomposed as the following Eq.2:

$$R_T = R_F + k \cdot R_F + R_W = (1 + k)R_F + R_W = R_V + R_W \qquad (2)$$

where, R_V is total viscous resistance, R_W is wave resistance and k is the form factor. The term, $k \cdot R_F$, is called viscous pressure resistance,R_{PV}, which will be discussed later. It can be seen that total viscous resistance consists of frictional resistance and viscous pressure resistance and both values can be obtained from simpleFoam solver based on double-body theory (see Figure 2).
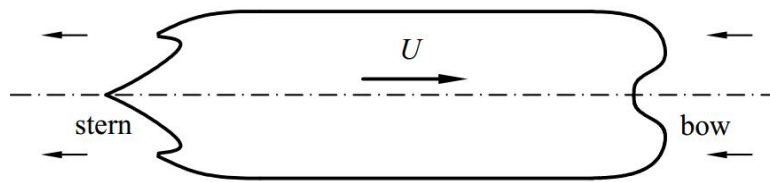
Figure 2. A Sketch of Double Body Model

While a ship is moving through the water, there is a viscous resistance acting on the whole wetted hull surface which causes a drag force opposing the direction of forward motion. This viscous resistance comes from the water viscosity although its value is low; it takes up a large proportion of the total resistance especially at lower speed. For viscous resistance, it consists of tangential and normal components, as can be seen in Figure 3.
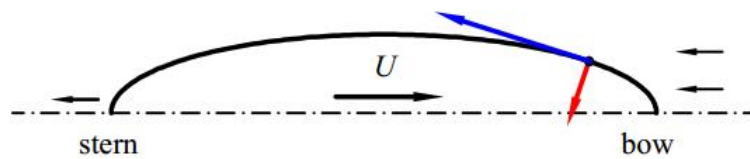
Figure 3. Tangential and Normal Resistance Components of a Local Point

1.   Superficial Frictional Resistance R_F

The tangential component (blue arrow in Figure 3) is called superficial frictional resistance. In essence, superficial frictional resistance is a shear force, so it is parallel to the ship hull at any point of wetted hull surface. Since it is a function of hull wetted surface area, fluid viscosity and surface roughness, it is assumed that its value can be calculated by the flat plate experiment. It can also be estimated by using the ITTC-57 formula (Eqs. 3-4):

$$C_f = \frac{0.075}{(log_{10}R_n - 2.0)^2} \qquad (3)$$

$$R_F = C_f \times \frac{1}{2}\rho S U_2 \tag{4}$$

where, C_f is the frictional resistance coefficient, R_n is Reynolds Number, ρ is the fluid density, S is the wetted surface area and U is the ship speed.

2. Viscous Pressure Resistance R_{PV}

The normal component (red arrow in Figure 3) is the viscous pressure resistance. This resistance component is related to the shape of the hull form. Figure 4 below shows a body completely submerged in two different conditions. The first case is in ideal flow, Point A at the bow and Point B at the stern can be considered as stagnation points, where the velocities are zero, so the pressure around these two points are higher based on Bernoulli's Equation. Finally the pressure distribution around the body can be depicted as the green curve. In the forward section, there is a normal force resisting the body motion while in the aft section, there is a normal force assisting the motion. In ideal flow case, these two forces are equal and there is no resistance acting on the body.
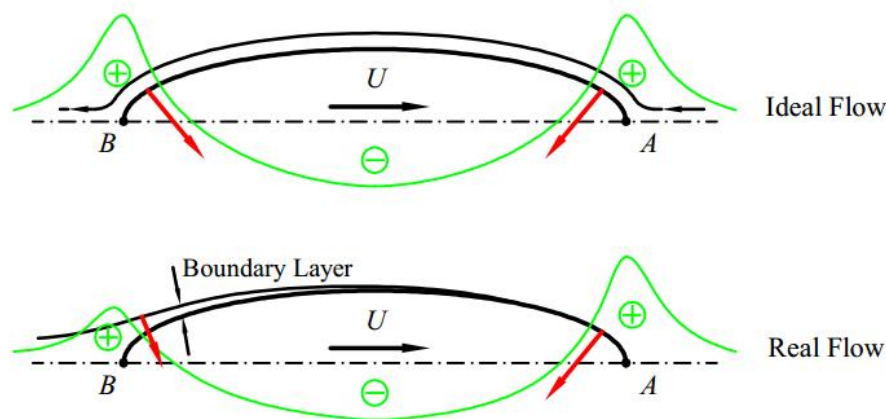
Figure 4. Different Pressure Distributions for Ideal Flow and Real Flow

However, the water in fact is not ideal flow, which is the second case of Figure 4. Therefore there are some differences in reality. Because of the existing of water velocity, the viscous boundary layer has been developed. Most of the time, the field of the aft portion is turbulence flow, and the velocity of the flow around Point B is not zero. So this time the aft pressure field will be lower than that of the first case, which means the pressure distribution along the underwater body has been modified. A high pressure is formed at the front part opposing the motion while a low pressure is formed at the back part assisting the motion. The difference of the forces due to the new pressure distribution is called Viscous Pressure Resistance, and this resistance component is influenced by the hull shape. Generally, the fuller shape ship has larger viscous pressure resistance than slender ship. As mentioned above, the viscous pressure

resistance is expressed as $k \cdot R_F$. In this whole project, the value of k is assumed to be independent to the ship speed.

After obtaining the values of both frictional resistance and viscous pressure resistance, it is possible and also necessary to take a look at the wave profile. If all the resistance components as well as the wave patterns are closer to the experimental data, then the CFD results can be considered to be reliable.

2.3. **Governing Equation**

In any domain, the fluid problems, including the turbulence flow, must be satisfied with the three basic conservative equations, namely mass, momentum and energy conservations, which are the theoretical foundations to investigate the ship hydrodynamics. Since the energy mentioned here is mainly mechanical energy and the thermodynamics is not considered in this thesis, then the energy conservation can be out of discussion.

1.   Mass Conservative Equation

Rate of mass increase in fluid element is equal to the net rate of mass flow into element. Mass conservative equation is also called the continuity equation that describes the transport of a conserved quantity, which can be expressed as Eq.5:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} + \frac{\partial (\rho w)}{\partial z} = 0 \tag{5}$$

where, ρ is the fluid density, u, v and w are the velocity components in Cartesian coordinate system.

In this project, the density of water is assumed to be constant because of the fluid incompressibility, so the first term can be negligible and the above equation can be simplified into Eq. 6:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{6}$$

2.   Momentum Conservative Equation

In essence, the conservation of momentum equation (i.e. Navier-Stokes equation) is derived from Newton's second law to the motion of viscous fluid. The change of the momentum per unit volume equals the sum of forces acting on this unit volume per unit time (Eqs.7).

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial x} + \frac{\mu}{\rho}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) + f_x \tag{7-1}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial y} + \frac{\mu}{\rho}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right) + f_y \qquad (7\text{-}2)$$

$$\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial z} + \frac{\mu}{\rho}\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) + f_z \qquad (7\text{-}3)$$

where, P is the fluid pressure, μ is the dynamic viscosity and f_x, f_y and f_z are the external force components in x, y and z direction, respectively. There is no f_x and f_y in this problem, and f_z is the gravitational force.

Given the initial and boundary conditions for laminar flow, the problem can be solved smoothly by using the above equations to obtain the unique exact solution. However, it is almost impossible to get the steady solution for turbulence model because of the mesh accuracy and computation time for eddies. In this reason, the RANS equations and turbulence model are applied to solve the turbulence case, which will be discussed later.

2.4. **VOF**

In two-phase interaction simulation, such as liquid and air, the main focus is on the numerical expression of their free surface interaction. In General, there are two main strategies to address this problem (Martin, 2014), one is to keep the mesh fixed and try to figure out the position of the interface which is called surface capturing method in Figure 5-(a) and the other is to use the deformed grids followed by the motion which is free tracking method in Figure 5-(b). The former approach is also suitable for modeling ship motion because of the complexity of the free surface, such as wave breaking, so it is more attractive than the later method in the scientific research areas.

Interface approximated by piecewise interpolation from cell volume fraction

Interface Modeled as a Boundary

(a) Volume of Fluid model   (b) Phase interface fitted grid
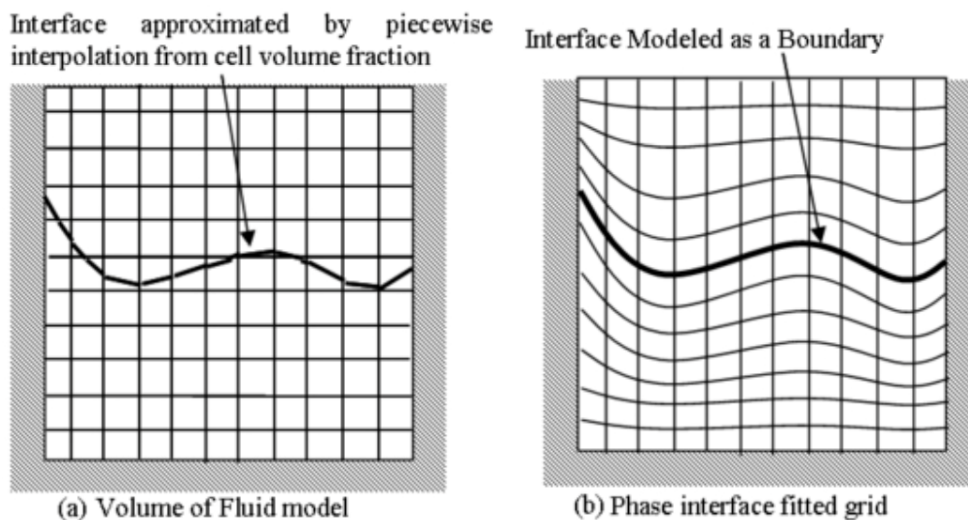
Figure 5. Interface Capturing and Interface Tracking Theories
Available From 6th *OpenFOAM Workshop* (Maki, 2011)

Volume of Fluid (VOF) method is a typical surface capturing method, which uses a volume fraction function α to record the ratio of the liquid volume to the whole volume of each cell. The volume fraction function α is specified to be 1 if the cell is full of liquid while it is set to be 0 if the cell is empty of liquid. When the value is between 0 and 1, it means there is an interface cutting the cell. The great advantage of this method is that the solution can reach acceptable accuracy with minimum storage requirements as well as short computational time. The two phase problem can be converted to the mixed single phase flow, and the density ρ and dynamic viscosity μ of the blended one can be expressed as Eqs. 8-9:

$$\rho = \alpha\rho_{water} + (1 - \alpha)\rho_{air} \tag{8}$$

$$\mu = \alpha\mu_{water} + (1 - \alpha)\mu_{air} \tag{9}$$

The volume fraction function α is governed by the following transport equation (Eq.10):

$$\frac{\partial\alpha}{\partial t} + u\frac{\partial\alpha}{\partial x} + v\frac{\partial\alpha}{\partial y} + w\frac{\partial\alpha}{\partial z} = 0 \tag{10}$$

where, (x, y, z) is the Cartesian coordinate system and (u, v, w) is the velocity components in Cartesian system.

In order to execute the VOF method, two important variables should be calculated, the volume fraction function α and the direction of normal vector of each cell. The α field at the 0 time step has already been initialized and it is known that the normal direction of the interface is the direction where the α changes most sharply. When both are properly computed, a planar cutting plane can be uniquely determined, see Figure 6. Finally all the planes can be constructed to approximate the entire free surface.
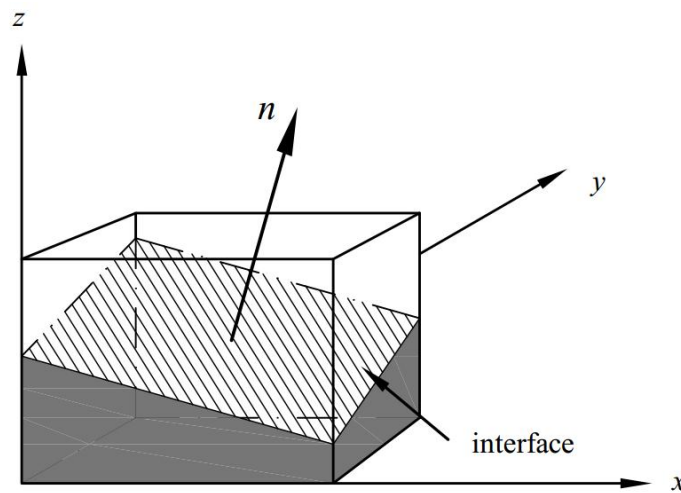
Figure 6. Reconstruction of the Interface for One Cell using VOF Method

Then this free surface is propagated by the flow, which makes it possible to calculate the variations of volume, mass and momentum fluxes from and into the neighboring cells. After that, a new volume fraction function α field is obtained as well as the normal vector of the interface, at the same time all the other parameters like mass and momentum fields are updated. In the end, the simulation is able to keep running for the next iteration.

The VOF method provides a fast and effective approach to solve the two-phase free boundaries in this thesis and the whole free surface can be obtained by reconstructing the interface based on the volume fraction function at iterations.

2.5. **RANS Equations**

It is well-known that the majority of natural flow is turbulent and it is easy to observe the turbulence phenomenon, which can occur in gas, liquid, or even solid. In fact, any flow will become unstable when the Reynolds number reaches to a certain value, which is the physical nature of all the flow problems. Since the study and simulation of turbulence models play a significant role in human life, the step in this area has never stopped for scientific researchers. Until now great progresses have been made in this area and there are three basic approaches to model turbulence:

1. RANSE (Reynolds-Averaged Navier-Stokes Equation)
2. LES (Large Eddy Simulation)
3. DNS (Direct Numerical Simulation)

In this thesis, the simulation will rely on the first approach, which is the most popular CFD simulation method nowadays in industrial and engineering areas.

The Comparison of Three Basic Turbulence Approaches
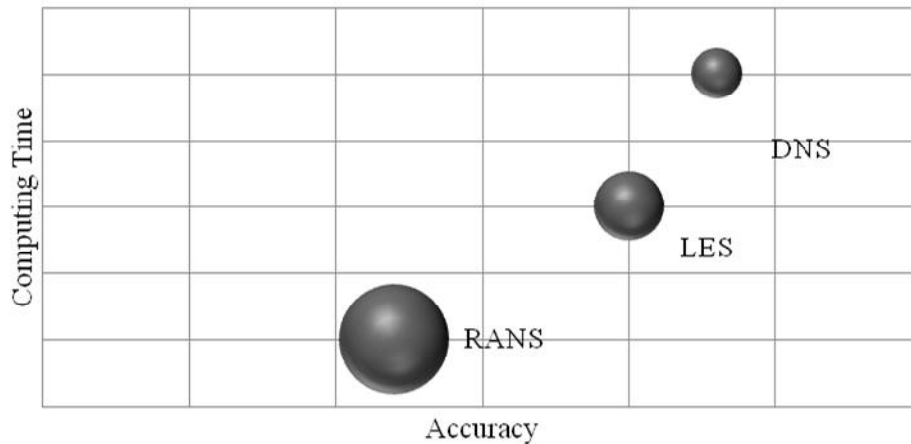
Figure 7. Computing Time vs. Accuary for RANS, LES and DNS

The differences of these three turbulence approaches are depicted in Figure 7 with respect to Accuracy and CPU requirements & Time. Direct Numerical Simulation (DNS) is the one that attempts to fully resolve the flow only by solving Navier-Stokes Equation, the problem of this approach is that it costs a huge computation resource and takes really a long time, which is not appropriate in the practical case. Large Eddy Simulation (LES) is the one that attempts to partially resolve the turbulence problem. The fundamental idea of this approach is to resolve the large scale eddies by the governing equations, for the small scale eddies, they are quite universal and can be filtered out and modeled by empirical methods.

Today the most practical approach for resolving the turbulence model is by way of the Reynolds Averaged Navier-Stokes Equations (RANSE), which is a kind of unresolved method. In RANSE solver, the conservation equations of mass and momentum are solved by using integral forms of Finite Volume Method. RANSE method helps to save a lot of computation time and the accuracy of the results is still acceptable. This is the main reason why most of the engineers and researchers are keen on this start-of-the-art approach.

For RANSE method, the main idea is the decomposition of the instantaneous velocity potential (see Figure 8). It is shown that the instantaneous velocity potential of the flow Φ can be separated into a mean value $\overline{\Phi}$ and a fluctuating component Φ' (Eq. 11).
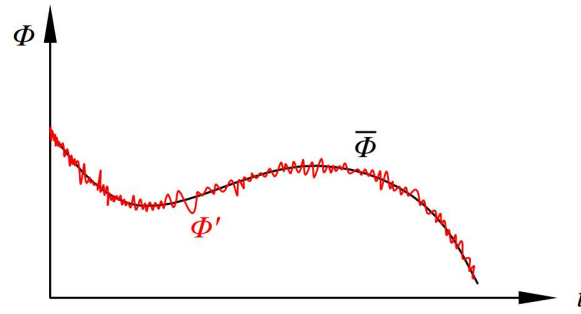
Figure 8. The Decomposition of the Instantaneous Velocity Potential

$$\phi(x,t) = \bar{\phi}(x,t) + \phi'(x,t) \tag{11}$$

The averaging process may be concerned as a time averaging steady-state situation while the latter term may be thought of repeatable transient conditions. So the final equations for the mean quantities are exactly identical to the original ones, the only difference is one additional term occurs, which is the term known as Reynolds Stress Tensor for fluctuating component. Then the original RANS equation can be transformed to the following type (Eq.12):

$$\rho\frac{\partial\vec{\bar{v}}}{\partial t} + \rho\overline{\overline{grad}\vec{\bar{v}}} \cdot \vec{\bar{v}} = \rho\vec{g} - \overline{grad}\bar{P} + \mu\vec{\Delta}\vec{\bar{v}} + \vec{R} \tag{12}$$

where, $\vec{R}$ is the so-called Reynolds Stress Tensor term and it is easy to explain this term. There is always stochastic, chaotic motion of the fluid particle and its velocity and pressure are always fluctuating around the average range thoroughly the whole simulation domain. Velocity fluctuation is the origin of the additional stresses of the fluid and this is expressed as Reynolds Stress $\vec{R}$. Reynolds Stress Models (RSM) includes several effects which are not easily carried out by Eddy-Viscosity models. On the other hand, RSM often demands a significant increase in computing time partly due to the additional equations but mostly due to reduced convergence. This additional effort is not always justified by increased accuracy.

So the key problem now is simplified to calculate the Reynolds Stress Tensor, which is a great idea to avoid the simulation of different sizes of eddies and still to be able to represent the turbulence characteristics. Although the real detailed turbulence pictures cannot be provided from any turbulences models, it can represent the main turbulence characteristics and help to get the suitable simulation results. Several mature turbulence models are found by scientists until now, including $k - \varepsilon$ model and $k - \omega$ model.

2.6. *k-ω* SST Model

As mentioned above, the main target of the application of RANSE method is to figure out the Reynolds Stresses Tensor term and it is known that a lot of popular turbulence models can be

applied for this turbulence closure. In this project, the SST (shear-stress transport) $k - \omega$ turbulence model is employed to express the turbulence behaviors. This model first came from (Menter, 1993), who discovered that the ε transport equation from the standard $k - \epsilon$ model could be converted into an ω transport equation by variable alternative.

As it is known, *k-ω* SST model integrates the advantages of both *k-ε* and *k-ω* models with more advanced features. *k-ω* SST turbulence model will behave better in the near wall areas with great near wall grids, especially for the large pressure gradient cases. In addition, the *k-ω* SST model can figure out the Low-Re turbulence model without any additional damping functions.

There are two main significant parameters needed while using the *k-ω* SST turbulent model, the turbulent kinetic energy *k* and specific dissipation rate *ω*, this is also the reason why this turbulence model is called *k-ω* SST model. But before calculating these, it is necessary to obtain two other parameters: the turbulence intensity *I* and the turbulence length *l*.

a) Turbulence Intensity *I*

The Table 1 below lists the reasonable values of turbulence intensity *I* applied in engineering. In this project, the ship is navigating in the calm water with low Reynolds number, which can be considered as medium-turbulence case. The turbulence intensity will be near the range of Medium Intensity, 1% to 5%.

Table 1. General Ranges of Different Turbulence Intensity Cases
Available from http://www.cfd-online.com/Wiki/Turbulence_intensity

Types	Ranges	Comments
Low-Turbulence	< 1%	Flow originating from a fluid that stands still, like external flow across cars
Medium-Turbulence	1%~5%	Flow in not-so-complex devices like large pipes or low speed flows(low Reynolds Number)
High-Turbulence	5%~20%	High-speed flow inside complex geometries like heat-exchangers

b) Turbulence Length *l*

The turbulence length describes only the size of the largest eddy in the flow, which can be estimated through the boundary layer thickness in Eqs. 13:

$$l = 0.4 \times \delta_{99} \qquad (13\text{-}1)$$

$$\delta_{99} = \frac{0.37 \cdot x}{Re_x^{-1/5}} \qquad (13\text{-}2)$$

$$Re_x = \frac{v \cdot x}{\nu} \qquad (13\text{-}3)$$

where, δ_{99} is the thickness of the boundary layer, Re_x is the local Reynolds number, v is the velocity of the inlet flow, x is the length between perpendiculars and ν is the kinematic viscosity of water, $\nu = 1.255 \times 10^{-6} \, m^2/s$ at 11.5°C.

After obtaining these two parameters, it is able to calculate the turbulent kinetic energy k and specific dissipation rate ω

1.   Turbulent Kinetic Energy k

The turbulence kinetic energy is estimated by using the turbulence intensity (Eq.14):

$$k = \frac{3}{2}(U_\infty \cdot I)^2 \qquad (14)$$

where, U_∞ is the velocity of the flow at infinite distance of the computation domain, the value can be considered as the inlet flow speed, and I is the turbulence intensity.

2.   Specific Dissipation Rate ω

The specific dissipation rate can be calculated through the turbulence kinetic energy and the turbulence length (Eq. 15):

$$\omega = \frac{k^{1/2}}{C_\mu^{1/4} \cdot l} \qquad \text{with } C_\mu = 0.09 \qquad (15)$$

The values of k and ω will be applied in the file initialization at the beginning and it is worth noting that both k and ω should be updated for different velocities of the flow in different cases.

3. CASE SETUP

OpenFOAM is a free, open source software package and the vision OpenFOAM-2.2.2 is adapted in this thesis. As a C++ library, OpenFOAM is subdivided into two main categories: solvers and utilities. Solvers are designed to figure out specific physical problems, like LTSInterFoam and simpleFoam used in the thesis, while utilities are created to execute tasks, such as setFields, createPatch etc. This makes OpenFOAM sustainable to reuse and extend its original functionality by users or other people throughout the world (Ghasemi, 2014).

The general operation procedures of an OpenFOAM case applied in fluid flow simulations are clearly presented in Figure 9. Similar to the majority of other CFD solvers, OpenFOAM consists of pre-processing utilities, solvers for partial differential equations and post-processing tools [14].

Pre-processing is the first stage for the simulation to set all the input information to the solver, such as the computational domain and the boundary conditions (Subsection 3.1), mesh generation (Subsection 3.2) as well as the file initializations (Subsection 3.3). All the input data should be in a suitable format, which can be read by the solvers.
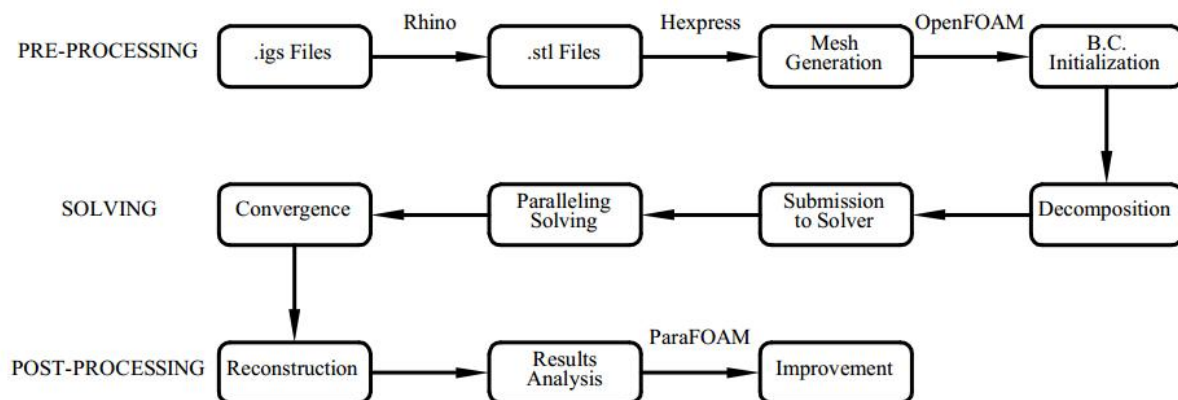
Figure 9. Typical Flow Chart for CFD Simulation by OpenFOAM

In the beginning, the ship hull form is expressed in *igs* format, which can be converted to *stl* format via 3D modeling software like Rhino. Then *stl* files will be read by commercial mesh generator HEXPRESS, in which the mesh generation is completed here. After obtaining the OpenFOAM mesh, the most important step is to set the correct initial and boundary conditions and start to run the simulation in parallel until reaching the convergence. In the end, the post-processing tool called ParaFOAM is used to view and analyze the simulation results from a

physical point of view. All the simulation results must be compared with the experimental data.

Since it is a numerical computation with thousands of mesh cells and iterations, it is unable complete the high demand of computing with the personal computers. To solve this problem, a super computer called VENUS-CLUSTER in the Computer Technology Center at University of Rostock is used.

3.1. **Geometry and Boundary Conditions**

The Geometry information of both JBC and KVLCC2 can be easily found on the websites: http://www.t2015.nmri.go.jp/ and http://www.simman2008.dk, respectively.

It is shown in Figure 10 and Table 2 that JBC is a typical bulk cargo ship with fat hull and the Froude number at design speed is $F_r = 0.142$. Only the full scale is given here, later the scale ratio will be calculated based on the Reynolds number similarity and Froude number similarity. To be more specific, the vortex behaviors of both KVLCC2 and JBC in model scales are the same because of Reynolds number similarity. In the meantime, the predicted results for JBC in model scales can be converted to the data in full scale due to Froude number similarity. More details about the scale ratio calculation can be found in Chapter 6.

Table 2. The Main Geometry and Design Speed of JBC

main particulars		Full scale
Length between perpendiculars	L_{PP}(m)	280.0
Length of waterline	L_{WL}(m)	285.0
Maximum beam of waterline	B_{WL}(m)	45.0
Depth	D(m)	25.0
Draft	T(m)	16.5
Displacement volume	∇(m^3)	178369.9
Wetted surface area w/o ESD	$S_{0\_w/oESD}$(m^2)	19556.1
Wetted surface area with ESD	S_{0_ESD}(m^2)	19633.9
Block coefficient (C_B)		0.858
Service speed		
U	knots	14.5
F_r		0.142

Figure 10. Side View of JBC Bulker Hull Form
Available from http://www.simman2008.dk/KVLCC/KVLCC2/tanker2.html

Figure 11 and Table 3 show the main dimensions and conditions of KVLCC2 resistance tests in model scale 1/58, which was performed by MOERI. The model test was conducted in the same Froude number of JBC.

Table 3. The Main Geometry and Design Speed of KVLCC2

main particulars		Model scale
Length between perpendiculars	L_{PP}(m)	5.517
Length of waterline	L_{WL}(m)	5.612
Maximum beam of waterline	B_{WL}(m)	1.000
Depth	D(m)	0.517
Draft	T(m)	0.359
Displacement volume	∇(m^3)	1.6023
Wetted surface area w/o ESD	$S_{0\_w/oESD}$(m^2)	8.0838
Wetted surface area with ESD	S_{0_ESD}(m^2)	-
Block coefficient (C_B)		0.8098
Service speed		
U	knots	1.047
F_r		0.142

Figure 11. Side View of KVLCC2 Bulker Hull Form
Available from http://www.t2015.nmri.go.jp/jbc.html

The dimension of the domain has a great influence to the computation process and it is vital to set the reasonable dimensions of computational region. Sufficient space around the ship hull should be given to better simulate the wake and lateral Kelvin wave. Since the ship hull is symmetric with respect to the XOZ plane, only half of the domain is considered. Figure 12-(a) displays the configuration of all the dimensions in model scale, so the simulation results can be directly compared with the experimental data. Figure 12-(b) gives the dimensions of the domain and the position of the ship model. The same information is listed in Table 4, where *L* is the length between perpendiculars.

(a)   (b)
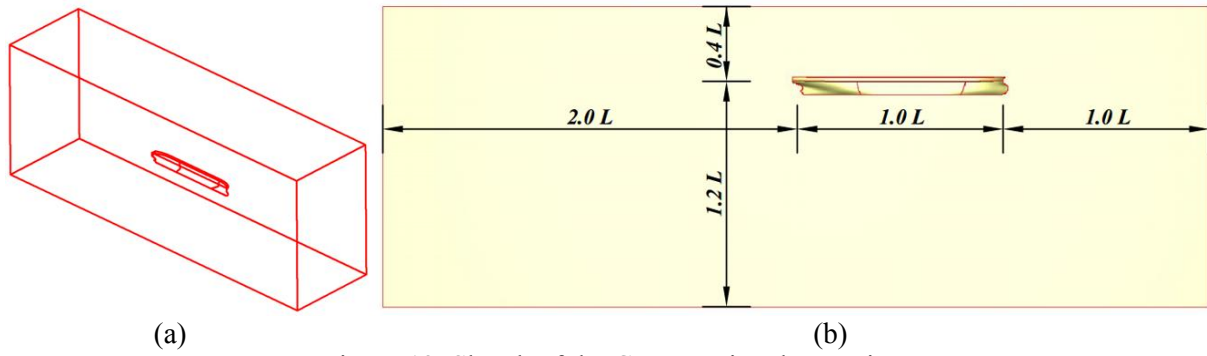
Figure 12. Sketch of the Computational Domain

Table 4. The Dimensions of the Compuational Domain

the dimensions of the computational domain			
Top	0.4L	Bottom	1.2L
Forward	1.0L	Aftward	2.0L
Side	1.5L		

The origin and the coordinate system remain the same as the one set in Rhino while the free surface is set at *z=0*. The stern (Aft Perpendicular) of the ship model is located at *x=0* and the bow (Fore Perpendicular) is situated at *x=L*. The computational domain finally shows the following extent:

$$-2.0L_{pp} < x < 2.0L_{pp}$$

$$0.0L_{pp} < y < 1.5L_{pp}$$

$$-1.2L_{pp} < z < 0.4L_{pp}$$

After the dimensions of computation domain have been defined, it is necessary to set the boundary condition for each patch. There are 7 different patches in total, which are named *in*, *out*, *top*, *bottom*, *farps*, *sym* and *ship* in this thesis. For all the patch files, the considered parameters are pressure, velocity and turbulence characteristics etc. Typically, there are two methods to define the boundary condition: one is to set the scalar value and the other is to specify the gradient value of a certain patch. Both methods can be seen in this project.

In terms of the *ship* patch, it is fixed in the whole simulation process and this is called no-slip boundary condition. In this case, the type of *wall* is given to the boundary type and the velocity is specified as *fixValue*, uniform (0 0 0). The boundary conditions of volume fraction and pressure are defined as *zeroGradient*. *buoyantPressure* is similar to *zeroGradient* but it is specially used for dynamic pressure. For LTSInterFoam solver, the *top* patch is an atmospheric boundary condition. The volume fraction is set to *zeroGradient* and pressure is set to *totalPressue* (Wortley, 2013). This is a balance between the pressure and velocity and the pressure will be accommodated based on the change of the velocity. A default boundary

type called *pressureInletOutVelocity* is applied in the velocity boundary condition. All the *top* boundary condition will change to *symmetryPlane* for simpleFoam case because *top* patch is also a symmetric plane for double-body model. As mentioned above, in order to simplify the problem and reduce the computing time, only half of the domain is modeled due to the symmetry about the XOZ plane. This symmetric plane is called *sym*, where all the boundary conditions are set as *symmetryPlane*. The velocity of the *in* patch is specified as *fixValue*, which is equal to the design speed of the model. *farps* represents the Far field boundary conditions, where all the boundary types are defined as *zeroGradient* to avoid any reflections of the wave effect. Detailed boundary conditions for both LTSInterFoam and simpleFoam are depicted in Table 5 and Table 6.

Table 5. Boundary Conditions for LTSInterFoam Solver

	Alpha1	p_rgh	U	Boundary Type
in	calculated	zeroGradient	fixedValue	patch
out	zeroGradient	fixedValue	zeroGradient	patch
top	zeroGradient	totalPressure	PressureInletOutletVelocity	patch
bottom	zeroGradient	zeroGradient	zeroGradient	patch
farps	zeroGradient	zeroGradient	zeroGradient	patch
ship	zeroGradient	buoyantPressure	fixedValue	wall
sym	symmetryPlane	symmetryPlane	symmetryPlane	symmetryPlane

Table 6. Boundary Conditions for simpleFoam Solver

	P	U	Boundary Type
in	zeroGradient	fixedValue	patch
out	fixedValue	zeroGradient	patch
top	symmetryPlane	symmetryPlane	symmetryPlane
bottom	zeroGradient	zeroGradient	wall
farps	zeroGradient	zeroGradient	patch
ship	zeroGradient	fixedValue	wall
sym	symmetryPlane	symmetryPlane	symmetryPlane

3.2. **Mesh Generation**

This part specially discusses the mesh generation of the ship and computational domain, which is difficult to be carried out by the build-in mesh generators in OpenFOAM, i.e. blockMesh, because of the complexity of the object. The accuracy of the simulation results is

greatly influenced by the quality of the mesh. In this paper a commercial mesher called Numeca HEXPRESS is introduced, which produces unstructured hex-dominant meshes in parallel with high-quality viscous layers.

After the determination of computational domain, a tool called Mesh Wizard inside HEXPRESS is adopted to complete the mesh generation. There are five main steps to build an unstructured hexahedral mesh into a valid HEXPRESS computational domain [16]. In each mesh-setting step the controlling parameters are adjusted to create different grid precisions of the model. The ship hull model and the computational domain are discredited mainly using '*quad*' and '*hex*', only at the special interfaces '*tri*' or '*tet*' grids will be used. Thanks to the principle of unstructured polyhedral cells, it can be deployed to address complex geometries. It is seen that HEXPRESS is a very versatile tool when applied to different boundary configurations.

1. Initial mesh

The HEXPRESS mesh generation process starts from the initial mesh step, where the bounding boxes are used for encompassing the whole computational domain. The subdivision numbers of the domain bounding box are input based on the cell sizes in x, y and z axises and it is better to get close cell lengths in three directions, see Figure 13. All the rest mesh procedures will be implemented based on this initial mesh.
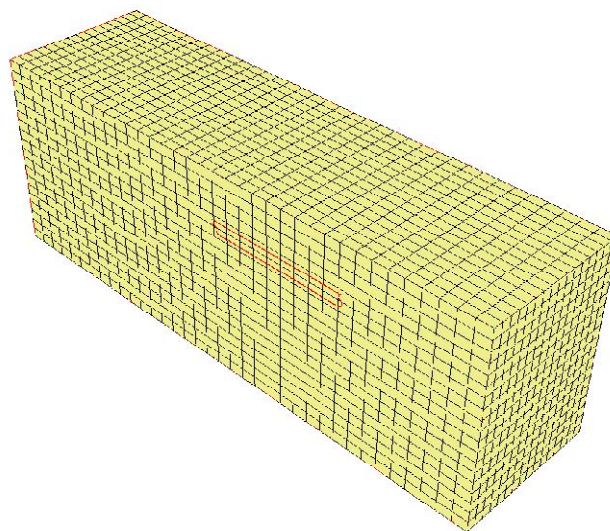
Figure 13. Initial Mesh of Mesh Wizard

2. Adapt to geometry

This mesh action is the most important in all the mesh generation procedures and a lot of parameters in this step directly influence the quality of the final mesh. In terms of curve,

surface and box refinements, the cells are gradually subdivided from the initial mesh based on the numbers of refinement level. Only the cells around free surface area, the refinement criteria is implemented by a specification of cell target sizes.

In addition, the anisotropic cells are used in order to largely decrease the number of total cells and this is done by increasing the aspect ratio from the default value 2 to 50 in free surface refinement part. Figure 14-(a) shows the mesh condition at the stern after running *Adapt to geometry*. It is observed that all the cells intersecting or located outside of the geometry are removed.
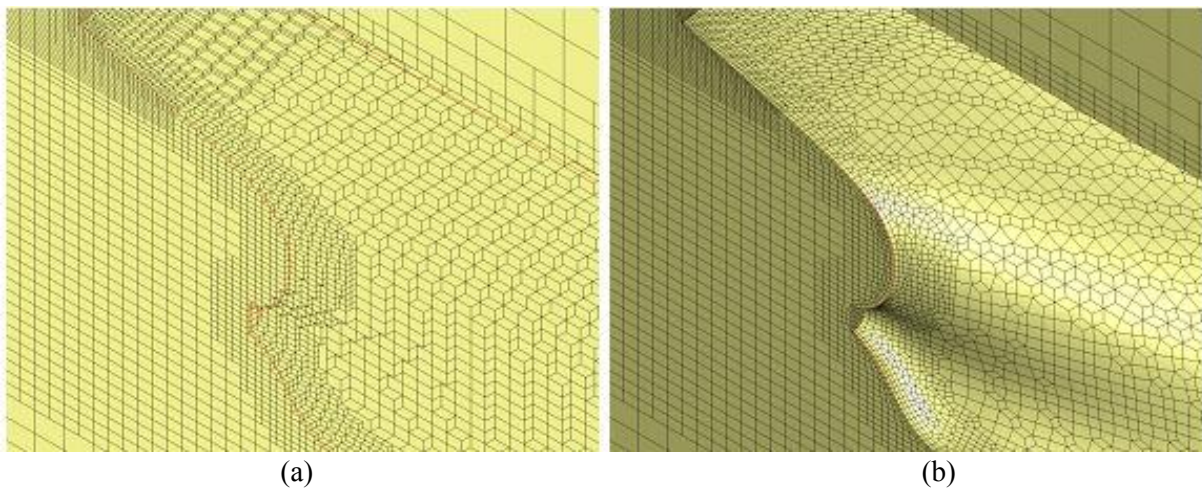
(a)   (b)

Figure 14. Adapt to Geometry and Snap to Geometry

3. Snap to geometry

The main purpose of this step is to project the staircase cells onto the geometry and preserve the small dimensional geometric features. It is not necessary to snap all the cells to the curves, for this case, some of the curves can be selected as *can be skipped* or *must be skipped* such as the intersection line between parallel middle body and the aft part. Moreover, it is easy to create highly distorted mesh while snapping and this is solved by layers of additional cells (buffer insertion). Figure 14-(b) presents the mesh condition after *snapping to geometry*.

4. Optimization

After the snapping action, it is still possible to create some distorted cell, especially around the corners or the regions with higher curvature. In most cases, concave or negative volume cells will appear in this area. *optimization* action is introduced to solve this problem by converting the concave cells into convex ones.

5. Viscous layers

Finally, high aspect ratio viscous layers are carried out to resolve the boundary layers near the ship hull. In this project, the $y+$ value is always set as 100, and around 3~5 viscous layers are used. The difference before and after the viscous layers action can be observed in Figure 15.
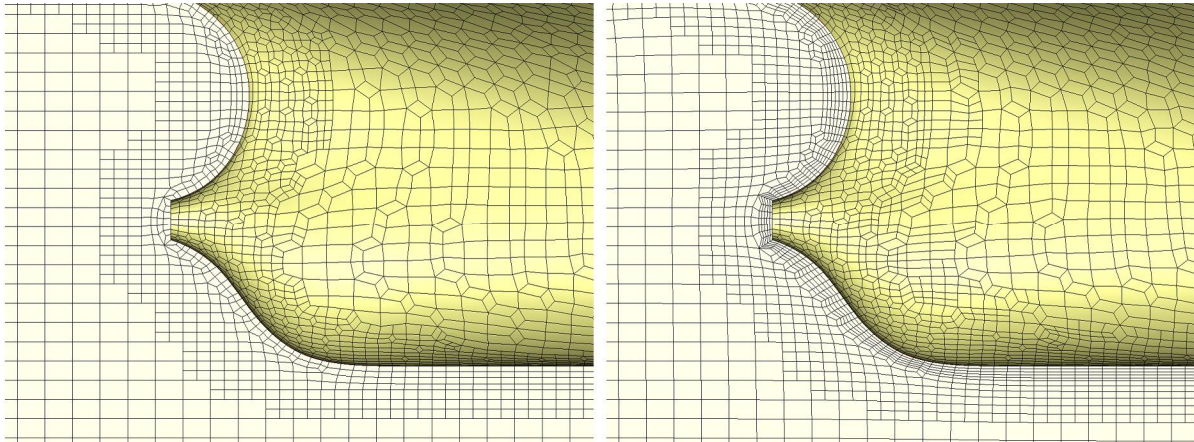
Figure 15. The Stern Part of the Mesh without / with Viscous Layers

Figure 16. The Final Panel of Mesh Wizard
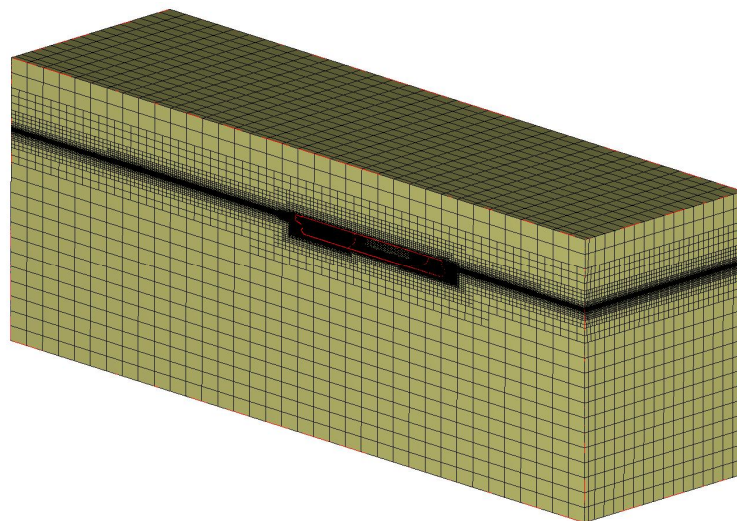
Figure 17. The Mesh Domain Generated from HEXPRESS

For all the five steps, each action should be successfully completed before moving to the next step. The final panel of Mesh Wizard is shown in Figure 16 with all five green ticks and Figure 17 gives the mesh domain after all the actions completely.

It is important to assess the quality of the completed meshes obtained from HEXPRESS. Orthogonality, aspect ratios and expansion ratios of the mesh are the top three aspects that should be taken into consideration. In theory, it is better that the orthogonality is closer to 90 degrees and both of aspect ratios and expansion ratios are closer to 1. The mesh quality assessment can be seen directly from HEXPRESS or with the help of checkMesh utility in OpenFOAM. The assessment for 2.65 M cells is reported in Figure 18:

(a)

(b)

(c)
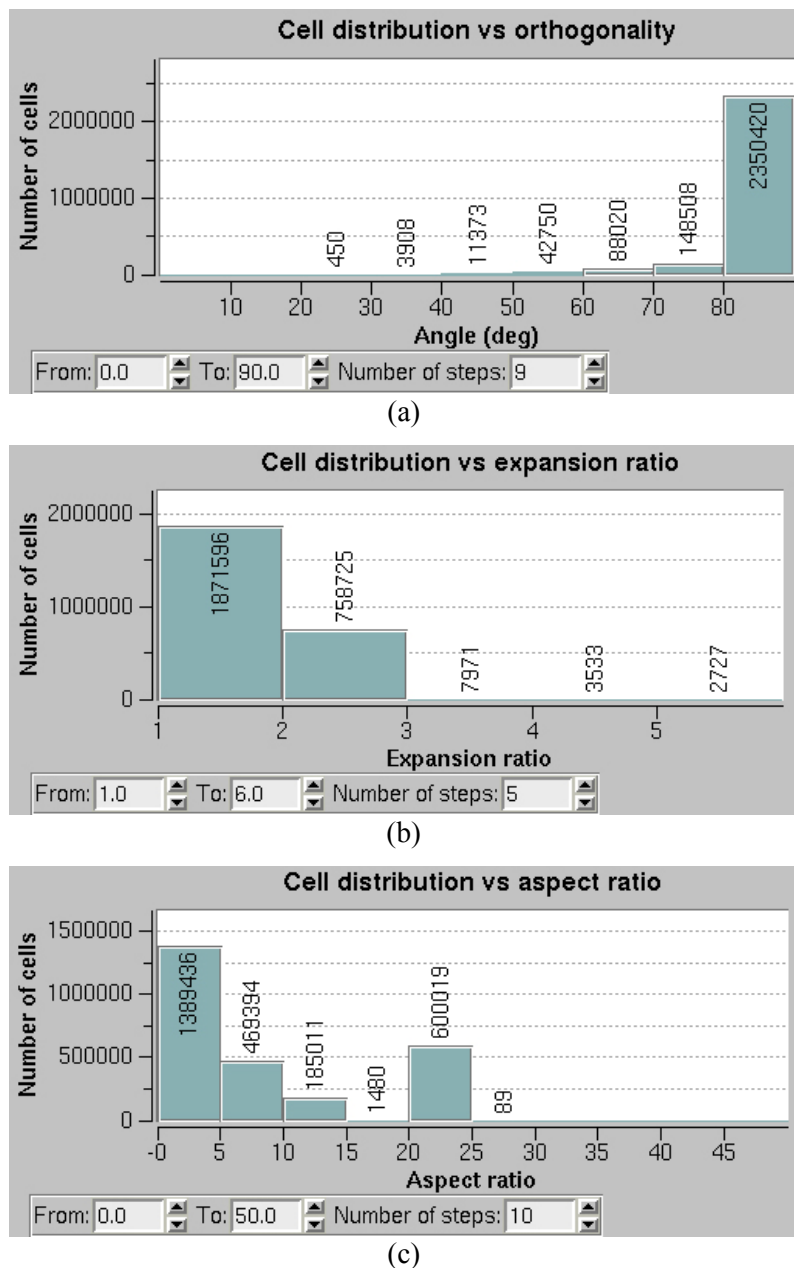
Figure 18. Cell Distributions vs. Orthogonality(a), Expansion Ratio(b) and Aspect Ratio(c)

From the figures, it is known that around 89% of the orthotropic angles are between 80 to 90 degrees and 71% of the expansion ratios are less than 2. For aspect ratios part, most of the ratios are within 5 while still a big portion, approximately 23% of the total cells are higher than 20. This is logical due to the discretization of free surface part in order to largely reduce the number of cells (the allowable maximum aspect ratio modified from 2 to 50). From the above mesh generation, it is seen that HEXPRESS is an advanced unstructured hexahedral mesh generator used to create the high-quality mesh for later CFD solver.

3.3. File Structure

OpenFOAM manages all the data with the expression of dictionary files and the main file structure is quite identical to that of any other cases, which can be seen in the following Figure 19. Thanks to the openness of OpenFOAM, experienced users enable to create new applications to meet their specific requirements. The best way to start a simulation case is to find and copy the original files from any other cases which include the necessary parameter files (e.g. from tutorial cases) and only modify them to satisfy the new case condition. This is a remarkable advantage of OpenFOAM applications. For both LTSInterFoam and simpleFoam solvers used in this project, they should contain the parameter files like k, U etc. one can easily get these files and most of the codes in these files will remain unchanged, except for a few scalar values (for example, kinematic turbulence energy k) and vector values (field speed (U_x, 0, 0) for instance).
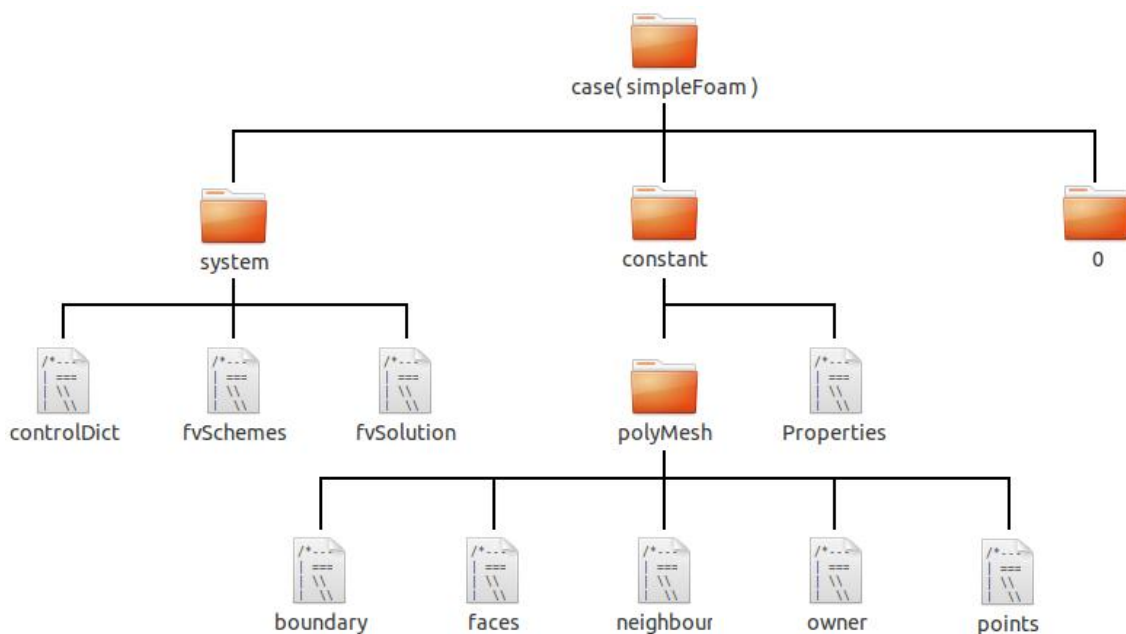
Figure 19. File Structure of OpenFOAM (simpleFoam case)

OpenFOAM manages all the dictionaries based on their own roles and functions. It can be observed that the original *<case>* dictionary consists of three sub-dictionaries: <0> (time dictionary), <constant> and <system>. After running the case, there will be other new time dictionaries, such as <0.1>, <0.2> and <0.3> etc. The names of the created dictionaries depend on the *deltaT* and *writeInterval* in <controlDict> file.

- Time dictionaries include the values of all the parameters calculated at each time step, only the data in <0> dictionary should be initialized in the beginning.
- The <constant> dictionary contains the mesh and all necessary files describe the properties of the flow as well as the turbulence models.
- The *<system>* dictionary embodies the settings for running, such as the discretizations of schemes and solutions.

OpenFOAM solver reads all the above files in the dictionaries and runs the step-by-step simulation. Before going to the details of each file, it is necessary to understand the dimensions in OpenFOAM files. From Table 7, it is known that all other quantities can be derived from the following seven fundamental physical quantities.

$$dimensions \quad [\ kg\ m\ s\ K\ mol\ A\ cd\ ];$$

Table 7. Seven Fundamental Physical Quantities

No.	Property	Symbol	Unit
1	Mass	kg	kilogram
2	Length	m	meter
3	Time	s	second
4	Temperature	K	Kelvin
5	Quantity	mol	moles
6	Current	A	ampere
7	Luminuous Intersity	cd	candela

In the next part, the primary files in each directory are introduced. Although different solvers contain different parameter files, most files in both LTSInterFoam and simpleFoam cases are similar to each other. LTSInterFoam is a local-time stepping (*LTS*) steady-state solver for 2 incompressible, isothermal immiscible fluids using a VOF (volume of fluid) phase-fraction based on interface capturing approach, while simpleFoam is also a steady-state solver for incompressible, turbulent flow of non-Newtonian fluids.

One thing to note here is that there are numbers of parameters in each OpenFOAM files, part of the parameters are kept default values, which are the same as the original values in the tutorial files. Only those parameters that strongly related to final results in each different case will be discussed in this thesis. For the sake of the convenient illustration of the each file content, just part of the key codes are presented here, the majority of the entire important files can be found in Appendix.

3.3.1 / case / 0

The <0> directory contains the initial and boundary conditions as well as the dimensions for all primary parameters, here are *k, ω, p, U* and *alpha1*(only for LTSInterFoam solver). The code for velocity *U* file is give here:

```
//********************************************************************//
demensions                        [0  1  -1  0  0  0  0];
internalField                     uniform (-U_x 0 0);
boundaryField
{
    in
    {
       type                       fixedValue;
            value                 uniform (-U_x 0 0);
    }
    …
}
//********************************************************************//
```

dimensions [0 1 -1 0 0 0 0]; states that the dimension of *U* is *m/s*.

internalField uniform(-U_x0 0); in the CFD simulation, the ship body is assumed to be fixed, then the fluid flows at the ship speed along –*x* direction (-U_x0 0). The same principle can be found for the boundary patch *in*, which is set as *fixedValue*, uniform (-U_x0 0).

The setting in *p* file is similar, but in LTSInterFoam case, the pressure file is *p_rgh*, namely dynamic pressure (p/ρ). *k* and *ω* files are the ones to describe the turbulence characteristics, which have already been discussed in Chapter 2. In general, the values of *k* and *ω* vary with the ship speed and the turbulence intensity. More detailed information about *k* and *ω* will be discussed in Chapter 4.

alpha1 file only exists in LTSInterFoam case. Two steps are involved in setting the initialα field with only values of 0 and 1. Firstly the initial entire computational domain for α field is

set as uniform 0, which means all the areas are considered as air in the beginning. Secondly *setFields* utility is used to specify the region of water by creating a box and the α value inside the box is 1.

```
//************************************************************//
regions
(
    boxToCell
        {
            box                 (-1000   -10000   -1000)  (1000   10000   0);
            fieldValues         (volScalarFieldValue      alpha1   1);
        }
);
//************************************************************//
```

3.3.2 / case / constant

Five sub-files are created from HEXPRESS after exporting to OpenFOAM format mesh and all of them are located in <ployMesh> file, namely: *boundary*, *faces*, *neighbor*, *owner* and *points*. *boundary* shows the definitions of the patches, each of which is specified as one boundary condition. The name of each patch should be identical to those in other files, like *k* and *U*.

```
//************************************************************//
{
    in
    {
      type                          patch;
            nFaces                  3298;
            startFace               16043963;
    }
    …
    ship
    {
      type                          wall;
            nFaces                  191565;
            startFace               16048898;
    }
    …
}
//************************************************************//
```

The example above shows the patch types for *in* and *ship*, which are *patch* and *wall*, respectively. *nFaces* gives the total number of faces for this patch while *startFace* shows the first start face in *face* list. The other files define the points, faces and *neighbor* and *owner* files contain the topological relations between the cells.

The dimensioned scalars *rho* and *nu* are given in *transportProperties* file.

```
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
twoPhase
{
        transportModel              twoPhase;
        phase1                      water;
        phase2                      air;
}
water
{
        transportModel              Newtonian;
        rho             rho     [1 -3 0 0 0 0 0]              1000;
        nu              nu      [0 2 -1 0 0 0 0]              1.255e-06;
}
air
{
        transportModel              Newtonian;
        rho             rho     [1 -3 0 0 0 0 0]              1;
        nu              nu      [0 2 -1 0 0 0 0]              1.48e-05;
}
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
```

It is shown that the main parameters of two-phase transport model are defined and *phase1* is for water while *phase2* is for air.

rho   rho   [1 -3 0 0 0 0 0] 1000; states the water density is 1000kg/m^3.

nu   nu   [0 2 -1 0 0 00] 1.255e-06; states the kinematic viscosity of water is $1.255 \times 10^{-6} m^2/s$

For simpleFoam case, the regions above the clam water plane are removed and the rest parts are doubled along the surface. This is called 'double-body' model and based on this assumption, there is only the definition of water phase in *transportProperties* file.

RASProperties file contains all the parameter coefficients for different turbulence models as well as the wall function and *turbulenceProperties* file states the selected $k - \omega \, SST$ turbulence model.

In terms of gravitational acceleration *g*, it is worth noting that there is no *g* file in simpleFoam case and the reason is the double-body hull and the fluid domain are symmetric with respect to the clam water plane. The entire fluid domain is full of water and the free surface effect is negligible, which is a kind of gravity wave. So there is no influence on gravity in simpleFoam case.

3.3.3   / case / system

controlDict file includes the detailed run-time information to run the case.

```
//************************************************************//
application              LTSInterFoam;
startFrom                latestTime;
startTime                0;
stopAt                   endTime;
endTime                  30000;
deltaT                   1;
writeControl             timeStep;
writeInterval            5000;
…
//************************************************************//
```

The start and end time of the simulation are specified respectively, and the *startTime* is 0 while the *endTime* is dependent on the *deltaT*, whose value is typically 1 for the steady-state cases in this project. In terms of *endTime*, the number should be big enough to assist the simulation to reach its convergence condition. *writeControl* and *writeInterval* are the two parameters to control the output information.

The most significant objective in this CFD simulation is to get different force components at steady state. This can be done by adding the force outputs at the last part of *controlDict* file. Then the force function will be employed automatically to calculate the different force components in each step and all the force information will be stored in a new folder located in /case/postProcessing/forces file.

fvSchemes file specifies different discretized schemes for all the terms in partial differential equations (see Table 8).

Table 8. Different Discretized Schemes

Ddt	time discretization
gradSchemes	gradient term discretizaitons
divSchemes	divergence term discretizations
laplacianSchemes	Laplacian term discretizations
interpolationSchemes	interpolation of values from cell centers to cell face centers
snGradSchemes	surface normal gradient evaluation at cell faces
fluxRequired	list fields where flux is generated in the applicant

```
//************************************************************//
ddtSchemes
{
        default                          steadyState;
```

```
}
gradSchemes
{
        default                         cellLimited Gauss linear 1;
        grad(U)                         cellLimited Gauss linear 1;
        grad(alpha)                     cellLimited Gauss linear 1;
}
divSchemes
{
        div(phi, U)                     bounded Gauss limitedLinearV 1 phi;
        div(phi, omega)                 bounded Gauss limitedLinear  1 phi;
        div(phi, k)                     bounded Gauss limitedLinear  1 phi;
        div((nuEff*dev(T(grad(U)))))    Gauss linear
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

The default for time discretizaion *ddt* Schemes is set to *steadyState* because simpleFoam is a steady-state solver. For gradients of *U* and *alpha*, *cellLimited Gauss Linear 1.0* is used. Then *divSchemes* solve the convective terms, for example, *div(phi, U)* is the convective velocity term and *div(phi, k)* and *div(phi, omega)* are the convective turbulence term, where *bounded Gauss limitedLinear 1.0 phi* is specified.

fvSolution file specifies the solvers to solve the discretized linear equation system. To be more specific, linear solver settings are given for pressure, velocity and turbulence viscosity. Further information regarding the following or other OpenFOAM methods can be found in (Saad, 2003)

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
solvers
{
    p
    {
        solver                  GAMG;
        smoother                GaussSeidel;
        nPreSweeps              0;
        nPostSweeps             2;
        cacheAgglomeration      true;
        nCellsInCoarsestLevel   10;
        agglomerator            faceAreaPair;
        mergeLevels             1;

        tolerance               1e-06;
        relTol                  0.1;
    };
    "(U|k|omega).*"
    {
        solver                  smoothSolver;
        smoother                GaussSeidel;
        nSweeps                 1;
```

```
        tolerance                      1e-7;
        relTol                         0.1;
        };
}
SIMPLE
{
    nNonOrthogonalCorrectors           2;
    pRefCell                           0;
    pRefValue                          0;
}
//*******************************************************************//
```

In this case, it is observed that the GAMG (Geometric Algebraic Multi Grid method) solver is specified for solving pressure p linear equation system while the smoothSolver is used for velocity U and turbulence viscosity (k and ω) linear equation systems. In addition, there are some corresponding parameters regarding each solver, including tolerance and relative tolerance settings. For this case above, the velocity solution is considered to achieve convergence condition while the residual is lower than the tolerance 1.0×10^{-7}, or the residual decreases within 10% by *relTol* at each time step.

The next consideration is SIMPLE algorithmic settings. It is known that SIMPLE algorithmic is used for steady-state solvers while PISO is used for transient solvers. The number of *nNonOrthogonalCorrectors* means how many times the pressure equation will be solved for a single iteration. *nNonOrthogonalCorrectors* can be increased to obtain better converged solutions if the pressure residual does not reduce or even diverge with the increase of iteration steps. In addition, it is observed that pressure residuals and time step continuity errors are reported twice at each step, this is because the *nCorrectors* is 2 in *fvSolution* file.

In the last part of this file, the relaxation factors are specified to determine the convergence rate of the simulation and normally the parameter values are set based on the users' experience. Lower relaxation factors are more likely to get convergence but longer computing time, while higher factors will speed up the running but may result in divergence. The empirical values for pressure are around 0 to 0.4 and the range of 0.6 to 1 is set for velocity relaxation factor.

4.   MODEL IMPROVEMENTS

Chapter 3 mainly focuses on the simulation case setup in order to start running the codes. However, in this part, the simulation model will be improved from several aspects to obtain more effective results. First the relatively accurate turbulence intensity will be determined based on the predicted resistance coefficients compared with experimental results in the early simulation stage. Then, multi blocks will be applied for local refinement at the bow and the stern. In addition, more refined meshes are used in the discretization of the free surface to capture the interface between water and air. Finally the simulation results are analyzed from the physical point of view with the help of post-processing tool paraFoam.

4.1. Determination of Turbulence Intensity

The experimental test of the KVLCC2 carrier was done by MOERI at 11.5 ℃ and the kinematic viscosity of the water was $1.225 \times 10^{-6} \, m^2/s$. Therefore it is easy to get the value of Reynolds number at a certain velocity. For instance, in the design speed of $U = 1.047 \, m/s$ ($F_r = 0.142$) case in model scale, the value of Reynolds number is 4.6×10^6. The resistance components will vary over the values of different Reynolds numbers, especially the frictional resistance. In general, higher Reynolds number means lower resistance and this can also be reflected by the estimated formula of ITTC-57 frictional coefficient C_f [18].

After the decomposition of instantaneous velocity, additional item of Navier-Stocks Equation can be solved by $k - \omega$ SST turbulence model. There are several parameters for expressing the turbulence characteristics, two of which are the most important ones, k and ω, whose values should be set in the initialization stages. For the case of $F_r = 0.142$, the value of Reynolds number will be fix at 4.6×10^6, which means during the whole process of the mesh study , the Reynolds number is equal to 4.6×10^6. On the other hand, turbulence intensity will also affect the values of resistance components. It is necessary to find a way to obtain the right turbulence intensity. This can be done by testing the model with different intensities in the same mesh configuration case, which should be reasonable and sufficient refined.

In general, the values of k and ω will vary with different turbulence intensities. The ranges of turbulence intensity from 0.01 to 0.05 are implemented based on Table 1 in this project and the calculated values of k and ω in $F_r = 0.142$ case are presented in Table 9.

Table 9. Values of k and ω for different Turbulence Intensities at $F_r = 0.142$

F_r	U (m/s)	Re	Theta (m)	Length (m)	I	k (m²/s²)	ω (1/s)
0.142	1.047	4.60E+06	9.49E-02	3.80E-02	0.01	0.000164	0.616692
					0.02	0.000658	1.233384
					0.03	0.00148	1.850075
					0.04	0.002631	2.466767
					0.05	0.004111	3.083459

After setting the values of k and ω, it is possible to restart the set of simulations again. In OpenFOAM there is an in-house force function to integrate the viscous force in each direction and its nature is the integration of the viscous force in x, y and z direction. The value of viscous force in x direction is exactly the frictional resistance of the model at each iteration step, which will be compared with the ITTC57 Frictional Coefficient C_f. Table 10 shows the test results for all cases and error percentages with respect to ITTC57 empirical C_f and experimental C_t, respectively, where the errors are calculated based on Eq. 16.

$$Error\%C_x = \frac{C_{x\_calculated} - C_{x\_experimental}}{C_{x\_experimental}} \times 100\% \qquad (16)$$

Table 10. Resistance Components and Errors with Different Turbulence Intensities

Turbulence Intensity	$C_v \times 10^3$	$C_t \times 10^3$	E%C_f	E%C_t	ITTC57 $C_f \times 10^3$	Experimental $C_t \times 10^3$
0.01	3.330	4.188	-3.48%	1.89%		
0.02	3.376	4.248	-2.13%	3.37%		
0.03	3.436	4.353	-0.42%	5.92%	3.45	4.11
0.04	3.499	4.470	1.43%	8.75%		
0.05	3.551	4.582	2.92%	11.49%		

Noting that the ITTC57 formula of C_f includes other resistance components, not only the viscous force. The calculated value from ITTC57 formula will be a little higher than the experimental case, which means the predicted viscous coefficients C_v from CFD methods are always lower. Then 0.01 0.02 and 0.03 cases can be considered from Table 10, where $C_v < C_f$.
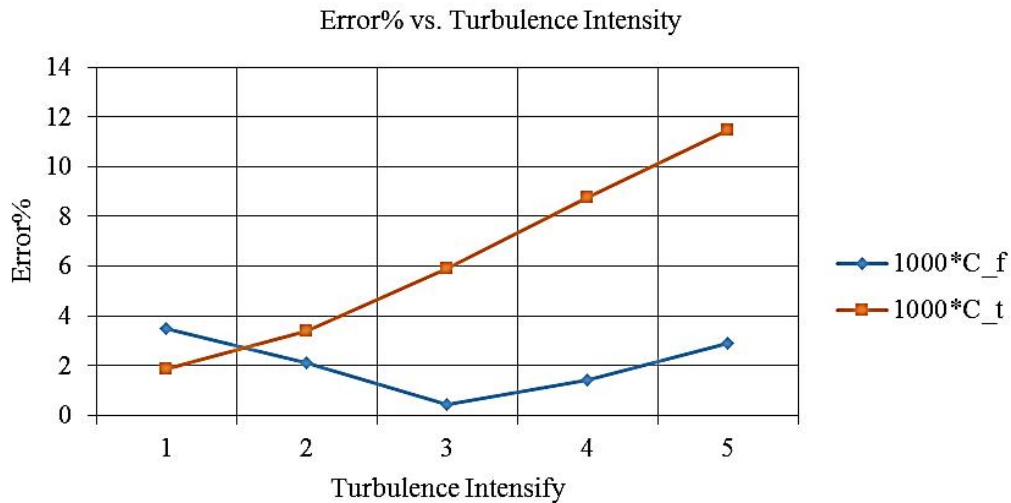
Figure 20. Errors of Viscous and Total Resistance vs. Turbulence Intensity

On the other hand, the total resistance coefficient C_t is also an important parameter that should be considered. Figure 20 represents all the error percentages of C_v (compared with ITTC57 empirical formula) and C_t (compared with experimental result), it is observed that,

- 0.01 Cace: the error of frictional resistance is large in all the three cases.
- 0.03 Case: the error of frictional resistance is the smallest, while the total resistance reaches 6%, which is not acceptable compared with the other two values.

In the end, the 2% turbulence intensity is selected, afterwards the values of the parameters, k and ω, are calculated. These values will be used in the following procedures.

4.2. Discretization of Free Surface

Similar to the determination of turbulence intensity, the model in this stage should also be tested several times. The main objective of free surface discretization is to better express the wave profiles and this can be done by estimating the minimal acceptable mesh sizes in x & y and z directions.

The wave is propagating in horizontal plane and it can be express by harmonic profile. Broadly speaking, it is the superposition of many harmonic functions. So for the discretization of x & y directions (see Figure 21), there should be at least 8 subdivisions in one wave length, in order to better express the harmonic characteristics.
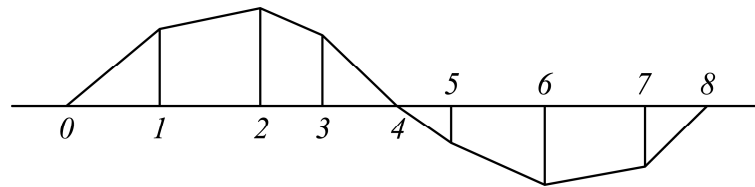
Figure 21. The Minimal Subdivisions in One Wave Length

Since the model is considered moving in depth water, the wave length λ can be calculated by the following equations (Eqs. 17-1, 17-2, 17-3, 17-4),

$$\omega^2 = gk \tag{17-1}$$

$$\omega = 2\pi/T \tag{17-2}$$

$$k = 2\pi/\lambda \tag{17-3}$$

$$UT = \lambda \tag{17-4}$$

where, ω is circular frequency; T is the wave period; k is the wave number and U is the design speed U=1.047m/s.

Then the estimation of the wave length is around 0.7 meter, which means the minimal target size in this case can be 0.7/8=0.087 m. The target size in x direction will be set 0.08m for reference ship KVLCC2 in the end. As x is the main propagating direction and the wave speed in y direction is lower, this will make the wave length shorter in y direction. So the target size in y direction is somehow shorter and the target size of 0.06m is given in y direction.

For vertical direction, The International Towing Tank Conference (ITTC, 2011) said that around 10 sub-cells should be provided for one wave amplitude. The simplest approach to get the wave amplitude is to run one case of simulation with reasonable mesh configuration, and then paraFoam is used to view the free surface distribution in converged situation. Figure 22-(a) and Figure 22-(b) show the wave profiles in the fore and aft parts and it is possible to measure the highest and lowest wave part except the local areas in the stern and bow, where can be locally refined with additional box later.
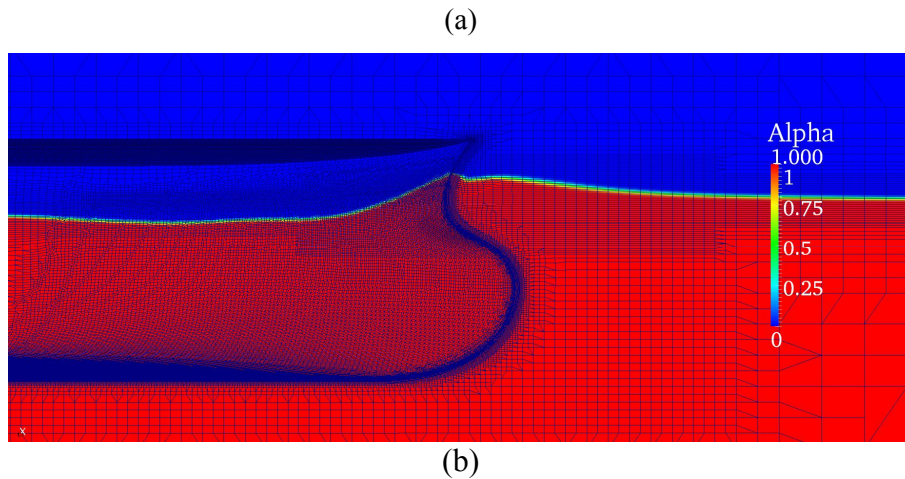
(a)

(b)
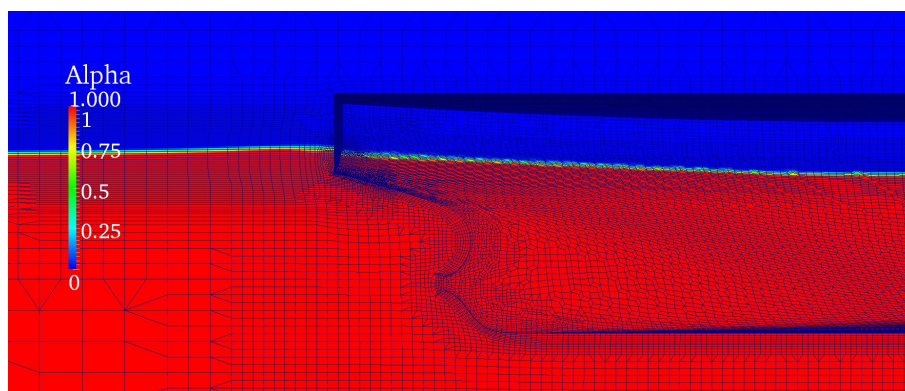
Figure 22. Free Surface Distribution at the Fore (a) and Aft (b) Parts of KVLCC2

After the measurement of two extreme parts (top and bottom parts of the free surface), it is necessary to reset the refinement box, which will decrease the number of cells efficiently. This can be seen in Figure 23 and the entire wave except the part round stern and bow can be limited in the yellow bar regions. In this thesis, the vertical range of this yellow bar is - $0.025m$~$0.025m$ and then the minimal cell in vertical direction is around $0.0035m$. After several tests, it is known that the target size in z direction of $0.004m$ is really enough.

Figure 23. Controlled Regions for Encompassing the Free Surface Elevation

The final target cell size in three directions for refinement mesh cases can be seen in following Figure 24.
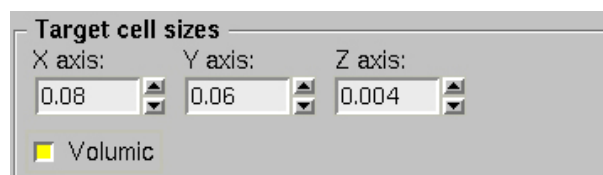

Figure 24. Final Target Cell Sizes in Three Directions for KVLCC2

4.3. **Local Box Refinement**

The meshes of the stern and stem parts are more important compared with the one in middle body due to the higher curvature. Two big refinement boxes and two small ones are used in these areas to get the local refined mesh, which can be seen in Figure 25 and Figure 26.
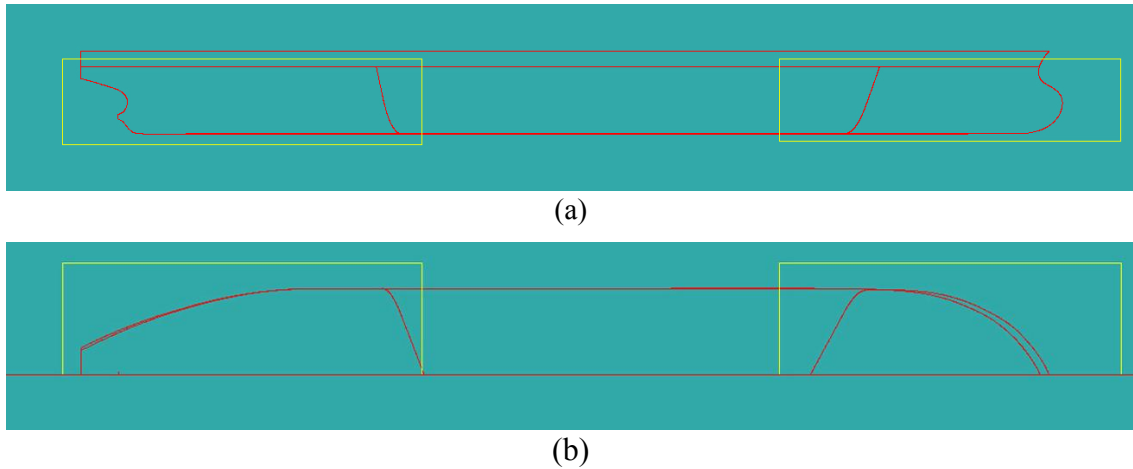

(a)

(b)

Figure 25. Side and Top Views of Large Refinement Boxes at the Stern and Bow

It is not necessary to set very high mesh level inside the big refinement boxes and one level lower than the small box is acceptable. However, the stagnation point at the fore perpendicular part will make the wave profile higher, where the highest point should be lower than the top of the small box. On the other hand, as the wave detaches at the transom, both the wake and the stern wave are sensitive to the resistance prediction. So the refinement small box at the stern is also required.
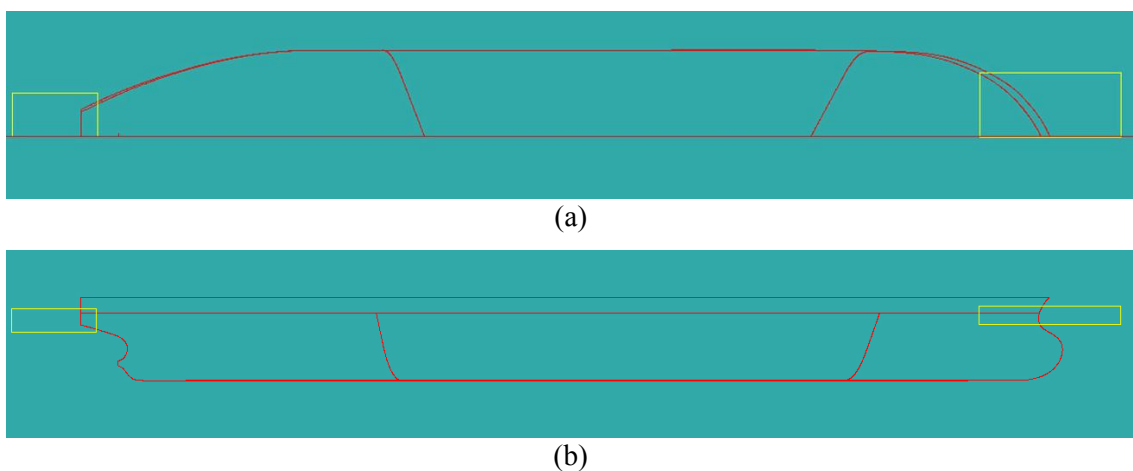

(a)

(b)

Figure 26. Side and Top Views of Small Refinement Boxes at the Stern and Bow

4.4. **Post-processing**

After the domain decomposition and parallel computing, it is necessary to view and analyze the calculated results and this can be done by post-processing tool paraFoam. paraFoam

provides a post-processing environment for users with the help of the Visualization Toolkit (VTK), which is able to read and process the calculated data from OpenFOAM cases visually with the rendering engine. Since resistance prediction is a steady state problem, only the last iterations after convergence need to be considered for post-processing. Alpha, pressure and velocity distributions along the ship hull are the top three parameters that should be investigated and these can be viewed easily by paraFoam. All the visual results will be analyzed from a physical point of view.
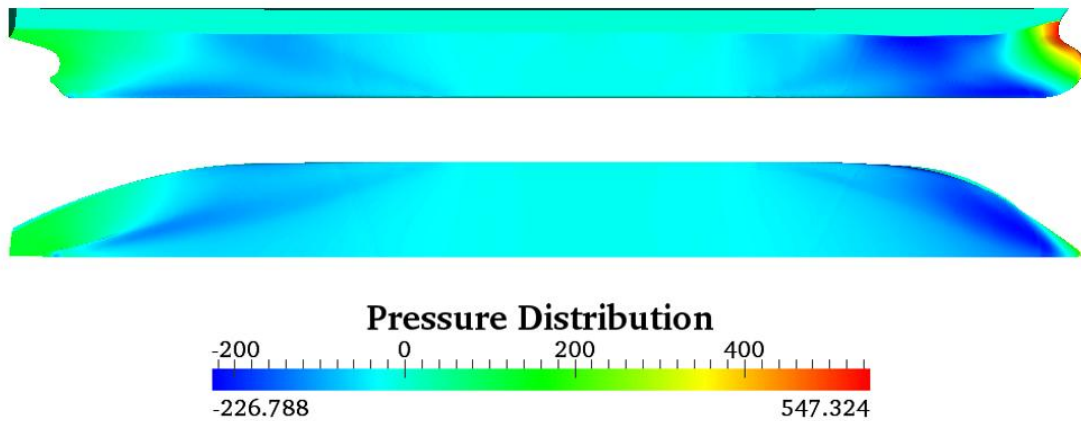

Figure 27. Pressure Distributions around KVLCC2 at $F_r = 0.142$ Case

Figure 27 depicts the pressure distribution over the KVLCC2 hull with the design speed ($F_r = 0.142$) and 0.02 turbulence intensity. The part around the bow gives the highest pressure due to the decrease of the wave speed and the highest value at the stagnation point will reach 547.3 N/m^2, which can be read from the color bar. It is also possible to check the value of the highest pressure from Bernoulli's Equation (Eq.18):

$$P + \frac{1}{2}\rho U^2 + \rho gh = P_{far} + \frac{1}{2}\rho U_{far}^2 + \rho gh_{far} \tag{18}$$

where, P is the pressure; ρ is the density of the water; U is the speed of the wave particle and h is the height of the wave. The subscript *far* means far away field of the domain.

Taking into account the horizontal streamline which crosses the stagnation point, the height of the streamline is constant and the speed of the water at the stagnation point is 0. So the pressure can be calculated from Eq.19

$$P = \frac{1}{2}\rho U_{far}^2 = 0.5 \times 1000 \times 1.047^2 = 548.1 \; N/m^2 \tag{19}$$

This value has a good agreement with the highest pressure shown in the color bar.

In addition, it is observed that the pressure will reduce from the bow to the shoulder of the hull and the reason is that the speed of the wave particle will increase from the stagnation point to the shoulder part. The same phenomenon can be seen in the aft part of the ship hull. For the parallel middle body, the speed of the water along x direction is near the incoming speed, so the dynamic pressure (p/ρ) is 0.

The corresponding results can also be seen from the velocity distribution around the hull. The velocity and its components at free surface are shown in Figure 28, it is observed that all the areas with lower pressure in Figure 27 have higher velocity and vice versa. To be more specific, the highest velocity appears at the shoulder and the lowest velocities are located at the stern and bow. It is also shown that the velocity at the wake field is lower than the lateral velocity due to the ship hull, but it will increase slowly to reach the value of incoming flow. In addition, it is also possible to see the boundary layer area close to the hull, where the velocity is very low and it will reach zero at the hull. However, from the x component in Figure 28-(b), it is seen that the velocity outside the boundary layer at the lateral middle body of the hull is more or less the same as the incoming flow velocity. This shows a great agreement of inversion with the pressure distribution due to the pressure-velocity coupling.
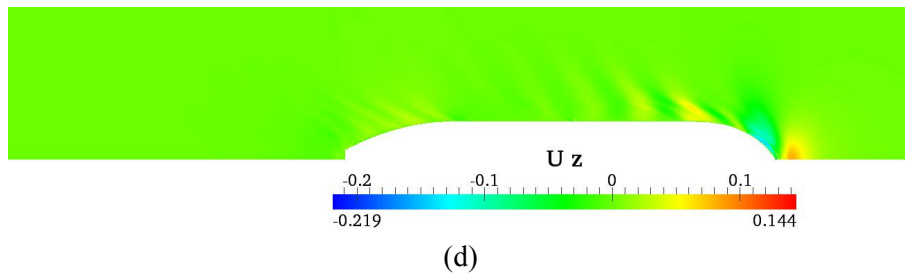

(a)

(b)

(c)

(d)
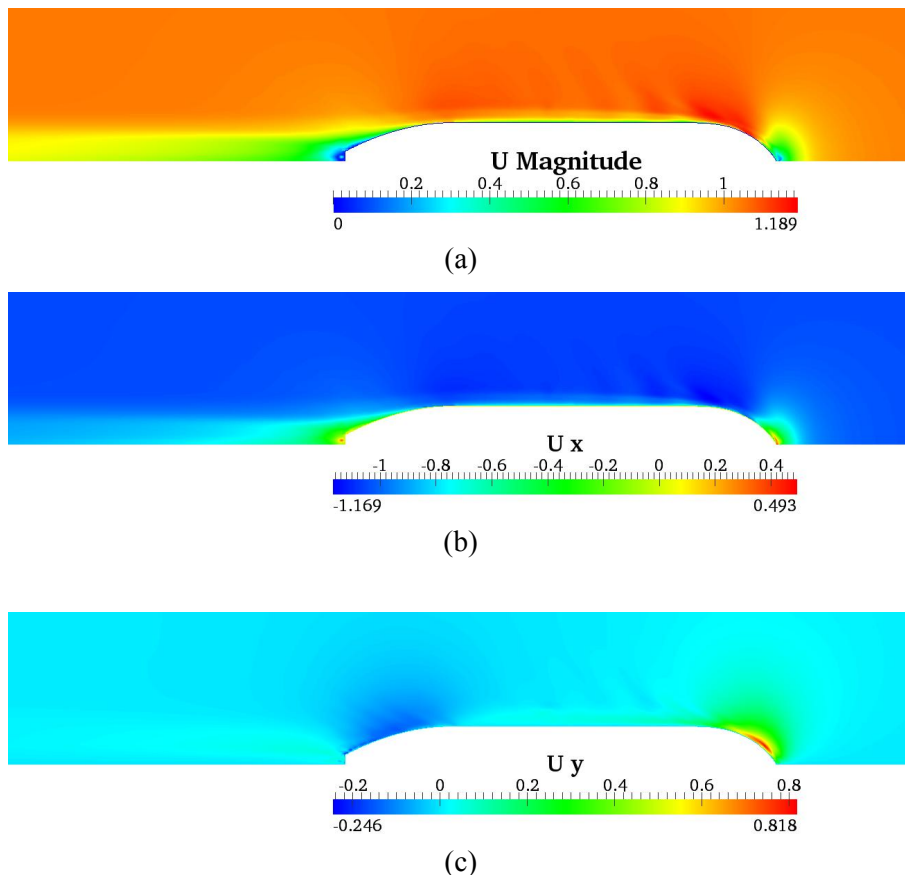
Figure 28. Veloctiy Magnatidue and Components at Free Surface

On the other hand, the simulation result can be checked from the resistance graph and the principle of the resistance is the integration of the force in *x* direction. Before showing a graphical representation of the residual developments of each parameter, the foamLog utility is applied to generate the new <logs> directory. <logs> file records the entire steps of the running case and it shows the information regarding the calculated values of all the parameters as well as their initial and final residuals at each iteration. Each file contains several columns, the first column is the iteration and the rest is/are the corresponding information at that iteration. It is necessary to check the pressure residual since force is calculation from the pressure at each local finite element. Figure 29 gives a good convergence case of the pressure residual and the final residual can go to less than 1×10^{-5}, whose values are low enough. The reason why the curve of the residual at the convergence is still fluctuating is due to the turbulence. Furthermore, Figure 30 depicts that both pressure resistance and the frictional resistance stay constant after 6000 iterations.
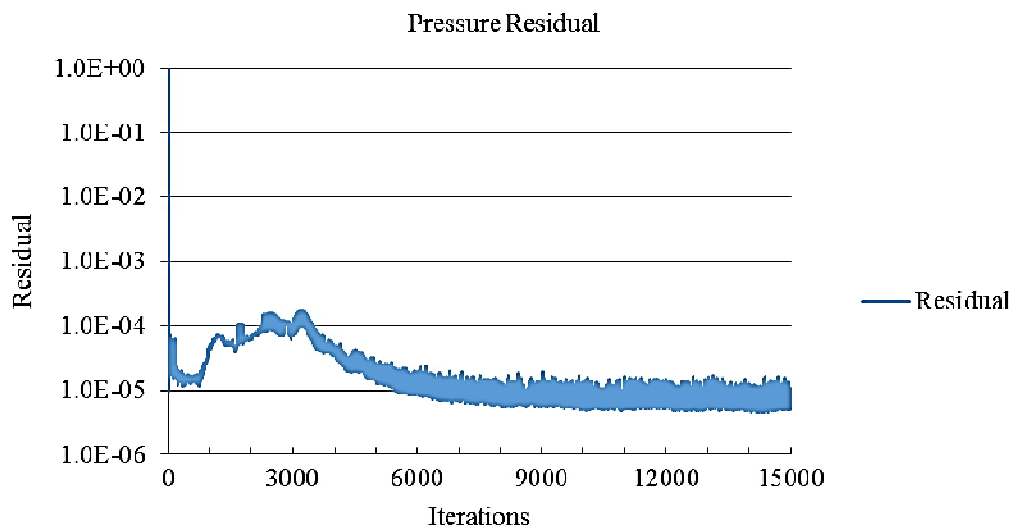

Figure 29. Pressure Residual over Iterations for KVLCC2
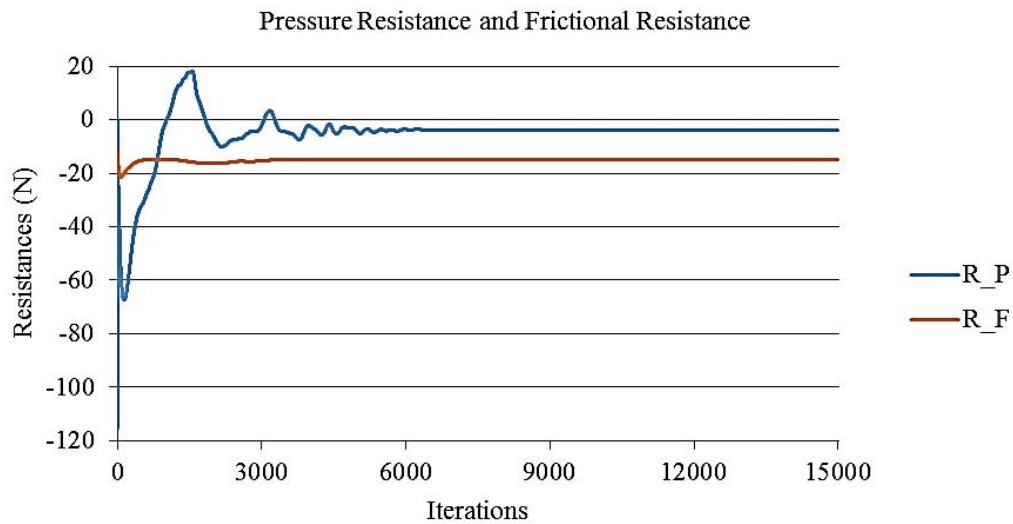
Pressure Resistance and Frictional Resistance

Figure 30. Pressure Resistance and Frictional Resistance over Iterations

Based on the above 4 steps of analysis and validation, the obtained mesh will work well enough. The only thing left is to do the grid refinement level study in order to get the most high-performance mesh configuration and this will be done in the next chapter.

The qualitative analyses above with respect to pressure and velocity distributions along the hull and converged resistances show that the calculated data from OpenFOAM give reasonable results. To some degree, it is proved that OpenFoam is a powerful solver to solve this fluid flow problem. More quantitative analyses will be implemented later in Chapter 5 with more detailed comparisons between the numerical results and experimental data.

5. MESH VALIDATION

As described in the previous chapter, in general the analytical solutions of the turbulence flow problems cannot be obtained directly and the temporal and spatial discretizations are applied for CFD methods. This means CFD approaches are all numerical calculations and it is unavoidable to cause errors. CFD errors consist of many factors and the main ones are the numerical error and the modeling error. For modeling error, it includes the determinations of the computational domain and the turbulence models as well as the initial settings of the simulations. It is assumed that the computational domain, the parameters of the turbulence model and the code for CFD solvers are correct and modeling error will not be considered in this project. The numerical error can be resulted from the time step sensitivity and the mesh sensitivity. Since the project is steady state simulation, there is no influence on time step sensitivity. In the following part, only the mesh sensitivity is conducted with systematical mesh refinements.

Mesh configuration should be investigated before and during the simulations in order to timely improve the parameter values or primitive meshes for later study. Once getting the acceptable mesh configuration from the previous chapter, five similar test cases are conducted with systematically refined girds in this thesis. Since there is no analytical solution of the fluid flow due to the chaotic features of the water, the verification of the results cannot process. However it is possible to explore the validated model which enables to get the converged result with a shorter computing time and lower errors and this converged model should be independent to the number of cells. In order to do this, all the cases are set in the same condition at constant Reynolds number $Re = 4.6 \times 10^{-6}$.

The grid convergence study is implemented with different mesh refinements and all the mesh information can be seen in Table 11. Only the initial meshes in the first mesh step of Mesh Wizard are expanded in five gradually refined levels and all the parameters in the rest four steps and the OpenFOAM file initializations are identical. For all the 5 grid refinements, the number of cells increases from 1.67M to 2.65M, 3.44M, 4.27M and 4.91M for Level 1-5 in free surface model condition, respectively. The number of cells in double body model case also increase respectively from 0.39M to 1.02M, 1.29M, 1.61M and 2.16M for 5 levels. Because there is no need to discretize the free surface of double body model, the total cells in this case will be much less than those in free surface model case. (Jin Kim *et al*, 2010) mentioned that the splitting speed of the mesh refinement in each axis cannot be larger than

$\sqrt{2}$, then an expansion ratio of 1.25 is used for the successive refined mesh sizes in *x* and *y* directions with the target size of (0.08*m*, 0.06*m*, 0.004*m*) for reference ship KVLCC2. The target sizes for both free surface model and double body model cases increase to (0.113*m*, 0.085*m*, 0.0056*m*) only at Level 1 case, while for double-body model of the finest Level 5, it reduces to (0.06*m*, 0.04*m*, 0.003*m*). The growth in the number of cells for LTSInterFoam is around 0.8M between two successive levels.

Table 11. Different Grid Refinement Levels for LTSInterFoam and simpleFoam

	initial mesh			minimal target size (*m*)			cell No.(FSM)	cell No.(DBM)
	x	*y*	*z*	*x*	*y*	*z*		
Level 1	23	10	12	0.113	0.085	0.0056	1.67*million*	0.39*million*
Level 2	36	16	19	0.08	0.06	0.004	2.65*million*	1.02*million*
Level 3	40	18	21	0.08	0.06	0.004	3.44*million*	1.29*million*
Level 4	45	20	23	0.08	0.06	0.004	4.27*million*	1.61*million*
Level 5	51	23	26	0.08	0.06	0.004	4.91*million*	2.16*million*
*FSM is for free surface model and DBM is for double body model								

In this thesis, the mesh validation is performed at the following two design conditions:

- Ship Speed $Fr = 0.142$ with the Draught $T = 20.8m$
- Ship Speed $Fr = 0.101$ with the Draught $T = 20.8m$

The draught $20.8m$ here is in full scale and it will be only $0.36m$ in model scale with a scale ratio of 1/58. Both two speed cases share the above test grids because their speeds are close, the same for the Reynolds numbers in model scale. First the above 5-level mesh configurations will be applied at $Fr = 0.142$ case to get the most proper mesh configuration. Then the predicted results, such as total resistance and wave elevations will be compared with the experimental data. If the predicted results with these mesh configurations show a good agreement with the experimental data, they will be converted to the $Fr = 0.101$ case for more tests. A good mesh configuration means it can give the good predicted and converged results at both $Fr = 0.142$ and $Fr = 0.101$ cases, not only in a single speed condition.

It is obvious that the mesh of the computational domains will change from Coarse to Fine for Level1 to Level 5 based on Table 11. In order to better observe the detailed mesh information, here only three levels of grids (Level 1, Level 3 and Level 5) for both free surface model and double body model of KVLCC2 are presented below.
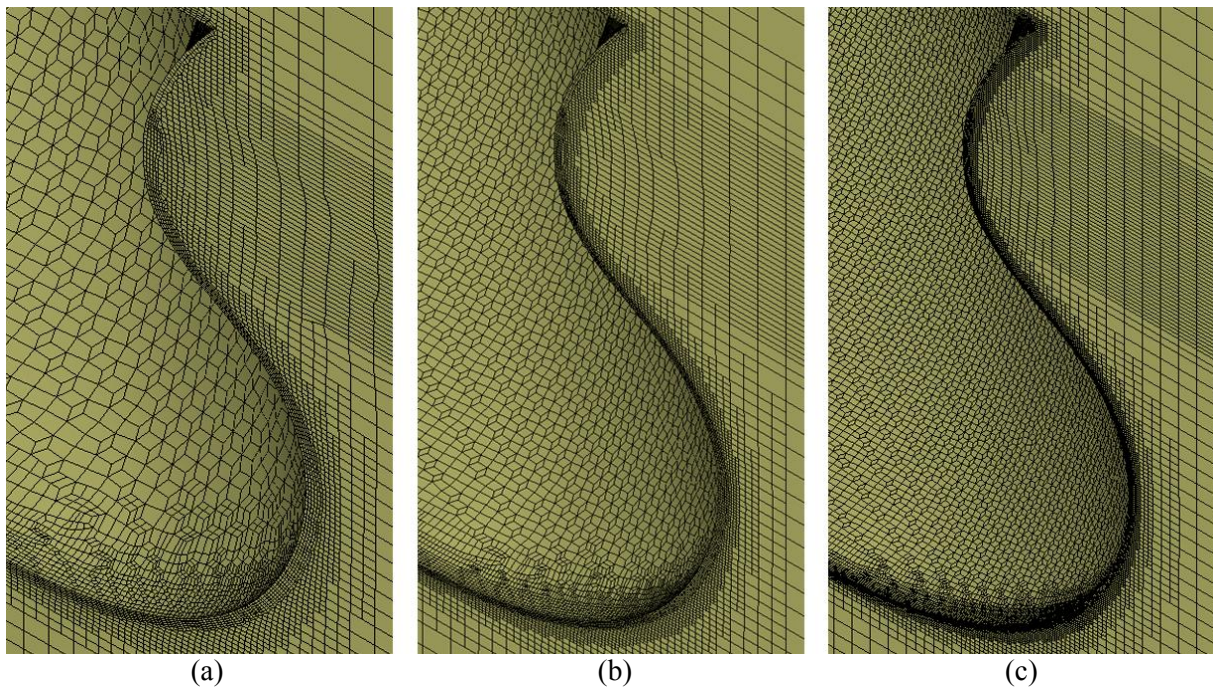
Figure 31. Different Mesh information at the bow for free surface model
((a) for coarse mesh, (b) for medium mesh and (c) for fine mesh)
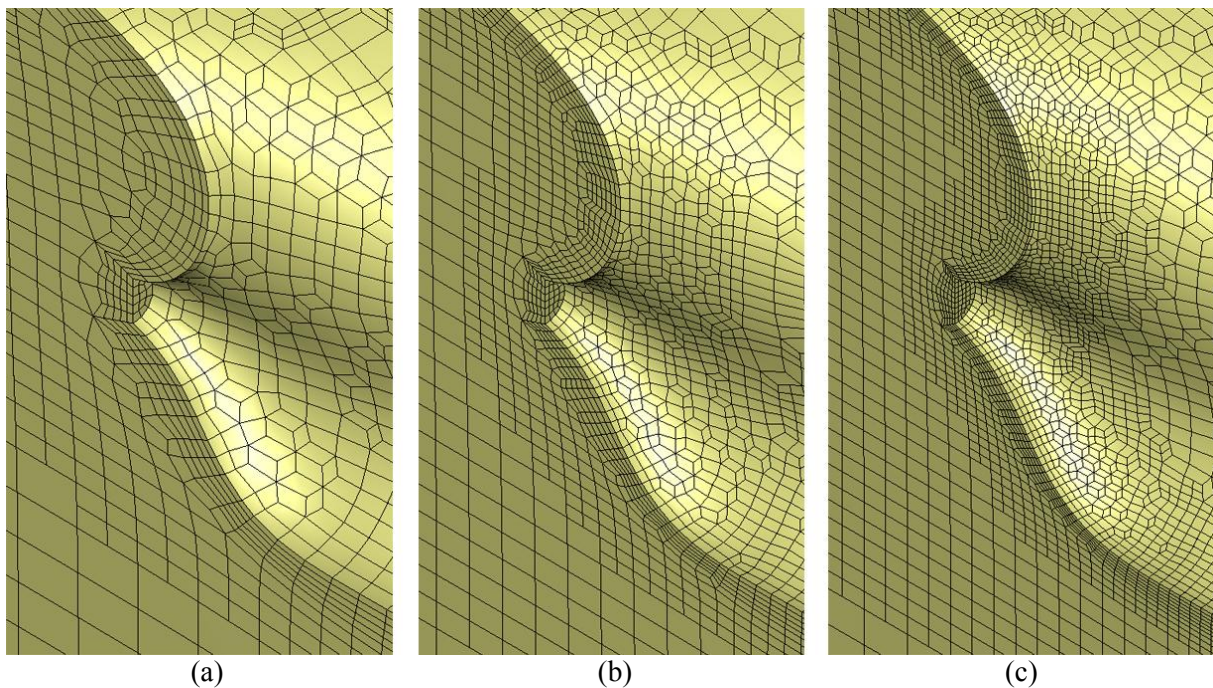

Figure 32. Different Mesh information at the stern for free surface model
((a) for coarse mesh, (b) for medium mesh and (c) for fine mesh)

Figure 31 and 32 depicts the grid information of three mesh cases at the bow and stern of KVLCC2 for free surface model, Level 1 coarse mesh with 1.67M cells in the left, Level 3 medium mesh with 3.44M cells in the middle and Level 5 fine mesh with 4.91M cells in the right. The same condition can be observed from Figure 33 and 34 for double body model. It is seen that the computational domain is cut at $z=0$ plane and the part above the plane is removed. No additional local refinement is set at the top surface of double body model.
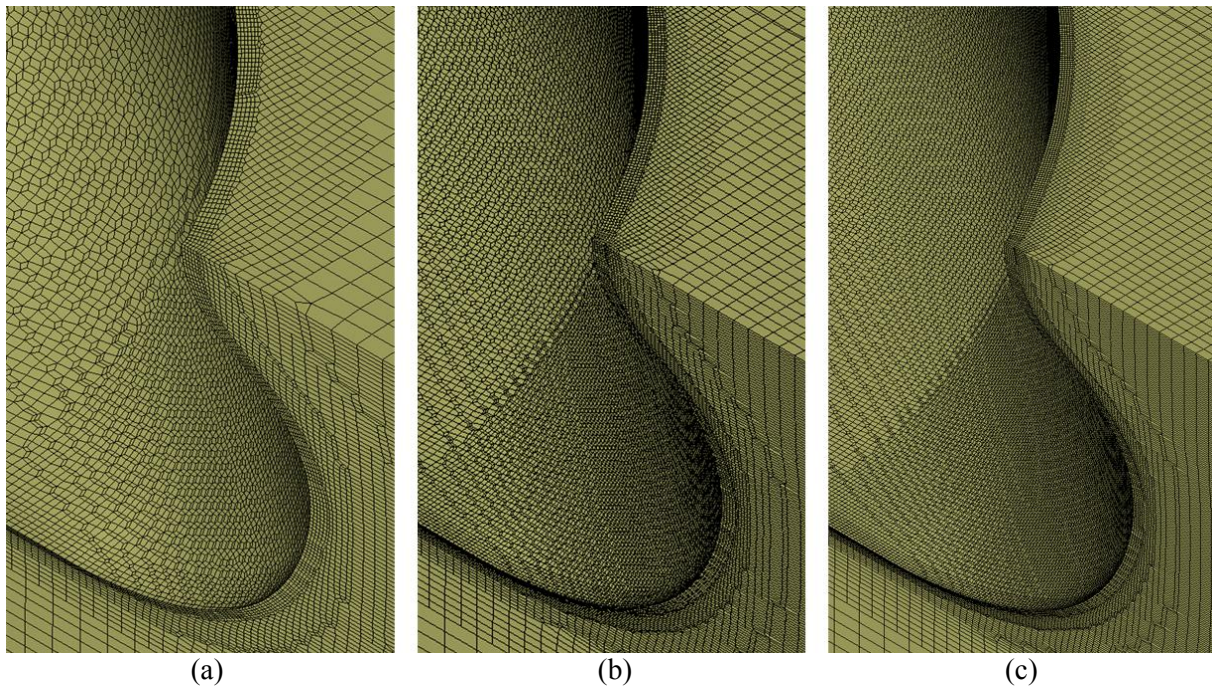
Figure 33. Different Mesh information at the bow for double body model
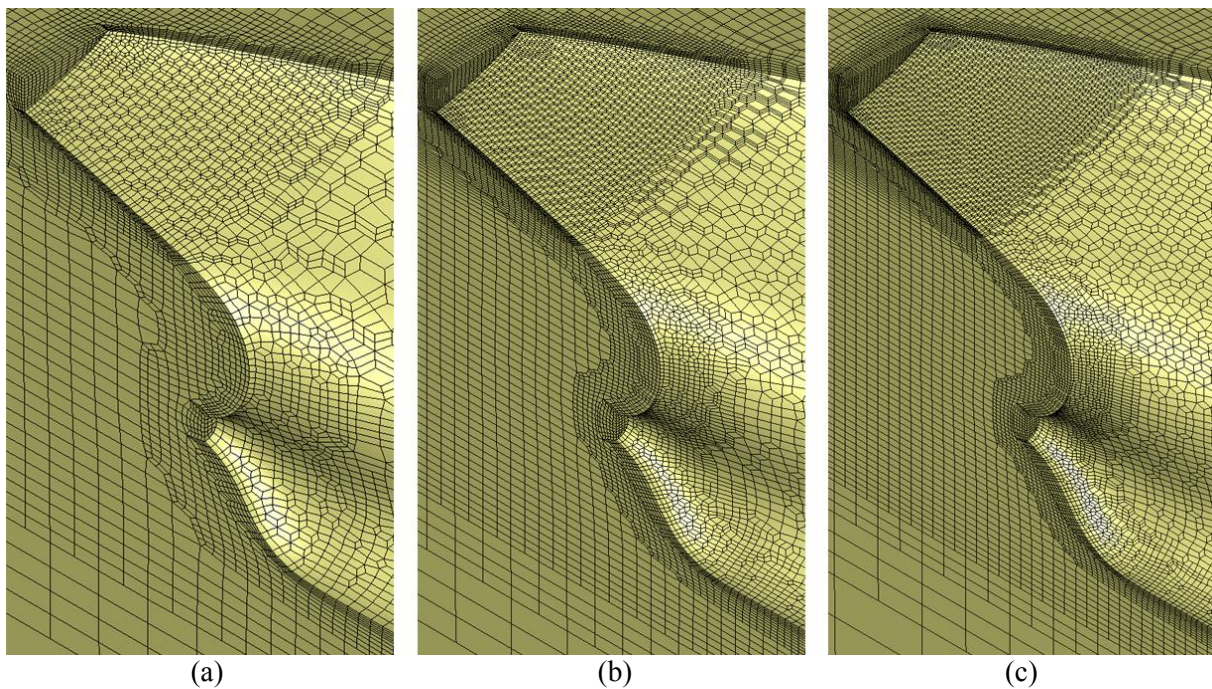((a) for coarse mesh, (b) for medium mesh and (c) for fine mesh)

Figure 34. Different Mesh information at the stern for double body model
((a) for coarse mesh, (b) for medium mesh and (c) for fine mesh)

The measurements of wave profiles by W. J. Kim were done at the design speed $Fr = 0.142$, however, the resistance towing test were implemented in several different speeds by MOERI. In this thesis, the predicted resistances at two ship speeds, $Fr = 0.142$ and $Fr = 0.101$, will be compared with the experimental data and only the wave profile obtained at $Fr = 0.142$ will be analyzed due to the lack of measured results at $Fr = 0.101$ condition. In the meanwhile, the wave elevations calculated by OpenFOAM will also be presented in the end.

5.1. $Fr = 0.142$ case

5.1.1 Grid Convergence

With the help of both LTSInterFoam and simpleFoam solvers, it is possible and easy to obtain all the resistance components. However, before analyzing the obtained results, it is necessary to check the convergence of different parameters, all of which should reach certain levels, especially the residuals of pressure and velocity. Take the LTSInterFoam residual results of Level 3 for example, Figure 35 plots the residuals of *k*, *w*, *p_rgh* and velocity components.
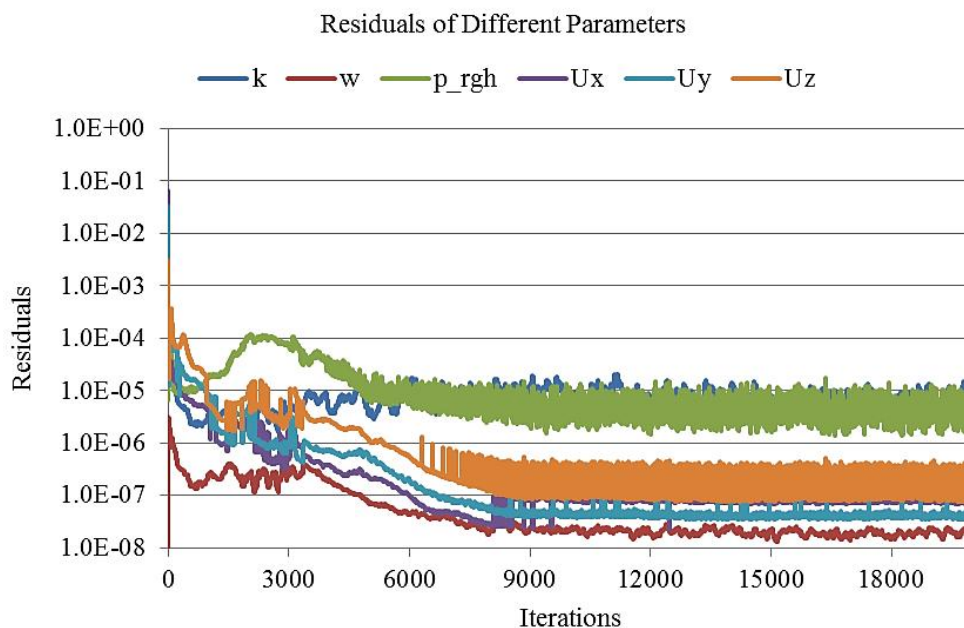

Figure 35. Parameter Residuals for LTSInterFoam Level 3 Case at $F_r = 0.142$

All of the residuals of the presented parameter reach quite low and relatively stable values after 8000 iterations approximately, lower than 1.0×10^{-7} for all the three velocity components and less than 1.0×10^{-5} for pressure. From Figure 35, it is seen that the simulation can be considered as convergence after 8000 iterations. As mentioned above, all the parameter cannot keep exactly flat and there are always some vibrations even at the convergence stage due to the turbulent feature.

Figure 36 gives the evolutions of different resistance coefficients over iterations. Viscous resistance and viscous pressure resistance can reach stable very fast, within 1000 iteration. However, the pressure resistance as well as total resistance is still fluctuating before 6000 iterations. But all the resistance components will be converged well after 6000 iterations.
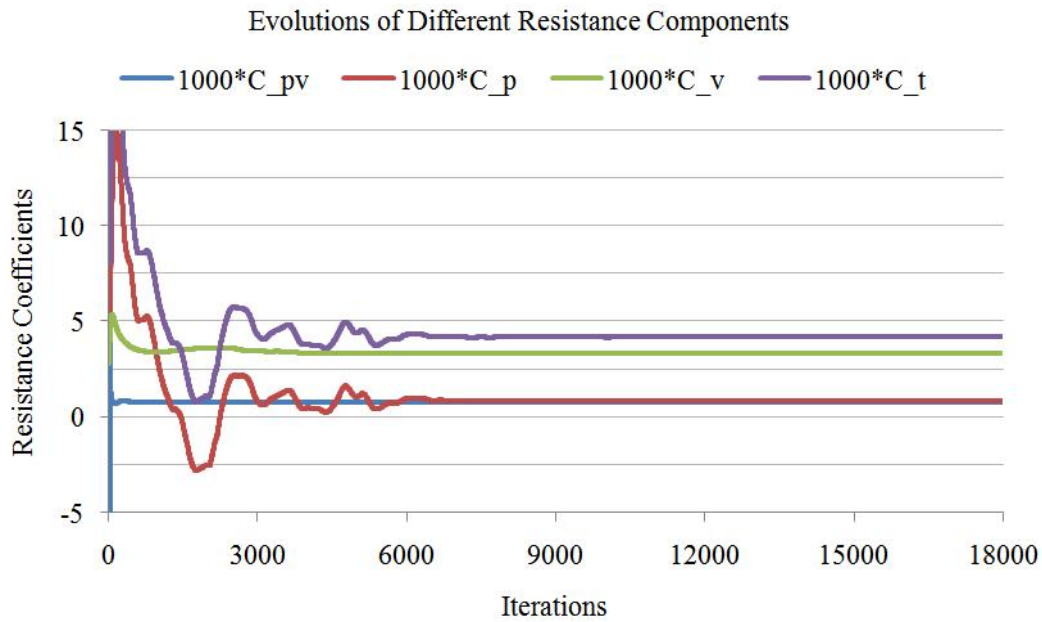
Evolutions of Different Resistance Components

Figure 36. Resistance Components over Iterations at $F_r = 0.142$ for Level 3 Case

On the other hand, different portions of the resistance can be observed clearly from Figure 36. Since the ship hull of KVLCC2 is fat and the design speed $Fr = 0.142$ is relatively low, the frictional resistance makes up a large part of the total resistance while the wake resistance is a very tiny proportion. The final five-step records of all the resistance components for Level 4 case are listed in Table 12. It is seen that there is a good convergence for all the components, especially for the viscous resistance calculated from LTSInterFoam solver, whose value is unchanged within the five records.

Table 12. Final Five Records of Resistances at $F_r = 0.142$ for Level 4 Case

No. of Iteration	$C_p \times 10^3$	$C_{pv} \times 10^3$	$C_w \times 10^3$	$C_v \times 10^3$	$C_t \times 10^3$
19989	0.837	0.779	0.058	3.338	4.175
19992	0.837	0.781	0.056	3.338	4.175
19995	0.837	0.781	0.056	3.338	4.175
19998	0.838	0.781	0.057	3.338	4.176
20001	0.838	0.778	0.060	3.338	4.176

In addition, it is also appreciated to take a look at the average percentages of different resistance components in final five records with respect to total resistance, see Table 13. The form factor is the percentage of the viscous pressure resistance with respect to the total resistance and the value is 0.187 in this case. So the predicted $1 + k = 1.187$ for Level 4 case. In the same condition, the measured form factor from MOERI is $1 + k = 1.191$ and the error

is only 2.1%. The simulation result of form factor shows a very good consistency with the experiment measurement.

Table 13. The Resistance Percentages with respect to C_t in $F_r = 0.142$ Case

C_p	C_{pv}	C_w	C_v	C_t
20.06%	18.68%	1.37%	79.94%	100.00%

After reaching the convergence for each case, the total resistance coefficient can be calculated from the simulation results. The predicted total resistance coefficients C_t are compared with the experimental values and the errors are given in the second last column in Table 14. It is obvious that the total resistances decrease with the increase of the mesh cells and the same information is given in Figure 37. With the increase of around 1.0 M cells from Level 1 to 2, the reduction of calculated total resistance error is more than 2%. From Level 2 to 3, as the growth in number of cells increases approximately 1.0 M, the decrease of predicted error is only 1%. Moreover, the decrease is only around 0.6% for the increase of 1.5 M cells from Level 3 to Level 5.

Table 14. Resistance ($\times 10^3$) of KVLCC2 at $F_r = 0.142$ with Different Grids

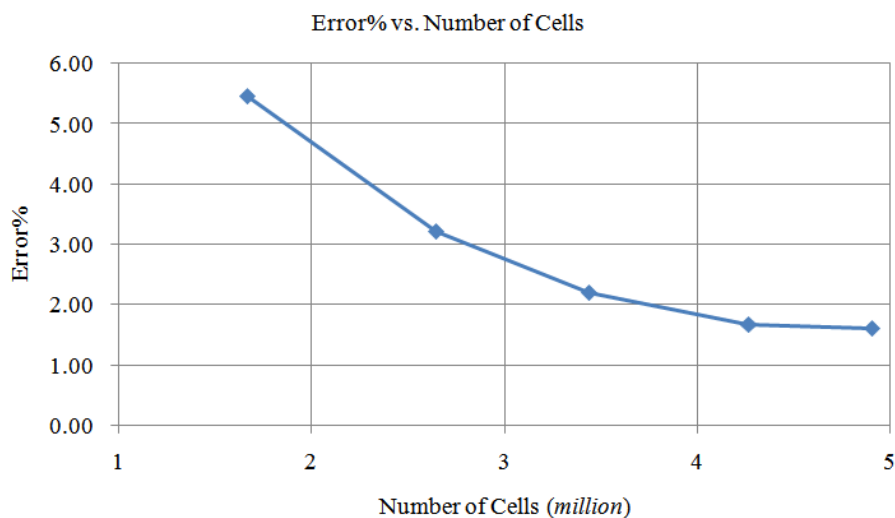|	No. of Cells	$C_v \times 10^3$	$C_p \times 10^3$	$C_t \times 10^3$	E%C_f	E%C_t	Exp. $C_t \times 10^3$
Level 1	1.67 M	3.394	0.940	4.334	-1.62	5.45	
Level 2	2.65 M	3.358	0.884	4.242	-2.67	3.21	
Level 3	3.44 M	3.336	0.865	4.201	-3.30	2.21	4.110
Level 4	4.27 M	3.339	0.840	4.179	-3.22	1.68	
Level 5	4.91 M	3.340	0.836	4.176	-3.19	1.61	

Figure 37. Total Resistance Errors over Cells for KVLCC2 at $F_r = 0.142$

For the frictional resistance coefficient C_f which derived from the ITTC57 empirical formula, it also provides a reference comparison with the calculated results from LTSInterFoam free-surface model. All the errors of frictional resistance coefficients are within 3.3%, whose values are low and acceptable.

The best mesh configuration for $Fr = 0.142$ case is in Level 4 case, which can be observed from Figure 37. From Level 4 to Level 5, the number of cells increases 0.64M, but the error of total resistance only reduces 0.07%, from where it can be considered that the mesh has reached the convergence condition. Finally the mesh configuration in Level 4 case is selected in this case.

5.1.2   *Wave Elevation*

For physical problems it is more important to analyze the field parameters such as the wave elevation and pressure distribution in the whole domain rather than a scalar value of total resistance. In this part, the same three cases (Level 1 for coarse mesh, Level 4 for Medium mesh and Level 5 for fine mesh) are selected for wave field investigation, such as wave contours and wave elevations in specific cutting planes.

After the analysis of different resistance coefficients, it is possible and necessary to compare the free surface wave field with the experimental data. However, accurate prediction of the free surface elevation is a big challenge for CFD approaches due to the low Kelvin wave amplitude (Deng *et al*, 2010), especially for the fat hull with low speed like the reference ship KVLCC2 in this project.
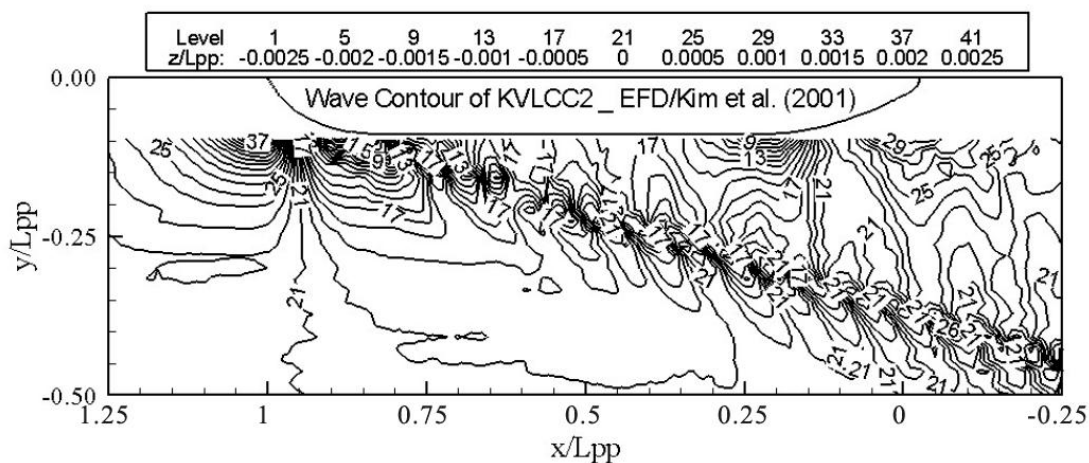

Figure 38. The measured Wave Pattern by W. J. Kim at $F_r = 0.142$
Available from *Gothenburg 2010-A Workshop on Numerical Ship Hydrodynamics*

During the experimental measurement from W. J. Kim, the wave elevations between $-0.0025 < z/L_{PP} < 0.0025$ were divided into 40 heights equally. Every four heights of the wave elevation would be recorded and the final wave contour can be seen in Figure 38. With the help of paraFoam, the wave contours of KVLCC2 in medium mesh are obtained in Figure 39. Still the wave field can be observed and it propagates outwards from the bow to the stern fields with the V-shaped wave.
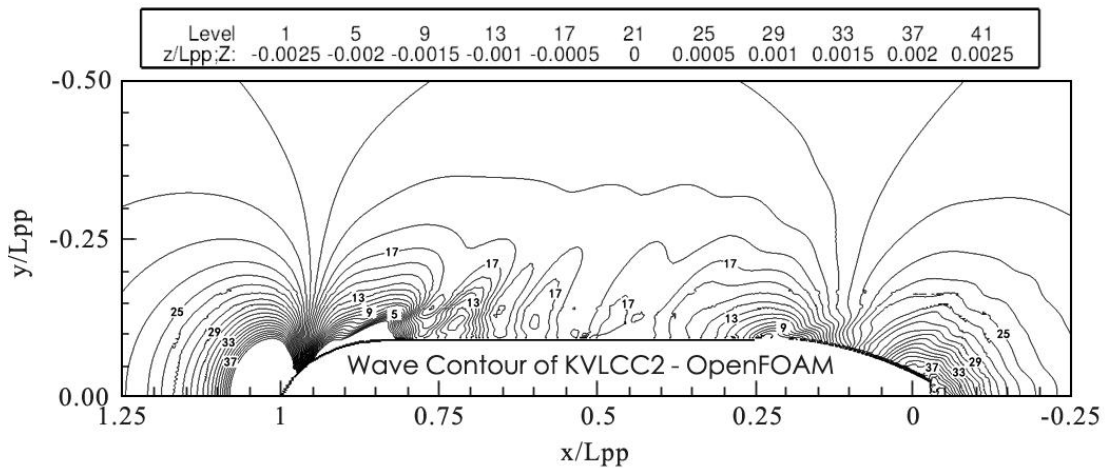

Figure 39. The Predicted Wave Pattern for KVLCC2 Medium Mesh at $F_r = 0.142$

According to Figure 38 and Figure 39, the calculated and measured wave contours are not so close, particularly in far fields. However, it is seen that the height level distributions in two figures near the hull with high mesh resolutions show a good prediction of the wave profile. In general, the results near the hull are more significant because the resistances are calculated from the integration of the pressure with each local cell area. On the other hand, the predicted wave contours in this thesis show a good consistency with the numerical results implemented by several other CFD solvers, such as MOERI-WAVIS and ECN-BEC-HO/ICARE. Even the wave contours out of the measured range and the far wave fields from ECN-BEC-HO/ICARE are close to the one from OpenFOAM solver.
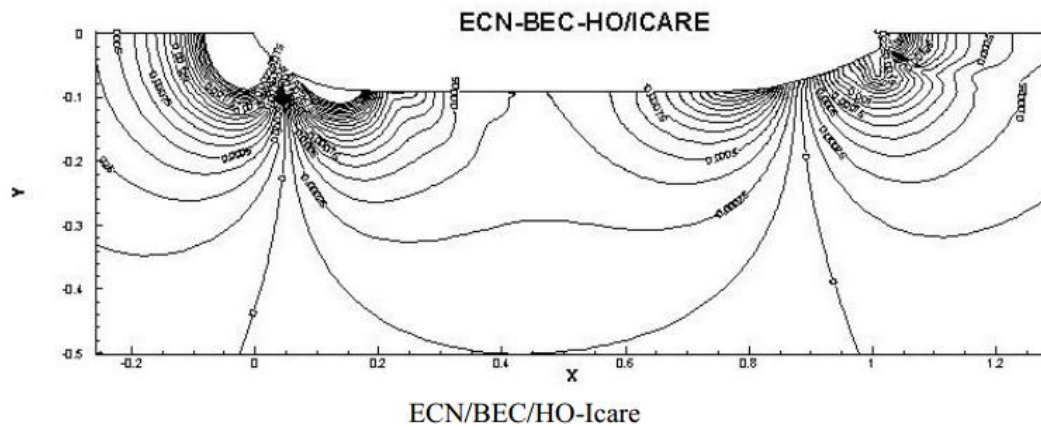
Figure 40. The Predicted Wave Pattern by ECN-BEC-HO/ICARE at $F_r = 0.142$
Available from *Gothenburg 2010-A Workshop on Numerical Ship Hydrodynamics*

Furthermore, precise comparison can be carried out by 2D wave elevation distributions at specified wave cuts between calculated and measured data. The lateral positions of the wave cuts are exactly followed the ones measured from W. J. Kim and they are $y/L_{PP} = -0.0964$, $y/L_{PP} = -0.1581$ and $y/L_{PP} = -0.2993$. The available measured data is between -1 to 1.5 x/L_{PP}. The regions out of this range are neglected due to the very small wave elevation.
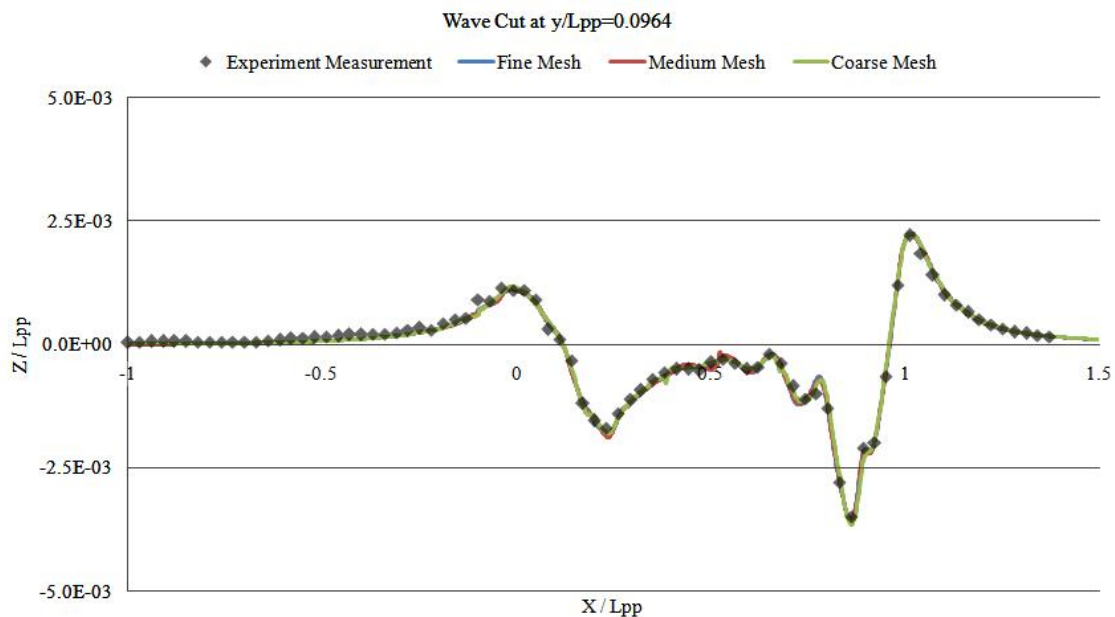
(a) $y/L_{PP} = -0.0964$

Figure 41. The Wave Elevation of KVLCC2 at $y/L_{PP} = -0.0964$ at $F_r = 0.142$

This wave cutting plane is located in the vicinity of KVLCC2 and the accuracy of the calculated results will have a great influence with the wave elevation distribution in this wave cut. Figure 41 shows a good agreement between predicted and measured wave patterns. The elevation reaches to the peak at the bow and decreases sharply to the trough at the shoulder of the hull, which is consistent with the pressure and velocity distributions discussed in the

previous chapter. The similar distribution can be observed at the stern of the hull, but the amplitudes of both crest and trough are smaller than those at the bow.

In addition, it is seen that the predicted wave elevations for both medium and fine mesh cases at the whole measurement range are close and a small difference occurs within the hull length zone, $0 < x < L$, in coarse condition. This can be explained by the different mesh refinement levels. The finer mesh helps to get better results and the grid has already converged in the medium mesh case.

(b) $y/L_{PP} = -0.1581$

In this case, the amplitudes of wave elevation at the bow and stern in Figure 42 are much lower than those in the first wave cut. However, the highest amplitudes still occur near the bow and the stern regions. The predicted and measured wave elevation at this case show a good agreement almost at all the measurement range and only a litter big discrepancy is located in the forward 1/3 part of the hull length, where the sharper fluctuation can be observed. The curve of the predicted elevation at this region is flatter than that of measured values and this phenomenon will be more obvious in the next wave cut, $y/L_{PP} = -0.2993$.
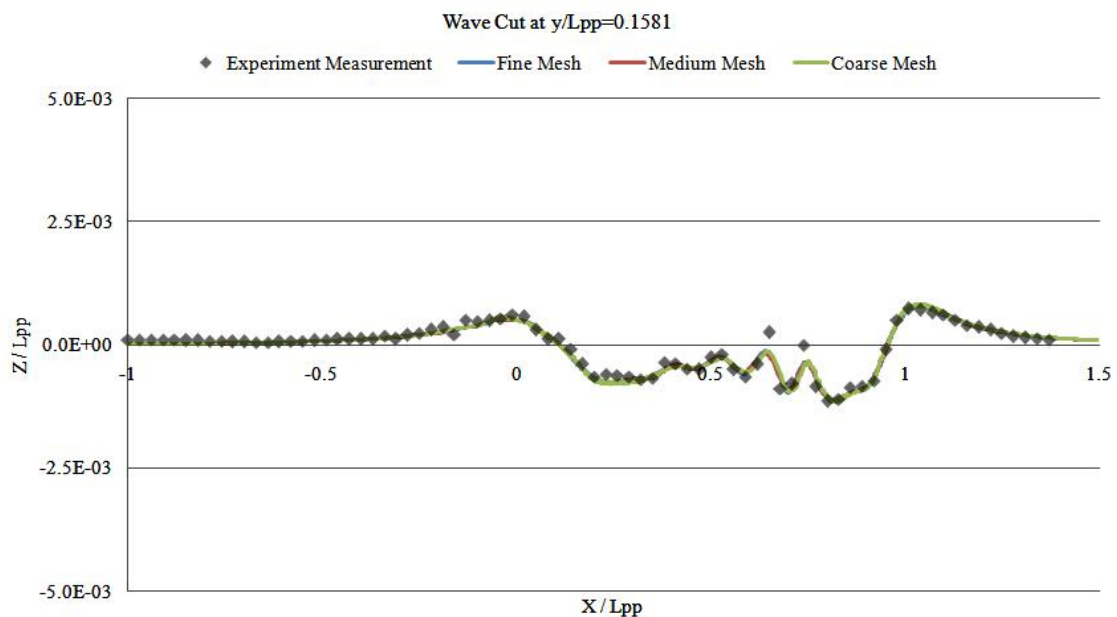

Figure 42. The Wave Elevation of KVLCC2 at $y/L_{PP} = -0.1581$ at $F_r = 0.142$

On the other hand, the differences of the elevation distribution in coarse, medium and fine meshes reduce. Even for the coarse case, the wave elevation is quite close to the other two cases. The reason is that the differences of the mesh discretizaion at $y/L_{PP} = -0.1581$ cut

are not larger than those at $y/L_{PP} = -0.0964$ cut case. All of mesh cases will give a relatively coarser mesh at this position, even for fine mesh case.

(c) $y/L_{PP} = -0.2993$

The wave elevations in the farthest cut, $y/L_{PP} = -0.2993$, are no longer in agreement with the measured data. The reasons of the discrepancies are mainly due to the coarse mesh in this place. Since it is far from the hull, the meshes are coarse in all three cases. There is a big gap between the neighboring gird points and neither point is able to detect the crest or trough of the wave easily. As the grid points lose the higher (crest) or lower (trough) wave information, most of the points will stay in the medium heights. This is the reason why the wave elevation is more flat in the cut plane of $y/L_{PP} = -0.2993$.

On the other hand, the predicted results for all coarse, medium and fine meshes are exactly the same from the Figure 43 due to the similar coarse mesh at this wave cutting plane, even the refined mesh around the hull.

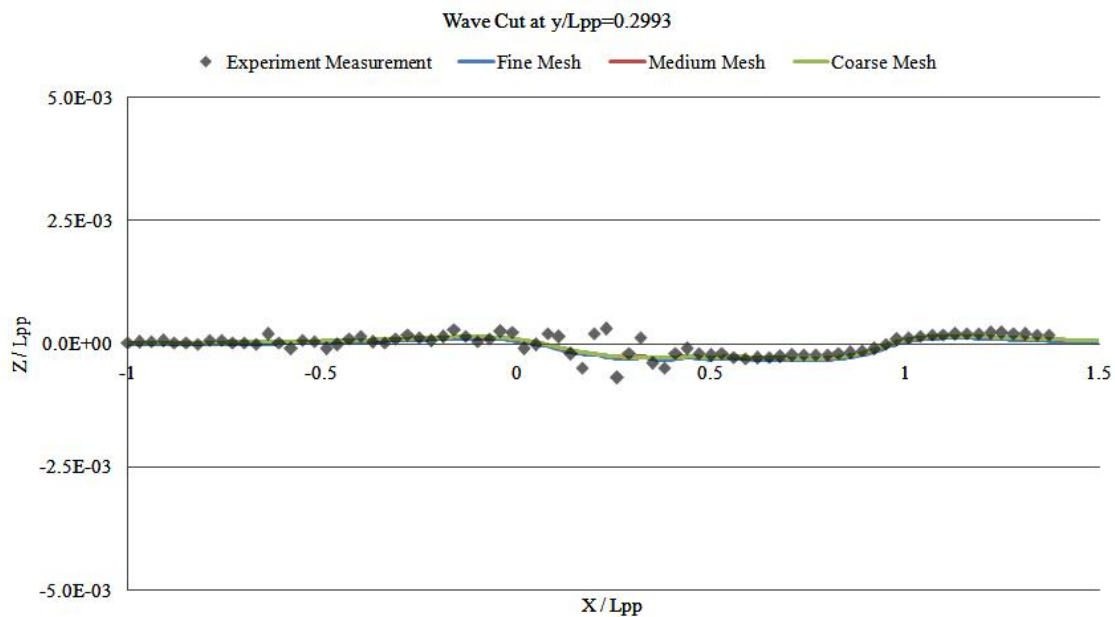Figure 43. The Wave Elevation of KVLCC2 at $y/L_{PP} = -0.2993$ at $F_r = 0.142$

5.2. $Fr = 0.101$ case

5.2.1   Grid Convergence

In this case, the same mesh configurations are applied for the grid convergence study after the determination of turbulence characteristics. However, only the resistance components are compared with the experimental data due to the missing measurements for free surface wave

profiles at present. The values of *k, ω* and the incoming flow speed with respect to this low ship speed are modified and the updated values are shown in Table 15.

Table 15. Updated Parameter Values at $F_r = 0.101$ case

F_r	U (m/s)	Re	Theta (m)	Length (m)	I	k (m^2/s^2)	ω (1/s)
0.101	0.743	3.27E+06	1.02E-01	4.07E-02	2%	0.000331	0.81728

All the other parameters and settings are the same as those in $F_r = 0.142$ case. Similarly, the identical test meshes are used for the grid convergence study, which were already presented in Table 11. Sufficient iterations are specified to help the simulation reach the convergence conditions. Figure 44 below gives the residual information of *k, w, p_rgh* and velocity components for LTSInterFoam Level 3 case, from which the simulation can be considered as convergence after around 6000 iterations. Moreover, the force histories of different resistance components also show a good convergence in Figure 45.
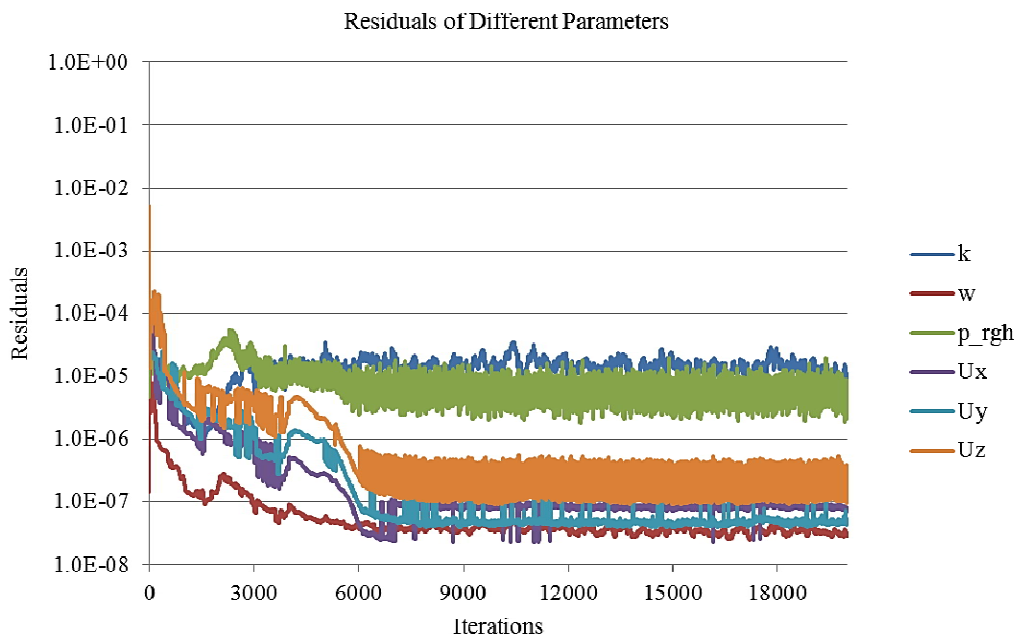

Figure 44. Parameter Residuals for LTSInterFoam Level 3 Case at $F_r = 0.101$
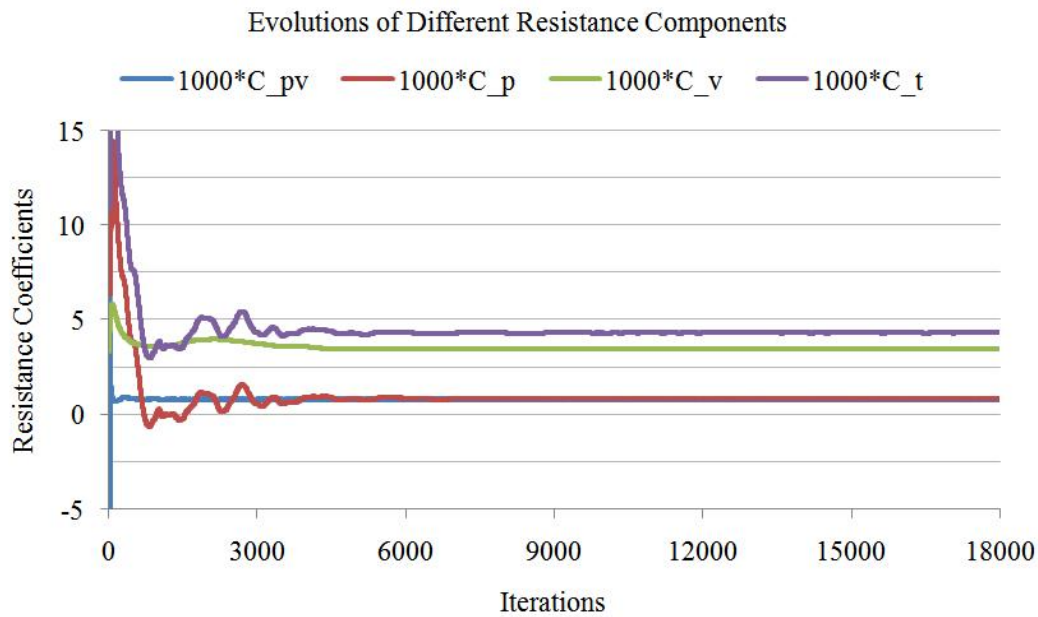
Figure 45. Resistance Components over Iterations at $F_r = 0.101$ for Level 3 Case

Table 16 gives the final five step records of all the resistance components in Level 4 grid case and the average portions of different components are calculated in Table 17. With the decrease of ship speed, it is obvious that the wave resistance decreases dramatically from 1.37% to only 0.65% and there is a slight increase of frictional resistance. In this case, the calculated form factor is $1 + k = 1.189$, which is only 1.05% error compared with experimental data.

Table 16. Final Five Records of Resistances at $F_r = 0.101$ for Level 4 Case

No. of Iteration	$C_p \times 10^3$	$C_{pv} \times 10^3$	$C_w \times 10^3$	$C_v \times 10^3$	$C_t \times 10^3$
19989	0.845	0.818	0.027	3.482	4.327
19992	0.846	0.819	0.027	3.482	4.327
19995	0.845	0.813	0.032	3.482	4.327
19998	0.845	0.816	0.029	3.482	4.327
20001	0.845	0.820	0.025	3.482	4.327

Table 17. The Percentages of Resistance with respect to C_t in $F_r = 0.101$ Case

C_p	C_{pv}	C_w	C_v	C_t
19.53%	18.89%	0.65%	80.47%	100.00%

Now it is the time to check the error percentages of the calculated results with the experimental data in $F_r = 0.101$ cases. Table 18 gives the values of each resistance coefficients for all five mesh levels and the errors of total resistance are calculated in the second last column.

Table 18. Resistance ($\times 10^3$) of KVLCC2 at $F_r = 0.101$ with Different Grids

	No. of Cells	$C_v \times 10^3$	$C_p \times 10^3$	$C_t \times 10^3$	E%C_t	Exp. $C_t \times 10^3$
Level 1	1.67 M	3.544	0.910	4.454	5.13%	
Level 2	2.65 M	3.496	0.867	4.362	2.96%	
Level 3	3.44 M	3.463	0.851	4.314	1.82%	4.237
Level 4	4.27 M	3.482	0.850	4.332	2.24%	
Level 5	4.91 M	3.483	0.847	4.330	2.20%	

Although the error in Level 3 is the lowest, it is still fluctuating and getting higher in Level 4. However, it can be seen that the differences of total resistance between Level 4 and 5 are quite small. With the increase of 0.64M cells, the error only decreases 0.04%, which can be considered as convergence. The same information can be found in Figure 46. It is worth noting that the result from Level 3 is not acceptable although it has lower error with less number of cells.
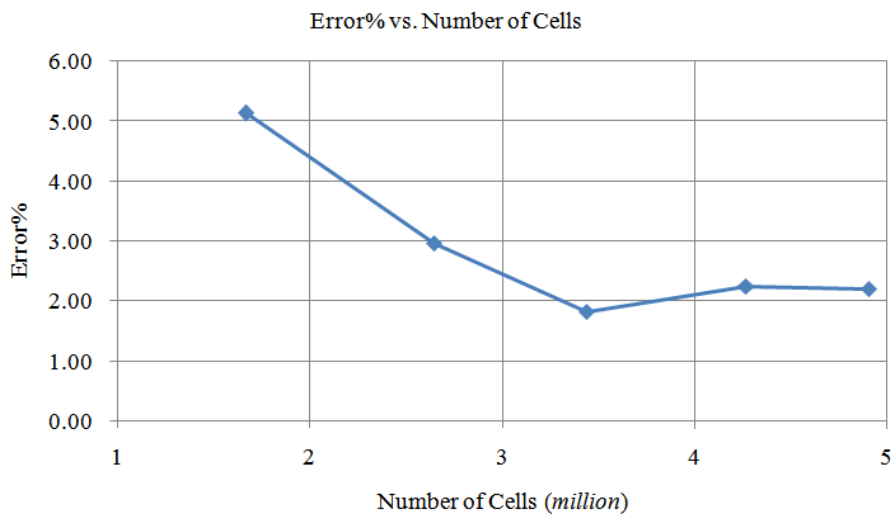

Figure 46. Total Resistance Errors over Cells at $F_r = 0.101$

From both speed cases, it is seen that the meshes will reach convergence at Level 4 case. With more increase of the cells, the calculated results will not be improved much more.

5.2.2 Wave Elevation

For wave elevation at $F_r = 0.101$ case, only the predicted elevation distributions are given in Figure 47, 48 and 49 by paraFoam at the same three wave cuts, $y/L_{PP} = -0.0964$, $y/L_{PP} = -0.1581$ and $y/L_{PP} = -0.2993$. Level 1, 4 and 5 grids represent the Coarse, Medium and Fine mesh cases, respectively. These predicted results can be the reference for future experiment or other CFD solvers.
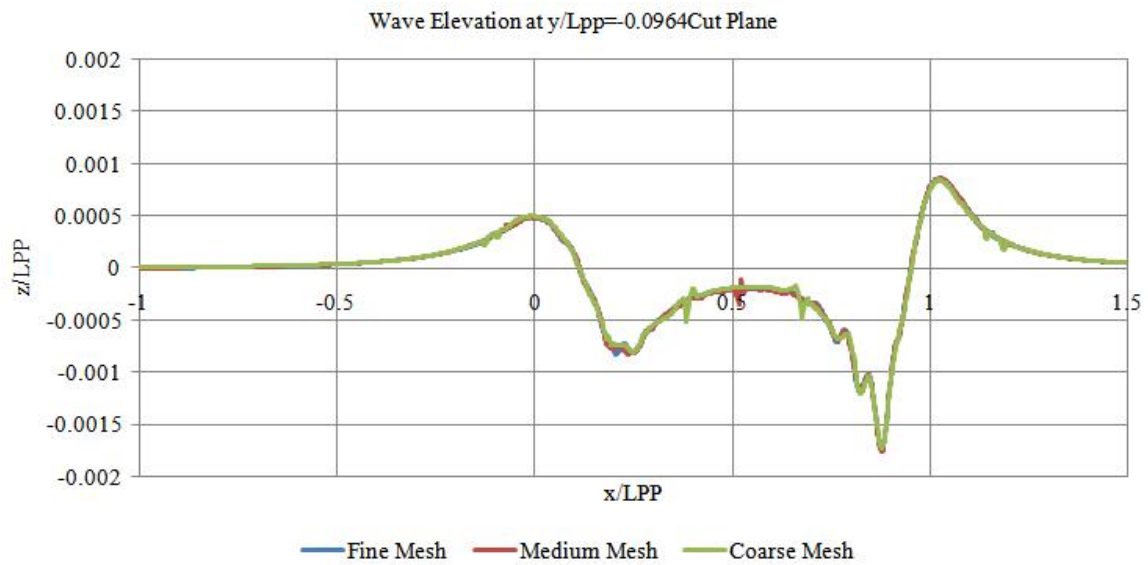
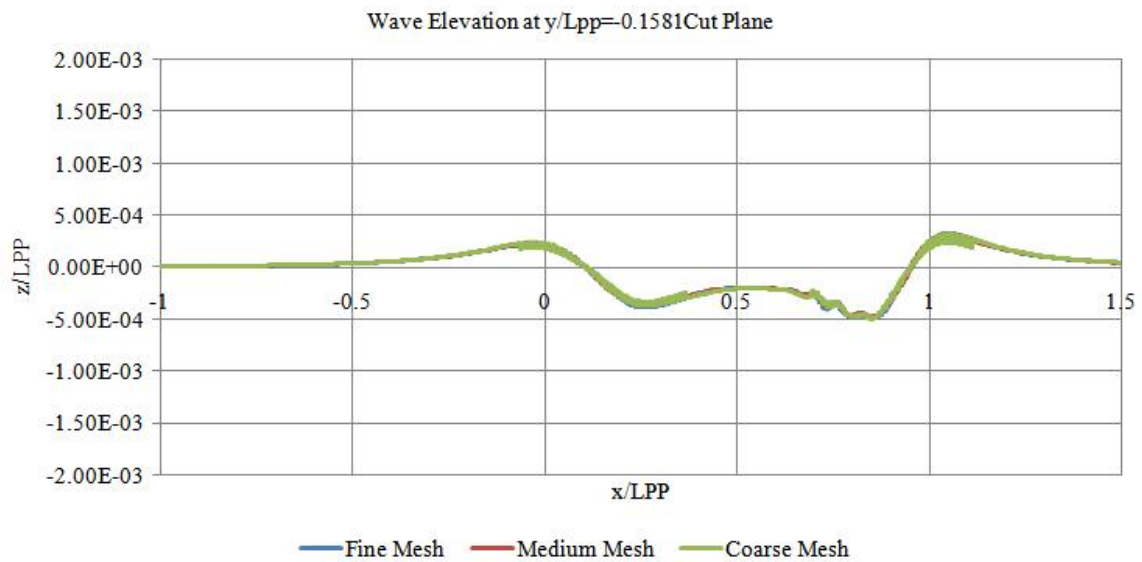Figure 47. The Wave Elevation of KVLCC2 at $y/L_{PP} = -0.0964$ at $F_r = 0.101$

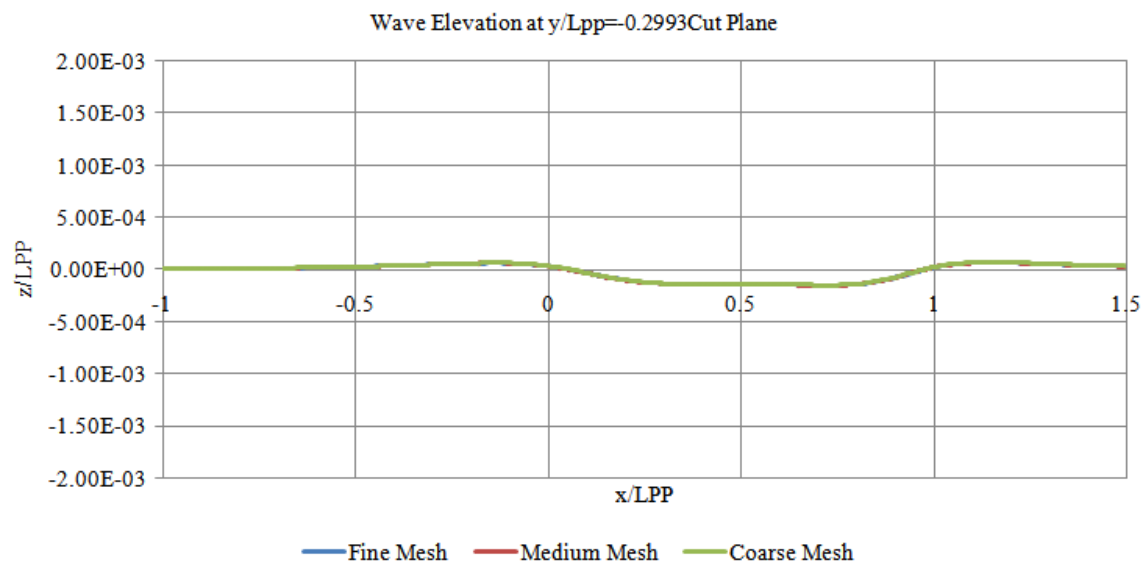Figure 48. The Wave Elevation of KVLCC2 at $y/L_{PP} = -0.1581$ at $F_r = 0.101$

Figure 49. The Wave Elevation of KVLCC2 at $y/L_{PP} = -0.2993$ at $F_r = 0.101$

6.   JBC BULKER

As mentioned in the introduction part, most of the work in this thesis has been done with the reference ship KVLCC2. However the main task of the thesis is to predict the resistance of JBC bulker, whose mesh validation and resistance prediction are also implemented in the same Froude numbers: $F_r = 0.142$ and $F_r = 0.101$. The most proper mesh configuration as well as the turbulence properties obtained from pervious chapters will be appied to JBC bulker at this time and it will be considered as medium mesh. Besides the medium one, one more coarse and fine mesh models will be created to validate the grid convergence. The simulations of JBC bulker are all running in the model scale. In addition, it is important to note that the Reynolds numbers should keep the same values as the ones of KVLCC2 cases: $Re = 4.60 \times 10^6$ for $F_r = 0.142$ and $Re = 3.72 \times 10^6$ for $F_r = 0.101$. The same Reynolds number will give the same behaviours of the turbulence effects.

The principle of this conversion from the reference KVLCC2 to the test ship JBC is based on the similarity of the two hull forms. Moreover, the flow behaviours around two hulls will be close if the Reynolds number is the same. The rest of the settings, such as the file initilizations (except the parameters related to the dimensions and velocity, like turbulent kinetic energy k and incoming flow speed U) and mesh configuration are the same for KVLCC2 and JBC cases. From the Reynolds similarity, the model scale in this thesis is calculated as 1/55.1 and the main dimensions of JBC bulker in model scale are given in Table 19.

Table 19. The Main Dimensions of JBC in Model Scale 1/55.1

main particulars_JBC		Model Scale
Length between perpendiculars	L_{PP}(m)	5.082
Length of waterline	L_{WL}(m)	5.172
Maximum beam of waterline	B_{WL}(m)	0.817
Depth	D(m)	0.454
Draft	T(m)	0.299
Displacement volume	∇(m^3)	1.066
Wetted surface area w/o ESD	$S_{0\_w/oESD}$(m^2)	6.441
Wetted surface area with ESD	S_{0_ESD}(m^2)	6.467
Block coefficient (CB)		0.858

The new simulations start from the Level 4 mesh configuration obtained from Chapter 5, which is concerned as medium mesh case. Similarity the coarse and fine mesh configurations are created based on the above study with an expansion ratio 1.25. Table 20 presents all the mesh information of JBC simulations in coarse, medium and fine cases.

The minimal target cell size in x direction is set as $0.064m$, which is a little lower than that of KVLCC2. The reason is that the wave length calculated in JBC case is shorter than that calculated in KVLCC2 case. The same situation can be seen in y direction, whose target cell is given as $0.05m$ for JBC ship.

Table 20. Different Levels of Grid Refinement for JBC

	initial mesh			minimal target size (m)			cell No. (FSM)	cell No. (DBM)
	x	y	z	x	y	z		
coarse	40	18	21				2.69 *million*	0.96 *million*
medium	45	20	23	0.064	0.050	0.003	3.28 *million*	1.2 *million*
fine	51	23	26				4.24 *million*	1.56 *million*
*FSM is for free surface model and DBM is for double body model								

6.1. $Fr = 0.142$ case

The model speed of JBC bulker in this case will be calculated and kinematic viscosity of water at experimental operation condition is given in http://www.t2015.nmri.go.jp/, whose value is $\nu = 1.107 \times 10^{-6} \, m^2/s$. However, there is no available experimental result of JBC in the website until now (10th December 2015). Part of the main initial parameters and turbulence properties of JBC bulker are given in Table 21 for $F_r = 0.142$ case.

Table 21. Parameter Values for JBC Bulker at $F_r = 0.142$ case

F_r	U (m/s)	ν (m^2/s)	Re	I	k (m^2/s^2)	ω ($1/s$)
0.142	1.003	1.107E-06	4.60E+06	0.02	0.0006032	1.282225

Figure 50 presents the parameter residuals over iterations for medium mesh case and it is seen that the simulation can be considered as convergence after around 6000 iterations. Then, the pressure distributions in three different views can be seen in Figure 51.
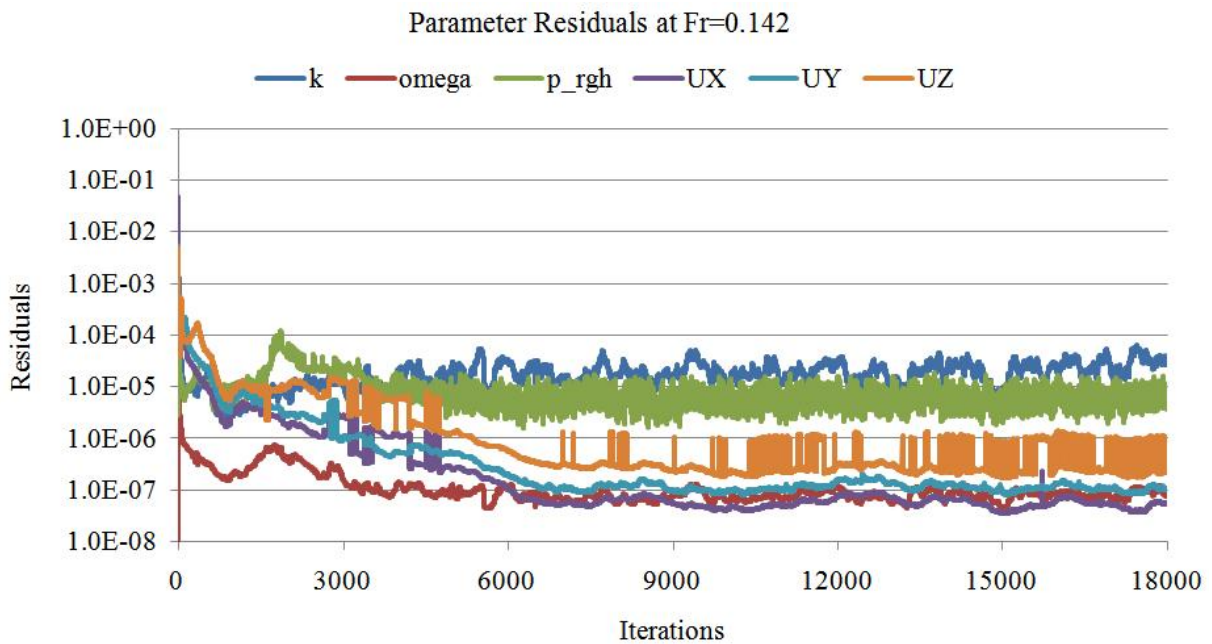
Cheng XU

Figure 50. Parameter Residuals for JBC Medium Case at $F_r = 0.142$
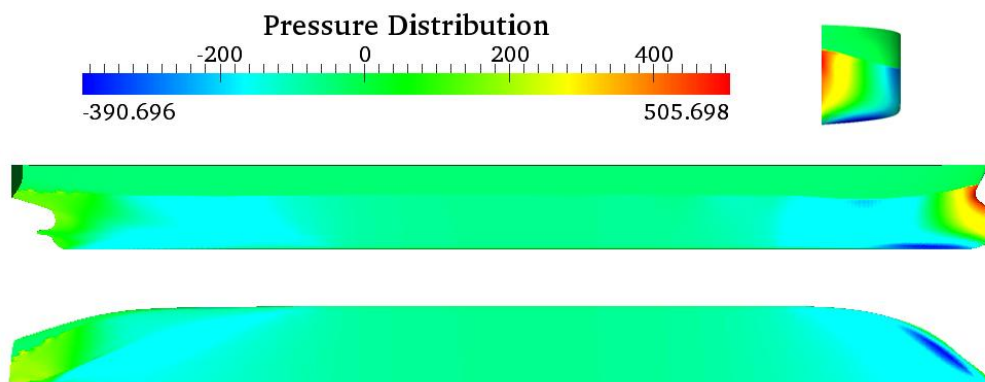

Figure 51. Pressure Distributions around JBC at $F_r = 0.142$ Case

The final resistance components of JBC are shown in Table 22 for coarse, medium and fine meshes. The total resistance coefficient C_t in coarse mesh is relatively higher compared with the ones in the other two cases. Although there is still a small difference between the results in medium mesh (4.476×10^{-3}) and fine mesh (4.451×10^{-3}) from Figure 52, the medium mesh case gives the best behavior: lower cell number with enough accuracy. The difference between medium and fine mesh cases is only 0.56%. Therefore, the mesh configuration in medium case is acceptable.

Table 22. Resistance $(\times 10^3)$ of JBC at $F_r = 0.142$ with Different Grids

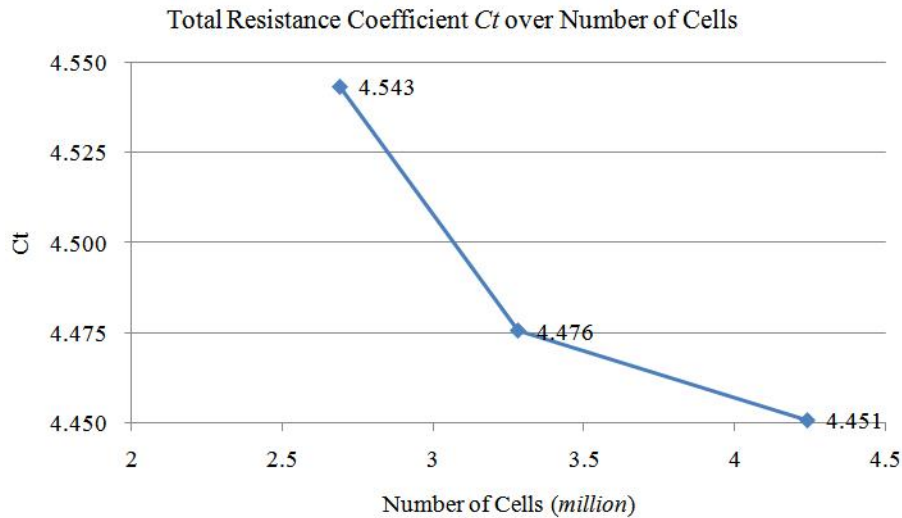|	cell No. (*million*)	$C_p \times 10^3$	$C_v \times 10^3$	$C_{pv} \times 10^3$	$C_t \times 10^3$
coarse	2.69	1.235	3.309	1.111	4.543
medium	3.28	1.154	3.321	1.117	4.476
fine	4.24	1.148	3.303	1.086	4.451

Figure 52. Total Resistance Errors over Cells for JBC at $F_r = 0.142$

At around the end of December 2015, NMRI gives the first experimental results of JBC for Fr=0.142 case, which can be seen in Table 23. Given the low values of sinkage and trim, the error percentage of the total resistance coefficient C_t between EFD data and the predicted result from this thesis for $F_r = 0.142$ case can be directly calculated as:

$$\text{Error}\%C_t = \frac{(C_{t\_CFD} - C_{t\_EFD})}{C_{t\_EFD}} \times 100\% = 4.3\%$$

Table 23. EFD data for JBC resistance w/o ESD, $F_r = 0.142$
Available from http://www.t2015.nmri.go.jp/Instructions_JBC/Case_1-1a.html

Parameters		EFD (D)
$C_T \times 10^3$	Value	**4.29**
	E%D	
$C_F \times 10^3$	Value	
	E%D	
$C_P \times 10^3$	Value	
	E%D	
sinkage [%LPP] upward positive	Value	**-0.086**
	E%D	
trim [%LPP] bow up positive	Value	**-0.180**
	E%D	

Figure 53 gives the wave contour of JBC bulker predicted from OpenFOAM. Double wave heights are used in order to plot the free surface wave field more clearly. Each eight height is marked and the range of Z/L_{PP} is between -0.0025 to 0.0025.
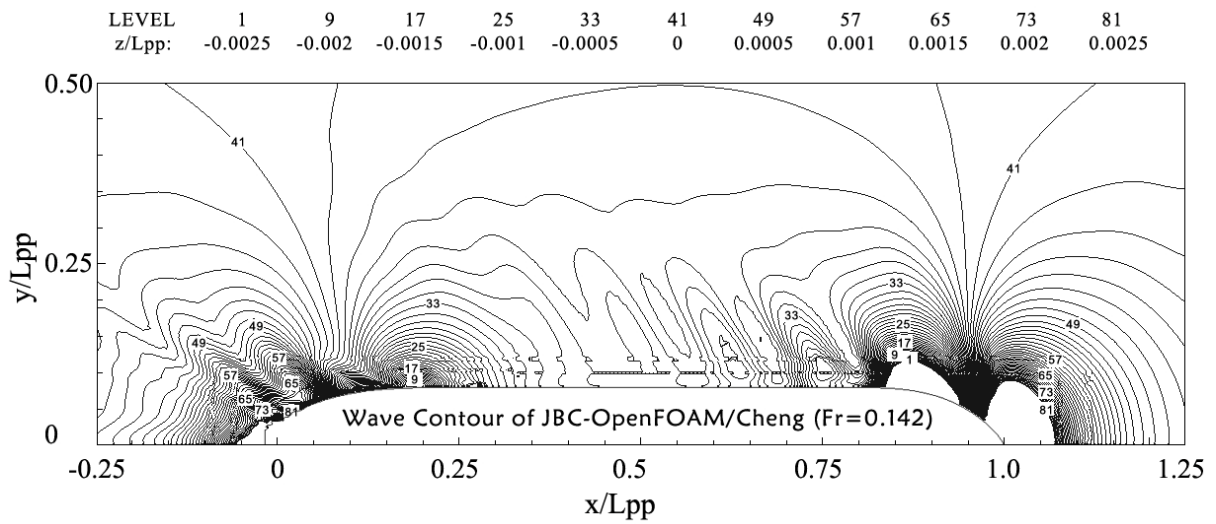
Figure 53. The Predicted Wave Pattern for JBC Medium Mesh at $F_r = 0.142$

Likewise, three new wave cuts are created in Figure 54 for further investigation of wave elevation and all of these three planes are parallel to XOZ plane. The distances of the three planes to *sym* are specified as $y/L_{PP} = 0.085$, $y/L_{PP} = 0.140$ and $y/L_{PP} = 0.280$, respectively.
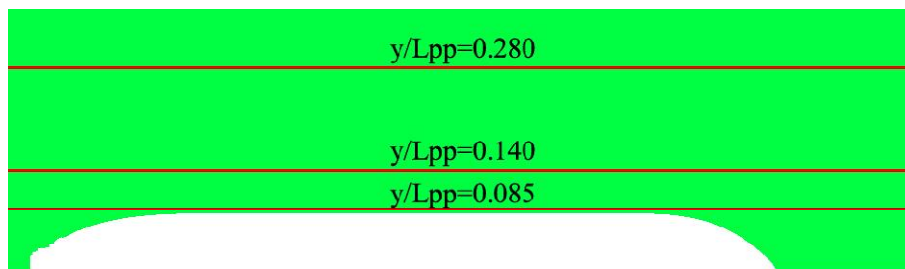

Figure 54. The Positions of Three Wave Cuts

More detailed wave elevations at three cutting positions are presented in Figure 55, Figure 56 and Figure 57.
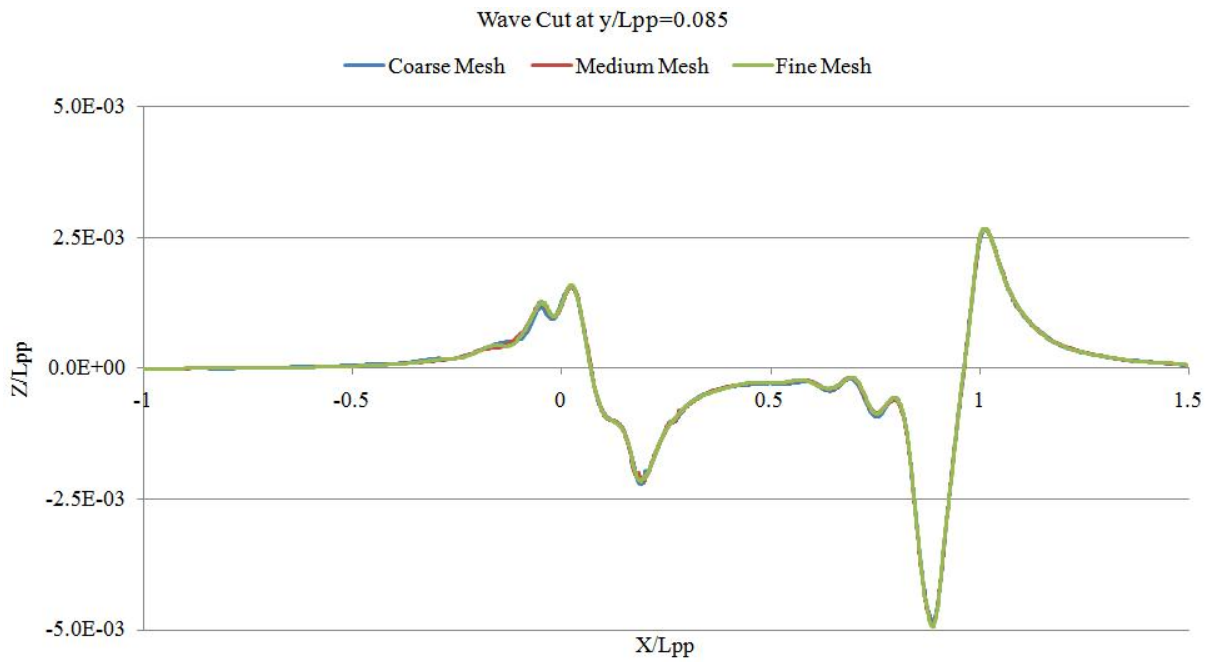
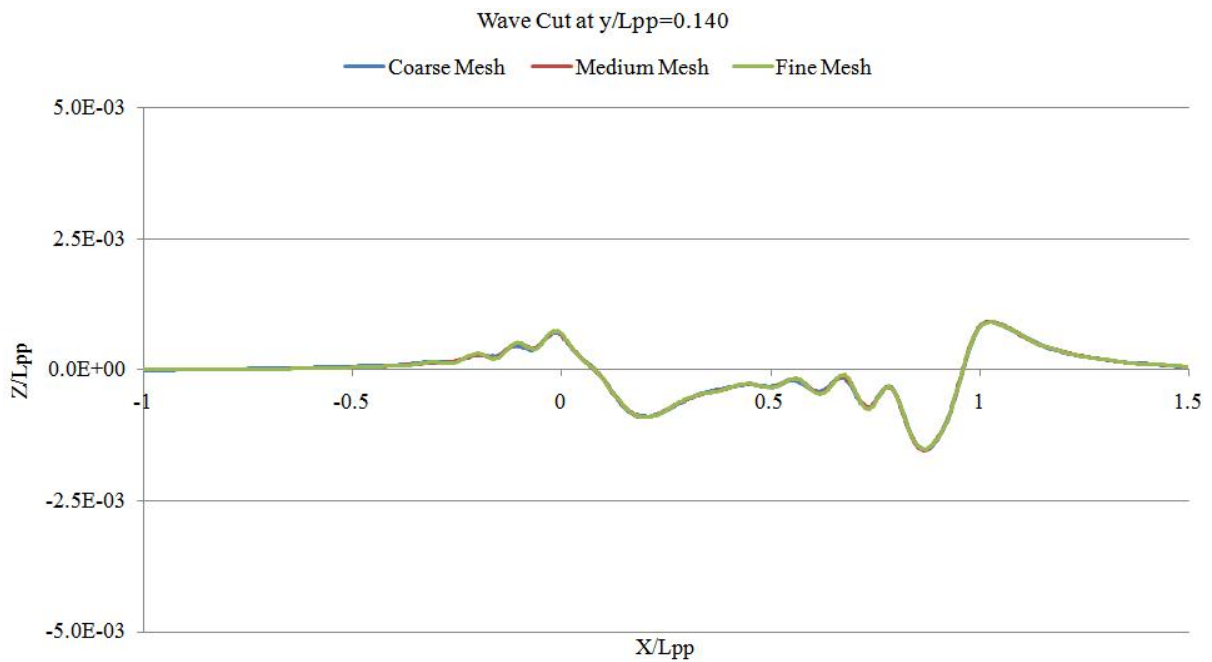Figure 55. The Wave Elevation of JBC at $y/L_{PP} = 0.085$ at $F_r = 0.142$

Figure 56. The Wave Elevation of JBC at $y/L_{PP} = 0.140$ at $F_r = 0.142$
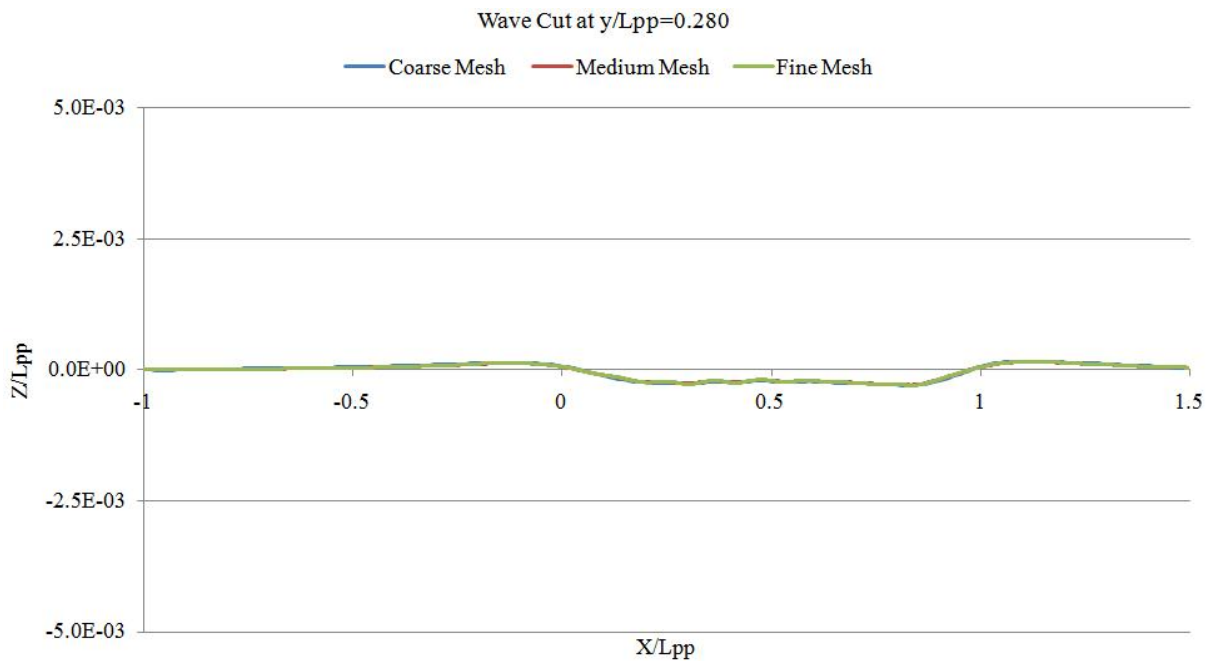
Figure 57. The Wave Elevation of JBC at $y/L_{PP} = 0.280$ at $F_r = 0.142$

6.2. $Fr = 0.101$ case

The main updated parameters used for JBC bulker in $F_r = 0.101$ case are listed in Table 23. The same as refernece ship, JBC mesh models in this case share the same configurations in $F_r = 0.142$ case.

Table 24. Parameter Values for JBC Bulker at $F_r = 0.101$ case

F_r	U (m/s)	ν (m²/s)	Re	I	k (m²/s²)	ω (1/s)
0.101	0.713	1.107E-06	3.27E+06	0.02	0.0003051	0.8519299

Also, the majority of the parameter residuals over iterations in medium case are shown in Figure 58 and the computation will converge at around 6000 iterations. In the end the pressure distribution around the JBC hull are given at convergence state in Figure 59

Figure 58. Parameter Residuals for JBC Medium Case at $F_r = 0.101$
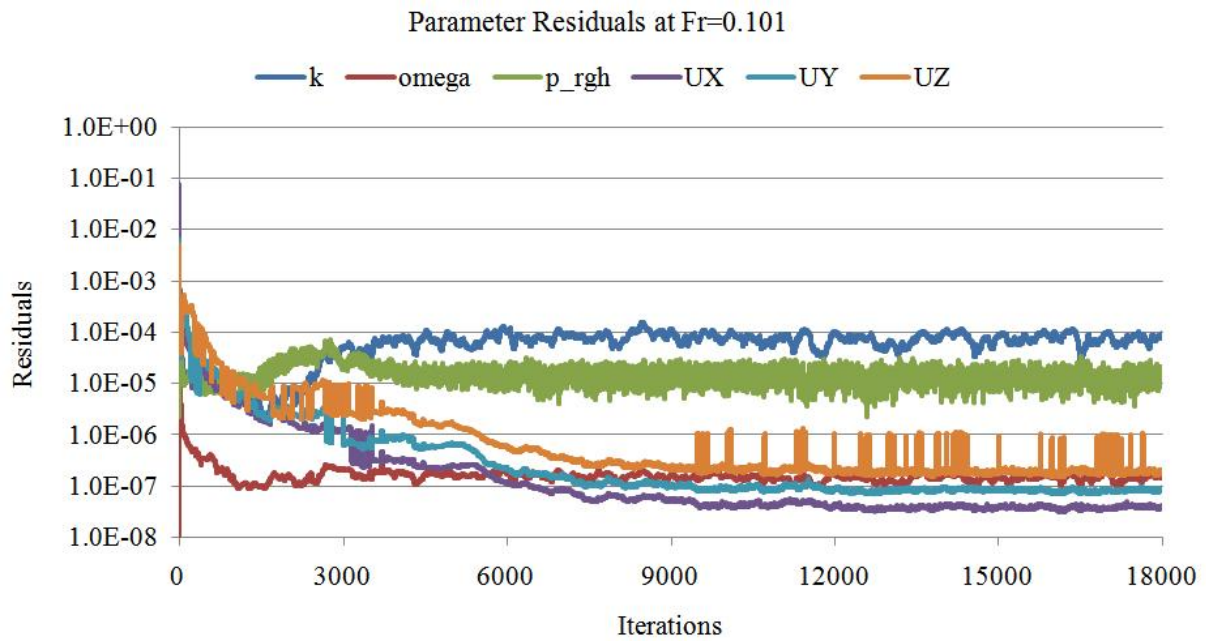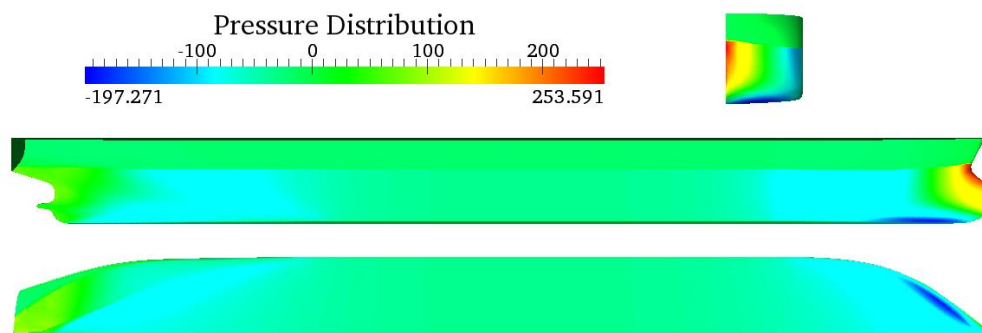

Figure 59. Pressure Distributions around JBC at $F_r = 0.101$ Case

Final resistance components of JBC in $F_r = 0.101$ case are calculated in Table 24 for all three mesh cases. The total resistance coefficient C_t in coarse mesh is relatively lower compared with the ones in the other two cases, where the results are close, 4.652×10^{-3} for medium mesh case and 4.642×10^{-3} for fine mesh case. The difference between these two cases is only 0.21%, which can be considered as mesh convergence. The same information can be seen in Figure 60. In the end, the medium mesh configuration in this case is also acceptable.

Table 25. Resistance ($\times 10^3$) of JBC at $F_r = 0.101$ with Different Grids

	cell No. (*million*)	$C_p \times 10^3$	$C_v \times 10^3$	$C_{pv} \times 10^3$	$C_t \times 10^3$
coarse	2.69	1.177	3.441	0.586	4.617
medium	3.28	1.198	3.453	0.587	4.652
fine	4.24	1.206	3.436	0.571	4.642

Total Resistance Coefficient *Ct* over Number of Cells
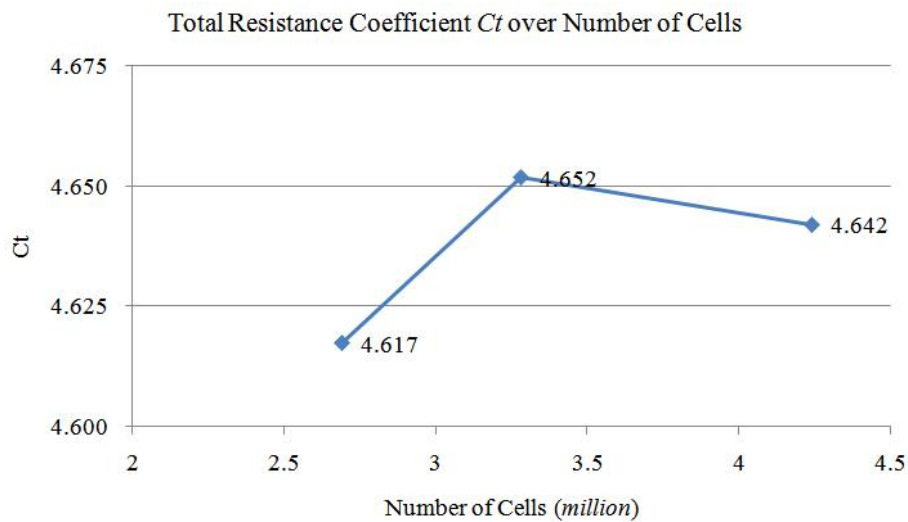
Figure 60. Total Resistance Errors over Cells for JBC at $F_r = 0.101$

As mentioned above, it is much difficult to measure the wave elevation of the vessel with high block coefficient C_B and not high speed. In this case, the model speed is even lower, which makes the measurement task worse. So no completed wave contour of JBC in $F_r = 0.101$ case will be given here and only a sketch of the wave field is shown in Figure 61. It is seen that the wave field around the JBC hull in this case is quite symmetric and only a few wave profiles occur in the stern of the hull. Furthermore, the wave elevations in the same three wave cutting planes can be seen in Figure 62, Figure 63 and Figure 64.
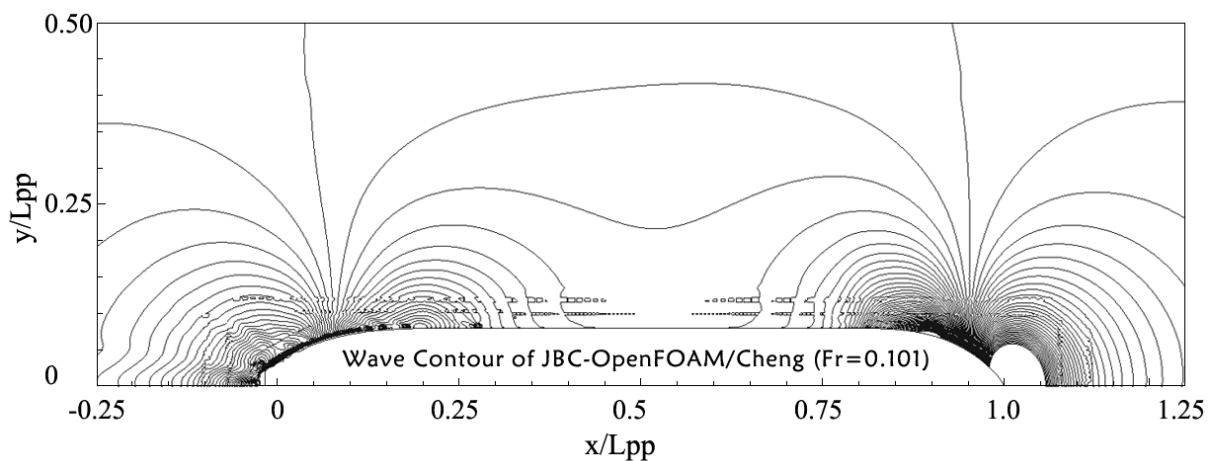
Figure 61. The Sketch of the Symmetric Wave Contour of JBC at $F_r = 0.101$ Case

Figure 62. The Wave Elevation of JBC at $y/L_{PP} = 0.085$ at $F_r = 0.101$

Figure 63. The Wave Elevation of JBC at $y/L_{PP} = 0.140$ at $F_r = 0.101$

Cheng XU

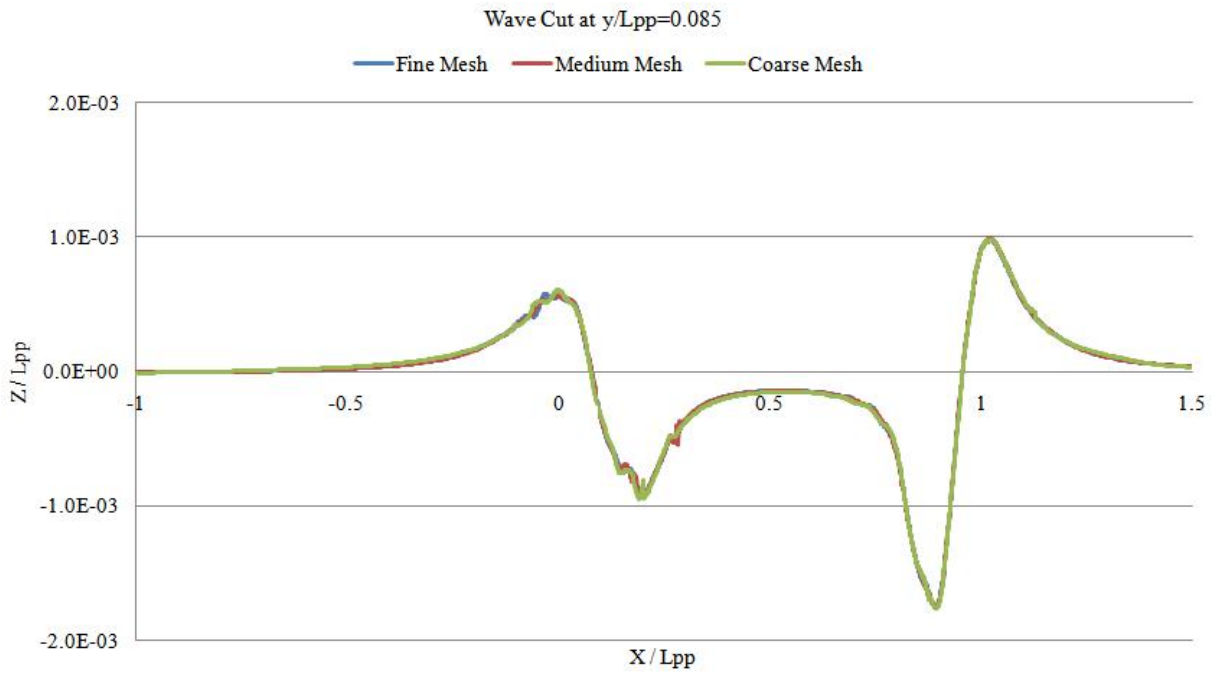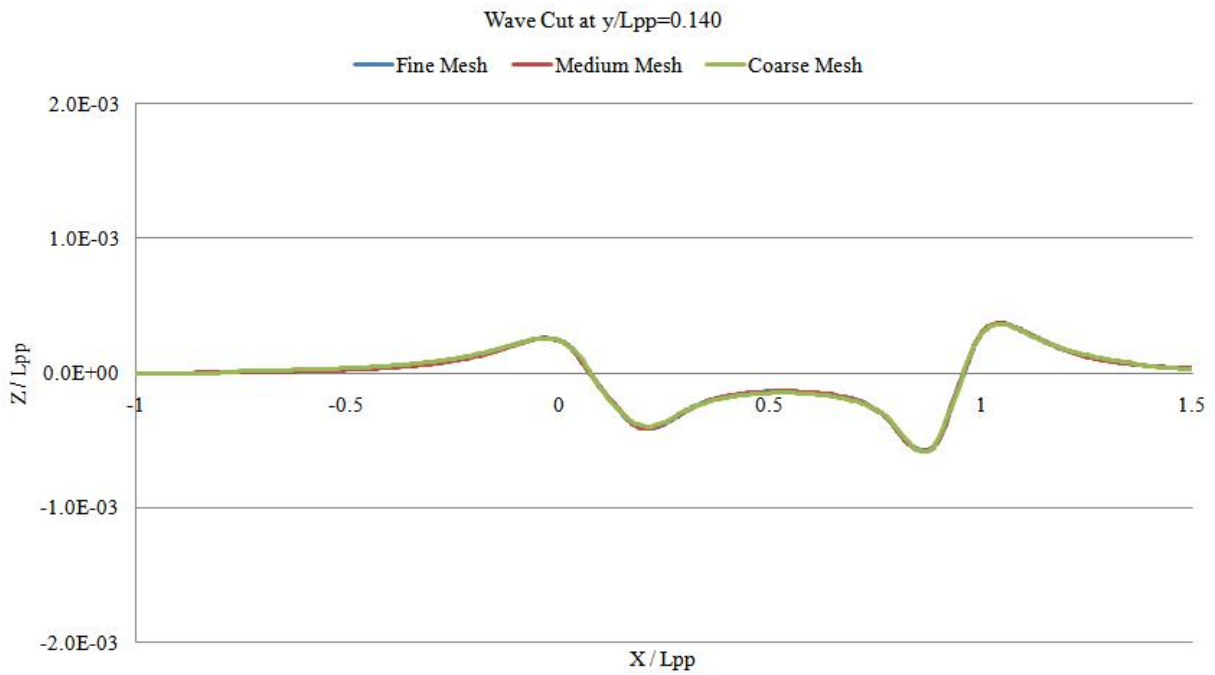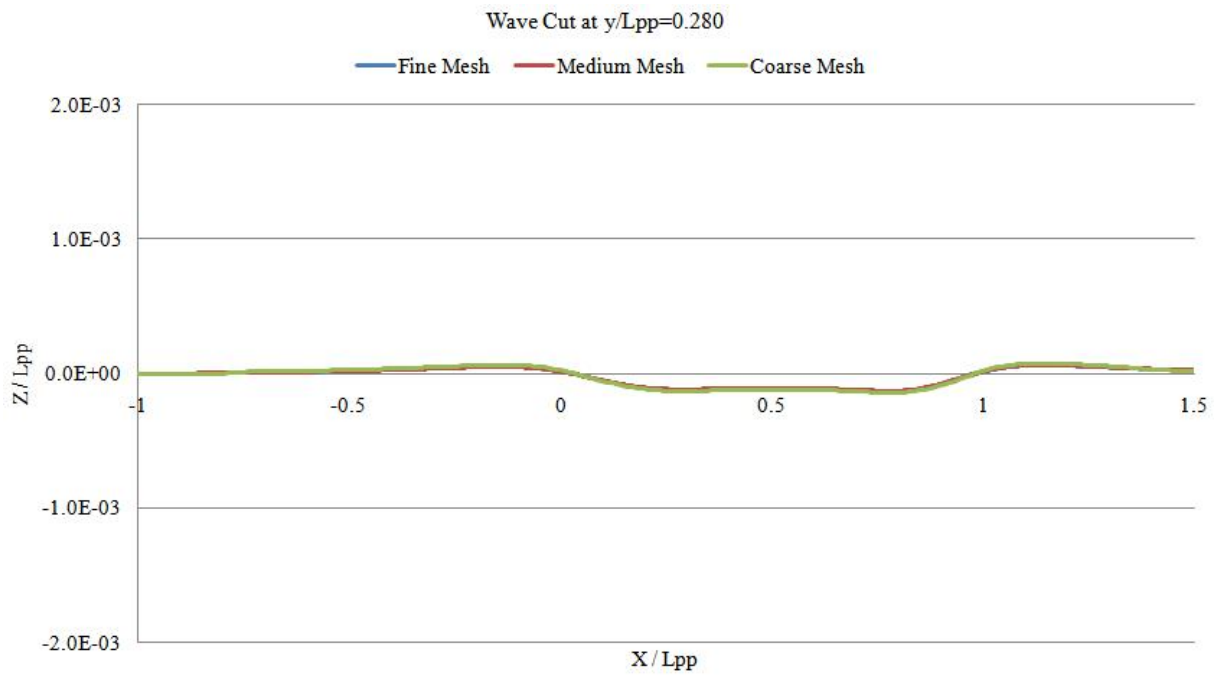Wave Cut at y/Lpp=0.280

Fine Mesh ——— Medium Mesh ——— Coarse Mesh

Figure 64. The Wave Elevation of JBC at $y/L_{PP} = 0.280$ at $F_r = 0.101$

7. CONCLUSIONS AND DISCUSSION

This thesis presents the resistance prediction and wave elevations of JBC bulker by solving Reynolds Average Navier-Stokes Equations with CFD OpenFoam solvers. The systematic investigations including the turbulence characteristics and the mesh convergence are implemented with the reference ship hull KVLCC2, which leads to improve the model parameters. Free surface model is applied for solving pressure and viscous forces by using LTSInterFoam solver and double body model is used for calculating the form drag and viscous force by simpleFoam solver. Afterwards, all the resistance components including wave resistance can be calculated.

Since resistance prediction with fixed hull operational condition is a kind of steady state problem, OpenFOAM performs quite well at the final simulation results. Except Level 1 coarsest mesh case, all the calculated results including total resistance and frictional resistance are within 3.3% errors compared with experimental data and empirical formula. As for the free surface around the hull, the predicted wave elevation distributions show a good agreement with the experimental measurement in the vicinity of the vessel, while they will no longer close to the measured values in the far lateral distance. Likewise, more precise wave contours from paraFoam are obtained in the vicinity of the hull while there is a discrepancy at the far field. The converged mesh configuration can be found in Level 4 case and very low errors of total resistance (1.7% error for $Fr = 0.142$ case and 2.2% error for $Fr = 0.101$ case) are obtained for reference vessel KVLCC2. Finally, a model scale of 1/55.1 is calculated for JBC bulker based on Reynolds similarity of turbulence flow around two hulls and Froude similarity of model and real JBC ship. All the settings of mesh and files based on the best configuration from KVLCC2 will be updated in order to calculate the resistance prediction of JBC bulker with the same solvers. Results are presented for resistances and wave elevations at three self-defined wave cuts, which can be the reference numerical CFD results for 2015 Workshop in Hydrodynamics.

In conclusion, the study presented in this project fulfilled all the objectives of this thesis. Nevertheless, there are still some shortcomings or improvements during the whole simulation procedures and several main aspects will be pointed out in the following parts:

1. As mentioned in Chapter 3, there are numbers of parameters in both initialization files and mesh files. Considering the short period, part of the parameters are kept unchanged

from the tutorial folders and some types of the discretized schemes such as *divSchemes* are recommended by the supervisor. For this case, it is also nice to test and study the influence of each parameter. In addition, for the determination of turbulence intensity, the interval step of 1% is implemented in this thesis. Although different turbulence intensity within a certain range will not affect the results that much, it is also necessary to get more precise turbulence intensity by either different determination methods or different turbulence models. After that, improved parameters of the simulation will help to obtain better results.

2. The EFD data of KVLCC2 and JBC is measured from different towing tanks, of course the working conditions like temperature or turbulence intensity cannot be exactly the same. This means, the turbulence intensity 0.02 obtained from reference ship is not 100% correct for that of JBC bulker, which also makes the error to some extent.

3. For KVLCC2 vessel, in general, more than 20,000 iterations are set in the *controlDict* file in order to get a fully converged result. However the simulations in all the cases will converge before 8,000 iterations. This means, the computations will take more than double time. For some finer meshes with more than 3.0 M cells, they will take another 20 hours to complete 20,000 iterations by using 48-Core CPUs after reaching convergence. It is appreciated to give termination criteria to detect whether the simulation should be stopped or not. This will largely save the computing time and accelerate the CFD investigation effectively.

4. VOF method only gives the average volume ratios of water inside each cell. With of help of paraFoam, the free surface can be contoured by *alpha* interpolation at the value of 0.5 and there is no problem to get accurate pressure and velocity fields in the vicinity of the hull. However for contouring the wave elevation, there is a large difference between the simulation and measurement. The wave contour is not easy to be captured precisely due to the very small gap between each two continuous contour lines, especially in the lateral field of the hull with coarse mesh. Using the same finer mesh at the lateral far field would improve the wave pattern, but it will need more computer resource and largely increase the computing time.

5. In mesh validation part, the converged mesh is determined from Figure 37 and Figure 46 by plotting the total resistances over number of cells. It is ok to get the converged mesh

visually in these cases because it is obvious that both points of Level 4 case in the polylines reach convergence conditions. However, more quantitative method called Richardson Extrapolation (Stern *et al*, 1999) can be adopted for examining the spatial convergence of CFD simulations. Richardson Extrapolation is not a new method now to analyze and improve the convergence rate and the accuracy of the numerical approximations in CFD mesh validation stages.

ACKNOWLEDGEMENT

REFERENCE

[1]. Chun Shen, Fengxian Sun and Xinlin Xia. 2014. Implementation of density-based solver for all speeds in the framework of OpenFOAM. *Computer Physics Communications*, 185(2014), 2730-2741.

[2]. Kevin W. Linfield and Robert G. Mudry, 2008. *Pros and Cons of CFD and Physical Flow Modeling* [online]. Airflow Sciences Corporation. Available from http://www.airflow sciences.com/sites/default/files/docs/Pros-and-Cons-of-CFD-and-physical-flow-modeling [Accessed 2 December 2015]

[3]. Kadir B. Korkmaz, 2015. *CFD Predictions of Resistance and Propulsion for the JAPAN Bulk Carrier (JBC) with and without an Energy Saving Device*. Thesis (Master). Chalmers University of Technology.

[4]. Eric Furbo, Janne Harju and Henric Nilsson, 2009. *Evaluation of Turbulence Models for Prediction of Flow Spearation at a Smooth Surface*. Report in Scientific Computing Advanced Course. Uppsala University.

[5]. L. Eca, M. Hoekstra and H. C. Raven, 2010. Calculation of the Flow around the KVLCC2 Tanker: Case 1.1a. *Gothenburg 2010-A Workshop on Numerical Ship Hydrodynamics*, Vol II, 547-552

[6]. Tao Xing, Pablo Carrica and Frederick Stern, 2010. Large-Scale RANS and DDES Computations of KVLCC2 at Drift Angle 0 Degree. *Gothenburg 2010-A Workshop on Numerical Ship Hydrodynamics*, Vol II, 511-516

[7]. M. Manzke, T. Rung, 2010. Resistance Prediction and Seakeeping Analysis with FreSCo+, *Gothenburg 2010-A Workshop on Numerical Ship Hydrodynamics*, Vol II, 639-644

[8]. Hamid Sadat-Hosseini *et al*, 2010. URANS Simulation and Validation of Added Resistance and Motions of the KVLCC2 Crude Carrier with Fixed and Free Surge Conditions. *Gothenburg 2010-A Workshop on Numerical Ship Hydrodynamics*, Vol II, 517-522

[9]. W.J.Kim, S.H. Van and D. H. Kim, 2001. Measurement of Flows around Modern Commerical Ship Models. *Experiments in Fluids*, 31(2001), 567-578.

[10]. Tobias Martin, 2014. *Validation of the Hybrid Turbulence Model for the Free Surface Ship Flows*. Thesis (Bachelor). University of Rostock.

[11]. Kevin M., 2011. *Ship Resistance Simulations with OpenFOAM-6th OpenFOAM Workship*. University of Michigan.

[12]. F. R. Menter, 1993. Zonal Two Equation $k - \omega$ Turbulence Models for Aerodynamic Flows. *American Institute of Aeronautics and Astronautics Paper*. 1993, 1993-2906.

[13]. E.Ghasemi, Soheil Soleimani and C.X. Lin, 2014. RANS Simulation of Methane-air Burner using Local Extinction Approach within Eddy Dissipation Concept by OpenFOAM. International Communications in Heat and Mass Transfer 54(2014), 96-102.

[14]. OpenFOAM The Open Source CFD Toolbox-User Guide [online], 2015. Version 3.0.0.

Available from http://foam.sourceforge.net/docs/Guides-a4/UserGuide.pdf [Accessed 7 July 2015].

[15]. Shaun Wortley, 2013. *CFD Analysis of Container Ship Sinkage, Trim and Resistance*. Thesis (Bachelor). Curtin University.

[16]. User Manual-HEXPRESS Unstructured Grid Generator, 2013. Version 3.1. Available from http://www.numeca.com/en [Accessed 29 June 2015]

[17]. Y. Saad, 2003. Iterative Methods for Sparse Linear Systems. *Society for Industrial and Applied Mathematics*. 2nd Edition.

[18]. ITTC Recommended Procedures-Resistance Uncertainty Analysis, Example for Resistance Test [online]. 2002, Version 7.5-02-02-02. Available from http://ittc.info /downloads/Archive%20of%20recommended%20procedures/2002%20Recommended% 20Procedures/7.5-02-02-02.pdf. [Accessed 26 October 2015]

[19]. ITTC, 2011. Recommended Procedures and Guidelines: Practical Guidelines for Ship CFD. Technical Report, International Towing Tank Conference.

[20]. Jin Kim *et al*, 2010. Feasibility Study on Numerical Towing Tank Application to Predictions of Resistance and Self-Propulsion Performances for a Ship. *Gothenburg 2010-A Workshop on Numerical Ship Hydrodynamics*, Vol II, 563-568

[21]. G.B.Deng *et al*, 2010. Verification and Validation for Unsteady Computation. *Gothenburg 2010-A Workshop on Numerical Ship Hydrodynamics*, Vol II, 447-452

[22]. Fred Stern et al, 1999. *Verification and Validation of CFD Simulation*. IIHR Report No. 407. Iowa Institute of Hydraulic Research.

APPENDIX

OpenFOAM Main Files (JBC Bulker)

A-1. Omega file

```
//********************************************************//
dimensions      [0 0 -1 0 0 0 0];

internalField   uniform 1.282225;

boundaryField
{
  out
  {
    type          zeroGradient;
  }
  farps
  {
    type          zeroGradient;
  }
  in
  {
    type          fixedValue;
    value         uniform 1.282225;
  }
  top
  {
    type          zeroGradient;
  }
  bottom
  {
    type          omegaWallFunction;
    Cmu           0.09;
    kappa         0.41;
    E             9.8;
    beta1         0.075;
    value         uniform 1.282225;
  }
  ship
  {
    type          omegaWallFunction;
    Cmu           0.09;
    kappa         0.41;
    E             9.8;
    beta1         0.075;
    value         uniform 1.282225;
  }
  sym
  {
    type          symmetryPlane;
  }
}
//********************************************************//
```

A-2. U file

```
//********************************************************//
dimensions      [ 0 1 -1 0 0 0 0 ];

internalField   uniform (-1.003 0 0);
```

```
boundaryField
{
 in
 {
  type            fixedValue;
  value           uniform (-1.003 0 0);
 }
 bottom
 {
  type            zeroGradient;
 }
 top
 {
  type            pressureInletOutletVelocity;
  value           uniform  (-1.003 0 0);
 }
 out
 {
  type            zeroGradient;
 }
 farps
 {
  type            zeroGradient;
 }
 ship
 {
  type            fixedValue;
  value           uniform (0 0 0);
 }
 sym
 {
  type            symmetryPlane;
 }
}
//************************************************************//
```

A-3.   transportProperties file

```
//************************************************************//
phase1
{
  transportModel  Newtonian;
  nu              nu      [ 0 2 -1 0 0 0 0 ]          1.107e-06;
  rho             rho     [ 1 -3 0 0 0 0 0 ]          1000;
}

phase2
{
  transportModel  Newtonian;
  nu              nu      [ 0 2 -1 0 0 0 0 ]          1.48e-05;
  rho             rho     [ 1 -3 0 0 0 0 0 ]          1;
}

sigma           sigma   [ 1 0 -2 0 0 0 0 ]           0;
//************************************************************//
```

A-4.   controlDict file

```
//************************************************************//
application     LTSInterFoam;
```

Master Thesis developed at University of Rostock, Germany

```
startFrom           latestTime;

startTime           0;

stopAt              endTime;

endTime             24000;

deltaT              1;

writeControl        timeStep;

writeInterval       3000;

purgeWrite          0;

writeFormat         ascii;

writePrecision      6;

writeCompression        compressed;

timeFormat          general;

timePrecision       7;

runTimeModifiable       yes;
adjustTimeStep          yes;
maxCo                   1;
maxAlphaCo              1;

maxMeshCo               0.01;

maxDeltaT               1;

libs
 (
   "libOpenFOAM.so"
   "libincompressibleRASModels.so"
   "libforces.so"
 );

functions
{
 forces
 {
   CofR                (0 0 0);
   UName               U;
   functionObjectLibs
     (
       "libforces.so"
     );
   log                 true;
   outputControl       timeStep;
   outputInterval      3;
   pName               p;
   patches
     (
       ship
     );
   rhoInf              1000;
   rhoName             rho;
```

```
  type                  forces;
 }
}
//**********************************************************//
```

A-5.   fvSchemes file
```
//**********************************************************//
ddtSchemes
{
   default               localEuler rDeltaT; //Euler;
}

gradSchemes
{
   default               cellLimited Gauss linear 1;
   grad(U)               cellLimited Gauss linear 1;
   grad(alpha)           Gauss linear;
   grad(p_rgh)           Gauss linear;
}

divSchemes
{
   div(rho*phi,U)        Gauss limitedLinearV 1 phi;
   div(phi,alpha)        Gauss vanLeer;
   div(phirb,alpha)      Gauss interfaceCompression;
   div(phi,omega)        Gauss limitedLinear 1 phi;
   div(phi,k)            Gauss limitedLinear 1 phi;
   div((muEff*dev(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
   default               Gauss linear corrected;
}

interpolationSchemes
{
   default               linear;
}

snGradSchemes
{
   default               corrected;

}

fluxRequired
{
  p_rgh;
  pcorr;
  alpha;
}
//**********************************************************//
```

A-6.   fvSolution file
```
//**********************************************************//
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    2;

    nAlphaCorr                  1;
    nAlphaSubCycles             2;
```

```
    cAlpha                      1;

    maxCo                       0.9;
    maxAlphaCo                  0.2;
    nAlphaSweepIter             1;

    rDeltaTSmoothingCoeff       0.1;
    rDeltaTDampingCoeff         1;
    maxDeltaT                   1;
}

relaxationFactors
{
  fields
  {
  p_rgh                         0.3;
  }
  equations
  {
  k                             0.5;
  omega                         0.5;
  U                             0.7;
  }
}
// ****************************************************************** //
```