CFD study of ship-to-bank interaction

Kim, Youngkook

2015

Kim, Y. (2015). CFD study of ship-to-bank interaction. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/65310>

<https://doi.org/10.32657/10356/65310>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

CFD STUDY OF SHIP-TO-BANK INTERACTION

KIM YOUNGKOOK

SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING

2015

CFD Study of Ship-To-Bank Interaction

KIM YOUNGKOOK

School of Mechanical & Aerospace Engineering

A thesis submitted to the Nanyang Technological University

in fulfillment of the requirement for the degree of

Master of Engineering

2015

Acknowledgement

I would like to express my gratitude to Associate Professor Eddie Ng Yin Kwee who is my supervisor for his excellent guidance and keen support. Prof. Ng was always patient and supportive for my problems and questions during the research period.

I also would like to extend my gratitude to NTU HPC support team, especially Soh Hwee Jin (Melvin) and Bong Zoebir who used to fix the problems and kindly updated the newer version of OpenFOAM in the HPC server so that I could use OpenFOAM 2.3.0 for my research.

Furthermore, I would like to appreciate to my wife, Cho Eunjin who supported my study in all.

Mar 2015

Kim Youngkook

Table of Contents

ACKNOWLEDGEMENT	2
TABLE OF CONTENTS	3
LIST OF FIGURES	5
LIST OF TABLES	8
ABSTRACT.....	9
NOMENCLATURE	10
CHAPTER 1 INTRODUCTION.....	15
1.1 BACKGROUND	15
1.2 OBJECTIVES	18
1.3 SCOPE OF RESEARCH	19
1.4 LITERATURE REVIEW.....	20
1.4.1 <i>Experimental approaches of ship-to-bank interaction</i>	<i>20</i>
1.4.2 <i>Numerical approaches of ship-to-bank interaction.....</i>	<i>22</i>
1.5 ORDER OF REPORT	26
CHAPTER 2 NUMERICAL METHODS.....	28
2.1 CFD THEORY	28
2.1.1 <i>Potential flow.....</i>	<i>28</i>
2.1.2 <i>RANS (Reynolds Averaged Navier-Stokes) equations</i>	<i>29</i>
2.1.3 <i>Finite Volume Method (FVM).....</i>	<i>33</i>
2.1.4 <i>Volume of fluid method.....</i>	<i>36</i>
2.2 CFD ALGORITHM	38
2.2.1 <i>Numerical discretisation and solution algorithm.....</i>	<i>38</i>
2.2.2 <i>Pressure corrections</i>	<i>44</i>
2.2.3 <i>Convergence and residuals.....</i>	<i>48</i>
2.2.4 <i>Courant number</i>	<i>50</i>
2.2.5 <i>Boundary conditions.....</i>	<i>52</i>
2.2.6 <i>Boundary layer treatment and y plus.....</i>	<i>54</i>
2.2.7 <i>Good practice of boundary layer treatment.....</i>	<i>58</i>
2.3 MATHEMATICAL MODEL OF SHIP-HANDLING SIMULATOR	61

CHAPTER 3	OPEN SOURCE CFD CODE	66
3.1	OPEN SOURCE CFD CODES AND OPENFOAM	66
3.2	OPENFOAM SOFTWARE STRUCTURES	68
3.2.1	<i>Architecture</i>	68
3.2.2	<i>Solvers</i>	69
3.2.3	<i>Numerical algorithm</i>	70
3.3	MULTI-PHASE SOLVER INTERFOAM AND LTSINTERFOAM	73
3.4	OPENFOAM EVALUATIONS IN OPEN WATER	76
3.4.1	<i>KVLCC2</i>	79
3.4.2	<i>KCS</i>	83
3.4.3	<i>Summary of the evaluations</i>	94
CHAPTER 4	ANALYSIS IN CONFINED WATER	96
4.1	COMPUTATIONAL SET-UP AND MESH	96
4.2	BOUNDARY CONDITIONS	105
CHAPTER 5	RESULTS AND DISCUSSION	109
5.1	COMPUTATION RESULTS	109
5.1.1	<i>Force and moment</i>	109
5.1.2	<i>Under keel space modelling of the case H</i>	116
5.1.3	<i>Pressure and velocity asymmetry</i>	122
5.1.4	<i>Results of the cases with propeller rotation</i>	128
5.1.5	<i>Comparison between single-phase and two-phase solutions</i>	129
5.1.6	<i>Free surface elevation</i>	130
5.2	DISCUSSION	136
CHAPTER 6	CONCLUSION	139
6.1	CONCLUSION	139
6.2	FUTURE WORK	141
APPENDIX	THE SOURCE CODES OF INTERFOAM	142
REFERENCES		152
PUBLICATION FROM THE PRESENT WORK		157

List of Figures

Figure 2-1: Finite volume discretisation [21].....	35
Figure 2-2: Explicit (upper) and implicit (upper) method [13]	43
Figure 2-3: Comparison of implicit and explicit solutions for same time step [7]	51
Figure 2-4: Velocity distribution near a solid wall [34]	57
Figure 2-5: Coordinate system of manoeuvring mathematical model	62
Figure 2-6: Six degrees of freedom of motion	63
Figure 2-7: The trajectory in ship-handling simulator with ship- to-bank interaction x and y axis: distance(m) from the initial point.....	65
Figure 3-1: OpenFOAM case structure.....	68
Figure 3-2: Profile and front view of KVLCC2.....	79
Figure 3-3: Convergence history of the resistance of KVLCC2	81
Figure 3-4: Comparison of wave elevation along the hull of KVLCC2 (Experiment result: Kim et al. [38])	82
Figure 3-5: Comparison of wave pattern of KVLCC2 of CFD (upper) and the experiment (Kim et al. [38]) (bottom)	83
Figure 3-6: Profile and front view of KCS.....	84
Figure 3-7: Mesh of KCS - (a) whole domain, (b) forward and (c) stern area	85
Figure 3-8: y^+ distributions on the hull.....	86
Figure 3-9: Convergence time history of the resistance (N) of KCS by (a) Gauss linear and (b) Least square gradient scheme	90
Figure 3-10: Comparison of wave elevation along the KCS hull (Experiment results: Kim et al. [38])	91

Figure 3-11: Comparison of wave pattern of KCS of OpenFOAM (upper) and the experiment (Kim et al., [38]) (bottom)	91
Figure 3-12: Wave pattern at the forward (a) and the stern (b)....	92
Figure 3-13: Wake distribution (axial velocity [m/s]) of KCS at (a) the propeller plane and (b) at the station no. 1 (left : experiment (Kim et al. [38]), right : OpenFOAM result)	94
Figure 4-1: The coordinate system.....	97
Figure 4-2: Schematic view of the bank arrangement (looking from the aft).....	97
Figure 4-3: Graphical view of the case C and the case H.....	100
Figure 4-4: Mesh of (a) whole domain and (b) near the hull (looking from the top), and (c) mesh on the hull surface of case H	103
Figure 5-1: Convergence history of (a) X and (b) Y forces (N), and (c) Z moment (Nm) from pressure of case F (Gradient scheme: Gauss linear, Max Co. reduced from 70s)	110
Figure 5-2: Convergence history of (a) X and (b) Y forces (N), and (c) Z moment (Nm) from pressure of case F (Gradient scheme: Least square)	111
Figure 5-3: Different meshing of case H - no layer (a), three layers (b), eight layers (c), fine mesh (d)	117
Figure 5-4: y^+ of the case H with different meshing no layer (a), three layers (b), eight layers (c), fine mesh (d)	118
Figure 5-5: Pressure [N] distribution in the water at the centreline of the case H with (b) three layers (c) eight layers	120
Figure 5-6: Pressure [N] distribution on the hull (looking from the bottom) of the case H with (b) three layers (c) eight layers	121

Figure 5-7: Pressure [N] distribution on the hull and bank surface of the cases F (a) and H (b) (looking from the top)	122
Figure 5-8: Dynamic pressure [N] distribution of OpenFOAM calculation of case A (a), F (b) and H (c) at the different sections (station 2.5, amidships and station 17.5)	124
Figure 5-9: Velocity magnitude (m/s) of the case F at (a) free surface, (b) 2/3 of draught from baseline, (c) 1/3 of draught from baseline and (d) baseline	125
Figure 5-10: Velocity magnitude (m/s) of the case H at (a) free surface, (b) 2/3 of draught from baseline, (c) 1/3 of draught from baseline and (d) baseline	126
Figure 5-11: Velocity magnitude (m/s) of the case I at (a) free surface, (b) 2/3 of draught from baseline, (c) 1/3 of draught from baseline and (d) baseline	127
Figure 5-12: The locations of the wave gauges 1, 2 and 3 of the cases (a) F, (b) I and (c) J	131
Figure 5-13: Wave elevation of the case F at (a) gauge 1, (b) gauge 2 and (c) gauge 3 (Lataire et al. [5])	132
Figure 5-14: Wave elevation of the case I at (a) gauge 1, (b) gauge 2 and (c) gauge 3 (Lataire et al. [5])	133
Figure 5-15: Wave elevation of the case J at (a) gauge 1, (b) gauge 2 and (c) gauge 3 (Lataire et al. [5])	134

List of Tables

Table 3-1: Principal dimensions of KVLCC2 model.....	79
Table 3-2: Grid convergence study (The experiment result: Kim et al. [38])	80
Table 3-3: Principal dimensions of KCS model.....	83
Table 3-4: The comparison of KCS resistance by different solvers (Experiment results: Kim et al. [38]).....	88
Table 4-1: Principal dimension of model ship	96
Table 4-2: The test cases without propeller thrust.....	99
Table 4-3: The test cases with propeller thrust	101
Table 4-4: Boundary conditions for confined water analysis of the single-phase solver	106
Table 4-5: Boundary conditions for confined water analysis of the multi-phase solver	107
Table 5-1: Comparison of the cases without thrust to the experiment results (Lataire et al. [5])	114
Table 5-2: Comparison between different meshing in the under keel space of the case H.....	120
Table 5-3: Comparison of the cases with thrust to the experiment results (Lataire et al. [4]).....	128
Table 5-4: The locations (m) of the wave gauges with referenrece to the centreline of the ship (Lataire et al. [5])	130

Abstract

Ship-to-bank interaction is a complex physical phenomenon that involves not only in the asymmetric pressure field near banks or channels but also shallow water effect. Traditionally many experimental studies were carried out in this field. As numerical method is getting popular, there were various computational approaches as well. In this study, CFD (Computational Fluid Dynamics) analysis is carried out to investigate ship-to-bank interaction. CFD analysis is also useful for finding hydrodynamic coefficients in manoeuvring mathematical model of the ship-handling simulator. For the numerical method, RANS based solvers with free surface capturing feature are used.

Flow around a container ship in confined water is investigated with the open source C++ toolbox, OpenFOAM. Computations with several bank arrangements and different settings are performed. The OpenFOAM results are also compared with experiment results for validation. Meshing, numerical scheme, numerical settings and calculation results are evaluated and discussed in this study.

Nomenclature

E	Integration constant of the roughness of the wall	-
Fn	Froude number ($Fn = U/\sqrt{gL_{pp}}$)	-
$F (F_i)$	Total force (in i coordinate)	N
g	Acceleration of gravity	m/s ²
H	H Operator (See Section 2.2.2)	-
I	Tensor matrix of inertia of body	kg·m ²
K	Roll moment	Nm
k	Sum of the mean kinetic energy of turbulence	m ² /s ²
L_{pp}	Length between perpendiculars	m
$M (M_i)$	Total moment (in i coordinate)	Nm
M	Pitch moment	Nm
$\dot{M}$	Mass flux	kg/(m ² ·s)
N	Yaw moment	Nm
p	Pressure	m ² /s ²
p_{rgh} (p_d)	Dynamic pressure	kg/(m·s ²)
S_f	Surface area of the cell face	m ²
t	time	s
U	Speed vector (u, v, w)	m/s

u_τ	Friction velocity	m/s
u^+	Non-dimensional velocity near the wall	-
w	Artificial velocity normal to the interface of volume fraction	m/s
X	Resistance / Longitudinal force	N
x_0	Origin of measurement	m
x_f	Distance from origin to the face centre	m
Y	Transverse force / Sway force	N
y_B	Distance from bank wall to the centreline of ship	m
y_p	Distance from the wall to the first grid point	m
y^+	Non-dimensional y_p	-
Z	Vertical force / Heave force	N
Γ	Diffusion coefficient	m ² /s
α	Scalar field of volume fraction	-
γ	Under-relaxation factor	-
ε	Rate of viscous dissipation of turbulent kinetic energy	m ² /s ³
κ	Surface curvature	1/m
κ	Von Karman's constant	-
μ	Dynamic viscosity	kg/(m·s)
ν	Kinematic viscosity	m ² /s

ρ	Density of fluid	kg/m^3
σ	Surface tension coefficient	N/m
τ_{ij}	Reynolds stress	$\text{kg}/(\text{m}\cdot\text{s}^2)$
τ_w	Wall shear stress	$\text{kg}/(\text{m}\cdot\text{s}^2)$
ϕ	Velocity potential	m^2/s
Ω	Angular velocity vector (p, q, r)	rad/s
ω	Turbulent frequency	$1/\text{s}$

Abbreviations

CFD	Computational Fluid Dynamics
CFL	Courant Friedrichs Lewy
CPU	Central Processing Unit
CV	Control Volume
DOF	Degree(s) of Freedom
EXP	Experiment
FDM	Finite Difference Method
FHR	Flanders Hydraulics Research
FVM	Finite Volume Method
HPMM	Horizontal Planar Motion Mechanism
KCS	KRISO Container Ship
KRISO	Korea Research Institute of Ship and Ocean engineering
KVLCC	KRISO Very Large Crude-oil Carrier
LES	Large Eddy Simulation
LNG	Liquefied Natural Gas
LTS	Local Time Stepping
MOERI	Maritime & Ocean Engineering Research Institute
MULES	Multi-dimensional Universal Limiter for Explicit Solution
ODE	Ordinary Differential Equation

OpenFOAM	Open Field Operation And Manipulation
OS	Operating System
PDE	Partial Differential Equation
PIMPLE	(Merged method of) PISO and SIMPLE
PISO	Pressure Implicit with Splitting of Operators
RANS	Reynolds Averaged Navier-Stokes (equation)
RPM	Revolution Per Minute
SIMPLE	Semi-Implicit Method for Pressure-Linked Equations
SST	Shear Stress Transport
STL	STereo-Lithography
TEU	Twenty-foot Equivalent Unit
VoF	Volume of Fluid

Chapter 1

Introduction

1.1 Background

When a ship navigates in the proximity of banks or harbours, the pressure field near the ship becomes asymmetric and ship experiences repelling or attracting forces. It makes hard for the ship to manoeuvre and can lead to an accident. In such cases, it is advised to travel at the middle of water to minimise any interaction from the bank or the harbour. However, as this is not always possible, it is important to understand the phenomenon to prevent any dangerous situation. Basically ship-to-bank interaction causes sway force and yaw moment on the ship when separation from the bank wall is small. Moreover, small under keel clearance in confined water makes the interaction even stronger due to the shallow water effect. In addition, there are several other factors that contribute such as propeller effect and bank shape. Because ships have been increased in size and length in recent years, the importance of prediction and understanding of ship-to-bank interaction are thus getting more significant.

Nowadays, ship-handling simulators play an important role in training captains and crew for safe operation and preventing from accidents. In the mathematical model of the ship-handling simulator, various forces and moments that arise from hull, rudder, propeller, wind, wave, bank effect, etc have to be carefully modelled for realistic simulations. The best way to obtain those forces and moments is to carry out model tests. Captive model test such as HPMM (Horizontal Planar Motion Mechanism) test enables measuring of each force and moment components. However, the test is costly and time consuming. Therefore, it is normal to use empirical methods that are based on the regression formulas from database to estimate these forces and moments. When the accuracy is concerned, CFD (Computational Fluid Dynamics) analysis could give better accuracy. Nowadays, it is not much to say that CFD technology became essential in marine industry. As CFD technology is developing, the accuracy and reliability of CFD analysis are getting higher. There were many studies to calculate hydrodynamic forces and moments of mathematical model of ship-handling simulator using CFD analysis.

As for ship-to-bank interaction, it is quite straightforward to use CFD calculations and model them into ship-handling simulator. However, careful approach is needed to calculate and model the

complex phenomenon due to the interaction with banks and shallow water effect. Recently, open source CFD codes have become popular due to their strong capability. They are totally free of charge to use and free to modify the source code. On the other hands, it is a challenge to evaluate the accuracy of the code despite the steep learning curve of open source codes. In this study, it is decided to use an open source CFD code to analyse ship-to-bank interaction from various bank arrangements and compare them to the experiment results.

1.2 Objectives

This study aims to predict ship-to-bank interaction with an open source CFD code. The numerical analysis of multiphase RANS solver will be evaluated so that the effect from free surface, turbulence and viscosity can be taken into account. In addition, the results will be compared with single-phase RANS solver. Making use of the open source CFD code, it is expected to know capability and limitation of the open source code. Moreover, it would be helpful to understand numerical algorithms of CFD codes.

By performing numerical simulation, ship-to-bank interaction with different bank arrangements will be investigated. This would also identify the benefit from CFD analysis compared to the experimental approach.

1.3 Scope of research

The open source CFD code, OpenFOAM will be used for numerical analysis. Among numerous solvers in OpenFOAM package, a VoF multiphase solver will be used to predict ship-to-bank interaction. The solver will be evaluated for open water cases before analysis is carried out in confined water. Resistance of a container ship and a large tanker ship will be calculated and compared to the experiment results.

The OpenFOAM solver will be used to calculate ship-to-bank interaction in confined water and the results will be compared to the experiment. CFD calculations will cover from the comparisons of forces and moment to the evaluation of meshing, modelling and wave elevation.

1.4 Literature Review

1.4.1 Experimental approaches of ship-to-bank interaction

From 1970s to these days, many experimental studies have been carried out to investigate ship-to-bank interaction and researchers developed empirical formulas that could predict ship-to-bank interaction. During 1970s, SSPA (Swedish maritime research centre) carried out various tank tests for different bank types and showed how the forces and moments vary over transient period. It was found that sway force and yaw moment vary in proportion to the square of ship speed. Using this test results, empirical formulas were also suggested from model test results (Norrbin [23]).

However, some studies that were carried out later showed that ship-to-bank effect occurs in a more complex manner. A continued study done about 30 years later in SSPA (Li et al [31]) proved that there is a point where the sway force is switched from suction to repulsive force when under keel clearance is extremely small, approximately at about 10% of water draught. Moreover, unlike the study in early times, it turns out that the sway force tends to increase in a higher order than two with respect to the ship speed when under keel clearance is very small.

Even in some cases, it became more than order of two at 40% of under keel clearance. At a sloped bank with 30 degrees angle, it tends to reduce force and moments at low speeds but bank effects increase at high speeds as compared to a vertical bank at a low under keel clearance. Moreover, a suction force arises from a rotating propeller compared to the cases without propeller according to the experiments. It was found the more propeller loading caused the larger contribution of suction force.

Expansive shallow water tank tests were carried out at FHR (Flanders Hydraulics Research) in Belgium (Vantorre et al. [21]) (Lataire et al. [4] [5]). More than 10,000 model tests of the bank effect were performed and many factors that influence the bank effect were investigated. Through the experiments, it was confirmed that there is a transition point from attraction sway force to repulsion force in shallow water. Small difference from the test results of SSPA is that it happened at slightly higher under keel clearance which is in range of 15% to 20%. Therefore, it is understood that the transition from attraction to repulsion force appears at the under keel clearance of 10~20% water draught. The test results also showed that propeller action increased the bank effect. Similarly to the results of SSPA, propeller action changed repulsion force into suction force in shallow water where under keel clearance is less than 35% of

water draught. These tests results are also developed into empirical formulas. Several sets of test data are made available for the bench marking purpose of simulation model (Lataire et al. [5]). In this study, these open test results are used for the comparison to the CFD calculations.

1.4.2 Numerical approaches of ship-to-bank interaction

As numerical technique was developing, there were many computational studies on the bank effect. In early times, numerical method based on potential theory was popular in marine industry.

In many studies, panel method was used to evaluate the bank effect. When the free surface condition is linearised by double model solution, it was good enough to show qualitative predictions only when the draught and water depth ratio is not too small, such that the ratio is greater than 1.5. (Q. Miao and J. Xia [32] and Lee and Lee [14]). When non-linear boundary condition is applied, panel method was also able to calculate ship-to-bank interaction in extreme shallow water with qualitative agreement (Park et al. [38]). With this non-linear potential method, even the transition point from suction force to repulsion force was found at 40% of the draught-water depth

ratio which is slightly higher than the experiment results reported by Li et al. [31] and Lataire et al. [4].

As computational capability is rapidly improving, RANS based finite volume methods were also getting popular. However, it was quite recent that RANS based method started to be used in this topic.

Lo et al. [2] carried out transient analysis on the bank effect. Trajectory of KCS container travelling near the bank was illustrated. Sway force and yaw moment with time variations at various velocities were calculated at about 30% of the draught-water depth ratio. However, there was no comparison with experiment result.

Wang et al. [10] studied viscous analysis around series 60 hull with SST $k-\omega$ turbulence model from 50% to 1000% of the water depth – draught ratio and showed good agreement with experiments in sway forces but over prediction with the very short distance to the bank wall.

Zou et al. [19] used validation and verification method including a grid convergence study to evaluate RANS based CFD prediction of ship-to-bank interaction. The bank effect between KVLCC hull and a sloped bank was analysed with an overlapping grid of

structured mesh and $k-\omega$ SST turbulence model. CFD calculations in the comparison with model tests were done at 50%, 35% and even 10% of the draught-water depth ratio. Furthermore, not only sway force and yaw moment but also rolling, sinkage and trim moment were calculated and compared with potential flow analysis. Because sinkage and trim moment did not show big difference between the potential method and CFD calculation result, free surface effect at slow speed is considered to be insignificant for sinkage and trim. In the study, slip boundary condition was applied on the bank wall instead of no slip condition. The rationale for this boundary condition is not clear and the author also stated that the boundary condition may not be the accurate. The bank effect analysis gave good agreement in tendencies of yaw and roll moments but with under prediction of sway force.

Mehdi et al. [22] carried out simulation of the viscous flow around LNG ship using RANS method with unstructured hybrid mesh and $k-\omega$ turbulence model. Yaw moment was compared with the experiment results at various velocities and distance from the bank wall but sway force results were not presented in their paper. Calculated yaw moments at different distance from the bank wall and 60% of the draught-water depth ratio were

compared to the tank test results and showed qualitative agreement with some under prediction.

From the various studies done in the past, it is still a big challenge to use CFD for ship-to-bank interaction analysis with extreme shallow water. Many studies failed to calculate extreme conditions such as 10% of under keel clearance or did not predict ship-to-bank correctly.

During the literature review about numerical studies of the bank effect, viscous analysis including free surface effect on the bank effect was not found so far. When a ship travels at slow speed, it is no doubt that free surface elevation is not so high and the importance from free surface might be relatively low. Furthermore, it was found that free surface also does not give significant contribution on the sinkage and the trim at low speed (Zou et al. [19]). Nevertheless, it is believed to be worthwhile to carry out viscous flow analysis with free surface for more accurate prediction of the flow near the proximity of banks. It is also expected to see the difference between the analysis with and without free surface.

1.5 Order of report

The structure of this report is as follows:

Chapter 1 introduces overview of this research and ship-to-bank interaction. It also includes literature review that covers ship-to-bank interaction study done both in experimental and numerical approaches.

Chapter 2 covers numerical methods of CFD including basic theory. It discusses numerical algorithms, boundary condition and boundary layer theory. Moreover, mathematical model of ship-handling simulator will be explained.

Chapter 3 introduces open source CFD codes and the solver used for this study. Moreover, the evaluation of the solver in open water is included.

Chapter 4 outlines numerical setup of CFD analysis in confined water and this covers computational setup and boundary conditions of CFD calculation.

Chapter 5 deals with the results of CFD calculations in confined water and discussions. Various comparisons and evaluations will be carried out.

Chapter 6 includes the summary of current work and discusses the future plan.

Appendix contains the main source codes of Open source CFD tool that was used for this study.

Lastly, References and Publication of this study are listed.

Chapter 2

Numerical Methods

2.1 CFD theory

2.1.1 Potential flow

In marine industry, traditional panel method based on potential theory as assumptions was popular and it is still used by many people due to its simplicity and quick computation in spite of inviscid and irrotational property of flow. High Reynolds number of many marine applications also enables potential flow remains useful and efficient for the area where viscous effect is not predominant. Nevertheless, potential flow is not applicable where the flow is dominated by viscous drag and turbulence, e.g. near the aft hull. Basic theory of potential flow will be briefly explained here.

In potential flow, velocity potential (ϕ) exists and governing equation is Laplace equation of continuity,

$$\nabla^2 \phi = 0 \quad (2.1)$$

This method must satisfy several boundary conditions.

Firstly, no flux on the body and the wall as such,

$$\nabla\phi \cdot \vec{\eta} = 0 \quad (2.2)$$

Secondly, fluid particle should not leave the free surface which is kinetic boundary condition. Thirdly, pressure on the free surface should be the same as atmospheric pressure, which is dynamic boundary condition.

Normally, the last two boundary conditions have non-linear terms and they are often linearised for simplification. One of the traditional methods is double model solution, which reflects hull form at waterline of calm water. With double model approximation, flow with rigid free surface can be calculated. Nowadays, non-linear potential method is also available to describe free surface elevation. Non-linear potential method is very useful and fast, hence, popular in minimising wave breaking resistance and ship hullform optimisations.

2.1.2 RANS (Reynolds Averaged Navier-Stokes) equations

When viscosity and turbulence cannot be ignored in the flow, potential method is no more accurate. The most popular method

is RANS equations with turbulence model and it gives better accuracy than potential flow that is irrotational and inviscid.

Basically turbulence has fluctuating property and needs very fine mesh with huge computing resources to solve it directly. RANS method takes average of fluctuating properties of turbulent flow and can save computational effort, memory and time. Therefore, RANS method is most appropriate for industrial cases which require huge computation mesh or domain.

The governing equations generally come from the continuity equation (2.3) and the momentum equations (2.4) ~ (2.5) of incompressible flow.

$$\text{div } \mathbf{u} = 0 \quad (2.3)$$

$$\frac{\partial u}{\partial t} + \text{div}(\mathbf{u}\mathbf{u}) = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \text{div}(\text{grad}(\mathbf{u})) \quad (2.4)$$

$$\frac{\partial v}{\partial t} + \text{div}(\mathbf{v}\mathbf{u}) = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \text{div}(\text{grad}(\mathbf{v})) \quad (2.5)$$

$$\frac{\partial w}{\partial t} + \text{div}(\mathbf{w}\mathbf{u}) = -\frac{1}{\rho} \frac{\partial p}{\partial z} + \nu \text{div}(\text{grad}(\mathbf{w})) \quad (2.6)$$

The first term on the left hand side of the momentum equations represents the time derivatives and is also called as unsteady

term. The second term is the convection term. On the right hand side, the first and the second terms are pressure term and viscous term, respectively.

RANS equations can be derived from the momentum equations by taking the time average.

Velocity vector and pressure vector in turbulent flow can be expressed as sum of steady mean component (U, V, W and P) and fluctuating component (u', v', w' and p').

$$u = U + u' \quad v = V + v' \quad w = W + w' \quad p = P + p' \quad (2.7)$$

As integration or differentiation of fluctuating flow is similar to time averaging operation, so the followings can be achieved.

$$\begin{aligned} \overline{\frac{\partial u}{\partial t}} &= \frac{\partial U}{\partial t} & \overline{\text{div}(\mathbf{u}\mathbf{u})} &= \text{div}(U\mathbf{U}) + \overline{\text{div}(\mathbf{u}'\mathbf{u}')} \\ -\frac{1}{\rho} \frac{\partial \overline{p}}{\partial x} &= -\frac{1}{\rho} \frac{\partial P}{\partial x} & \overline{\nu \text{div}(\text{grad}(\mathbf{u}))} &= \nu \text{div}(\text{grad}(U)) \end{aligned} \quad (2.8)$$

Therefore, momentum equations can be written as time averaged momentum equations. e.g. x-momentum equation becomes,

$$\frac{\partial U}{\partial t} + \text{div}(U\mathbf{U}) + \text{div}(\overline{u'\mathbf{u}'}) = -\frac{1}{\rho} \frac{\partial P}{\partial x} + \nu \text{div}(\text{grad}(U)) \quad (2.9)$$

Move the third term on the left hand side to the right hand side as it plays a role as turbulent stresses, then we have:

$$\frac{\partial U}{\partial t} + \text{div}(U\mathbf{U}) = -\frac{1}{\rho} \frac{\partial P}{\partial x} + \nu \text{div}(\text{grad}(U)) + \frac{1}{\rho} \left[\frac{\partial(-\rho \overline{u'^2})}{\partial x} + \frac{\partial(-\rho \overline{u'v'})}{\partial y} + \frac{\partial(-\rho \overline{u'w'})}{\partial z} \right] \quad (2.10)$$

$$\frac{\partial V}{\partial t} + \text{div}(V\mathbf{U}) = -\frac{1}{\rho} \frac{\partial P}{\partial y} + \nu \text{div}(\text{grad}(V)) + \frac{1}{\rho} \left[\frac{\partial(-\rho \overline{u'v'})}{\partial x} + \frac{\partial(-\rho \overline{v'^2})}{\partial y} + \frac{\partial(-\rho \overline{v'w'})}{\partial z} \right] \quad (2.11)$$

$$\frac{\partial W}{\partial t} + \text{div}(W\mathbf{U}) = -\frac{1}{\rho} \frac{\partial P}{\partial z} + \nu \text{div}(\text{grad}(W)) + \frac{1}{\rho} \left[\frac{\partial(-\rho \overline{u'w'})}{\partial x} + \frac{\partial(-\rho \overline{v'w'})}{\partial y} + \frac{\partial(-\rho \overline{w'^2})}{\partial z} \right] \quad (2.12)$$

These are called Reynolds Averaged Navier-Stoke (RANS) equations of incompressible flow.

The additional stress terms on the right hand side are named as Reynolds stresses, and they have three normal stresses

$$\tau_{xx} = -\rho \overline{u'^2} \quad \tau_{yy} = -\rho \overline{v'^2} \quad \tau_{zz} = -\rho \overline{w'^2} \quad (2.13)$$

and three shear stresses.

$$\tau_{xy} = \tau_{yx} = -\rho \overline{u'v'} \quad \tau_{xz} = \tau_{zx} = -\rho \overline{u'w'} \quad \tau_{yz} = \tau_{zy} = -\rho \overline{v'w'} \quad (2.14)$$

Turbulence fluctuations require extra terms in the time averaged or Reynolds averaged equations, and they have to be properly modelled with turbulence models. Therefore, RANS equations focus on the properties of the mean flow rather than computing fluctuating turbulent velocity directly. This can save numeral cost significantly.

2.1.3 Finite Volume Method (FVM)

Finite Volume Method (FVM) is one of the discretisation methods that complex equations can be solved with finite space using control volumes. Governing equations such as RANS equations include PDEs (Partial Differential Equations) and a non-linear term, therefore, it is very hard to solve them directly. Hence, it is important to apply appropriate discretization method to solve those equations for complex geometry. The discretisation process includes discretisation of the calculation domain and equations. The solution domain, which is generally physical space, is divided into a finite number of cells which are termed as control

volumes. Equation discretisation transforms partial differential equations into sets of linear algebraic equations and they are applied on the computational mesh. The equations should be also discretised for a finite number of time steps, if transient simulation is required.

The control volumes fill the computational domain without overlapping each other. In finite volume method, by principal, all the numerical properties are computed at the centroid of cells although they use the values on the vertices or faces during the computations. Cells are connected to each other by faces which are the boundary surfaces of each cell. The faces also form the boundaries of the computational space domain and called boundary faces. Others inside the domain are called internal faces. Figure 2-1 shows parameters in finite volume discretisation. Here f stands for the cell face, P the cell centre, N centre of the neighbouring cell, d is orthogonal direction through the cell face and S_f is the surface area of the cell face.

Figure 2-1: Finite volume discretisation [21]

The advantages of FVM are firstly, mass, momentum and energy are well conserved. Secondly, it can also handle unstructured mesh compared to traditional FDM (Finite Difference Method). The unstructured mesh is getting popular because it can handle various shapes more easily and efficiently. Thirdly, the solvers are efficient for iterative methods.

Disadvantage is the occurrence of false diffusion when numerical scheme is simple or mesh is not fine enough. For this reason, large computational resources should be applied to industrial cases, however, high computational cost of FVM is getting affordable due to the development of computer CPU and parallel computing nowadays.

2.1.4 Volume of fluid method

In many of marine CFD applications, prediction of free surface is very important. Traditionally, potential flow was popular for the free surface flow. Nowadays, viscous flow analysis with free surface is common as the computer performance is getting powerful.

There are several numerical modelling techniques to describe the interface of multiphase systems which deal with water and air together. As for free surface capturing techniques in finite volume method, VoF (Volume of Fluid) and level set method are most popular – even the two methods can be combined. VoF method that was introduced by Hirt and Nichols [8] is a free surface modelling technique applying volume fraction to convection schemes. It uses a scalar field to express the volume fraction (α) in whole domain, which represents water when it is 1 and air when it is 0, hence, the region where the volume fraction is between 0 and 1 becomes the free surface. It satisfies Eulerian material derivative.

$$\frac{D\alpha}{Dt} = 0 \quad (2.15)$$

, and this is

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot \mathbf{u} \alpha = 0 \quad (2.16)$$

VoF method is known to show high accuracy in the three dimensional applications. However, the mesh has to be very fine in order to capture the free surface properly as α is transitional from 0 to 1. Otherwise the interface will show smearing results. As a remedy, OpenFOAM uses modified governing equation for α by adding an artificial compression velocity which is a vector normal to the interface.

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot \mathbf{u} \alpha + \nabla \cdot \mathbf{w}(\alpha(1 - \alpha)) = 0 \quad (2.17)$$

As shown in the formula, the artificial term goes to zero if α goes to zero or one, which means that the artificial term does not affect to the result where the phase is completely water or air. The term only works within the interface zone to make it possible to describe sharp free surface by working to the normal direction to the interface. Magnitude of the compression, $\mathbf{w}$ (*cAlpha* in OpenFOAM) can be used to turn on or off the compression term.

2.2 CFD algorithm

In this chapter, brief algorithm and numerical techniques are described. All the CFD theory in this chapter may be limited to the followings:

- Finite Volume Method
- Incompressible Navier-Stokes equations with Mach number < 1
- General fluid flow simulations

2.2.1 Numerical discretisation and solution algorithm

The generic form of the transport equation for a fluid property ϕ is like the following:

$$\frac{\partial}{\partial t}(\rho\phi) + \text{div}(\rho u\phi) = \text{div}(\Gamma \text{grad } \phi) + S_\phi \quad (2.18)$$

Here, u is the velocity vector, ρ is the density, Γ is the diffusion coefficient and S_ϕ is a source term. For the simplicity, a one-dimensional case with a uniform grid is explained here. For the finite volume discretization, integrating this equation over the control volume (CV) and time,

$$\begin{aligned}
& \int_t^{t+\Delta t} \int_{CV} \frac{\partial}{\partial t} (\rho\phi) dV dt + \int_t^{t+\Delta t} \int_{CV} \text{div}(\rho u\phi) dV dt \\
& = \int_t^{t+\Delta t} \left[\int_{CV} \text{div}(\Gamma \text{grad } \phi) dV + \int_{CV} S_\phi dV \right] dt \quad (2.19)
\end{aligned}$$

Using the Gauss' theorem (2.20), the volume integrals of divergent terms can be expressed as surface integrals.

$$\int_V \nabla \phi dV = \oint_A \phi dS \quad (2.20)$$

Re-arranging time integration and volume integration, this can be written as follows:

$$\begin{aligned}
& \int_{CV} \int_t^{t+\Delta t} \frac{\partial}{\partial t} (\rho\phi) dt dV + \int_t^{t+\Delta t} \int_A n \cdot (\rho u\phi) dA dt \\
& = \int_t^{t+\Delta t} \left(\int_A n \cdot \left(\Gamma \frac{d\phi}{dx} \right) dA \right) dt + \int_t^{t+\Delta t} \int_{CV} S_\phi dV dt \quad (2.21)
\end{aligned}$$

Here, n is the unit vector normal to the face and A stands for the area of the cell face.

All the terms in the equation have to be approximated by various schemes in finite space and time to discretise the equations. The first term on the left hand side in the unsteady term becomes,

$$\int_{CV} \int_t^{t+\Delta t} \frac{\partial}{\partial t} (\rho\phi) dt dV = \int_{CV} \rho(\phi^n - \phi^0) dV = \rho(\phi_P^n - \phi_P^0) V_P \quad (2.22)$$

where 0 denotes current time step, n new value in the next time step, p cell point, V_P is the control volume and

$$\begin{aligned} \phi^n &= \phi(t + \Delta t) \\ \phi^0 &= \phi(t) \end{aligned} \quad (2.23)$$

The second term which is convection term and can be discretised as,

$$\int_t^{t+\Delta t} \int_A n \cdot (\rho u \phi) dA dt = \int_t^{t+\Delta t} (F\phi_E - F\phi_W) dt \quad (2.24)$$

F stands for the mass flux passes through the face and defined as $F = A(\rho u)$. E and W stand for the nodal point to the east and the west, respectively.

Depending on the time integrations whether defining $\int_{\Delta t} \phi_P dt \approx \phi_P^0$ or ϕ_P^n , this becomes,

$$[F\phi_E^n - F\phi_W^n]\Delta t \quad \text{or} \quad [F\phi_E^0 - F\phi_W^0]\Delta t \quad (2.25)$$

Similarly, the third term which is diffusion term becomes:

$$\int_t^{t+\Delta t} \left(\int_A n \cdot \left(\Gamma \frac{d\phi}{dx} \right) dA \right) dt = \int_t^{t+\Delta t} \left[\Gamma A \left(\frac{\phi_E - \phi_P}{\delta x} \right) - \Gamma A \left(\frac{\phi_P - \phi_W}{\delta x} \right) \right] dt \quad (2.26)$$

Then integrate this in time gives:

$$\begin{aligned} & \left[\Gamma A \left(\frac{\phi_E^0 - \phi_P^0}{\delta x} \right) - \Gamma A \left(\frac{\phi_P^0 - \phi_W^0}{\delta x} \right) \right] \Delta t \\ & \text{or} \\ & \left[\Gamma A \left(\frac{\phi_E^n - \phi_P^n}{\delta x} \right) - \Gamma A \left(\frac{\phi_P^n - \phi_W^n}{\delta x} \right) \right] \Delta t \end{aligned} \quad (2.27)$$

The fourth term is the source term and becomes:

$$\begin{aligned} \int_t^{t+\Delta t} \int_{CV} S_\phi dV dt &= \int_t^{t+\Delta t} \bar{S} V_P dt = \int_t^{t+\Delta t} (S_u V_P + S_P \phi_P V_P) dt \\ &= (S_u V_P + S_P \phi_P^0 V_P) \Delta t \quad \text{or} \quad (S_u V_P + S_P \phi_P^n V_P) \Delta t \end{aligned} \quad (2.28)$$

Finally, discretization in time can express the equations in two different ways depending on how we choose unknowns from the different time step.

The convection and diffusion terms are simplified by taking surface integrals over the cell and by using Gauss theorem.

These surface integrals can be expressed by the summation of the mass flux through the faces to describe general form.

$$\int_V \nabla \phi dV = \sum_f \left(\int_f \phi dS \right) \approx \sum_f A \phi \quad (2.29)$$

First method is explicit approach which can be written as:

$$\frac{\rho(\phi_P^n - \phi_P^0)V_P}{\Delta t} + \sum_f F \phi_f^0 = \sum_f \Gamma A \text{grad } \phi_f^0 + S_u V_P + S_P \phi_P^0 V_P \quad (2.30)$$

Another method is implicit approach that involves more complex calculations:

$$\frac{\rho(\phi_P^n - \phi_P^0)V_P}{\Delta t} + \sum_f F \phi_f^n = \sum_f \Gamma A \text{grad } \phi_f^n + S_u V_P + S_P \phi_P^n V_P \quad (2.31)$$

Here, A represents the area of the cell face and V the control volume. The surface integrals can be expressed as face summation where f stands for cell faces. For the surface integrals, the implicit approach involves neighbouring faces in the new time step. This makes calculations more complex. In the explicit form, there is only one unknown nodal point which is straightforward, while the implicit form has three unknown points.

Then, what is the reason for using the implicit method in spite of the complex algorithm? The implicit approach allows larger time step than the explicit method. More details will be discussed in Ch 2.2.4.

Figure 2-2 illustrates how grid points are deployed in two methods for time marching solution.

Figure 2-2: Explicit (upper) and implicit (upper) method [13]

Therefore, the momentum equations are discretised into linear algebraic equations that normally becomes,

$$a_P \phi_P = \sum a_{nb} \phi_{nb} + b \quad (2.32)$$

a is a function of velocity and $_{nb}$ represents neighbouring cells.

Then it can be arranged into the form of

$$[A] [\phi] = [B] \quad (2.33)$$

$[A]$ is a square matrix and $[B]$ is the source vector in column. Exact linear algebraic equations and forms in the matrix depend on the governing equation and the numerical scheme used for the discretisation. Applying appropriate matrix solving algorithm, the matrix equation can be solved to obtain the solution.

2.2.2 Pressure corrections

For incompressible flow, the continuity equation itself does not contribute to the pressure solution, as density is constant. Thus, the pressure gradient is unknown and has to be solved separately while solving the momentum equations.

Adding the pressure term in the momentum equation, the equation (2.32) can be written as,

$$a_p U_p = H(U) - \nabla p \quad (2.34)$$

where $H(U)$ term contains all the terms for the neighbours and all the source parts of transient terms. Writing this for the velocity U becomes:

$$U_p = \frac{H(U)}{a_p} - \frac{1}{a_p} \nabla p \quad (2.35)$$

The continuity equation is also discretised as,

$$\nabla \cdot U = \sum_f A \cdot U_f = 0 \quad (2.36)$$

where A denotes area of the cell face.

The velocity of the equation (2.35) can be expressed on the cell face:

$$U_f = \left(\frac{H(U)}{a_p} \right)_f - \left(\frac{1}{a_p} \right)_f (\nabla p)_f \quad (2.37)$$

Substitute the equation (2.37) into discretised continuity equation (2.36):

$$\nabla \cdot \left(\frac{1}{a_p} \nabla p \right) = \nabla \cdot \left(\frac{H(U)}{a_p} \right) = \sum_f A \cdot \left(\frac{H(U)}{a_p} \right)_f = 0 \quad (2.38)$$

The equation (2.38) has to be satisfied to guarantee the conservation of the face fluxes.

By expressing the gradient into the summation of mass flux at the cell face, the final form of Navier-Stokes equation becomes,

$$a_p U_p = H(U) - \sum_f A \cdot (p)_f \quad (2.39)$$

The face flux F is the velocity at the face (2.36) multiplied by the area of the cell face:

$$F = S \cdot U_f = S \cdot \left[\left(\frac{H(U)}{a_p} \right)_f - \left(\frac{1}{a_p} \right)_f (\nabla p)_f \right] \quad (2.40)$$

Normally pressure is solved by the pressure-velocity coupling algorithms such as SIMPLE (Patankar [29]) and PISO (Issa [12]).

They are iterative methods of guess and correction.

SIMPLE (Semi-Implicit Method for Pressure Linked Equations) process is in the following sequences:

- 1) Discretised momentum equations are solved with the initial guessed pressure which is denoted by p^* to find velocity components (u^* , v^* and w^*). This is a momentum predictor step.
- 2) Discretised equations including the $H(U)$ operator are used to solve pressure correction equation.

- 3) Then the pressure correction is defined from the difference between the guessed pressure and correct pressure field. The corrected pressure (p) is obtained by the guessed pressure (p^*) and pressure correction (p_{corr}).

$$p = p^* + p_{\text{corr}} \quad (2.41)$$

The same applies to the velocity corrections from the pressure correction:

$$u = u^* + u_{\text{corr}} \quad (2.42)$$

$$v = v^* + v_{\text{corr}}$$

$$w = w^* + w_{\text{corr}}$$

- 4) Solve other scalar equations (e.g. turbulence properties)
- 5) Repeat this process until the velocity field satisfies the continuity equation.

PISO (Pressure Implicit with Splitting of Operators) algorithm is also similar to SIMPLE but has one more corrector step. First, in PISO, the same steps with SIMPLE are performed and there is one more correction for pressure and velocities to be carried out. These steps are repeated until it reaches convergence. In spite of the additional step for pressure correction, PISO is known to be efficient and fast. However, this also depends on the case. It is not clear which method is the best for general flow case.

For this iterative method, under-relaxation factors help stability of the calculations. Especially for the pressure correction

methods, the equations are prone to diverge easily during the iterations. Generally, when a new value of variable (ϕ) is calculated, under-relaxation is applied as follows:

$$\phi_P^{new} = \phi_P^{old} + \gamma(\phi_P^{predicted} - \phi_P^{old}) \quad (2.43)$$

The under-relaxation γ is normally applied between 0 and 1 so that the iterative solution can be changed slowly and it increases stability of the calculation. Basically it does not affect the converged values but helps reduce the oscillation. However, this may slow down the convergence. Under-relaxation factors for all the variables are case dependent and there is no general recommendation.

If the factor equals to 1, it means no relaxation applied. Predicted value will be used without under-relaxation.

Sometimes the factor can be more than 1 to accelerate the convergence. In this case, the stability should be ensured during the calculations.

2.2.3 Convergence and residuals

The discretised governing equations are solved by iterative algorithm because the initial solution will not satisfy them. The left hand side and the right hand side of the matrix form of linear algebraic equations will not be equal with the initial values in

the first iteration that are assumed from the boundary conditions. The difference between the left and the right hand side is called momentum residual. During the iterations, the absolute values of residuals should be reduced so that the left and right hand sides of the equation can get closer. By this guessing and correction procedure, the variables get nearer to the actual solutions. Therefore, the residuals are important measures on how the solution satisfies the governing equations and can be used to estimate the error. In other words, the solution would not be correct and diverge if the absolute values of residuals are becomes bigger during the iterations.

There are many things that can affect the convergence of the solution. They are relaxation factors, quality of the mesh, numerical schemes, diagonal dominance of the matrix, etc. Wrong selection of mesh, numerical settings or boundary conditions would cause the solution to diverge. Nevertheless, this is always case dependent and there are no optimum settings. In some cases, convergence should be judged by the force values rather than the momentum residuals. The oscillation of force values will be stabilised or will not change any more when the solution converges. In this case, however, monitoring the residuals still can give a good indication of whether the solution diverges or not.

2.2.4 Courant number

The Courant number is a stability criterion of computation and defined as follows.

$$Co = \frac{\delta t |U|}{\delta x} \quad (2.44)$$

δt is the time step, U is the magnitude of velocity at the cell and δx is the size of the cell in the direction of the velocity.

In explicit approach, the Courant number must be less than one to ensure the numerical stability while running the computation. This is also called as CFL (Courant-Friedrichs-Lewy) condition. As can be seen in the formula, the time step of transient calculations has to be small enough if the velocity is high or the cell size in the direction of the velocity is very small in order to make sure CFL condition is met. CFL condition must be ensured in all the cells throughout the computation domain. Especially Courant number has to be far smaller at the interface region of multi-phase flow for the stability of the solution.

However, the Courant number can easily exceed one in the implicit approach. In some of the commercial codes, the Courant number is allowed even beyond hundreds. This means much larger time step can be used in the implicit approach. Although

implicit method allows larger time step, an appropriate Courant number depends on the type of the calculations and should be selected carefully for the accurate solution. Figure 2-3 is an example that shows the comparison between the implicit and explicit method. When the equal time step is used for the implicit and explicit algorithms, the solution of implicit approach gives more stable and accurate results.

Figure 2-3: Comparison of implicit and explicit solutions for same time step [7]

2.2.5 Boundary conditions

Boundary conditions are essential to find the unknowns in the governing equations. Wrong or poor boundary conditions will lead the computation to wrong solution or divergence. There are several types of boundary conditions and they are mainly divided into fixed value and gradient types. The fixed value type boundary conditions set the specific value of ϕ on the cell face, and this is called Dirichlet type. The fixed gradient boundary condition applies gradient of ϕ to the normal direction pointing outward on the boundary faces and is known as Von Neumann type.

For the general fluid flow simulation, boundary conditions are normally set for inlets, outlets, walls and/or symmetry planes.

Inlet: A fixed value is set for the velocity and zero gradient for the pressure at the inlet boundary.

Outlet: It is important for an outlet boundary to be set in such a way that it satisfies the overall mass conservation. Therefore, the outlet boundary condition is set in conjunction with the inlet boundary condition and velocity boundary condition.

For the mass conservation, the total mass flux which are going out (M_{out}) and coming in (M_{in}) must be the equal. In order to achieve this, the flux ratio (M_{in}/M_{out}) is multiplied to outlet velocity to ensure the continuity and subsequently multiplied to the East neighboring velocities (Versteeg and Malalasekera, 2007). Therefore, normally zero gradient is set for the velocity and fixed value for the pressure at the outlet boundary.

It is also important to locate the outlet boundary far enough from the region where the flow is expected to be developed so that the flow gradients become zero in the direction normal to the outlet boundary.

Wall: The wall boundary conditions are used on the solid objects and on the walls. On the wall boundaries, no-slip condition has to be met so that the velocity on the wall becomes the equal to that of the wall. For instance, velocity should be zero on the wall of the static object. When the object is moving, many CFD codes provide moving wall velocity boundary condition to be able to apply no-slip wall. On the wall boundaries, the mesh should be defined carefully to resolve the viscous flow characteristics in the boundary layer. More details for this will be discussed in the next sections.

Slip wall: When this boundary condition is set, the flow property normal to the wall becomes zero and tangential components zero gradient. This is normally applied to the wall where flow is just passing along.

Symmetry plane: The boundary conditions for symmetry planes give no flow gradient or flux across this boundary. This can be applied to save computational cost where the geometry and the flow field are exactly symmetric across the boundary.

2.2.6 Boundary layer treatment and y plus

y plus (y^+) is a non-dimensionalised distance by the friction velocity and kinematic viscosity and represents the distance from the solid wall to the first grid point from the wall.

$$y^+ = \frac{\Delta y_p}{\nu} \sqrt{\frac{\tau_w}{\rho}} = \frac{\Delta y_p \cdot u_\tau}{\nu} \quad (2.45)$$

, where Δy_p is distance from the wall to the first grid point, τ_w is wall shear stress, u_τ is friction velocity, ρ is the density of fluid and ν is the velocity component perpendicular to the wall.

It is known that the flow is laminar near the wall and is transitional from laminar to turbulence at about $y^+ = 11.225$. In

order to calculate viscous flow properties in the boundary layer correctly, mesh needs to be carefully designed to locate y^+ in the proper range.

On the surface of a wall, no slip condition is applicable which means the fluid on the wall has the same velocity with the wall itself. For instance, if a wall is stationary, the fluid on the wall is also stationary. At the very next to the wall, there is a very thin layer where viscous effect dominates and the flow is laminar. It is named viscous sub-layer or laminar sub-layer. It is also called as linear sub-layer because of the linear relation between y^+ and u^+ . u^+ is non-dimensional velocity with friction velocity, u_τ .

$$u^+ = \frac{U}{u_\tau} \quad (2.46)$$

$$u_\tau = \sqrt{\tau_w/\rho} \quad (2.47)$$

, where τ_w is wall shear stress

In this region, the shear stress is approximately equal to the wall shear stress and $y^+ = u^+$ is satisfied. When wall bounded effects are highly significant in a flow, viscous sub-layer has to be resolved, hence, y^+ must be in order of 1 to have at least one point in the viscous sub-layer. For a good result, y^+ should be less than 1 in all the area. In other words, mesh near the wall

has to be well refined to resolve the whole boundary layer, and turbulence model also needs to be of low Reynolds number model such as SST (Shear Stress Transport) type that can resolve viscous sub-layer properly. Epsilon based turbulence models are not designed for the viscous sub-layer. Generally, cases with low Reynolds number should have mesh with the first grid point from the wall within this sub-layer.

Outside this sub-layer, there is a turbulence region where u^+ and y^+ vary in a logarithmic relation as such,

$$u^+ = \frac{1}{\kappa} \ln(Ey^+) \quad (2.48)$$

where κ is von Karman's constant and E is an integration constant from the roughness of the wall.

This is the region where $30 < y^+ < 500$ and called log-law layer. In this region, the shear stress is no more constant and varies with the distance from the wall. When wall bounded effects are not priority and Reynolds number is high, the first grid point can be put in this log-law layer region to apply wall functions. Wall function is a semi-empirical method that can predict the velocity profile near-wall region. In other words, y^+ has to be in the range of 30 to 500 in order to use wall function. However, it is

important to make sure that the first point is not placed outside of the boundary layer when using wall function. By using wall functions, mesh resolution can be reduced, hence, saving computational costs and efforts.

If y^+ is between 5 and 30, it is a region where laminar flow transits to turbulent flow and calculated shear stress is neither constant nor logarithmic. This region is called buffer layer and has to be avoided when generating mesh with certain y^+ value. Therefore, the user should determine whether viscous sub-layer would be resolved or a wall function would be used to calculate near-wall region for the grid design near the wall. Figure 2-4 shows velocity distribution near a solid wall with respect to $\log y^+$.

Figure 2-4: Velocity distribution near a solid wall [34]

2.2.7 Good practice of boundary layer treatment

As discussed, for the grid design, special care should be taken at the boundary layer region in which flow properties vary rapidly. Wrong design of boundary mesh layer will give wrong calculation results. When one designs the grid to target certain range of y^+ value, valid y^+ value can be known only after calculation is carried out with enough iterations. It is because y^+ value is different between the initial field and after flow is developed sufficiently. Therefore, it is recommended to monitor y^+ values during the computations.

Normally marine CFD applications are high Reynolds number flow and wall functions can be used to calculate the near wall region. In this case, y^+ value is recommended to be below 300 and above 30 when flow is developed.

If the user wants to resolve whole boundary layer, enough number of mesh layers (i.e. at least 20 layers) needs to be designed in this laminar region without sudden increase in the size of mesh between the layers. Several marine CFD codes do not use wall function but resolve whole boundary layer from the wall where the velocity is set to zero although the marine applications has high Reynolds number. In order to achieve this,

it is recommended that the mesh near the wall needs to be very fine with $y^+ \leq 1$ and turbulence model must be valid in all regions of the boundary layer. This is known to be very accurate in calculating ship resistance (Larsson et al. [18])

This practice is also important for full scale analysis. When it comes to the experiment, it is well known that there is a big discrepancy of Reynolds number between model scale and full scale when scaling down with geometrical similarity. This causes relatively thicker boundary layer in the model scale compared to full scale due to the different Reynolds number and brings forth different results. For this reason, there were many efforts to carry out full scale CFD analysis. Traditionally CFD technology was limited to the model scale not only because of limitation of computational resources but also because the first cell from the hull surface gets to have very large aspect ratio in full scale. In order to have y^+ within 1, too large aspect ratio of grid causes solvers to be broken down easily. Nowadays, many commercial solvers are able to handle high aspect ratio. Another problem comes from full scale analysis is when turbulence model is used with wall function. General wall functions are mainly developed and validated with the relatively small scale but not robust enough for full scale ships with slender hullform where Reynolds number easily goes above 10^9 . However, improvement of CFD

techniques as well as increasing hardware capability enables full scale analysis nowadays (Larsson et al. [18]).

2.3 Mathematical model of ship-handling simulator

This chapter will cover mathematical model of ship-handling simulator and discuss how to make use of CFD results into ship-handling simulator.

According to the Euler's equations of rigid body dynamics, the motion of a rigid body can be written as follows,

$$\vec{F} = m\dot{U} + m\Omega \times U \quad (2.49)$$

$$\vec{M} = I\dot{\Omega} + \Omega \times I\Omega \quad (2.50)$$

Here, ship's centre of gravity is assumed to be at the midship.

Figure 2-5 shows coordinate system of mathematical model. Considering three dimensions, force vector and moment vector are,

$$\vec{F} = (X, Y, Z), \quad \vec{M} = (K, M, N)$$

Here, m is mass of the ship, I is moment of inertia of the body, linear speed vector is $U(u, v, w)$ and angular velocity vector is $\Omega(p, q, r)$.

Figure 2-5: Coordinate system of manoeuvring mathematical model

Substituting all these vectors into the equation (2.50),

$$I_{xx}\dot{p} - (I_{zz} - I_{yy})qr + I_{xz}(\dot{r} - qp) + I_{yz}(r^2 - q^2) + I_{xy}(pr + \dot{q}) = K \quad (2.51)$$

$$I_{yy}\dot{q} - (I_{xx} - I_{zz})rp + I_{xy}(\dot{p} - rq) - I_{xz}(p^2 - r^2) + I_{yz}(qp + \dot{r}) = M \quad (2.52)$$

$$I_{zz}\dot{r} - (I_{yy} - I_{xx})pq - I_{yz}(\dot{q} - pr) + I_{xy}(q^2 - p^2) - I_{xz}(rq + \dot{p}) = N \quad (2.53)$$

Assuming to be $(I_{xy} = I_{yz} = I_{xz} = 0)$ from the tensor matrix,

$$I = \begin{pmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{pmatrix}$$

Hence, the equations (2.49) and (2.50) can be expanded as follows:

$$m(\dot{u} - vr + wq) = X \quad (2.54)$$

$$m(\dot{v} - wp + ur) = Y \quad (2.55)$$

$$m(\dot{w} - uq - vp) = Z \quad (2.56)$$

$$I_{xx}\dot{p} + (I_{zz} - I_{yy})qr = K \quad (2.57)$$

$$I_{yy}\dot{q} + (I_{xx} - I_{zz})rp = M \quad (2.58)$$

$$I_{zz}\dot{r} + (I_{yy} - I_{xx})pq = N \quad (2.59)$$

These are the motion equations for 6 degrees of freedom (DOF) of motion in the general ship-handling simulator when the centre of mass is at the midship. Figure 2-6 illustrates 6 DOF of motions.

Figure 2-6: Six degrees of freedom of motion

The right hand side represents forces (X, Y, Z) and moments (K, M, N) in 6 DOF of motions which are surge, sway, heave, roll, pitch and yaw, respectively. Each of them represents the sum of

all the external forces and moments from hull, propeller, rudder, aerodynamic, wave, mooring lines, etc.

Likewise, the forces and moments from ship-to-bank interaction can be added into the additional terms in the right hand side, too. For this reason, sway force and yaw moment from the bank effect can be non-dimensionalised and modelled for different water depth and separation from the bank wall.

$$C_{BY} = \frac{F_Y}{mgFn^2} \quad : \quad \begin{array}{l} \text{non-dimensionalised sway force from the} \\ \text{bank effect} \end{array} \quad (2.60)$$

$$C_{BN} = \frac{M_Z}{mgLFn^2} \quad : \quad \begin{array}{l} \text{non-dimensionalised yaw moment from the} \\ \text{bank effect} \end{array} \quad (2.61)$$

Even if CFD calculations are carried out for steady state without mesh motion, the results can be easily implemented in the ship-handling simulator for time variant simulations. When the ship moves near the bank during the real time simulation, modelled sway force and yaw moment of the ship at the specific separation from the bank and water depth can be added in the right hand side of motion equations of simulator (Equation 2.55 and 2.59) as an additional force and a moment components. As the ship is repelled from or attracted to the bank due to the added force and moment, force and moment can be updated by interpolating at a new distance that varies at each time step. By this, simulations

for the irregular bank and the bathymetry will also be possible, hence, training of crew in confined water can be more realistic.

Figure 2-7 shows a simulation result of a vertical bank case in ship-handling simulator after modelling the forces and moments at various distances but at a fixed water depth. Kongsberg system [15] has been used for the testing of ship-handling simulator.

Figure 2-7: The trajectory in ship-handling simulator with ship-to-bank interaction

x and y axis: distance(m) from the initial point

Chapter 3

Open source CFD Code

3.1 Open Source CFD Codes and OpenFOAM

CFD technology has become essential in many fields including marine industry. There are popular commercial CFD codes with huge user groups due to the convenient and strong functions. Meanwhile, open source CFD codes are also improving in functions and it is also becoming popular in many applications these days. They are free to use and even free for modification under the GNU General Public License. The users need not worry about the license and can run it anywhere. They also have good pre- and post- processors that are also free to use, and parallel computing interface is useful to speed up the computations. Although they are still not as convenient as commercial codes and it takes some time to leverage, it is decided to use an open source CFD code to build its capability in this study.

Among many open source CFD codes, OpenFOAM has been chosen for this study due to the massive user group and numerous tutorials.

OpenFOAM (Open Field Operation And Manipulation) is a C++ library toolbox based on finite volume method. It is first released by Prof. Hrvoje Jasak and Mr. Henry G. Weller in 2004 after several studies and projects. Various solvers with different governing equations and numerical schemes can be used to solve different problems. The user can even create a new solver because the libraries in OpenFOAM toolbox are designed to solve ODE (ordinary differential equations) or PDE (partial differential equations). The latest version 2.3.0 was released in Feb 2014. There is another group called OpenFOAM extend-project who develops modified OpenFOAM solvers by themselves and discusses related issues. There are many seminars and conferences in many countries to discuss about the applications. The capability and applications are known to be comparable to commercial codes. Initially OpenFOAM 2.1.0 was used in this study, and then changed to the version 2.3.0 in the middle of the study when it was released.

3.2 OpenFOAM software structures

3.2.1 Architecture

Basically, an OpenFOAM case is composed of various files in several folders. Figure 3-1 shows the structure of a general fluid flow case of OpenFOAM by the folders and the major input files.

Figure 3-1: OpenFOAM case structure

The 'system' folder has numerical schemes and their options. Users can set discretisation schemes, pressure solvers, matrix

solver, time step, the numerical settings, etc. to be used for governing equations.

The 'constant' folder has mesh and physical properties such as density, kinematic viscosity, acceleration of gravity, turbulence model information. The mesh information are saved individually as points, faces, cell, boundary, etc. in the sub-folder, 'PolyMesh'.

The time directories are the folders where all the variables are stored. Initial time folder, normally '0', has initial and boundary conditions in it. As computation is run, folders for other time steps with output files are saved by the solver according to the time step set by the user.

3.2.2 Solvers

There are several solver groups in the package such as incompressible flow, compressible flow, multiphase flow, combustion, heat transfer and buoyancy-driven flow, electromagnetics, stress analysis of solids, etc. Under each group, there are several solvers too. This study will focus on the multiphase solvers to include free surface effect. Single-phase solver has also been used to compare with multi-phase solutions.

3.2.3 Numerical algorithm

OpenFOAM is a toolbox that can solve partial differential equations. For example, the follow momentum equation,

$$\frac{\partial \rho U}{\partial t} + \nabla \cdot \phi U - \nabla \cdot \mu \nabla U = -\nabla p$$

is expressed as:

```
solve
(
    fvm::ddt(rho, U)
    + fvm::div(phi, U)
    - fvm::laplacian(mu, U)
    ==
    - fvc::grad(p)
);
```

Here ‘fvm’ and ‘fvc’ are the classes to discretise differential operators such as ddt ($\partial / \partial t$), div ($\nabla \cdot$) and laplacian (∇^2). The class ‘fvm’ involves implicit derivatives to generate matrix. ‘fvc’ stands for Finite Volume Calculus and calculates explicit operations to return geometric field data.

OpenFOAM gives options for each numerical scheme. For the gradient scheme, 2nd order schemes (Gauss linear and Least square) are used in this study to maintain the accuracy of the solution. Gradient scheme is the numerical scheme that interpolates the value to the cell faces since the variables are basically stored at the cell centre. Gauss linear gradient scheme means Gaussian integration to interpolate the value from the cell centre to the face centre and the interpolation is done linearly.

A limiter can be used for gradient calculations especially when high order scheme is applied. There are cell type and face type limiters. Basically cell type limiter is applied to the gradient between the cell centres by clipping the gradient equally, whereas face limiter applies limiting gradient on the face of the cells. MD stands for multi-dimensional which clips the flux in the direction normal to the cell faces. In the simulation of this study, *cellMDLimited* was applied to maintain certain accuracy of numerical scheme. Available limiters are followings.

- *cellMDLimited* : Least diffusive and more accuracy
- *cellLimited* : Less dissipative
- *faceMDLimited* : More dissipative
- *faceLimited* : Most diffusive but robust scheme

Another pressure-velocity solving algorithm, PIMPLE is provided by OpenFOAM. It is a hybrid method of PISO (Pressure Implicit with Splitting of Operators) and SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm. It is, in fact, not different from the PISO procedure except for the outer correctors. If the outer corrector parameter is set to be 1 or not set, it is simply same as the PISO algorithm. Outer iteration option is known to give more stable and accurate calculations.

3.3 Multi-phase solver interFOAM and LTSInterFOAM

In many marine applications of CFD, free surface cannot be neglected, hence, multi-phase solvers are very useful for interface capturing. For this reason, interFoam and LTSInterFoam were chosen for this study. interFoam is a solver for transient solutions of two incompressible, isothermal and immiscible fluids using a VoF (volume of fluid) phase-fraction. LTSInterFoam adopts local time stepping (LTS) that uses different time steps for each cell depending on the different velocity and cell size, and enables to reach steady state quickly. Since this study will focus on capturing forces and moments at steady state, LTSInterFoam is useful to obtain them in relatively shorter time. The governing equations used in the solver are the continuity equation (3.1) and momentum equation (3.2) for incompressible flow:

$$\nabla \cdot \mathbf{U} = 0 \quad (3.1)$$

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = -\nabla p^* - \mathbf{g} \cdot h \nabla \rho + F_S + \nabla \cdot (\mu \nabla \mathbf{U}) + (\nabla \mathbf{U}) \cdot \nabla \mu \quad (3.2)$$

where μ is dynamic viscosity, p^* is the dynamic pressure, $\mathbf{g}$ is acceleration of gravity, ρ is density, $\mathbf{U}$ is velocity vector, h is hydrostatic height of the fluid, hence, first term in the right hand

side represents dynamic pressure and second term hydrostatic pressure. F_S is the source momentum of surface tension, which is expressed as;

$$F_S = \sigma \kappa \nabla \alpha \quad (3.3)$$

, where σ represents the surface tension coefficient, κ is the surface curvature and α is volume fraction which is bounded in $0 \leq \alpha \leq 1$ (0 for air and 1 for water). Surface curvature can be modelled as the boundary condition of wall surfaces setting contact angle. Surface tension can be neglected in the simulation in which huge mass of fluid develops the flow fast in large domain and the effect from surface tension is not important. Therefore, in this simulation, surface tension between the interface and walls are neglected.

To make sure scalar fields are bounded in multi-phase flow, OpenFOAM uses MULES (Multi-dimensional Universal Limiter for Explicit Solution) algorithm. This handles free surface developments by limiting the face fluxes multi-directionally for the boundedness of hyperbolic equations. Although Euler implicit unsteady method is used in OpenFOAM, this explicit limiter requires Courant number below 1. In OpenFOAM 2.3.0, semi-implicit version of MULES has been introduced. This

executes an implicit predictor first and followed by explicit correction step. This allows quite bigger time step with larger Courant number and reduces computational time from older versions of OpenFOAM.

The main source codes of interFoam are included in the Appendix. They include most of the CFD algorithm and techniques discussed in this study.

3.4 OpenFOAM evaluations in open water

Starting from the tutorial on open water, the mesh, numerical settings, boundary conditions, etc. are tested to determine the most suitable settings.

Prior to studying the bank effect using OpenFOAM, it would be beneficial to evaluate the solver with open water analysis to know its accuracy and find the robust computational settings. Compared to flow in confined water, open water analysis is relatively simpler, therefore, it is a good start before moving on to the confined water analysis that involves with shallow water effect as well as interaction with banks. KVLCC2 (KRISO Very Large Crude-oil Carrier 2) and KCS (KRISO Container Ship) that are popular for the benchmarking are chosen to compare with CFD analysis. The test results of these two ships were released by MOERI (Maritime & Ocean Engineering Research Institute), formerly known as KRISO (Korea Research Institute Ship and Ocean engineering) for validation purpose.

Considering the numerical cost and the accuracy, the minimum size of the mesh near the hull was determined. The mesh was created by *snappyHexMesh* combining several blocks to make graded mesh efficiently. As the ship and the domain are

symmetric, half hull was modelled and calculated. To save the computational cost, the mesh is created to be finer near the hull and getting coarse towards the outside of the domain. Turbulence model used for simulation was $k-\omega$ SST with wall function and the average y^+ was kept below 200 on the most of the surface area.

The boundary conditions are set as fixed value at the velocity inlet and pressure outlet, and zero gradient at the velocity outlet and pressure inlet. No slip condition was applied on the ship hull and symmetry plane on the side, bottom and middle symmetry plane.

In OpenFOAM 2.1.0, it turns out that the free surface level tends to be changed as the computation develops. This means water fraction is not well conserved. The problem started to occur after some iterations when the solution is getting stabilised. This probably happens because different inlet and outlet velocity could cause continuity correction as discussed in the chapter 2.2.5. This could affect total water fraction and make the water level changes. This is nearly certain because the problem does not appear if the outlet velocity is set to be equal to the inlet velocity. Therefore, when the inlet velocity is high, the phenomenon is more obvious. This problem has been reported

many times in the OpenFOAM forum (www.cfd-online.com), however, the problem disappeared in the newer version of OpenFOAM. In version 2.3.0, there was an improvement in the outlet boundary condition. Improving mesh quality, that is, reducing aspect ratio of the mesh at the inlet and outlet was also able to mitigate this problem. The problem was not significant in KVLCC2 case presumably due to the relatively slow speed, whereas it was obvious in KCS.

Forces and moments on the hull surface can be calculated by following formulas:

$$F_{pressure} = \int p \cdot \vec{n} dS \quad (3.4)$$

$$F_{viscous} = \int \tau \cdot \vec{n} dS \quad (3.5)$$

$$M_{pressure} = \int (x_f - x_o) \times p \cdot \vec{n} dS \quad (3.6)$$

$$M_{viscous} = \int (x_f - x_o) \times \tau \cdot \vec{n} dS \quad (3.7)$$

, where p is pressure, τ is wall shear stress, x_o is origin of measurement and x_f is the distance from origin to the face centre of the force component.

3.4.1 KVLCC2

Table 3-1 shows the principal dimension of KVLCC2 model ship that is geometrically scaled down by ratio of 58.0.

Table 3-1: Principal dimensions of KVLCC2 model

Length between perpendiculars (m)	5.5172
Breadth (m)	1.0000
Draught (m)	0.3586
Block coefficient	0.8098
Speed (m/s)	1.047
Froude number	0.142

Figure 3-2: Profile and front view of KVLCC2

Figure 3-2 is the profile and the front view of KVLCC2 hullform. OpenFOAM 2.1.0 and LTSInterFoam solver was used for this computation. Since the resistance at the steady velocity where flow is fully developed is important, transient solver has no advantage to predict the resistance. Flow analysis around KVLCC2 was carried out with grid convergence study (Table 3-2).

Table 3-2: Grid convergence study
(The experiment result: Kim et al. [38])

No. of mesh (K)	Pressure Force (N)	Viscous Force (N)	Total Force (N)	Limiter
690	11.16	10.45	21.61	Not used
1100	9.3	13.39	22.69	Not used
1470	7.17	12.08	19.25	100% -> 50%
1742	6.87	11.94	18.82	100% -> 50%
2434	6.78	11.74	18.52	100% -> 50%
3556	6.95	11.50	18.45	100% -> 50%
EXP	-	-	18.12	-

The number of mesh was tested up to 3.5 million and longitudinal force that is resistance was monitored. The force value did not change much after the mesh is getting finer to some extent, i.e. 1.7 million mesh. The mesh with 3.5 million cells showed 1.8% deviation from the experiment result. When the mesh was very fine, a limiter on the gradient scheme was necessary for the convergence. Otherwise the computation diverged before the iteration number 2000. In order to obtain better accuracy, 100% limiter was used first, and reduced to 50% when the force value is stabilised. Figure 3-3 shows the convergence history of resistance when the number of mesh was

3.5 million. With the iteration of 15000 (750 in the horizontal axis of Figure 3-3), the limiter has been reduced to 50%.

Figure 3-3: Convergence history of the resistance of KVLCC2

Wave elevation along the hull is compared with experiment result (Kim et al. [37]) in Figure 3-4 and it shows very good agreement. The horizontal axis stands for the distance from the forward perpendicular to aft direction, which means zero is at the forward perpendicular of the ship. Due to the relatively slow speed of tanker ship, the amount of changing water level was not significant.

Figure 3-4: Comparison of wave elevation along the hull of KVLCC2 (Experiment result: Kim et al. [38])

Wave pattern was compared to the experiment measurement in Figure 3-5. When limiter was used for very fine mesh, it resulted in oscillating behaviour in free surface pattern. For this reason, contour line of wave elevation became messy and it was hard to compare with experiment result, thus, the contour line was not presented in this study. This oscillation can be observed near the stern area of wave pattern in Figure 3-5. However, predicted wave pattern is in good tendency compared to the experiment. More comparisons were made in KCS container ship with the newer version of OpenFOAM.

Figure 3-5: Comparison of wave pattern of KVLCC2 of CFD (upper) and the experiment (Kim et al. [38]) (bottom)

3.4.2 KCS

Table 3-3 is the principal dimension of KCS model which is scaled down by the ratio of 31.599 from the full scale.

Table 3-3: Principal dimensions of KCS model

Length between perpendiculars (m)	7.2786
Breadth (m)	1.0190
Draught (m)	0.3418
Block coefficient	0.651
Speed (m/s)	2.196
Froude number	0.260

Figure 3-6 is the profile and the front view of KCS hull.

Figure 3-6: Profile and front view of KCS

The number of mesh used for the simulation was about 1.7 million. Figure 3-7 shows generated mesh around and on KCS hull. Trimmed mesh was generated for the mesh near the ship to be fine and for the outer region to be coarse. The grid convergence study was not carried out again for KCS as it has been done for KVLCC2, but reasonable number of mesh size was used based on the results from KVLCC2. One problem from the very fine mesh is y^+ value. When the mesh at the hull surface is too small, it was difficult to keep y^+ value all over the surface above 30. This would cause problem with wall function as discussed in Chapter 2.2.6 and the solution would not be correct.

Figure 3-7: Mesh of KCS - (a) whole domain, (b) forward and (c) stern area

Figure 3-8 shows y^+ distributions on the hull surface when the flow is fully developed. Due to the bulbous bow area where y^+ appeared low, it was hard to use finer mesh than that was used at the hull surface. Additional mesh refinement was applied near the stern boss (Mesh refinement in (c) of Figure 3-7) where y^+ appeared higher than other under water area. By this, it was possible to model the mesh with reasonable y^+ value. Moreover, finer mesh was also used around the free surface region for better free surface capturing because relatively high velocity would create more wave elevation.

Figure 3-8: y^+ distributions on the hull

For KCS analysis in open water, the problem of rising free surface level was quite significant due to the relatively higher velocity. After testing different settings, it was decided to use OpenFOAM version 2.3.0 that was released during that time. In the newer version of OpenFOAM, the problem did not appear anymore. In this version, there was significant improvement of

the outlet boundary condition to maintain the conservation of volume fraction. Another problem was over-prediction of force in LTSInterFoam solver although free surface elevation and the pattern were well predicted with this solver. It is presumed that the different time step throughout the domain leads to wrong solution. Several different grid types were tested but the problem was not solved. When the transient solver, interFoam was used, the obtained force value showed about 3.5% deviation from the experiment results. For the comparison purpose, the result of single-phase steady-state solver, simpleFoam has been added. From the result of the steady-state solver, it is assumed that free surface affects about 7.5% of the resistance in this KCS case. The comparison between the solvers and the experiment is presented in Table 3-4. The total resistance has been non-dimensionalised as below:

$$C_T = \frac{R_T}{\frac{1}{2}\rho S U^2}$$

R_T : Total resistance (N)

ρ : density of Water (kg/m³)

S : Wetted surface area (m²)

U : Velocity (m/s)

Table 3-4: The comparison of KCS resistance by different solvers (Experiment results: Kim et al. [38])

Solver	Resistance coefficient	Deviation	Gradient scheme
simpleFoam	3.415e-3	-3.99%	gauss linear
interFoam	3.681e-3	3.49%	cellLimited gauss linear 1.0
LTSinterFoam	5.108e-3	43.60%	gauss linear
EXP	3.557e-3	-	-

Although high maximum Courant number (Max. 25 and 10 for the interface) used for the simulation due to the semi-implicit algorithm adopted in OpenFOAM 2.3.0, the convergence of the force value was quite stable. Figure 3-9 is the convergence history of the resistance value of interFoam solver for the different gradient schemes.

A limiter was needed for the stability of the calculation. Because of the relatively big oscillation of Gauss linear gradient scheme, the limiter was removed after the computation became stable at 50 second. This suppressed the oscillation of the gradient scheme and also improved the accuracy. Least squares method showed less oscillating behaviour but slightly less accuracy of the resistance value. Gauss linear scheme resulted in 31.492 N for

the viscous drag force while Least squares scheme 32.157 N. Gauss Linear showed slightly better result by 2% when the limiter was removed. In Figure 3-9, it is found that the pressure force gets lower after 50 seconds when the limiter is removed and this reduces over-prediction. The solution using Least squares method without the limiter was not converged. Hereafter, the result with Gauss linear scheme is used for other validations in this chapter.

In Figure 3-10, wave elevation of OpenFOAM calculation was compared with experiment (Kim et al. [38]). As for the horizontal axis that shows x/L_{pp} , 0 stands for forward perpendicular, 0.5 for midship and 1 for aft perpendicular. Except for the crest at the bow wave, it shows very good agreement with the experiment.

In Figure 3-11, wave pattern also shows good agreement with the experiment result (Kim et al. [38]). 0 stands for the forward perpendicular and 1 for the aft perpendicular. Some strange shapes at the side of the divergent wave behind the stern are appeared due to the mesh refinement.

Figure 3-9: Convergence time history of the resistance (N) of KCS by (a) Gauss linear and (b) Least square gradient scheme

Figure 3-10: Comparison of wave elevation along the KCS hull (Experiment results: Kim et al. [38])

Figure 3-11: Comparison of wave pattern of KCS of OpenFOAM (upper) and the experiment (Kim et al., [38]) (bottom)

Figure 3-12: Wave pattern at the forward (a) and the stern (b)

In Figure 3-12, wave pattern at the forward and the stern are presented. It illustrates the maximum (red) and the minimum (blue) of the wave elevations. Kelvin wave which consists of diverging wave and transverse wave is well described in the wave pattern.

Stern flow, which is wake distribution, becomes important information for propeller design. Due to the aft hull shape, wake velocities entering into the propeller plane become inhomogeneous on the propeller plane and they cause pressure fluctuation on the propeller blades while they are rotating in high RPM. This fluctuating pressure causes cavitation on the propeller

blades and on the rudder, and leads to many problems. Cavitation also affects the propeller efficiency, vibration and erosion on the propeller blades and rudder. Therefore, it is important to design the aft hull in such a way that the wake velocity becomes evenly distributed at the same blade radius to minimise pressure fluctuation especially for the container ship which has bigger propeller loading with powerful main engine.

Wake velocities that are calculated by OpenFOAM are visualised at the couple of sections in Figure 3-13. In the experiment, a dummy hub is attached on the stern boss while it is absent in the CFD calculation. Therefore, difference is unavoidable near the stern boss on the propeller plane. Overall tendencies are well predicted. One reason for the difference is the mesh fineness. Behind the ship stern, the flow separation, drag and turbulence make the flow very complex, hence, very fine mesh is needed to predict the flow accurately. In author's opinion and from the experience with commercial marine CFD codes SHIPFLOW and WAVIS, these are reasonable results due to the complexity of flow.

Figure 3-13: Wake distribution (axial velocity [m/s]) of KCS at (a) the propeller plane and (b) at the station no. 1 (left : experiment (Kim et al. [38]), right : OpenFOAM result)

3.4.3 Summary of the evaluations

OpenFOAM was able to solve multi-phase problem in open water and calculate resistance force value with reasonable size of mesh. The converged force value showed good accuracy with the wall function. For the gradient scheme, Gauss linear scheme without a limiter gave slightly better accuracy than Least squares

method, however, the latter had advantage of less oscillation in the converged force value. The free surface pattern and the elevation were well captured too. The wake distribution also showed reasonable accuracy in spite of medium fineness of mesh. Several problems in the OpenFOAM 2.1.0 such as conservation of volume fraction, oscillations on the free surface level, etc. were all fixed in the new version.

The local time stepping solver computed a wrong force value for KCS whereas the problem did not occur with KVLCC. This will be further tested in the next chapter on confined water analyses.

Chapter 4

Analysis in Confined Water

4.1 Computational set-up and mesh

Numerical setup for confined water analyses was established according to the experimental setup done by FHR (Lataire et al. [5]). The test cases with several bank arrangements and different under keel clearance were chosen for numerical analysis. As found in the experimental research, a 8000TEU container ship in model scale was used for the simulations with a scale ratio of 80. Table 4-1 is the principle dimension of the model ship.

Table 4-1: Principal dimension of model ship

Length overall (m)	4.332
Breadth (m)	0.530
Draught (m)	0.149 ~ 0.180
Block Coefficient	0.65 ~ 0.66 (depending on the draught)

Figure 4-1 shows the coordinate system of the cases.

Figure 4-1: The coordinate system

Figure 4-2 shows the bank arrangements of the test case.

Figure 4-2: Schematic view of the bank arrangement (looking from the aft)

As can be seen in Figure 4-2, three types of the bank were used. The arrangement (a) has a straight vertical bank wall. The bank type (b) has an inclined bank wall and type (c) has a bank wall that is inclined and submerged. The coordinate system used for

the simulation is defined as body fixed and right handed system and that is different from the experiment. The origin is located at the amidships, centreline and keel for x, y and z, respectively. Longitudinal axis (x) is towards the bow, transverse axis (y) towards port side and vertical axis (z) upwards. Therefore, positive sway force directs to the outside the bank and yaw moment is positive when the bow is away and the stern towards the bank. The negative heading angle of the case A in Table 4-2 means that bow of the ship is toward the bank and the stern is away from the bank.

Four cases were chosen among the experiment results for CFD simulations. Table 4-2 includes the details of the case arrangements. Under keel clearance is 10% or 100% of the draught and various lateral position of the ship from the bank are included in the test cases. The test cases without propeller thrust have been chosen firstly to see the bank effect merely from asymmetric flow field. If propeller is rotating, it may affect the force and the moment values. In these test cases, propeller was basically prevented from the rotations, therefore, measured RPM is -1 and propeller thrusts are all slight negative because the propeller worked like a 'parachute'. The case A that has an heading angle toward the inflow and the measured negative propeller torque was relatively big.

Table 4-2: The test cases without propeller thrust

Test title		A	C	F	H
Under keel clearance	[%]	100	10	100	10
Draught	[m]	0.180	0.180	0.180	0.149
Water depth	[m]	0.360	0.198	0.360	0.163
Forward component of speed vector	[m/s]	0.6842	0.4578	0.8012	0.4578
Transverse component of speed vector	[m/s]	-0.0599	0.0	0.0	0.0
Lateral separation	[m]	0.265	0.265	1.435	1.965
Heading angle	[°]	-5.0	0.0	0.0	0.0
Propeller rate	[rpm]	-1	-1	-1	-1
Propeller thrust	[N]	-1.10	-0.20	-0.93	-0.37
Propeller torque	[Nmm]	-27.09	-3.05	0.84	7.28
Type of bank	-	(b)	(c)	(b)	(a)
Bank inclination		8:1	5:1	3:1	1

The case A has slight heading angle with small separation from the bank wall. The case C is the most severe case where both under keel clearance and lateral separation from the bank wall are very small. The minimum distance from the hull to the sloped bank wall is very small. The case F does not have extreme clearance from the bank wall or the bank bottom, however, the velocity of the ship is relatively high. The case H has very small under keel clearance but moderate distance to the bank wall.

Figure 4-3 is the graphical view of the test cases as modelled in OpenFOAM. It shows the bank shape, the location of the ship and water depth.

Figure 4-3: Graphical view of the case C and the case H

There is a need to find out how much propeller contributes to the bank effect. Although propeller rotation is not modelled in this study, another set of simulations has been done for the test cases with propeller thrust but without modelling the propeller. The purpose of this set of simulations is to compare the CFD results without propellers to the experiment results that have propeller thrust. This will indicate the significance of propeller contribution to the bank effect. They will show the difference as

much as propeller rotation contributes to the bank. Table 4-3 is the detailed experiment conditions for this set of test cases. The only difference in the CFD simulations is that the propeller is not modelled.

Table 4-3: The test cases with propeller thrust

Test title		D	I	J
Under keel clearance	[%]	100	35	100
Draught	[m]	0.149	0.149	0.149
Water depth	[m]	0.297	0.200	0.297
Forward component of speed vector	[m/s]	0.6868	0.5723	0.8012
Transverse component of speed vector	[m/s]	0.0	0.0	0.0
Lateral separation	[m]	1.340	0.565	0.530
Heading angle	[°]	0.0	0.0	0.0
Propeller rate	[rpm]	540	359	539
Propeller thrust	[N]	3.30	1.46	3.16
Propeller torque	[Nmm]	53.78	20.65	62.35
Type of bank	-	(b)	(c)	(c)
Bank inclination		8:1	8:1	5:1

For the confined water analysis, the mesh size and settings were determined from the open water analysis results. Efforts were made to model the mesh finer or prism boundary layers properly because a certain number of mesh layers would be important for

the narrow gap between the bank and the ship as well as the under keel clearance. At least, two boundary layers have been modelled. The OpenFOAM mesher was prone to fail to generate more than three layers even boundary layers at curvature of the hull. Nevertheless, the total number of meshes becomes less than the open water analysis due to the confined geometry of the computational domain. The size of the mesh are 0.8~1.4 million. Figure 4-4 is the mesh of whole domain, near the hull and hull surface mesh of case H. The computation domain has been modelled up to the deck height of the ship so that the bottom can be seen from the top. Freeboard height is enough to cover air domain because the confined water cases have relatively slow speed.

Too small size of the mesh at the hull surface was not appropriate because it caused y^+ value too small. A compromise was important between the cell sizes next to the hull surface and the y^+ value to model the narrow gap properly. The meshes were modelled in such a way that y^+ values ranged between 30 and 300 on all the area of the hull surface except the case H. This will be discussed in the next chapter. For the single-phase simulations, number of mesh was about 50~70% of the multi-phase case depending on the under keel clearance and water depth because the domain was modelled up to the flat free

surface level. Under water meshes are basically equal to those used for two-phase solver.

Figure 4-4: Mesh of (a) whole domain and (b) near the hull (looking from the top), and (c) mesh on the hull surface of case H

Turbulence model was determined as $k-\omega$ SST after simple comparison between $k-\epsilon$ and $k-\omega$ SST. Both gave similar force

value but $k-\epsilon$ showed a bit larger oscillation especially for sway force value though showed little difference in the converged force values. Since bigger oscillation makes it difficult to judge convergence, $k-\omega$ SST model was chosen for the analysis.

Just as the KCS open water analysis, three solvers – interFoam, LTSInterFoam and simpleFoam, are used for the confined water analysis. The comparison of the single-phase and the two-phase solvers will show how free surface contributes to the ship-to-bank interaction. As it is normally known that the VoF solver is much more sensitive and instable, this will also help finding out the stability of the VoF solver of OpenFOAM and how reliable results are in comparison to the single-phase solver.

4.2 Boundary conditions

As for the boundary conditions, a fixed value is applied at the inlet for velocity and turbulence properties. At the outlet, zero gradient is applied for velocity and turbulence properties. No slip condition is applied to the hull. For the velocity at the bank boundary, the same fixed value with the inlet velocity was used so that no-slip condition can be applied on the bank wall. This is reasonable for no-slip condition on the bank wall of body fixed coordinate system because the point of view is on the ship and undisturbed water and the bank should have the same velocity.

For the single-phase solver, a slip condition is used for the velocity on the top which is free surface level. Zero gradient can be used to the walls, top and outlet of the pressure boundary. Fixed value is applied at the inlet pressure. Standard wall function of OpenFOAM was used for turbulence properties. Table 4-4 is the summarised boundary conditions applied for the single-phase solver.

**Table 4-4: Boundary conditions for confined water analysis
of the single-phase solver**

Flow properties	inlet	outlet	top	bank	ship
U [m/s]	fixed value (u 0 0)	zero gradient	zero gradient	fixed value (u 0 0)	no slip (0 0 0)
p [m ² /s ²]	zero gradient	fixed value 0	slip	zero gradient	zero gradient
k [m ² /s ²]	fixed value 0.00015	zero gradient	zero gradient	kqRWallFunction 0.00015	
ω [1/s]	fixed value 2	zero gradient	Zero gradient	omegaWallFunction 2	

In the two-phase solver in which air and water exist together, top boundary plane becomes the boundary for atmosphere and some boundary conditions are different from the single-phase solver. The main differences are in the pressure boundary conditions and for the atmospheric boundary. The boundary conditions used for the multi-phase solver are listed in Table 4-5.

The alpha boundary conditions are needed for the volume fraction. In OpenFOAM 2.3.0, new alpha boundary condition has been implemented at the outlet for the conservation of the volume fraction.

Table 4-5: Boundary conditions for confined water analysis of the multi-phase solver

Flow properties	inlet	outlet	atmosphere	bank	ship
U [m/s]	fixed value (u 0 0)	zero gradient	pressureInlet-Outlet-Velocity	Fixed value (u 0 0)	Fixed value (0 0 0)
p _{rgh} (p _d) [kg / ms ²]	zero gradient	fixedFlux-pressure 0	Total pressure 0	fixedFluxPressure 0	
k [m ² /s ²]	fixed value 0.00015	zero gradient	inletOutlet inletValue 0.00015	kqRWallFunction 0.00015	
ω [1/s]	fixed value 2	zero gradient	inletOutlet inletValue 2	omegaWallFunction n 2	
Alpha [-]	calculated	Variable Height FlowRate	inletOutlet	zero gradient	

For the pressure boundary conditions of two phase solver, dynamic pressure is set at the boundary as discussed in the governing equation of the multi-phase solver. The pressure used for multi-phase solver has different unit from single-phase solver as density is being multiplied. ‘fixedFluxPressure’ is an improved boundary condition that sets pressure gradient based on the fixed flux, hence, determined from the velocity condition at the boundary. The flux equation (2.40) is used for this boundary condition to set pressure gradient. For instance, pressure condition becomes no gradient to apply zero flux on the no-slip wall boundary. Therefore, it is equivalent to zero gradient at the

no-slip walls. This was used for the pressure boundary conditions at the inlet and the wall boundaries. Zero gradient is used at the outlet and total pressure is used at the top. Total pressure adjust pressure based on U:

$$p_0 = p + \frac{1}{2}\rho|U|^2 \quad (4.1)$$

Chapter 5

Results and Discussion

5.1 Computation results

For the most of calculations in this study, High Performance Computing (HPC) system in Nanyang Technological University was used for parallel computation. It took about 3 hours for a 100 second simulation of about one million-mesh case with 32 processors. Because the semi-implicit method was implemented to the two-phase solver of OpenFOAM 2.3.0, the computation time was affordable with maximum Courant number 25 and interface Courant number 10. When both Courant numbers were set to be 0.4, it took about one day. For the single-phase steady-state calculations of the steady-state solver, a MacBook Pro with 2.3GHz Intel Core i7 CPU (8 cores) was used. Computation time was all less than 10 minutes.

5.1.1 Force and moment

Figure 5-1 and Figure 5-2 are the convergence time history of forces and moment by two different pressure gradient schemes.

Figure 5-1: Convergence history of (a) X and (b) Y forces (N), and (c) Z moment (Nm) from pressure of case F (Gradient scheme: Gauss linear, Max Co. reduced from 70s)

Figure 5-2: Convergence history of (a) X and (b) Y forces (N), and (c) Z moment (Nm) from pressure of case F (Gradient scheme: Least square)

The computation results of two-phase solver show that pressure component dominates over the viscosity with large oscillations for ship-to-bank interaction. It is presumed that the oscillation is caused from the interaction between the bank and the ship because the oscillation indicates quite different behaviour from the open water analysis in spite of relatively slow speed. Even the longitudinal force showed different oscillating behaviour from the open water analysis that was discussed in the previous chapter. Gauss linear and Least squares schemes that have 2nd order accuracy were evaluated as the gradient scheme. As Gauss Linear method showed big oscillation, Courant number was reduced to 0.4 when the pressure value is stabilised at 70 second. This oppressed the oscillation and helped in averaging values. However, the computation took about one day for the additional 50 seconds simulation with the low Courant number. The force values were less oscillating with Least Square method and with high Courant number. The under relaxation factors were also used for stable solutions as follows:

p	0.3
U	0.7
k	0.7
omega	0.7

Because the values showed oscillating behaviour, the computation had to be monitored over the sufficient time duration and the converged force values were obtained from the average over the time where the oscillation is stabilised.

Three different solvers have been used for the simulations. As discussed in the Chapter 3.3, `simpleFoam` is a single-phase steady-state solver, `interFoam` is a VoF transient solver and `LTSInterFoam` is a local time stepping VoF solver. Table 5-1 is the comparison table of forces and moments between the CFD results and the experiment (Lataire et al., 2009). From the comparison results, `LTSInterFoam` gave wrong values just as shown for KCS analysis in open water. Different mesh type and mesh fineness were tested to rectify this, however, all did not improve the results much although other numerical settings were almost the same with `interFoam`. As it gave wrong results, Least square scheme was not tested for the LTS solver, moreover, the LTS solver was excluded from the evaluation hereafter. `interFoam` gave reasonable accuracy. Comparing the different gradient schemes, least square generally showed better results compared to Gauss linear. `simpleFoam` also allowed good accuracy in spite of single-phase simulation without free surface. Least square method also showed better accuracy for single-phase calculations.

**Table 5-1: Comparison of the cases without thrust to the experiment results
(Lataire et al. [5])**

Case		A		C		F		H	
Solver	Value	Gauss linear	Least Squares	Gauss linear	Least Squares	Gauss linear	Least Squares	Gauss linear	Least Squares
LTSInter-Foam	Fx (N)	-6.960	-	Diverged		-9.345	-	-3.211	-
	Fy (N)	-8.190				-1.029		0.539	
	Mz (Nm)	-11.652				1.259		4.600	
Inter-Foam	Fx (N)	-4.799	-4.866			-5.465	-5.512	-1.559	-2.239
	Fy (N)	-7.122	-5.646			-1.111	-1.008	0.581	1.963
	Mz (Nm)	-8.112	-9.067			1.982	1.305	2.159	5.561
Simple-Foam	Fx (N)	-4.193	-3.922	-4.062	-4.094	-4.395	-3.794	-1.839	-1.880
	Fy (N)	-4.518	-5.343	-0.789	0.709	-1.106	-1.300	0.807	1.245
	Mz (Nm)	-9.948	-8.665	4.145	6.810	0.629	1.053	3.296	4.017
EXP	Fx (N)	-4.130		-2.618		-5.354		-2.935	
	Fy (N)	-5.139		0.962		-1.648		2.545	
	Mz (Nm)	-8.939		4.300		1.804		6.959	

The followings are the comparison results of each case.

Case A The two-phase solver over-predicted longitudinal force slightly, but sway force and yaw moment showed reasonable accuracy with Least square method. The single-phase solver showed good accuracy compared to the experiment. Least Square scheme gave better result of sway force and yaw moment for the both solvers with the reasonable accuracy.

Case C This case did not converge for the VoF solvers in spite of many trials and efforts. However, the single-phase solution was able to converge. Single-phase solver also did not show good accuracy. Even Gauss linear gradient scheme predicted sway force to the opposite direction.

Case F The two-phase solution with Gauss linear gradient scheme gave the good accuracy for longitudinal force and yaw moment, however, sway force was under-predicted. Single-phase solver predicted better sway force but not for other values. The reason for this is not clear.

Case H This case has small under keel clearance which is 10% of water draught, but the separation from the bank wall is however, not small. When same meshing type (2 layers of boundary layer) was used from other cases, all the values were under-predicted. The details will be discussed in the next Section 5.1.2. The presented values in Table 5-1 were when mesh was further refined to have more boundary layers.

5.1.2 Under keel space modelling of the case H

Because of small under keel clearance of the case H, it was difficult to model the mesh in the narrow space properly. If many layers are modelled in the under keel space, y^+ value becomes too low, i.e. below 30. In that case, the wall function may not be valid. To keep y^+ value bigger than 30, the optimum number of layer in the under keel clearance had to be only about two in the case H. The small under keel clearance is known to affect the bank effect significantly, therefore, too few number of layers would not be enough to capture the effect. In this case, the best thing is to model y^+ value less than 1 throughout the hull surface and resolve the whole boundary layer without using wall functions. In the OpenFOAM mesher by which meshing is very slow compared to commercial meshers, this was not efficient because number of mesh becomes too huge.

To compromise between the number of layers and y^+ value in the under keel space, a comparison has been made. Four different meshes are tested for the case H. They have same mesh size in other area but different boundary layers or different mesh in the under keel space. The tested meshes are shown in Figure 5-3 that illustrates the midship section near the bilge radius.

**Figure 5-3: Different meshing of case H -
no layer (a), three layers (b), eight layers (c), fine mesh (d)**

Figure 5-4 shows y^+ distribution for each of the four meshing types.

**Figure 5-4: y^+ of the case H with different meshing
no layer (a), three layers (b), eight layers (c), fine mesh (d)**

(a) has no specific modelling in the boundary layer, (b) has two prism boundary layers and total three layers in the under keel, (c) has eight prism boundary layers and total nine in the under keel, and (d) has very fine refinement in the under keel clearance without modelling boundary layer. (b) type meshing is a general boundary modelling for all other test cases used in this study.

(a) and (b) have y^+ values above 30 and all in good range. OpenFOAM mesher has problem when many prism layers have to be created at the boundary layer especially at large curvature. For this reason, prism layers have not been generated as much as amidships at fore and aft area of (c) where curvature appears. Instead, many mesh layers have been created in the boundary layer at the parallel middle body area where under keel clearance is most significant. In (d), very fine mesh was used in under keel space, hence, very low y^+ distribution can be seen on the bottom hull surface.

The computed results for the different meshes of the case H are tabulated in Table 5-2. Least Square methods are used for the gradient scheme. From the comparison results, (c) type meshing showed the best results. Therefore, it is considered that number of layers in the under keel clearance are important to predict

bank effect. Many prism layers at the boundary layer are more effective than the (d) type meshing.

Table 5-2: Comparison between different meshing in the under keel space of the case H

	(a)	(b)	(c)	(d)	EXP
No. of mesh (mil.)	1.11	1.13	1.15	5.21	-
F _x (N)	1.484	2.239	2.236	2.247	2.935
F _y (N)	0.368	1.104	1.963	1.037	2.545
M _z (Nm)	4.004	3.734	5.561	3.647	6.959

More comparisons are made for the meshing (b) and (c). Figure 5-5 shows pressure distribution in the water at the centreline of the ship.

Figure 5-5: Pressure [N] distribution in the water at the centreline of the case H with (b) three layers (c) eight layers

The pressure distribution on the hull was compared in Figure 5-6. For better visualisation, the range of the pressure applied became different for the two cases.

Figure 5-6: Pressure [N] distribution on the hull (looking from the bottom) of the case H with (b) three layers (c) eight layers

When eight layers are used (c), the overall pressure is higher both in the water (Figure 5-5) and on the hull (Figure 5-6). In Figure 5-5, the pressure difference between the low pressure region (under keel space) and the high pressure region (bow) of the case (c) (ranged in 1380~1475N) is bigger compared to the case (b) (ranged in 1350~1425 N).

5.1.3 Pressure and velocity asymmetry

The computation shows clear pressure asymmetry that causes the bank effect. Figure 5-7 shows the computed pressure distributions of the cases on the hull surface and the bank surface. The deck is not shown as the domain is modelled up to the freeboard height.

Figure 5-7: Pressure [N] distribution on the hull and bank surface of the cases F (a) and H (b) (looking from the top)

Because the incoming velocity is much faster in the case F (0.8012 m/s) than in the case H (0.4578 m/s), the pressure range and its variation are much higher in the case F. However,

the measured sway force and yaw moment from the experiment appeared much bigger in the case H than in the case F. Therefore, it is obvious that under keel clearance affects more significantly although bank effect is known to be almost proportional to velocity square.

Next, three sectional planes were cut into the water to investigate the pressure distributions in the flow. Sections were cut at amidships, near the bow (station 17.5) and near the stern (station 2.5) were cut and presented in Figure 5-8. As force is directed from high to low pressure, the direction of yaw moment can be guessed from this sectional views. High pressure is observed near the bow and this would contribute to the yaw moment. The calculated negative yaw moment of the case A can be observed from the very high pressure at the port side of the bow. On the other hand, the cases F and H where positive yaw moments were calculated have strong high pressure at the starboard side.

Figure 5-9, Figure 5-10 and Figure 5-11 show the velocity magnitude of the cases F, H and I at various water levels. The case I that has small clearance between the ship and the bank was added for better comparison and this case illustrates the most asymmetry of velocity field. The case F that has less

extreme condition than the cases H and I shows the least asymmetry among the three cases. The case F reveals biggest asymmetric velocity at the 1/3 draught level, and the cases H and I reveal at the 2/3 draught level.

Figure 5-8: Dynamic pressure [N] distribution of OpenFOAM calculation of case A (a), F (b) and H (c) at the different sections (station 2.5, amidships and station 17.5)

Figure 5-9: Velocity magnitude [m/s] of the case F at (a) free surface, (b) 2/3 of draught from baseline, (c) 1/3 of draught from baseline and (d) baseline

Figure 5-10: Velocity magnitude [m/s] of the case H at (a) free surface, (b) 2/3 of draught from baseline, (c) 1/3 of draught from baseline and (d) baseline

Figure 5-11: Velocity magnitude [m/s] of the case I at (a) free surface, (b) 2/3 of draught from baseline, (c) 1/3 of draught from baseline and (d) baseline

5.1.4 Results of the cases with propeller rotation

Table 5-3 is the comparison results of the cases with propeller thrust. Sway forces and yaw moments are compared with the experiment results. Since local time stepping solver showed wrong results for most of the cases, it has not been included in these cases. Although the case D and the case I had small clearance of the sloped bank and the hull surface, however the solution converged well with reasonable accuracy of yaw moment.

Table 5-3: Comparison of the cases with thrust to the experiment results (Lataire et al. [5])

Solver	Case	D		I		J	
	Value	Gauss Linear	Least Square	Gauss Linear	Least Square	Gauss Linear	Least Square
inter-Foam	Fy (N)	-2.011	-0.754	-0.201	0.137	-0.894	-0.696
	Mz (Nm)	19.28	15.932	9.750	12.60	2.169	1.838
simple-Foam	Fy (N)	-2.683	-2.354	-1.080	-0.699	-1.427	Diverged
	Mz (Nm)	4.692	5.108	1.960	2.117	1.688	
EXP	Fy (N)	-8.056		-3.682		-2.264	
	Mz (Nm)	16.407		8.312		2.065	

As can be seen in the comparison, when the propeller was rotating, all the sway forces showed under prediction compared

to the experiment. In these cases, Gauss linear method showed relatively better results than least square method for yaw moment. The direction of the sway forces was mostly correct but the magnitudes were all wrong. When the propeller is rotating, sway forces were all in suction direction to the bank wall. However, yaw moment values showed reasonable accuracy. This implies the propeller rotation significantly affects the magnitude of sway forces. Single-phase solver showed poor accuracy even for yaw moments. Further comparison for these cases are not reliable but will become speculation, hence, it is not meaningful.

The single-phase calculations with Least Squares scheme for the case J diverged.

5.1.5 Comparison between single-phase and two-phase solutions

It was expected that the difference between the calculated values and the experiment single-phase solver and two-phase solver would show how free surface affects on the bank effect. However, the single-phase solution of the cases A and F showed even better accuracy than two-phase solution. In other cases, single-phase solution showed under prediction compared to the two-phase solutions. Especially the cases with propulsion showed

poor accuracy. From the case C, single-phase solver is more stable than two-phase solver.

5.1.6 Free surface elevation

Free surface elevations were compared with experiment results. The wave measurements were done in three points of the cases C, F, I and J. The comparison was done for these available cases except the case C that was failed to converge. Table 5-4 is the location of the wave gauges from centreline of the ship according to the experiment. Wave gauges measure the wave elevation at the defined transverse distances from the centreline of the ship.

Table 5-4: The locations (m) of the wave gauges with reference to the centreline of the ship (Lataire et al. [5])

	Wave gauge 1	Wave gauge 2	Wave gauge 3
Case F	0.475	1.005	1.535
Case I	0.875	1.215	2.115
Case J	0.920	1.320	1.785

Figure 5-12 is the graphical views of the measuring locations of the wave gauges and also gives a brief idea about the water depth at each wave gauge.

Figure 5-12: The locations of the wave gauges 1, 2 and 3 of the cases (a) F, (b) I and (c) J

Figure 5-13, Figure 5-14 and Figure 5-15 show the comparison of wave elevations with the experiment measurements. In the horizontal scale, 0 stands for amidships and positive direction is toward the stern.

Figure 5-13: Wave elevation of the case F

at (a) gauge 1, (b) gauge 2 and (c) gauge 3 (Lataire et al. [5])

Figure 5-14: Wave elevation of the case I

at (a) gauge 1, (b) gauge 2 and (c) gauge 3 (Lataire et al. [5])

Figure 5-15: Wave elevation of the case J

at (a) gauge 1, (b) gauge 2 and (c) gauge 3 (Lataire et al. [5])

The further wave gauges from the ship showed less accuracy of the wave elevation. A steep wave is observed in the case I and it is continued from the hull to the bank wall. This was also found in the experiment but over-predicted in the computation result at the gauges 1 and 2. In addition, phase delay was observed, especially at the furthest wave gauge from the ship, i.e. wave gauge 3. This phase delay was also found in the case F. The phase of wave trough was deviated by about a quarter of wave length. The case J showed relatively good accuracy. As can be seen in Figure 5-12, the wave elevation is considered less accurate at shallow water region – at the wave gauges 3 of the case F, and at the wave gauge 2 and 3 of the case I. Lastly, the computations did not capture the oscillation of the wave elevation correctly. The experiment showed more oscillation but this was not captured or captured differently in the computation results.

5.2 Discussion

OpenFOAM was able to predict ship-to-bank interaction with the different bank types in extremely shallow water. Two-phase solver failed to converge from finding the solution of one severe case that has extreme under keel clearance and small separation from the sloped bank wall.

For the stable calculations and convergence, good quality of meshing was particularly important in OpenFOAM compared to other commercial CFD codes. Another reason for this is pressure becomes sensitive in the multi-phase solver compared to the single-phase solver. Special care is required for the aspect ratio and high skewness of the grid.

Another challenge is a compromise between the mesh fineness and y^+ value to model the narrow gap between the hull and the bank. Boundary layer and the narrow gap modelling are important although y^+ values have to go slightly below 30 that falls to buffer layer. For the case with small under keel clearance, good meshing improves sway force significantly.

The pressure values of the calculations in confined water often show more oscillating behaviour as compared to the open water

analysis when converging. It is considered to occur from the reflections and interactions between the bank and the ship. Reducing Courant number was helpful to suppress the oscillation. Least Square method of gradient scheme showed less oscillating behaviour than Gauss linear. Overall, Gauss linear scheme showed better accuracy, however Courant Number had to be reduced to very small, e.g. 0.4 in order to capture the force and moment correctly due to the oscillation.

The computation tends to under-predict forces and moment. This is most obvious in sway force. There would be several reasons for this. Firstly, there is limitation of meshing in the narrow gap. One thing can try is resolving whole boundary layer by modelling y^+ below 1. However, this is not easy with OpenFOAM mesher and a commercial mesher should be employed for efficient boundary layer modelling. Secondly, fixed heave and pitch motion of CFD calculations could have caused this under-prediction because the two motions were free in the model tests. This assumption of fixed motion neglects squat effect, therefore, more accuracy can be expected if the 2 DOF motions are set to be free. However, the calculated sway forces and yaw moments from the bank effect are all in correct directions for all the types of bank shapes and arrangement although ship-to-bank interaction is very complex and difficult to predict the directions.

The under-prediction was more obvious in the cases with propeller rotation while yaw moments were predicted with reasonable accuracy. This explains how propeller rotation affects the sway force. More study is needed whether propeller intervention cause additional sway force to specific direction. In this set of experiments, all the cases with propeller rotation resulted in suction forces and the computation without propeller modelling computed the sway forces to the correct direction but in smaller magnitude. However, it is hard to conclude this at this stage. From this comparison, it is noted that propeller rotation seems to affect the magnitude of sway forces but does not give significant contribution to yaw moment.

Free surface elevation was less accurate than the open water analysis. In addition, difference was found in wave phase. Such a phase delay was more obvious in the case where the water depth is shallow. Less oscillation on the free surface level was observed in the computation than the experiment result.

Chapter 6

Conclusion

6.1 Conclusion

In this study, CFD theory and numerical techniques were discussed. In order to use open source CFD code, the discretisation of governing equations, solution algorithm, stability of the computation, boundary condition and boundary layer treatment were explained. Several features and the solvers of OpenFOAM were also investigated.

In this study, flow around ships in open water and in confined water with several bank arrangements was simulated by incompressible RANS solver of OpenFOAM. The results from single-phase and two-phase solver were compared. The hydrodynamic forces and moment due to the bank, wave elevation were compared to the experiment results.

Under keel clearance modelling was evaluated to compromise between mesh fineness and y^+ distribution. The gradient schemes of OpenFOAM were compared in terms of accuracy and stable computation. OpenFOAM was able to predict the bank

effect in confined water that causes the bank effect as well as the open water analysis. Asymmetric pressure and velocity fields were well presented in connection to the calculated forces and moment. In order to find the propeller contribution to the bank effect, the cases with propeller thrust were compared to the computations without propeller.

The reason for under prediction of sway force has to be further investigated with careful meshing and numerical schemes. Unfortunately, the experiment results open to the public were only available to those particular cases, therefore, systematic parametric study was not possible.

6.2 Future work

As for the CFD analysis of the bank effect, only one case which is the vertical bank arrangement was successful in this report. Analysis for other cases with sloped banks and with different under keel clearances have to be tested, too. There are two types of sloped bank in the experiments. One type is surface piercing sloped bank and the other is where the slope ends below the waterline.

Since good meshing affects the calculations of multiphase solver of OpenFOAM significantly, there will be intensive study in meshing options especially for the refinement at near wall region. Otherwise, commercial software can be used for the meshing. Meshing software that can export compatible file type to OpenFOAM can be used for this. In addition, it will be valuable to compare the results of OpenFOAM to that of a commercial code.

OpenFOAM has strong dynamic mesh options with adaptive mesh refinement capability. This can add pitch and heave motion of ship, hence, it is expected that the inclusion of shallow water effect will give better accuracy.

Appendix

The source codes of interFoam

interFoam.C: The main codes of the solver. It handles PIMPLE loop and connects all the related header files to solve the governing equations.

```
1    #include "fvCFD.H"
2    #include "CMULES.H"
3    #include "subCycle.H"
4    #include "interfaceProperties.H"
5    #include "incompressibleTwoPhaseMixture.H"
6    #include "turbulenceModel.H"
7    #include "pimpleControl.H"
8    #include "fvIOoptionList.H"
9    #include "fixedFluxPressureFvPatchScalarField.H"
10
11    // * * * * *
12
13    int main(int argc, char *argv[])
14    {
15        #include "setRootCase.H"
16        #include "createTime.H"
17        #include "createMesh.H"
18
19        pimpleControl pimple(mesh);
20
21        #include "initContinuityErrs.H"
22        #include "createFields.H"
23        #include "readTimeControls.H"
24        #include "createPrghCorrTypes.H"
25        #include "correctPhi.H"
26        #include "CourantNo.H"
27        #include "setInitialDeltaT.H"
28
29        // * * * * *
30
```

```

31 Info<< "\nStarting time loop\n" << endl;
32
33 while (runTime.run())
34 {
35     #include "readTimeControls.H"
36     #include "CourantNo.H"
37     #include "alphaCourantNo.H"
38     #include "setDeltaT.H"
39
40     runTime++;
41
42     Info<< "Time = " << runTime.timeName() << nl << endl;
43
44     // --- Pressure-velocity PIMPLE corrector loop
45     while (pimple.loop())
46     {
47         #include "alphaControls.H"
48
49         if (pimple.firstIter() || alphaOuterCorrectors)
50         {
51             twoPhaseProperties.correct();
52
53             #include "alphaEqnSubCycle.H"
54             interface.correct();
55         }
56
57         #include "UEqn.H"
58
59         // --- Pressure corrector loop
60         while (pimple.correct())
61         {
62             #include "pEqn.H"
63         }
64
65         if (pimple.turbCorr())
66         {
67             turbulence->correct();
68         }
69     }
70
71     runTime.write();
72
73     Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
74         << " ClockTime = " << runTime.elapsedClockTime() << " s"
75         << nl << endl;

```

```

76     }
77
78     Info<< "End\n" << endl;
79
80     return 0;
81 }

```

AlphaEqn.H: It contains equations for the volume fraction α including MULES algorithm (Refer to Chapter 3.3)

```

1  {
2      word alphaScheme("div(phi,alpha)");
3      word alphasScheme("div(phirb,alpha)");
4
5      // Standard face-flux compression coefficient
6      surfaceScalarField phic(interface.cAlpha()*mag(phi/mesh.magSf()));
7
8      // Add the optional isotropic compression contribution
9      if (icAlpha > 0)
10     {
11         phic *= (1.0 - icAlpha);
12         phic += (interface.cAlpha()*icAlpha)*fvc::interpolate(mag(U));
13     }
14
15     // Do not compress interface at non-coupled boundary faces
16     // (inlets, outlets etc.)
17     forAll(phic.boundaryField(), patchi)
18     {
19         fvsPatchScalarField& phicp = phic.boundaryField()[patchi];
20
21         if (!phicp.coupled())
22         {
23             phicp == 0;
24         }
25     }
26
27     tmp<surfaceScalarField> tphiAlpha;
28
29     if (MULESCorr)

```

```

30  {
31      fvScalarMatrix alpha1Eqn
32      (
33          #ifdef LTSSOLVE
34      fv::localEulerDdtScheme<scalar>(mesh,rDeltaT.name()).fvmDdt(alpha1)
35          #else
36      fv::EulerDdtScheme<scalar>(mesh).fvmDdt(alpha1)
37          #endif
38      + fv::gaussConvectionScheme<scalar>
39      (
40          mesh,
41          phi,
42          upwind<scalar>(mesh, phi)
43      ).fvmDiv(phi, alpha1)
44      );
45
46      alpha1Eqn.solve();
47
48      Info<< "Phase-1 volume fraction = "
49          << alpha1.weightedAverage(mesh.Vsc()).value()
50          << " Min(alpha1) = " << min(alpha1).value()
51          << " Max(alpha1) = " << max(alpha1).value()
52          << endl;
53
54      tmp<surfaceScalarField> tphiAlphaUD(alpha1Eqn.flux());
55      tphiAlpha = tmp<surfaceScalarField>
56      (
57          new surfaceScalarField(tphiAlphaUD())
58      );
59
60      if (alphaApplyPrevCorr && tphiAlphaCorr0.valid())
61      {
62          Info<< "Applying the previous iteration compression flux" << endl;
63          #ifdef LTSSOLVE
64          MULES::LTScorrect(alpha1, tphiAlpha(), tphiAlphaCorr0(), 1, 0);
65          #else
66          MULES::correct(alpha1, tphiAlpha(), tphiAlphaCorr0(), 1, 0);
67          #endif
68
69          tphiAlpha() += tphiAlphaCorr0();
70      }
71
72      // Cache the upwind-flux
73      tphiAlphaCorr0 = tphiAlphaUD;
74

```

```

75     alpha2 = 1.0 - alpha1;
76
77     interface.correct();
78 }
79
80 for (int aCorr=0; aCorr<nAlphaCorr; aCorr++)
81 {
82     surfaceScalarField phir(phic*interface.nHatf());
83
84     tmp<surfaceScalarField> tphiAlphaUn
85     (
86         fvc::flux
87         (
88             phi,
89             alpha1,
90             alphaScheme
91         )
92     + fvc::flux
93     (
94         -fvc::flux(-phir, alpha2, alphasScheme),
95         alpha1,
96         alphasScheme
97     )
98 );
99
100 if (MULESCorr)
101 {
102     tmp<surfaceScalarField> tphiAlphaCorr(tphiAlphaUn() - tphiAlpha());
103     volScalarField alpha10(alpha1);
104
105     #ifdef LTSSOLVE
106     MULES::LTScorrect(alpha1, tphiAlphaUn(), tphiAlphaCorr(), 1, 0);
107     #else
108     MULES::correct(alpha1, tphiAlphaUn(), tphiAlphaCorr(), 1, 0);
109     #endif
110
111     // Under-relax the correction for all but the 1st corrector
112     if (aCorr == 0)
113     {
114         tphiAlpha() += tphiAlphaCorr();
115     }
116     else
117     {
118         alpha1 = 0.5*alpha1 + 0.5*alpha10;
119         tphiAlpha() += 0.5*tphiAlphaCorr();

```



```

120     }
121 }
122 else
123 {
124     tphiAlpha = tphiAlphaUn;
125
126     #ifdef LTSSOLVE
127     MULES::explicitLTSSolve(alpha1, phi, tphiAlpha(), 1, 0);
128     #else
129     MULES::explicitSolve(alpha1, phi, tphiAlpha(), 1, 0);
130     #endif
131 }
132
133     alpha2 = 1.0 - alpha1;
134
135     interface.correct();
136 }
137
138     rhoPhi = tphiAlpha()*(rho1 - rho2) + phi*rho2;
139
140     if (alphaApplyPrevCorr && MULESCorr)
141     {
142         tphiAlphaCorr0 = tphiAlpha() - tphiAlphaCorr0;
143     }
144
145     Info<< "Phase-1 volume fraction = "
146         << alpha1.weightedAverage(mesh.Vsc()).value()
147         << " Min(alpha1) = " << min(alpha1).value()
148         << " Max(alpha1) = " << max(alpha1).value()
149         << endl;
150 }

```

AlphaEqnSubCycle.H: This is for additional sub-cycles in the given time step in order to compute stable α field.

```

1     if (nAlphaSubCycles > 1)
2     {
3         dimensionedScalar totalDeltaT = runTime.deltaT();
4         surfaceScalarField rhoPhiSum

```

```

5      (
6      IOobject
7      (
8      "rhoPhiSum",
9      runTime.timeName(),
10     mesh
11     ),
12     mesh,
13     dimensionedScalar("0", rhoPhi.dimensions(), 0)
14     );
15
16     for
17     (
18         subCycle<volScalarField> alphaSubCycle(alpha1, nAlphaSubCycles);
19         !(++alphaSubCycle).end();
20     )
21     {
22         #include "alphaEqn.H"
23         rhoPhiSum += (runTime.deltaT()/totalDeltaT)*rhoPhi;
24     }
25
26     rhoPhi = rhoPhiSum;
27 }
28 else
29 {
30     #include "alphaEqn.H"
31 }
32
33 rho == alpha1*rho1 + alpha2*rho2;

```

UEqn.H: The velocity equation of the governing equation. This is coupled with pressure equation in “PEqn.H”.

```

1      fvVectorMatrix UEqn
2      (
3          fvm::ddt(rho, U)
4          + fvm::div(rhoPhi, U)
5          + turbulence->divDevRhoReff(rho, U)
6      );
7

```

```

8      UEqn.relax();
9
10     if (pimple.momentumPredictor())
11     {
12         solve
13         (
14             UEqn
15             ==
16             fvc::reconstruct
17             (
18                 (
19                     fvc::interpolate(interface.sigmaK())*fvc::snGrad(alpha1)
20                     - ghf*fvc::snGrad(rho)
21                     - fvc::snGrad(p_rgh)
22                 ) * mesh.magSf()
23             )
24         );
25     }

```

PEqn.H: The pressure equation of the governing equation as discussed in Chapter 2.2.2. “phiHbyA” in the source codes stands for $\frac{H(U)}{a_p}$. See the equations (2.35) ~ (2.39).

```

1      {
2          volScalarField rAU("rAU", 1.0/UEqn.A());
3          surfaceScalarField rAUf("rAUf", fvc::interpolate(rAU));
4
5          volVectorField HbyA("HbyA", U);
6          HbyA = rAU*UEqn.H();
7
8          surfaceScalarField phiHbyA
9          (
10             "phiHbyA",
11             (fvc::interpolate(HbyA) & mesh.Sf())
12             + fvc::interpolate(rho*rAU)*fvc::ddtCorr(U, phi)
13         );
14
15         adjustPhi(phiHbyA, U, p_rgh);

```

```

16
17     surfaceScalarField phig
18     (
19         (
20             fvc::interpolate(interface.sigmaK())*fvc::snGrad(alpha1)
21             - ghf*fvc::snGrad(rho)
22         )*rAUf*mesh.magSf()
23     );
24
25     phiHbyA += phig;
26
27     // Update the fixedFluxPressure BCs to ensure flux consistency
28     setSnGrad<fixedFluxPressureFvPatchScalarField>
29     (
30         p_rgh.boundaryField(),
31         (
32             phiHbyA.boundaryField()
33             - (mesh.Sf()).boundaryField() & U.boundaryField())
34         )/(mesh.magSf()).boundaryField()*rAUf.boundaryField()
35     );
36
37     while (pimple.correctNonOrthogonal())
38     {
39         fvScalarMatrix p_rghEqn
40         (
41             fvm::laplacian(rAUf, p_rgh) == fvc::div(phiHbyA)
42         );
43
44         p_rghEqn.setReference(pRefCell, getRefCellValue(p_rgh, pRefCell));
45
46         p_rghEqn.solve(mesh.solver(p_rgh.select(pimple.finalInnerIter())));
47
48         if (pimple.finalNonOrthogonalIter())
49         {
50             phi = phiHbyA - p_rghEqn.flux();
51
52             p_rgh.relax();
53
54             U = HbyA + rAU*fvc::reconstruct((phig - p_rghEqn.flux())/rAUf);
55             U.correctBoundaryConditions();
56             fvOptions.correct(U);
57         }
58     }
59
60     #include "continuityErrs.H"

```

```

61
62     p == p_rgh + rho*gh;
63
64     if (p_rgh.needReference())
65     {
66         p += dimensionedScalar
67         (
68             "p",
69             p.dimensions(),
70             pRefValue - getRefCellValue(p, pRefCell)
71         );
72         p_rgh = p - rho*gh;
73     }
74 }

```

References

- [1] Alberto Ghione, Development and Validation of a two-phase CFD model using OpenFOAM, Master Thesis, Royal Institute of Technology, Stockholm, Sweden, 2012
- [2] C. Lo, Dong-Taur Su and I-Fu Lin, Applying Computational Fluid Dynamics to Simulate Bank Effects, 19th International Offshore and Polar Engineering Conference, page 466~471, Osaka, Japan, 2009
- [3] D. W. Park, H. J. Choi and K. J. Paik, Bank Effect of a Ship Operating in a Shallow Water and Channel, Journal of Navigation and Port, Volume 38, No. 1, page 19~27, 2014
- [4] E. Lataire, M. Vantorre, E. Laforce, K. Eloot and G. Delefortrie, Navigation in Confined Waters: Influence of Bank Characteristics on Ship-Bank Interaction, International Conference on Ship Manoeuvring in Shallow and Confined Water, page 135~143, Abtwerp, Belgium, 2007
- [5] E. Lataire, M. Vantorre and K. Eloot, Systematic Model Tests on Ship-Bank Interaction Effects, International Conference on Ship Manoeuvring in Shallow and Confined Water: Bank Effect, Abtwerp, Belgium, 2009
- [6] Eric Furbo, Evaluation of RANS turbulence models for flow problems with significant impact of boundary layers, Master thesis, Uppsala University, Sweden, 2010
- [7] H. K. Versteeg and W. Malalasekera, An Introduction to Computational Fluid Dynamics, 2nd Edition, Pearson Education Limited, UK, 2007

- [8] Hirt C. W., Nichols B. D, Volume of Fluid (VoF) method for the dynamics of free boundaries, J. of Computer Physics, 1981, 39(1), page 201-225
- [9] Hrvoje Jasak, Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows, Thesis for Doctor of Philosophy, University of London and Imperial College, UK, 1996
- [10] Huaming Wang, Zaojian Zou, Yonghe Xie and Wanlun Kong, Numerical Study of Viscous Hydrodynamic Forces on a Ship Navigating near Bank in Shallow Water, Proceedings of the 20th International Offshore and Polar Engineering Conference, page 523~528, Beijing, China, 2010
- [11] Iranzo, J. Domingo, I. Trejo, M. Terceno and J. Valle, Calculation of the resistance and the profile of a 3600 TEU container ship, Canal de Experiencias Hidrodinamicas, El Pardo, Ministerio de Defensa, Barcelona, Spain, 2007
- [12] Issa, R.I., Solution of the implicitly discretized fluid flow equations by operator-splitting, J. Comp. Physics, 62:40-65, 1986
- [13] John D. Anderson, Computational Fluid Dynamics, The Basics with Applications, McGraw-Hill Inc., New York, USA, 1995
- [14] K. Lee and S. G. Lee, Investigation of ship maneuvering with hydrodynamic effects between ship and bank, Journal of Mechanical Science and Technology, No. 22, page 1230~1236, 2008
- [15] Kevin Maki, Ship Resistance Simulation with OpenFOAM, 6thOpenFOAM workshop, PA, USA, 2011

- [16] Kongsberg Maritime, Hydrodynamic Modelling Tool Report SU-0689, Mathematical Model of Ship Dynamics, Norway, 2012
- [17] Knut Inge Engelbreth, Flow Around a Circular Cylinder Near a Plate Wall, Master Thesis, Norwegian University of Science and Technology, Norway, 2011
- [18] L. Larsson, B. Regnstrom, L. Broberg, D. Li, C. Janson, Features, Fantasies, and Feats in the Theoretical/Numerical Prediction of Ship Performance, Proceedings of the 22nd Symposium on Naval Hydrodynamics, page 11-32, Washington D.C, USA, 1998
- [19] L. Zou, L. Larsson, G. Delefortrie and E. Lataire, CFD Prediction and Validation of Ship-bank Interaction in a Canal, 2nd International Conference on Ship Manoeuvring in Shallow and Confined Water: Ship to Ship Interaction, May 18-20, Trondheim, Norway, 2011
- [20] Liu Jing, Investigation of multiphase liquid system in microchannels, PhD thesis, School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore, 2014
- [21] M. Vantorre, and G. Delefortrie, Experimental Investigation of Ship-Bank Interaction Forces, Proceeding of International Conference on Marine Simulation and Ship Manoeuvrability, MARSIM'03, page RC-31-1~9, Kanazawa, Japan, 2003
- [22] Mehdi Nakisa, Adi Maimun, Yasser M. Ahmed, A.Y.Sian, A. Priyanto, Jaswaramd F. Behrouzi, RANS Simulation of the Viscous Flow around Hull of LNG Ship in Confined Water, 15th Mathematical and Computational Methods in Science and Engineering, page 133~137, Kuala Lumpur, Malaysia, 2013

- [23] Norrbin, Bank Effect on a Ship Moving through a Short Dredged Channel, Proceedings of 10th Symposium on Naval Hydrodynamics, page 71~88, Cambridge, MA, USA, 1974
- [24] OpenFOAM®, The Open source CFD toolbox, User Guide 2.3.0
- [25] OpenFOAM®, The Open source CFD toolbox, Programmers Guide 2.3.0
- [26] OpenFOAM® v2.0.0 official website about Steady-State VoF solver, <http://www.openfoam.org/version2.0.0/steady-VoF.php> (Last accessed in Jan 2015).
- [27] OpenFOAM® v2.3.0 official website about Multiphase modelling,
<http://www.openfoam.org/version2.3.0/mutiphase.php>
(Last accessed in Mar 2015).
- [28] OpenFOAM wiki website, <http://openfoamwiki.net> (Last accessed in May 2014).
- [29] Patankar, S.V., Numerical heat transfer and fluid flow, Hemisphere Publishing Corporation, Taylor & Francis Group, New York, USA, 1980
- [30] Pedro Miguel Borges Lopes, Free-surface flow interface and air-entrainment modelling using OpenFOAM, Doctorial Program Report, Universidade de Coimbra, Portugal, 2013
- [31] Q. Li, M. Leer-Andersen, P. Ottosson, P. Trägårdh, Experimental Investigation of Bank Effects Under Extreme Conditions, Proceeding of PRADS 2001, page 541-546, Shanghai, China, 2001

- [32] Quanming Miao and Jinzhu Xia, Numerical Study of Bank Effect on a Ship Traveling in a Channel, The 8th International Conference on Numerical Ship Hydrodynamics, Busan, Korea, 2003
- [33] Santiago Marquez Damian, An Extended Mixture Model for the Simultaneous Treatment of Short and Long Scale Interfaces, PhD Thesis, Universidad Nacional Del Litoral, Argentina, 2013
- [34] Schlichting H., Boundary Layer Theory, 7th edition, McGraw-Hill, USA, 1979
- [35] S. G. Lee, Ship motion and manoeuvring, Lecture Note, page 116~119, Department of Naval Architecture and Ocean Engineering, Pusan National University, Korea, 1998
- [36] The manoeuvring Committee of ITTC, Final Report and Recommendations to the 23rd ITTC, Proceedings of the 23rd International Towing Tank Conference - Volume I, page 201~234, Venice, Italy, 2002
- [37] The manoeuvring Committee of ITTC, Final Report and Recommendations to the 26th ITTC, Proceedings of the 26th International Towing Tank Conference - Volume I, page 129~154, Rio de Janeiro, Brazil, 2011
- [38] W. J. Kim, S. H. Van and D. H. Kim, Measurements of Flows Around Modern Commercial Ship Models, Experiments in Fluids, page 567~578, 2001

Publication from the present work

Youngkook Kim, E. Y. K. Ng and Gopala Krishnan, CFD Prediction of Ship-to-bank Interaction for Shiphhandling Simulator, MARTECH 2014, Singapore, 29~30 April, 2014

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

CFD STUDY OF SHIP-TO-BANK INTERACTION

KIM YOUNGKOOK

SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING

2015

CFD Study of Ship-To-Bank Interaction

KIM YOUNGKOOK

School of Mechanical & Aerospace Engineering

A thesis submitted to the Nanyang Technological University

in fulfillment of the requirement for the degree of

Master of Engineering

2015

Acknowledgement

I would like to express my gratitude to Associate Professor Eddie Ng Yin Kwee who is my supervisor for his excellent guidance and keen support. Prof. Ng was always patient and supportive for my problems and questions during the research period.

I also would like to extend my gratitude to NTU HPC support team, especially Soh Hwee Jin (Melvin) and Bong Zoebir who used to fix the problems and kindly updated the newer version of OpenFOAM in the HPC server so that I could use OpenFOAM 2.3.0 for my research.

Furthermore, I would like to appreciate to my wife, Cho Eunjin who supported my study in all.

Mar 2015

Kim Youngkook

Table of Contents

ACKNOWLEDGEMENT	2
TABLE OF CONTENTS	3
LIST OF FIGURES	5
LIST OF TABLES	8
ABSTRACT.....	9
NOMENCLATURE	10
CHAPTER 1 INTRODUCTION.....	15
1.1 BACKGROUND	15
1.2 OBJECTIVES	18
1.3 SCOPE OF RESEARCH	19
1.4 LITERATURE REVIEW.....	20
1.4.1 <i>Experimental approaches of ship-to-bank interaction</i>	<i>20</i>
1.4.2 <i>Numerical approaches of ship-to-bank interaction.....</i>	<i>22</i>
1.5 ORDER OF REPORT	26
CHAPTER 2 NUMERICAL METHODS.....	28
2.1 CFD THEORY	28
2.1.1 <i>Potential flow.....</i>	<i>28</i>
2.1.2 <i>RANS (Reynolds Averaged Navier-Stokes) equations</i>	<i>29</i>
2.1.3 <i>Finite Volume Method (FVM).....</i>	<i>33</i>
2.1.4 <i>Volume of fluid method.....</i>	<i>36</i>
2.2 CFD ALGORITHM	38
2.2.1 <i>Numerical discretisation and solution algorithm.....</i>	<i>38</i>
2.2.2 <i>Pressure corrections</i>	<i>44</i>
2.2.3 <i>Convergence and residuals.....</i>	<i>48</i>
2.2.4 <i>Courant number</i>	<i>50</i>
2.2.5 <i>Boundary conditions.....</i>	<i>52</i>
2.2.6 <i>Boundary layer treatment and y plus.....</i>	<i>54</i>
2.2.7 <i>Good practice of boundary layer treatment.....</i>	<i>58</i>
2.3 MATHEMATICAL MODEL OF SHIP-HANDLING SIMULATOR	61

CHAPTER 3	OPEN SOURCE CFD CODE	66
3.1	OPEN SOURCE CFD CODES AND OPENFOAM	66
3.2	OPENFOAM SOFTWARE STRUCTURES	68
3.2.1	<i>Architecture</i>	68
3.2.2	<i>Solvers</i>	69
3.2.3	<i>Numerical algorithm</i>	70
3.3	MULTI-PHASE SOLVER INTERFOAM AND LTSINTERFOAM	73
3.4	OPENFOAM EVALUATIONS IN OPEN WATER	76
3.4.1	<i>KVLCC2</i>	79
3.4.2	<i>KCS</i>	83
3.4.3	<i>Summary of the evaluations</i>	94
CHAPTER 4	ANALYSIS IN CONFINED WATER	96
4.1	COMPUTATIONAL SET-UP AND MESH	96
4.2	BOUNDARY CONDITIONS	105
CHAPTER 5	RESULTS AND DISCUSSION	109
5.1	COMPUTATION RESULTS	109
5.1.1	<i>Force and moment</i>	109
5.1.2	<i>Under keel space modelling of the case H</i>	116
5.1.3	<i>Pressure and velocity asymmetry</i>	122
5.1.4	<i>Results of the cases with propeller rotation</i>	128
5.1.5	<i>Comparison between single-phase and two-phase solutions</i>	129
5.1.6	<i>Free surface elevation</i>	130
5.2	DISCUSSION	136
CHAPTER 6	CONCLUSION	139
6.1	CONCLUSION	139
6.2	FUTURE WORK	141
APPENDIX	THE SOURCE CODES OF INTERFOAM	142
REFERENCES		152
PUBLICATION FROM THE PRESENT WORK		157

List of Figures

Figure 2-1: Finite volume discretisation [21].....	35
Figure 2-2: Explicit (upper) and implicit (upper) method [13]	43
Figure 2-3: Comparison of implicit and explicit solutions for same time step [7]	51
Figure 2-4: Velocity distribution near a solid wall [34]	57
Figure 2-5: Coordinate system of manoeuvring mathematical model	62
Figure 2-6: Six degrees of freedom of motion	63
Figure 2-7: The trajectory in ship-handling simulator with ship- to-bank interaction x and y axis: distance(m) from the initial point.....	65
Figure 3-1: OpenFOAM case structure.....	68
Figure 3-2: Profile and front view of KVLCC2.....	79
Figure 3-3: Convergence history of the resistance of KVLCC2	81
Figure 3-4: Comparison of wave elevation along the hull of KVLCC2 (Experiment result: Kim et al. [38])	82
Figure 3-5: Comparison of wave pattern of KVLCC2 of CFD (upper) and the experiment (Kim et al. [38]) (bottom)	83
Figure 3-6: Profile and front view of KCS.....	84
Figure 3-7: Mesh of KCS - (a) whole domain, (b) forward and (c) stern area	85
Figure 3-8: y^+ distributions on the hull.....	86
Figure 3-9: Convergence time history of the resistance (N) of KCS by (a) Gauss linear and (b) Least square gradient scheme	90
Figure 3-10: Comparison of wave elevation along the KCS hull (Experiment results: Kim et al. [38])	91

Figure 3-11: Comparison of wave pattern of KCS of OpenFOAM (upper) and the experiment (Kim et al., [38]) (bottom)	91
Figure 3-12: Wave pattern at the forward (a) and the stern (b)....	92
Figure 3-13: Wake distribution (axial velocity [m/s]) of KCS at (a) the propeller plane and (b) at the station no. 1 (left : experiment (Kim et al. [38]), right : OpenFOAM result)	94
Figure 4-1: The coordinate system.....	97
Figure 4-2: Schematic view of the bank arrangement (looking from the aft).....	97
Figure 4-3: Graphical view of the case C and the case H.....	100
Figure 4-4: Mesh of (a) whole domain and (b) near the hull (looking from the top), and (c) mesh on the hull surface of case H	103
Figure 5-1: Convergence history of (a) X and (b) Y forces (N), and (c) Z moment (Nm) from pressure of case F (Gradient scheme: Gauss linear, Max Co. reduced from 70s)	110
Figure 5-2: Convergence history of (a) X and (b) Y forces (N), and (c) Z moment (Nm) from pressure of case F (Gradient scheme: Least square)	111
Figure 5-3: Different meshing of case H - no layer (a), three layers (b), eight layers (c), fine mesh (d)	117
Figure 5-4: y^+ of the case H with different meshing no layer (a), three layers (b), eight layers (c), fine mesh (d)	118
Figure 5-5: Pressure [N] distribution in the water at the centreline of the case H with (b) three layers (c) eight layers	120
Figure 5-6: Pressure [N] distribution on the hull (looking from the bottom) of the case H with (b) three layers (c) eight layers	121

Figure 5-7: Pressure [N] distribution on the hull and bank surface of the cases F (a) and H (b) (looking from the top)	122
Figure 5-8: Dynamic pressure [N] distribution of OpenFOAM calculation of case A (a), F (b) and H (c) at the different sections (station 2.5, amidships and station 17.5)	124
Figure 5-9: Velocity magnitude (m/s) of the case F at (a) free surface, (b) 2/3 of draught from baseline, (c) 1/3 of draught from baseline and (d) baseline	125
Figure 5-10: Velocity magnitude (m/s) of the case H at (a) free surface, (b) 2/3 of draught from baseline, (c) 1/3 of draught from baseline and (d) baseline	126
Figure 5-11: Velocity magnitude (m/s) of the case I at (a) free surface, (b) 2/3 of draught from baseline, (c) 1/3 of draught from baseline and (d) baseline	127
Figure 5-12: The locations of the wave gauges 1, 2 and 3 of the cases (a) F, (b) I and (c) J	131
Figure 5-13: Wave elevation of the case F at (a) gauge 1, (b) gauge 2 and (c) gauge 3 (Lataire et al. [5])	132
Figure 5-14: Wave elevation of the case I at (a) gauge 1, (b) gauge 2 and (c) gauge 3 (Lataire et al. [5])	133
Figure 5-15: Wave elevation of the case J at (a) gauge 1, (b) gauge 2 and (c) gauge 3 (Lataire et al. [5])	134

List of Tables

Table 3-1: Principal dimensions of KVLCC2 model.....	79
Table 3-2: Grid convergence study (The experiment result: Kim et al. [38])	80
Table 3-3: Principal dimensions of KCS model.....	83
Table 3-4: The comparison of KCS resistance by different solvers (Experiment results: Kim et al. [38]).....	88
Table 4-1: Principal dimension of model ship	96
Table 4-2: The test cases without propeller thrust.....	99
Table 4-3: The test cases with propeller thrust	101
Table 4-4: Boundary conditions for confined water analysis of the single-phase solver	106
Table 4-5: Boundary conditions for confined water analysis of the multi-phase solver	107
Table 5-1: Comparison of the cases without thrust to the experiment results (Lataire et al. [5])	114
Table 5-2: Comparison between different meshing in the under keel space of the case H.....	120
Table 5-3: Comparison of the cases with thrust to the experiment results (Lataire et al. [4]).....	128
Table 5-4: The locations (m) of the wave gauges with referenrece to the centreline of the ship (Lataire et al. [5])	130

Abstract

Ship-to-bank interaction is a complex physical phenomenon that involves not only in the asymmetric pressure field near banks or channels but also shallow water effect. Traditionally many experimental studies were carried out in this field. As numerical method is getting popular, there were various computational approaches as well. In this study, CFD (Computational Fluid Dynamics) analysis is carried out to investigate ship-to-bank interaction. CFD analysis is also useful for finding hydrodynamic coefficients in manoeuvring mathematical model of the ship-handling simulator. For the numerical method, RANS based solvers with free surface capturing feature are used.

Flow around a container ship in confined water is investigated with the open source C++ toolbox, OpenFOAM. Computations with several bank arrangements and different settings are performed. The OpenFOAM results are also compared with experiment results for validation. Meshing, numerical scheme, numerical settings and calculation results are evaluated and discussed in this study.

Nomenclature

E	Integration constant of the roughness of the wall	-
Fn	Froude number ($Fn = U/\sqrt{gL_{pp}}$)	-
$F (F_i)$	Total force (in i coordinate)	N
g	Acceleration of gravity	m/s ²
H	H Operator (See Section 2.2.2)	-
I	Tensor matrix of inertia of body	kg·m ²
K	Roll moment	Nm
k	Sum of the mean kinetic energy of turbulence	m ² /s ²
L_{pp}	Length between perpendiculars	m
$M (M_i)$	Total moment (in i coordinate)	Nm
M	Pitch moment	Nm
$\dot{M}$	Mass flux	kg/(m ² ·s)
N	Yaw moment	Nm
p	Pressure	m ² /s ²
p_{rgh} (p_d)	Dynamic pressure	kg/(m·s ²)
S_f	Surface area of the cell face	m ²
t	time	s
U	Speed vector (u, v, w)	m/s

u_τ	Friction velocity	m/s
u^+	Non-dimensional velocity near the wall	-
w	Artificial velocity normal to the interface of volume fraction	m/s
X	Resistance / Longitudinal force	N
x_0	Origin of measurement	m
x_f	Distance from origin to the face centre	m
Y	Transverse force / Sway force	N
y_B	Distance from bank wall to the centreline of ship	m
y_p	Distance from the wall to the first grid point	m
y^+	Non-dimensional y_p	-
Z	Vertical force / Heave force	N
Γ	Diffusion coefficient	m ² /s
α	Scalar field of volume fraction	-
γ	Under-relaxation factor	-
ε	Rate of viscous dissipation of turbulent kinetic energy	m ² /s ³
κ	Surface curvature	1/m
κ	Von Karman's constant	-
μ	Dynamic viscosity	kg/(m·s)
ν	Kinematic viscosity	m ² /s

ρ	Density of fluid	kg/m^3
σ	Surface tension coefficient	N/m
τ_{ij}	Reynolds stress	$\text{kg}/(\text{m}\cdot\text{s}^2)$
τ_w	Wall shear stress	$\text{kg}/(\text{m}\cdot\text{s}^2)$
ϕ	Velocity potential	m^2/s
Ω	Angular velocity vector (p, q, r)	rad/s
ω	Turbulent frequency	$1/\text{s}$

Abbreviations

CFD	Computational Fluid Dynamics
CFL	Courant Friedrichs Lewy
CPU	Central Processing Unit
CV	Control Volume
DOF	Degree(s) of Freedom
EXP	Experiment
FDM	Finite Difference Method
FHR	Flanders Hydraulics Research
FVM	Finite Volume Method
HPMM	Horizontal Planar Motion Mechanism
KCS	KRISO Container Ship
KRISO	Korea Research Institute of Ship and Ocean engineering
KVLCC	KRISO Very Large Crude-oil Carrier
LES	Large Eddy Simulation
LNG	Liquefied Natural Gas
LTS	Local Time Stepping
MOERI	Maritime & Ocean Engineering Research Institute
MULES	Multi-dimensional Universal Limiter for Explicit Solution
ODE	Ordinary Differential Equation

OpenFOAM	Open Field Operation And Manipulation
OS	Operating System
PDE	Partial Differential Equation
PIMPLE	(Merged method of) PISO and SIMPLE
PISO	Pressure Implicit with Splitting of Operators
RANS	Reynolds Averaged Navier-Stokes (equation)
RPM	Revolution Per Minute
SIMPLE	Semi-Implicit Method for Pressure-Linked Equations
SST	Shear Stress Transport
STL	STereo-Lithography
TEU	Twenty-foot Equivalent Unit
VoF	Volume of Fluid

Chapter 1

Introduction

1.1 Background

When a ship navigates in the proximity of banks or harbours, the pressure field near the ship becomes asymmetric and ship experiences repelling or attracting forces. It makes hard for the ship to manoeuvre and can lead to an accident. In such cases, it is advised to travel at the middle of water to minimise any interaction from the bank or the harbour. However, as this is not always possible, it is important to understand the phenomenon to prevent any dangerous situation. Basically ship-to-bank interaction causes sway force and yaw moment on the ship when separation from the bank wall is small. Moreover, small under keel clearance in confined water makes the interaction even stronger due to the shallow water effect. In addition, there are several other factors that contribute such as propeller effect and bank shape. Because ships have been increased in size and length in recent years, the importance of prediction and understanding of ship-to-bank interaction are thus getting more significant.

Nowadays, ship-handling simulators play an important role in training captains and crew for safe operation and preventing from accidents. In the mathematical model of the ship-handling simulator, various forces and moments that arise from hull, rudder, propeller, wind, wave, bank effect, etc have to be carefully modelled for realistic simulations. The best way to obtain those forces and moments is to carry out model tests. Captive model test such as HPMM (Horizontal Planar Motion Mechanism) test enables measuring of each force and moment components. However, the test is costly and time consuming. Therefore, it is normal to use empirical methods that are based on the regression formulas from database to estimate these forces and moments. When the accuracy is concerned, CFD (Computational Fluid Dynamics) analysis could give better accuracy. Nowadays, it is not much to say that CFD technology became essential in marine industry. As CFD technology is developing, the accuracy and reliability of CFD analysis are getting higher. There were many studies to calculate hydrodynamic forces and moments of mathematical model of ship-handling simulator using CFD analysis.

As for ship-to-bank interaction, it is quite straightforward to use CFD calculations and model them into ship-handling simulator. However, careful approach is needed to calculate and model the

complex phenomenon due to the interaction with banks and shallow water effect. Recently, open source CFD codes have become popular due to their strong capability. They are totally free of charge to use and free to modify the source code. On the other hands, it is a challenge to evaluate the accuracy of the code despite the steep learning curve of open source codes. In this study, it is decided to use an open source CFD code to analyse ship-to-bank interaction from various bank arrangements and compare them to the experiment results.

1.2 Objectives

This study aims to predict ship-to-bank interaction with an open source CFD code. The numerical analysis of multiphase RANS solver will be evaluated so that the effect from free surface, turbulence and viscosity can be taken into account. In addition, the results will be compared with single-phase RANS solver. Making use of the open source CFD code, it is expected to know capability and limitation of the open source code. Moreover, it would be helpful to understand numerical algorithms of CFD codes.

By performing numerical simulation, ship-to-bank interaction with different bank arrangements will be investigated. This would also identify the benefit from CFD analysis compared to the experimental approach.

1.3 Scope of research

The open source CFD code, OpenFOAM will be used for numerical analysis. Among numerous solvers in OpenFOAM package, a VoF multiphase solver will be used to predict ship-to-bank interaction. The solver will be evaluated for open water cases before analysis is carried out in confined water. Resistance of a container ship and a large tanker ship will be calculated and compared to the experiment results.

The OpenFOAM solver will be used to calculate ship-to-bank interaction in confined water and the results will be compared to the experiment. CFD calculations will cover from the comparisons of forces and moment to the evaluation of meshing, modelling and wave elevation.

1.4 Literature Review

1.4.1 Experimental approaches of ship-to-bank interaction

From 1970s to these days, many experimental studies have been carried out to investigate ship-to-bank interaction and researchers developed empirical formulas that could predict ship-to-bank interaction. During 1970s, SSPA (Swedish maritime research centre) carried out various tank tests for different bank types and showed how the forces and moments vary over transient period. It was found that sway force and yaw moment vary in proportion to the square of ship speed. Using this test results, empirical formulas were also suggested from model test results (Norrbin [23]).

However, some studies that were carried out later showed that ship-to-bank effect occurs in a more complex manner. A continued study done about 30 years later in SSPA (Li et al [31]) proved that there is a point where the sway force is switched from suction to repulsive force when under keel clearance is extremely small, approximately at about 10% of water draught. Moreover, unlike the study in early times, it turns out that the sway force tends to increase in a higher order than two with respect to the ship speed when under keel clearance is very small.

Even in some cases, it became more than order of two at 40% of under keel clearance. At a sloped bank with 30 degrees angle, it tends to reduce force and moments at low speeds but bank effects increase at high speeds as compared to a vertical bank at a low under keel clearance. Moreover, a suction force arises from a rotating propeller compared to the cases without propeller according to the experiments. It was found the more propeller loading caused the larger contribution of suction force.

Expansive shallow water tank tests were carried out at FHR (Flanders Hydraulics Research) in Belgium (Vantorre et al. [21]) (Lataire et al. [4] [5]). More than 10,000 model tests of the bank effect were performed and many factors that influence the bank effect were investigated. Through the experiments, it was confirmed that there is a transition point from attraction sway force to repulsion force in shallow water. Small difference from the test results of SSPA is that it happened at slightly higher under keel clearance which is in range of 15% to 20%. Therefore, it is understood that the transition from attraction to repulsion force appears at the under keel clearance of 10~20% water draught. The test results also showed that propeller action increased the bank effect. Similarly to the results of SSPA, propeller action changed repulsion force into suction force in shallow water where under keel clearance is less than 35% of

water draught. These tests results are also developed into empirical formulas. Several sets of test data are made available for the bench marking purpose of simulation model (Lataire et al. [5]). In this study, these open test results are used for the comparison to the CFD calculations.

1.4.2 Numerical approaches of ship-to-bank interaction

As numerical technique was developing, there were many computational studies on the bank effect. In early times, numerical method based on potential theory was popular in marine industry.

In many studies, panel method was used to evaluate the bank effect. When the free surface condition is linearised by double model solution, it was good enough to show qualitative predictions only when the draught and water depth ratio is not too small, such that the ratio is greater than 1.5. (Q. Miao and J. Xia [32] and Lee and Lee [14]). When non-linear boundary condition is applied, panel method was also able to calculate ship-to-bank interaction in extreme shallow water with qualitative agreement (Park et al. [38]). With this non-linear potential method, even the transition point from suction force to repulsion force was found at 40% of the draught-water depth

ratio which is slightly higher than the experiment results reported by Li et al. [31] and Lataire et al. [4].

As computational capability is rapidly improving, RANS based finite volume methods were also getting popular. However, it was quite recent that RANS based method started to be used in this topic.

Lo et al. [2] carried out transient analysis on the bank effect. Trajectory of KCS container travelling near the bank was illustrated. Sway force and yaw moment with time variations at various velocities were calculated at about 30% of the draught-water depth ratio. However, there was no comparison with experiment result.

Wang et al. [10] studied viscous analysis around series 60 hull with SST $k-\omega$ turbulence model from 50% to 1000% of the water depth – draught ratio and showed good agreement with experiments in sway forces but over prediction with the very short distance to the bank wall.

Zou et al. [19] used validation and verification method including a grid convergence study to evaluate RANS based CFD prediction of ship-to-bank interaction. The bank effect between KVLCC hull and a sloped bank was analysed with an overlapping grid of

structured mesh and $k-\omega$ SST turbulence model. CFD calculations in the comparison with model tests were done at 50%, 35% and even 10% of the draught-water depth ratio. Furthermore, not only sway force and yaw moment but also rolling, sinkage and trim moment were calculated and compared with potential flow analysis. Because sinkage and trim moment did not show big difference between the potential method and CFD calculation result, free surface effect at slow speed is considered to be insignificant for sinkage and trim. In the study, slip boundary condition was applied on the bank wall instead of no slip condition. The rationale for this boundary condition is not clear and the author also stated that the boundary condition may not be the accurate. The bank effect analysis gave good agreement in tendencies of yaw and roll moments but with under prediction of sway force.

Mehdi et al. [22] carried out simulation of the viscous flow around LNG ship using RANS method with unstructured hybrid mesh and $k-\omega$ turbulence model. Yaw moment was compared with the experiment results at various velocities and distance from the bank wall but sway force results were not presented in their paper. Calculated yaw moments at different distance from the bank wall and 60% of the draught-water depth ratio were

compared to the tank test results and showed qualitative agreement with some under prediction.

From the various studies done in the past, it is still a big challenge to use CFD for ship-to-bank interaction analysis with extreme shallow water. Many studies failed to calculate extreme conditions such as 10% of under keel clearance or did not predict ship-to-bank correctly.

During the literature review about numerical studies of the bank effect, viscous analysis including free surface effect on the bank effect was not found so far. When a ship travels at slow speed, it is no doubt that free surface elevation is not so high and the importance from free surface might be relatively low. Furthermore, it was found that free surface also does not give significant contribution on the sinkage and the trim at low speed (Zou et al. [19]). Nevertheless, it is believed to be worthwhile to carry out viscous flow analysis with free surface for more accurate prediction of the flow near the proximity of banks. It is also expected to see the difference between the analysis with and without free surface.

1.5 Order of report

The structure of this report is as follows:

Chapter 1 introduces overview of this research and ship-to-bank interaction. It also includes literature review that covers ship-to-bank interaction study done both in experimental and numerical approaches.

Chapter 2 covers numerical methods of CFD including basic theory. It discusses numerical algorithms, boundary condition and boundary layer theory. Moreover, mathematical model of ship-handling simulator will be explained.

Chapter 3 introduces open source CFD codes and the solver used for this study. Moreover, the evaluation of the solver in open water is included.

Chapter 4 outlines numerical setup of CFD analysis in confined water and this covers computational setup and boundary conditions of CFD calculation.

Chapter 5 deals with the results of CFD calculations in confined water and discussions. Various comparisons and evaluations will be carried out.

Chapter 6 includes the summary of current work and discusses the future plan.

Appendix contains the main source codes of Open source CFD tool that was used for this study.

Lastly, References and Publication of this study are listed.

Chapter 2

Numerical Methods

2.1 CFD theory

2.1.1 Potential flow

In marine industry, traditional panel method based on potential theory as assumptions was popular and it is still used by many people due to its simplicity and quick computation in spite of inviscid and irrotational property of flow. High Reynolds number of many marine applications also enables potential flow remains useful and efficient for the area where viscous effect is not predominant. Nevertheless, potential flow is not applicable where the flow is dominated by viscous drag and turbulence, e.g. near the aft hull. Basic theory of potential flow will be briefly explained here.

In potential flow, velocity potential (ϕ) exists and governing equation is Laplace equation of continuity,

$$\nabla^2 \phi = 0 \quad (2.1)$$

This method must satisfy several boundary conditions.

Firstly, no flux on the body and the wall as such,

$$\nabla\phi \cdot \vec{\eta} = 0 \quad (2.2)$$

Secondly, fluid particle should not leave the free surface which is kinetic boundary condition. Thirdly, pressure on the free surface should be the same as atmospheric pressure, which is dynamic boundary condition.

Normally, the last two boundary conditions have non-linear terms and they are often linearised for simplification. One of the traditional methods is double model solution, which reflects hull form at waterline of calm water. With double model approximation, flow with rigid free surface can be calculated. Nowadays, non-linear potential method is also available to describe free surface elevation. Non-linear potential method is very useful and fast, hence, popular in minimising wave breaking resistance and ship hullform optimisations.

2.1.2 RANS (Reynolds Averaged Navier-Stokes) equations

When viscosity and turbulence cannot be ignored in the flow, potential method is no more accurate. The most popular method

is RANS equations with turbulence model and it gives better accuracy than potential flow that is irrotational and inviscid.

Basically turbulence has fluctuating property and needs very fine mesh with huge computing resources to solve it directly. RANS method takes average of fluctuating properties of turbulent flow and can save computational effort, memory and time. Therefore, RANS method is most appropriate for industrial cases which require huge computation mesh or domain.

The governing equations generally come from the continuity equation (2.3) and the momentum equations (2.4) ~ (2.5) of incompressible flow.

$$\text{div } \mathbf{u} = 0 \quad (2.3)$$

$$\frac{\partial u}{\partial t} + \text{div}(\mathbf{u}\mathbf{u}) = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \text{div}(\text{grad}(\mathbf{u})) \quad (2.4)$$

$$\frac{\partial v}{\partial t} + \text{div}(\mathbf{v}\mathbf{u}) = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \text{div}(\text{grad}(\mathbf{v})) \quad (2.5)$$

$$\frac{\partial w}{\partial t} + \text{div}(\mathbf{w}\mathbf{u}) = -\frac{1}{\rho} \frac{\partial p}{\partial z} + \nu \text{div}(\text{grad}(\mathbf{w})) \quad (2.6)$$

The first term on the left hand side of the momentum equations represents the time derivatives and is also called as unsteady

term. The second term is the convection term. On the right hand side, the first and the second terms are pressure term and viscous term, respectively.

RANS equations can be derived from the momentum equations by taking the time average.

Velocity vector and pressure vector in turbulent flow can be expressed as sum of steady mean component (U, V, W and P) and fluctuating component (u', v', w' and p').

$$u = U + u' \quad v = V + v' \quad w = W + w' \quad p = P + p' \quad (2.7)$$

As integration or differentiation of fluctuating flow is similar to time averaging operation, so the followings can be achieved.

$$\begin{aligned} \overline{\frac{\partial u}{\partial t}} &= \frac{\partial U}{\partial t} & \overline{\text{div}(\mathbf{u}\mathbf{u})} &= \text{div}(U\mathbf{U}) + \overline{\text{div}(\mathbf{u}'\mathbf{u}')} \\ -\frac{1}{\rho} \frac{\partial \overline{p}}{\partial x} &= -\frac{1}{\rho} \frac{\partial P}{\partial x} & \overline{\nu \text{div}(\text{grad}(\mathbf{u}))} &= \nu \text{div}(\text{grad}(U)) \end{aligned} \quad (2.8)$$

Therefore, momentum equations can be written as time averaged momentum equations. e.g. x-momentum equation becomes,

$$\frac{\partial U}{\partial t} + \text{div}(U\mathbf{U}) + \text{div}(\overline{u'\mathbf{u}'}) = -\frac{1}{\rho} \frac{\partial P}{\partial x} + \nu \text{div}(\text{grad}(U)) \quad (2.9)$$

Move the third term on the left hand side to the right hand side as it plays a role as turbulent stresses, then we have:

$$\frac{\partial U}{\partial t} + \text{div}(U\mathbf{U}) = -\frac{1}{\rho} \frac{\partial P}{\partial x} + \nu \text{div}(\text{grad}(U)) + \frac{1}{\rho} \left[\frac{\partial(-\rho \overline{u'^2})}{\partial x} + \frac{\partial(-\rho \overline{u'v'})}{\partial y} + \frac{\partial(-\rho \overline{u'w'})}{\partial z} \right] \quad (2.10)$$

$$\frac{\partial V}{\partial t} + \text{div}(V\mathbf{U}) = -\frac{1}{\rho} \frac{\partial P}{\partial y} + \nu \text{div}(\text{grad}(V)) + \frac{1}{\rho} \left[\frac{\partial(-\rho \overline{u'v'})}{\partial x} + \frac{\partial(-\rho \overline{v'^2})}{\partial y} + \frac{\partial(-\rho \overline{v'w'})}{\partial z} \right] \quad (2.11)$$

$$\frac{\partial W}{\partial t} + \text{div}(W\mathbf{U}) = -\frac{1}{\rho} \frac{\partial P}{\partial z} + \nu \text{div}(\text{grad}(W)) + \frac{1}{\rho} \left[\frac{\partial(-\rho \overline{u'w'})}{\partial x} + \frac{\partial(-\rho \overline{v'w'})}{\partial y} + \frac{\partial(-\rho \overline{w'^2})}{\partial z} \right] \quad (2.12)$$

These are called Reynolds Averaged Navier-Stoke (RANS) equations of incompressible flow.

The additional stress terms on the right hand side are named as Reynolds stresses, and they have three normal stresses

$$\tau_{xx} = -\rho \overline{u'^2} \quad \tau_{yy} = -\rho \overline{v'^2} \quad \tau_{zz} = -\rho \overline{w'^2} \quad (2.13)$$

and three shear stresses.

$$\tau_{xy} = \tau_{yx} = -\rho \overline{u'v'} \quad \tau_{xz} = \tau_{zx} = -\rho \overline{u'w'} \quad \tau_{yz} = \tau_{zy} = -\rho \overline{v'w'} \quad (2.14)$$

Turbulence fluctuations require extra terms in the time averaged or Reynolds averaged equations, and they have to be properly modelled with turbulence models. Therefore, RANS equations focus on the properties of the mean flow rather than computing fluctuating turbulent velocity directly. This can save numeral cost significantly.

2.1.3 Finite Volume Method (FVM)

Finite Volume Method (FVM) is one of the discretisation methods that complex equations can be solved with finite space using control volumes. Governing equations such as RANS equations include PDEs (Partial Differential Equations) and a non-linear term, therefore, it is very hard to solve them directly. Hence, it is important to apply appropriate discretization method to solve those equations for complex geometry. The discretisation process includes discretisation of the calculation domain and equations. The solution domain, which is generally physical space, is divided into a finite number of cells which are termed as control

volumes. Equation discretisation transforms partial differential equations into sets of linear algebraic equations and they are applied on the computational mesh. The equations should be also discretised for a finite number of time steps, if transient simulation is required.

The control volumes fill the computational domain without overlapping each other. In finite volume method, by principal, all the numerical properties are computed at the centroid of cells although they use the values on the vertices or faces during the computations. Cells are connected to each other by faces which are the boundary surfaces of each cell. The faces also form the boundaries of the computational space domain and called boundary faces. Others inside the domain are called internal faces. Figure 2-1 shows parameters in finite volume discretisation. Here f stands for the cell face, P the cell centre, N centre of the neighbouring cell, d is orthogonal direction through the cell face and S_f is the surface area of the cell face.

Figure 2-1: Finite volume discretisation [21]

The advantages of FVM are firstly, mass, momentum and energy are well conserved. Secondly, it can also handle unstructured mesh compared to traditional FDM (Finite Difference Method). The unstructured mesh is getting popular because it can handle various shapes more easily and efficiently. Thirdly, the solvers are efficient for iterative methods.

Disadvantage is the occurrence of false diffusion when numerical scheme is simple or mesh is not fine enough. For this reason, large computational resources should be applied to industrial cases, however, high computational cost of FVM is getting affordable due to the development of computer CPU and parallel computing nowadays.

2.1.4 Volume of fluid method

In many of marine CFD applications, prediction of free surface is very important. Traditionally, potential flow was popular for the free surface flow. Nowadays, viscous flow analysis with free surface is common as the computer performance is getting powerful.

There are several numerical modelling techniques to describe the interface of multiphase systems which deal with water and air together. As for free surface capturing techniques in finite volume method, VoF (Volume of Fluid) and level set method are most popular – even the two methods can be combined. VoF method that was introduced by Hirt and Nichols [8] is a free surface modelling technique applying volume fraction to convection schemes. It uses a scalar field to express the volume fraction (α) in whole domain, which represents water when it is 1 and air when it is 0, hence, the region where the volume fraction is between 0 and 1 becomes the free surface. It satisfies Eulerian material derivative.

$$\frac{D\alpha}{Dt} = 0 \quad (2.15)$$

, and this is

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot \mathbf{u} \alpha = 0 \quad (2.16)$$

VoF method is known to show high accuracy in the three dimensional applications. However, the mesh has to be very fine in order to capture the free surface properly as α is transitional from 0 to 1. Otherwise the interface will show smearing results. As a remedy, OpenFOAM uses modified governing equation for α by adding an artificial compression velocity which is a vector normal to the interface.

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot \mathbf{u} \alpha + \nabla \cdot \mathbf{w}(\alpha(1 - \alpha)) = 0 \quad (2.17)$$

As shown in the formula, the artificial term goes to zero if α goes to zero or one, which means that the artificial term does not affect to the result where the phase is completely water or air. The term only works within the interface zone to make it possible to describe sharp free surface by working to the normal direction to the interface. Magnitude of the compression, $\mathbf{w}$ (*cAlpha* in OpenFOAM) can be used to turn on or off the compression term.

2.2 CFD algorithm

In this chapter, brief algorithm and numerical techniques are described. All the CFD theory in this chapter may be limited to the followings:

- Finite Volume Method
- Incompressible Navier-Stokes equations with Mach number < 1
- General fluid flow simulations

2.2.1 Numerical discretisation and solution algorithm

The generic form of the transport equation for a fluid property ϕ is like the following:

$$\frac{\partial}{\partial t}(\rho\phi) + \text{div}(\rho u\phi) = \text{div}(\Gamma \text{grad } \phi) + S_\phi \quad (2.18)$$

Here, u is the velocity vector, ρ is the density, Γ is the diffusion coefficient and S_ϕ is a source term. For the simplicity, a one-dimensional case with a uniform grid is explained here. For the finite volume discretization, integrating this equation over the control volume (CV) and time,

$$\begin{aligned}
& \int_t^{t+\Delta t} \int_{CV} \frac{\partial}{\partial t} (\rho\phi) dV dt + \int_t^{t+\Delta t} \int_{CV} \text{div}(\rho u\phi) dV dt \\
& = \int_t^{t+\Delta t} \left[\int_{CV} \text{div}(\Gamma \text{grad } \phi) dV + \int_{CV} S_\phi dV \right] dt \quad (2.19)
\end{aligned}$$

Using the Gauss' theorem (2.20), the volume integrals of divergent terms can be expressed as surface integrals.

$$\int_V \nabla \phi dV = \oint_A \phi dS \quad (2.20)$$

Re-arranging time integration and volume integration, this can be written as follows:

$$\begin{aligned}
& \int_{CV} \int_t^{t+\Delta t} \frac{\partial}{\partial t} (\rho\phi) dt dV + \int_t^{t+\Delta t} \int_A n \cdot (\rho u\phi) dA dt \\
& = \int_t^{t+\Delta t} \left(\int_A n \cdot \left(\Gamma \frac{d\phi}{dx} \right) dA \right) dt + \int_t^{t+\Delta t} \int_{CV} S_\phi dV dt \quad (2.21)
\end{aligned}$$

Here, n is the unit vector normal to the face and A stands for the area of the cell face.

All the terms in the equation have to be approximated by various schemes in finite space and time to discretise the equations. The first term on the left hand side in the unsteady term becomes,

$$\int_{CV} \int_t^{t+\Delta t} \frac{\partial}{\partial t} (\rho\phi) dt dV = \int_{CV} \rho(\phi^n - \phi^0) dV = \rho(\phi_P^n - \phi_P^0) V_P \quad (2.22)$$

where 0 denotes current time step, n new value in the next time step, p cell point, V_P is the control volume and

$$\begin{aligned} \phi^n &= \phi(t + \Delta t) \\ \phi^0 &= \phi(t) \end{aligned} \quad (2.23)$$

The second term which is convection term and can be discretised as,

$$\int_t^{t+\Delta t} \int_A n \cdot (\rho u \phi) dA dt = \int_t^{t+\Delta t} (F\phi_E - F\phi_W) dt \quad (2.24)$$

F stands for the mass flux passes through the face and defined as $F = A(\rho u)$. E and W stand for the nodal point to the east and the west, respectively.

Depending on the time integrations whether defining $\int_{\Delta t} \phi_P dt \approx \phi_P^0$ or ϕ_P^n , this becomes,

$$[F\phi_E^n - F\phi_W^n]\Delta t \quad \text{or} \quad [F\phi_E^0 - F\phi_W^0]\Delta t \quad (2.25)$$

Similarly, the third term which is diffusion term becomes:

$$\int_t^{t+\Delta t} \left(\int_A n \cdot \left(\Gamma \frac{d\phi}{dx} \right) dA \right) dt = \int_t^{t+\Delta t} \left[\Gamma A \left(\frac{\phi_E - \phi_P}{\delta x} \right) - \Gamma A \left(\frac{\phi_P - \phi_W}{\delta x} \right) \right] dt \quad (2.26)$$

Then integrate this in time gives:

$$\begin{aligned} & \left[\Gamma A \left(\frac{\phi_E^0 - \phi_P^0}{\delta x} \right) - \Gamma A \left(\frac{\phi_P^0 - \phi_W^0}{\delta x} \right) \right] \Delta t \\ & \text{or} \\ & \left[\Gamma A \left(\frac{\phi_E^n - \phi_P^n}{\delta x} \right) - \Gamma A \left(\frac{\phi_P^n - \phi_W^n}{\delta x} \right) \right] \Delta t \end{aligned} \quad (2.27)$$

The fourth term is the source term and becomes:

$$\begin{aligned} \int_t^{t+\Delta t} \int_{CV} S_\phi dV dt &= \int_t^{t+\Delta t} \bar{S} V_P dt = \int_t^{t+\Delta t} (S_u V_P + S_P \phi_P V_P) dt \\ &= (S_u V_P + S_P \phi_P^0 V_P) \Delta t \quad \text{or} \quad (S_u V_P + S_P \phi_P^n V_P) \Delta t \end{aligned} \quad (2.28)$$

Finally, discretization in time can express the equations in two different ways depending on how we choose unknowns from the different time step.

The convection and diffusion terms are simplified by taking surface integrals over the cell and by using Gauss theorem.

These surface integrals can be expressed by the summation of the mass flux through the faces to describe general form.

$$\int_V \nabla \phi dV = \sum_f \left(\int_f \phi dS \right) \approx \sum_f A \phi \quad (2.29)$$

First method is explicit approach which can be written as:

$$\frac{\rho(\phi_P^n - \phi_P^0)V_P}{\Delta t} + \sum_f F \phi_f^0 = \sum_f \Gamma A \text{grad } \phi_f^0 + S_u V_P + S_P \phi_P^0 V_P \quad (2.30)$$

Another method is implicit approach that involves more complex calculations:

$$\frac{\rho(\phi_P^n - \phi_P^0)V_P}{\Delta t} + \sum_f F \phi_f^n = \sum_f \Gamma A \text{grad } \phi_f^n + S_u V_P + S_P \phi_P^n V_P \quad (2.31)$$

Here, A represents the area of the cell face and V the control volume. The surface integrals can be expressed as face summation where f stands for cell faces. For the surface integrals, the implicit approach involves neighbouring faces in the new time step. This makes calculations more complex. In the explicit form, there is only one unknown nodal point which is straightforward, while the implicit form has three unknown points.

Then, what is the reason for using the implicit method in spite of the complex algorithm? The implicit approach allows larger time step than the explicit method. More details will be discussed in Ch 2.2.4.

Figure 2-2 illustrates how grid points are deployed in two methods for time marching solution.

Figure 2-2: Explicit (upper) and implicit (upper) method [13]

Therefore, the momentum equations are discretised into linear algebraic equations that normally becomes,

$$a_P \phi_P = \sum a_{nb} \phi_{nb} + b \quad (2.32)$$

a is a function of velocity and $_{nb}$ represents neighbouring cells.

Then it can be arranged into the form of

$$[A] [\phi] = [B] \quad (2.33)$$

$[A]$ is a square matrix and $[B]$ is the source vector in column. Exact linear algebraic equations and forms in the matrix depend on the governing equation and the numerical scheme used for the discretisation. Applying appropriate matrix solving algorithm, the matrix equation can be solved to obtain the solution.

2.2.2 Pressure corrections

For incompressible flow, the continuity equation itself does not contribute to the pressure solution, as density is constant. Thus, the pressure gradient is unknown and has to be solved separately while solving the momentum equations.

Adding the pressure term in the momentum equation, the equation (2.32) can be written as,

$$a_p U_p = H(U) - \nabla p \quad (2.34)$$

where $H(U)$ term contains all the terms for the neighbours and all the source parts of transient terms. Writing this for the velocity U becomes:

$$U_p = \frac{H(U)}{a_p} - \frac{1}{a_p} \nabla p \quad (2.35)$$

The continuity equation is also discretised as,

$$\nabla \cdot U = \sum_f A \cdot U_f = 0 \quad (2.36)$$

where A denotes area of the cell face.

The velocity of the equation (2.35) can be expressed on the cell face:

$$U_f = \left(\frac{H(U)}{a_p} \right)_f - \left(\frac{1}{a_p} \right)_f (\nabla p)_f \quad (2.37)$$

Substitute the equation (2.37) into discretised continuity equation (2.36):

$$\nabla \cdot \left(\frac{1}{a_p} \nabla p \right) = \nabla \cdot \left(\frac{H(U)}{a_p} \right) = \sum_f A \cdot \left(\frac{H(U)}{a_p} \right)_f = 0 \quad (2.38)$$

The equation (2.38) has to be satisfied to guarantee the conservation of the face fluxes.

By expressing the gradient into the summation of mass flux at the cell face, the final form of Navier-Stokes equation becomes,

$$a_P U_P = H(U) - \sum_f A \cdot (p)_f \quad (2.39)$$

The face flux F is the velocity at the face (2.36) multiplied by the area of the cell face:

$$F = S \cdot U_f = S \cdot \left[\left(\frac{H(U)}{a_P} \right)_f - \left(\frac{1}{a_P} \right)_f (\nabla p)_f \right] \quad (2.40)$$

Normally pressure is solved by the pressure-velocity coupling algorithms such as SIMPLE (Patankar [29]) and PISO (Issa [12]).

They are iterative methods of guess and correction.

SIMPLE (Semi-Implicit Method for Pressure Linked Equations) process is in the following sequences:

- 1) Discretised momentum equations are solved with the initial guessed pressure which is denoted by p^* to find velocity components (u^* , v^* and w^*). This is a momentum predictor step.
- 2) Discretised equations including the $H(U)$ operator are used to solve pressure correction equation.

- 3) Then the pressure correction is defined from the difference between the guessed pressure and correct pressure field. The corrected pressure (p) is obtained by the guessed pressure (p^*) and pressure correction (p_{corr}).

$$p = p^* + p_{\text{corr}} \quad (2.41)$$

The same applies to the velocity corrections from the pressure correction:

$$u = u^* + u_{\text{corr}} \quad (2.42)$$

$$v = v^* + v_{\text{corr}}$$

$$w = w^* + w_{\text{corr}}$$

- 4) Solve other scalar equations (e.g. turbulence properties)
- 5) Repeat this process until the velocity field satisfies the continuity equation.

PISO (Pressure Implicit with Splitting of Operators) algorithm is also similar to SIMPLE but has one more corrector step. First, in PISO, the same steps with SIMPLE are performed and there is one more correction for pressure and velocities to be carried out. These steps are repeated until it reaches convergence. In spite of the additional step for pressure correction, PISO is known to be efficient and fast. However, this also depends on the case. It is not clear which method is the best for general flow case.

For this iterative method, under-relaxation factors help stability of the calculations. Especially for the pressure correction

methods, the equations are prone to diverge easily during the iterations. Generally, when a new value of variable (ϕ) is calculated, under-relaxation is applied as follows:

$$\phi_P^{new} = \phi_P^{old} + \gamma(\phi_P^{predicted} - \phi_P^{old}) \quad (2.43)$$

The under-relaxation γ is normally applied between 0 and 1 so that the iterative solution can be changed slowly and it increases stability of the calculation. Basically it does not affect the converged values but helps reduce the oscillation. However, this may slow down the convergence. Under-relaxation factors for all the variables are case dependent and there is no general recommendation.

If the factor equals to 1, it means no relaxation applied. Predicted value will be used without under-relaxation.

Sometimes the factor can be more than 1 to accelerate the convergence. In this case, the stability should be ensured during the calculations.

2.2.3 Convergence and residuals

The discretised governing equations are solved by iterative algorithm because the initial solution will not satisfy them. The left hand side and the right hand side of the matrix form of linear algebraic equations will not be equal with the initial values in

the first iteration that are assumed from the boundary conditions. The difference between the left and the right hand side is called momentum residual. During the iterations, the absolute values of residuals should be reduced so that the left and right hand sides of the equation can get closer. By this guessing and correction procedure, the variables get nearer to the actual solutions. Therefore, the residuals are important measures on how the solution satisfies the governing equations and can be used to estimate the error. In other words, the solution would not be correct and diverge if the absolute values of residuals are becomes bigger during the iterations.

There are many things that can affect the convergence of the solution. They are relaxation factors, quality of the mesh, numerical schemes, diagonal dominance of the matrix, etc. Wrong selection of mesh, numerical settings or boundary conditions would cause the solution to diverge. Nevertheless, this is always case dependent and there are no optimum settings. In some cases, convergence should be judged by the force values rather than the momentum residuals. The oscillation of force values will be stabilised or will not change any more when the solution converges. In this case, however, monitoring the residuals still can give a good indication of whether the solution diverges or not.

2.2.4 Courant number

The Courant number is a stability criterion of computation and defined as follows.

$$Co = \frac{\delta t |U|}{\delta x} \quad (2.44)$$

δt is the time step, U is the magnitude of velocity at the cell and δx is the size of the cell in the direction of the velocity.

In explicit approach, the Courant number must be less than one to ensure the numerical stability while running the computation. This is also called as CFL (Courant-Friedrichs-Lewy) condition. As can be seen in the formula, the time step of transient calculations has to be small enough if the velocity is high or the cell size in the direction of the velocity is very small in order to make sure CFL condition is met. CFL condition must be ensured in all the cells throughout the computation domain. Especially Courant number has to be far smaller at the interface region of multi-phase flow for the stability of the solution.

However, the Courant number can easily exceed one in the implicit approach. In some of the commercial codes, the Courant number is allowed even beyond hundreds. This means much larger time step can be used in the implicit approach. Although

implicit method allows larger time step, an appropriate Courant number depends on the type of the calculations and should be selected carefully for the accurate solution. Figure 2-3 is an example that shows the comparison between the implicit and explicit method. When the equal time step is used for the implicit and explicit algorithms, the solution of implicit approach gives more stable and accurate results.

Figure 2-3: Comparison of implicit and explicit solutions for same time step [7]

2.2.5 Boundary conditions

Boundary conditions are essential to find the unknowns in the governing equations. Wrong or poor boundary conditions will lead the computation to wrong solution or divergence. There are several types of boundary conditions and they are mainly divided into fixed value and gradient types. The fixed value type boundary conditions set the specific value of ϕ on the cell face, and this is called Dirichlet type. The fixed gradient boundary condition applies gradient of ϕ to the normal direction pointing outward on the boundary faces and is known as Von Neumann type.

For the general fluid flow simulation, boundary conditions are normally set for inlets, outlets, walls and/or symmetry planes.

Inlet: A fixed value is set for the velocity and zero gradient for the pressure at the inlet boundary.

Outlet: It is important for an outlet boundary to be set in such a way that it satisfies the overall mass conservation. Therefore, the outlet boundary condition is set in conjunction with the inlet boundary condition and velocity boundary condition.

For the mass conservation, the total mass flux which are going out (M_{out}) and coming in (M_{in}) must be the equal. In order to achieve this, the flux ratio (M_{in}/M_{out}) is multiplied to outlet velocity to ensure the continuity and subsequently multiplied to the East neighboring velocities (Versteeg and Malalasekera, 2007). Therefore, normally zero gradient is set for the velocity and fixed value for the pressure at the outlet boundary.

It is also important to locate the outlet boundary far enough from the region where the flow is expected to be developed so that the flow gradients become zero in the direction normal to the outlet boundary.

Wall: The wall boundary conditions are used on the solid objects and on the walls. On the wall boundaries, no-slip condition has to be met so that the velocity on the wall becomes the equal to that of the wall. For instance, velocity should be zero on the wall of the static object. When the object is moving, many CFD codes provide moving wall velocity boundary condition to be able to apply no-slip wall. On the wall boundaries, the mesh should be defined carefully to resolve the viscous flow characteristics in the boundary layer. More details for this will be discussed in the next sections.

Slip wall: When this boundary condition is set, the flow property normal to the wall becomes zero and tangential components zero gradient. This is normally applied to the wall where flow is just passing along.

Symmetry plane: The boundary conditions for symmetry planes give no flow gradient or flux across this boundary. This can be applied to save computational cost where the geometry and the flow field are exactly symmetric across the boundary.

2.2.6 Boundary layer treatment and y plus

y plus (y^+) is a non-dimensionalised distance by the friction velocity and kinematic viscosity and represents the distance from the solid wall to the first grid point from the wall.

$$y^+ = \frac{\Delta y_p}{\nu} \sqrt{\frac{\tau_w}{\rho}} = \frac{\Delta y_p \cdot u_\tau}{\nu} \quad (2.45)$$

, where Δy_p is distance from the wall to the first grid point, τ_w is wall shear stress, u_τ is friction velocity, ρ is the density of fluid and ν is the velocity component perpendicular to the wall.

It is known that the flow is laminar near the wall and is transitional from laminar to turbulence at about $y^+ = 11.225$. In

order to calculate viscous flow properties in the boundary layer correctly, mesh needs to be carefully designed to locate y^+ in the proper range.

On the surface of a wall, no slip condition is applicable which means the fluid on the wall has the same velocity with the wall itself. For instance, if a wall is stationary, the fluid on the wall is also stationary. At the very next to the wall, there is a very thin layer where viscous effect dominates and the flow is laminar. It is named viscous sub-layer or laminar sub-layer. It is also called as linear sub-layer because of the linear relation between y^+ and u^+ . u^+ is non-dimensional velocity with friction velocity, u_τ .

$$u^+ = \frac{U}{u_\tau} \quad (2.46)$$

$$u_\tau = \sqrt{\tau_w/\rho} \quad (2.47)$$

, where τ_w is wall shear stress

In this region, the shear stress is approximately equal to the wall shear stress and $y^+ = u^+$ is satisfied. When wall bounded effects are highly significant in a flow, viscous sub-layer has to be resolved, hence, y^+ must be in order of 1 to have at least one point in the viscous sub-layer. For a good result, y^+ should be less than 1 in all the area. In other words, mesh near the wall

has to be well refined to resolve the whole boundary layer, and turbulence model also needs to be of low Reynolds number model such as SST (Shear Stress Transport) type that can resolve viscous sub-layer properly. Epsilon based turbulence models are not designed for the viscous sub-layer. Generally, cases with low Reynolds number should have mesh with the first grid point from the wall within this sub-layer.

Outside this sub-layer, there is a turbulence region where u^+ and y^+ vary in a logarithmic relation as such,

$$u^+ = \frac{1}{\kappa} \ln(Ey^+) \quad (2.48)$$

where κ is von Karman's constant and E is an integration constant from the roughness of the wall.

This is the region where $30 < y^+ < 500$ and called log-law layer. In this region, the shear stress is no more constant and varies with the distance from the wall. When wall bounded effects are not priority and Reynolds number is high, the first grid point can be put in this log-law layer region to apply wall functions. Wall function is a semi-empirical method that can predict the velocity profile near-wall region. In other words, y^+ has to be in the range of 30 to 500 in order to use wall function. However, it is

important to make sure that the first point is not placed outside of the boundary layer when using wall function. By using wall functions, mesh resolution can be reduced, hence, saving computational costs and efforts.

If y^+ is between 5 and 30, it is a region where laminar flow transits to turbulent flow and calculated shear stress is neither constant nor logarithmic. This region is called buffer layer and has to be avoided when generating mesh with certain y^+ value. Therefore, the user should determine whether viscous sub-layer would be resolved or a wall function would be used to calculate near-wall region for the grid design near the wall. Figure 2-4 shows velocity distribution near a solid wall with respect to $\log y^+$.

Figure 2-4: Velocity distribution near a solid wall [34]

2.2.7 Good practice of boundary layer treatment

As discussed, for the grid design, special care should be taken at the boundary layer region in which flow properties vary rapidly. Wrong design of boundary mesh layer will give wrong calculation results. When one designs the grid to target certain range of y^+ value, valid y^+ value can be known only after calculation is carried out with enough iterations. It is because y^+ value is different between the initial field and after flow is developed sufficiently. Therefore, it is recommended to monitor y^+ values during the computations.

Normally marine CFD applications are high Reynolds number flow and wall functions can be used to calculate the near wall region. In this case, y^+ value is recommended to be below 300 and above 30 when flow is developed.

If the user wants to resolve whole boundary layer, enough number of mesh layers (i.e. at least 20 layers) needs to be designed in this laminar region without sudden increase in the size of mesh between the layers. Several marine CFD codes do not use wall function but resolve whole boundary layer from the wall where the velocity is set to zero although the marine applications has high Reynolds number. In order to achieve this,

it is recommended that the mesh near the wall needs to be very fine with $y^+ \leq 1$ and turbulence model must be valid in all regions of the boundary layer. This is known to be very accurate in calculating ship resistance (Larsson et al. [18])

This practice is also important for full scale analysis. When it comes to the experiment, it is well known that there is a big discrepancy of Reynolds number between model scale and full scale when scaling down with geometrical similarity. This causes relatively thicker boundary layer in the model scale compared to full scale due to the different Reynolds number and brings forth different results. For this reason, there were many efforts to carry out full scale CFD analysis. Traditionally CFD technology was limited to the model scale not only because of limitation of computational resources but also because the first cell from the hull surface gets to have very large aspect ratio in full scale. In order to have y^+ within 1, too large aspect ratio of grid causes solvers to be broken down easily. Nowadays, many commercial solvers are able to handle high aspect ratio. Another problem comes from full scale analysis is when turbulence model is used with wall function. General wall functions are mainly developed and validated with the relatively small scale but not robust enough for full scale ships with slender hullform where Reynolds number easily goes above 10^9 . However, improvement of CFD

techniques as well as increasing hardware capability enables full scale analysis nowadays (Larsson et al. [18]).

2.3 Mathematical model of ship-handling simulator

This chapter will cover mathematical model of ship-handling simulator and discuss how to make use of CFD results into ship-handling simulator.

According to the Euler's equations of rigid body dynamics, the motion of a rigid body can be written as follows,

$$\vec{F} = m\dot{U} + m\Omega \times U \quad (2.49)$$

$$\vec{M} = I\dot{\Omega} + \Omega \times I\Omega \quad (2.50)$$

Here, ship's centre of gravity is assumed to be at the midship.

Figure 2-5 shows coordinate system of mathematical model. Considering three dimensions, force vector and moment vector are,

$$\vec{F} = (X, Y, Z), \quad \vec{M} = (K, M, N)$$

Here, m is mass of the ship, I is moment of inertia of the body, linear speed vector is $U(u, v, w)$ and angular velocity vector is $\Omega(p, q, r)$.

Figure 2-5: Coordinate system of manoeuvring mathematical model

Substituting all these vectors into the equation (2.50),

$$I_{xx}\dot{p} - (I_{zz} - I_{yy})qr + I_{xz}(\dot{r} - qp) + I_{yz}(r^2 - q^2) + I_{xy}(pr + \dot{q}) = K \quad (2.51)$$

$$I_{yy}\dot{q} - (I_{xx} - I_{zz})rp + I_{xy}(\dot{p} - rq) - I_{xz}(p^2 - r^2) + I_{yz}(qp + \dot{r}) = M \quad (2.52)$$

$$I_{zz}\dot{r} - (I_{yy} - I_{xx})pq - I_{yz}(\dot{q} - pr) + I_{xy}(q^2 - p^2) - I_{xz}(rq + \dot{p}) = N \quad (2.53)$$

Assuming to be $(I_{xy} = I_{yz} = I_{xz} = 0)$ from the tensor matrix,

$$I = \begin{pmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{pmatrix}$$

Hence, the equations (2.49) and (2.50) can be expanded as follows:

$$m(\dot{u} - vr + wq) = X \quad (2.54)$$

$$m(\dot{v} - wp + ur) = Y \quad (2.55)$$

$$m(\dot{w} - uq - vp) = Z \quad (2.56)$$

$$I_{xx}\dot{p} + (I_{zz} - I_{yy})qr = K \quad (2.57)$$

$$I_{yy}\dot{q} + (I_{xx} - I_{zz})rp = M \quad (2.58)$$

$$I_{zz}\dot{r} + (I_{yy} - I_{xx})pq = N \quad (2.59)$$

These are the motion equations for 6 degrees of freedom (DOF) of motion in the general ship-handling simulator when the centre of mass is at the midship. Figure 2-6 illustrates 6 DOF of motions.

Figure 2-6: Six degrees of freedom of motion

The right hand side represents forces (X, Y, Z) and moments (K, M, N) in 6 DOF of motions which are surge, sway, heave, roll, pitch and yaw, respectively. Each of them represents the sum of

all the external forces and moments from hull, propeller, rudder, aerodynamic, wave, mooring lines, etc.

Likewise, the forces and moments from ship-to-bank interaction can be added into the additional terms in the right hand side, too. For this reason, sway force and yaw moment from the bank effect can be non-dimensionalised and modelled for different water depth and separation from the bank wall.

$$C_{BY} = \frac{F_Y}{mgFn^2} \quad : \quad \begin{array}{l} \text{non-dimensionalised sway force from the} \\ \text{bank effect} \end{array} \quad (2.60)$$

$$C_{BN} = \frac{M_Z}{mgLFn^2} \quad : \quad \begin{array}{l} \text{non-dimensionalised yaw moment from the} \\ \text{bank effect} \end{array} \quad (2.61)$$

Even if CFD calculations are carried out for steady state without mesh motion, the results can be easily implemented in the ship-handling simulator for time variant simulations. When the ship moves near the bank during the real time simulation, modelled sway force and yaw moment of the ship at the specific separation from the bank and water depth can be added in the right hand side of motion equations of simulator (Equation 2.55 and 2.59) as an additional force and a moment components. As the ship is repelled from or attracted to the bank due to the added force and moment, force and moment can be updated by interpolating at a new distance that varies at each time step. By this, simulations

for the irregular bank and the bathymetry will also be possible, hence, training of crew in confined water can be more realistic.

Figure 2-7 shows a simulation result of a vertical bank case in ship-handling simulator after modelling the forces and moments at various distances but at a fixed water depth. Kongsberg system [15] has been used for the testing of ship-handling simulator.

Figure 2-7: The trajectory in ship-handling simulator with ship-to-bank interaction

x and y axis: distance(m) from the initial point

Chapter 3

Open source CFD Code

3.1 Open Source CFD Codes and OpenFOAM

CFD technology has become essential in many fields including marine industry. There are popular commercial CFD codes with huge user groups due to the convenient and strong functions. Meanwhile, open source CFD codes are also improving in functions and it is also becoming popular in many applications these days. They are free to use and even free for modification under the GNU General Public License. The users need not worry about the license and can run it anywhere. They also have good pre- and post- processors that are also free to use, and parallel computing interface is useful to speed up the computations. Although they are still not as convenient as commercial codes and it takes some time to leverage, it is decided to use an open source CFD code to build its capability in this study.

Among many open source CFD codes, OpenFOAM has been chosen for this study due to the massive user group and numerous tutorials.

OpenFOAM (Open Field Operation And Manipulation) is a C++ library toolbox based on finite volume method. It is first released by Prof. Hrvoje Jasak and Mr. Henry G. Weller in 2004 after several studies and projects. Various solvers with different governing equations and numerical schemes can be used to solve different problems. The user can even create a new solver because the libraries in OpenFOAM toolbox are designed to solve ODE (ordinary differential equations) or PDE (partial differential equations). The latest version 2.3.0 was released in Feb 2014. There is another group called OpenFOAM extend-project who develops modified OpenFOAM solvers by themselves and discusses related issues. There are many seminars and conferences in many countries to discuss about the applications. The capability and applications are known to be comparable to commercial codes. Initially OpenFOAM 2.1.0 was used in this study, and then changed to the version 2.3.0 in the middle of the study when it was released.

3.2 OpenFOAM software structures

3.2.1 Architecture

Basically, an OpenFOAM case is composed of various files in several folders. Figure 3-1 shows the structure of a general fluid flow case of OpenFOAM by the folders and the major input files.

Figure 3-1: OpenFOAM case structure

The 'system' folder has numerical schemes and their options. Users can set discretisation schemes, pressure solvers, matrix

solver, time step, the numerical settings, etc. to be used for governing equations.

The 'constant' folder has mesh and physical properties such as density, kinematic viscosity, acceleration of gravity, turbulence model information. The mesh information are saved individually as points, faces, cell, boundary, etc. in the sub-folder, 'PolyMesh'.

The time directories are the folders where all the variables are stored. Initial time folder, normally '0', has initial and boundary conditions in it. As computation is run, folders for other time steps with output files are saved by the solver according to the time step set by the user.

3.2.2 Solvers

There are several solver groups in the package such as incompressible flow, compressible flow, multiphase flow, combustion, heat transfer and buoyancy-driven flow, electromagnetics, stress analysis of solids, etc. Under each group, there are several solvers too. This study will focus on the multiphase solvers to include free surface effect. Single-phase solver has also been used to compare with multi-phase solutions.

3.2.3 Numerical algorithm

OpenFOAM is a toolbox that can solve partial differential equations. For example, the follow momentum equation,

$$\frac{\partial \rho U}{\partial t} + \nabla \cdot \phi U - \nabla \cdot \mu \nabla U = -\nabla p$$

is expressed as:

```
solve
(
    fvm::ddt(rho, U)
    + fvm::div(phi, U)
    - fvm::laplacian(mu, U)
    ==
    - fvc::grad(p)
);
```

Here ‘fvm’ and ‘fvc’ are the classes to discretise differential operators such as ddt ($\partial / \partial t$), div ($\nabla \cdot$) and laplacian (∇^2). The class ‘fvm’ involves implicit derivatives to generate matrix. ‘fvc’ stands for Finite Volume Calculus and calculates explicit operations to return geometric field data.

OpenFOAM gives options for each numerical scheme. For the gradient scheme, 2nd order schemes (Gauss linear and Least square) are used in this study to maintain the accuracy of the solution. Gradient scheme is the numerical scheme that interpolates the value to the cell faces since the variables are basically stored at the cell centre. Gauss linear gradient scheme means Gaussian integration to interpolate the value from the cell centre to the face centre and the interpolation is done linearly.

A limiter can be used for gradient calculations especially when high order scheme is applied. There are cell type and face type limiters. Basically cell type limiter is applied to the gradient between the cell centres by clipping the gradient equally, whereas face limiter applies limiting gradient on the face of the cells. MD stands for multi-dimensional which clips the flux in the direction normal to the cell faces. In the simulation of this study, *cellMDLimited* was applied to maintain certain accuracy of numerical scheme. Available limiters are followings.

- *cellMDLimited* : Least diffusive and more accuracy
- *cellLimited* : Less dissipative
- *faceMDLimited* : More dissipative
- *faceLimited* : Most diffusive but robust scheme

Another pressure-velocity solving algorithm, PIMPLE is provided by OpenFOAM. It is a hybrid method of PISO (Pressure Implicit with Splitting of Operators) and SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm. It is, in fact, not different from the PISO procedure except for the outer correctors. If the outer corrector parameter is set to be 1 or not set, it is simply same as the PISO algorithm. Outer iteration option is known to give more stable and accurate calculations.

3.3 Multi-phase solver interFOAM and LTSInterFOAM

In many marine applications of CFD, free surface cannot be neglected, hence, multi-phase solvers are very useful for interface capturing. For this reason, interFoam and LTSInterFoam were chosen for this study. interFoam is a solver for transient solutions of two incompressible, isothermal and immiscible fluids using a VoF (volume of fluid) phase-fraction. LTSInterFoam adopts local time stepping (LTS) that uses different time steps for each cell depending on the different velocity and cell size, and enables to reach steady state quickly. Since this study will focus on capturing forces and moments at steady state, LTSInterFoam is useful to obtain them in relatively shorter time. The governing equations used in the solver are the continuity equation (3.1) and momentum equation (3.2) for incompressible flow:

$$\nabla \cdot \mathbf{U} = 0 \quad (3.1)$$

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = -\nabla p^* - \mathbf{g} \cdot h \nabla \rho + \mathbf{F}_S + \nabla \cdot (\mu \nabla \mathbf{U}) + (\nabla \mathbf{U}) \cdot \nabla \mu \quad (3.2)$$

where μ is dynamic viscosity, p^* is the dynamic pressure, $\mathbf{g}$ is acceleration of gravity, ρ is density, $\mathbf{U}$ is velocity vector, h is hydrostatic height of the fluid, hence, first term in the right hand

side represents dynamic pressure and second term hydrostatic pressure. F_S is the source momentum of surface tension, which is expressed as;

$$F_S = \sigma \kappa \nabla \alpha \quad (3.3)$$

, where σ represents the surface tension coefficient, κ is the surface curvature and α is volume fraction which is bounded in $0 \leq \alpha \leq 1$ (0 for air and 1 for water). Surface curvature can be modelled as the boundary condition of wall surfaces setting contact angle. Surface tension can be neglected in the simulation in which huge mass of fluid develops the flow fast in large domain and the effect from surface tension is not important. Therefore, in this simulation, surface tension between the interface and walls are neglected.

To make sure scalar fields are bounded in multi-phase flow, OpenFOAM uses MULES (Multi-dimensional Universal Limiter for Explicit Solution) algorithm. This handles free surface developments by limiting the face fluxes multi-directionally for the boundedness of hyperbolic equations. Although Euler implicit unsteady method is used in OpenFOAM, this explicit limiter requires Courant number below 1. In OpenFOAM 2.3.0, semi-implicit version of MULES has been introduced. This

executes an implicit predictor first and followed by explicit correction step. This allows quite bigger time step with larger Courant number and reduces computational time from older versions of OpenFOAM.

The main source codes of interFoam are included in the Appendix. They include most of the CFD algorithm and techniques discussed in this study.

3.4 OpenFOAM evaluations in open water

Starting from the tutorial on open water, the mesh, numerical settings, boundary conditions, etc. are tested to determine the most suitable settings.

Prior to studying the bank effect using OpenFOAM, it would be beneficial to evaluate the solver with open water analysis to know its accuracy and find the robust computational settings. Compared to flow in confined water, open water analysis is relatively simpler, therefore, it is a good start before moving on to the confined water analysis that involves with shallow water effect as well as interaction with banks. KVLCC2 (KRISO Very Large Crude-oil Carrier 2) and KCS (KRISO Container Ship) that are popular for the benchmarking are chosen to compare with CFD analysis. The test results of these two ships were released by MOERI (Maritime & Ocean Engineering Research Institute), formerly known as KRISO (Korea Research Institute Ship and Ocean engineering) for validation purpose.

Considering the numerical cost and the accuracy, the minimum size of the mesh near the hull was determined. The mesh was created by *snappyHexMesh* combining several blocks to make graded mesh efficiently. As the ship and the domain are

symmetric, half hull was modelled and calculated. To save the computational cost, the mesh is created to be finer near the hull and getting coarse towards the outside of the domain. Turbulence model used for simulation was $k-\omega$ SST with wall function and the average y^+ was kept below 200 on the most of the surface area.

The boundary conditions are set as fixed value at the velocity inlet and pressure outlet, and zero gradient at the velocity outlet and pressure inlet. No slip condition was applied on the ship hull and symmetry plane on the side, bottom and middle symmetry plane.

In OpenFOAM 2.1.0, it turns out that the free surface level tends to be changed as the computation develops. This means water fraction is not well conserved. The problem started to occur after some iterations when the solution is getting stabilised. This probably happens because different inlet and outlet velocity could cause continuity correction as discussed in the chapter 2.2.5. This could affect total water fraction and make the water level changes. This is nearly certain because the problem does not appear if the outlet velocity is set to be equal to the inlet velocity. Therefore, when the inlet velocity is high, the phenomenon is more obvious. This problem has been reported

many times in the OpenFOAM forum (www.cfd-online.com), however, the problem disappeared in the newer version of OpenFOAM. In version 2.3.0, there was an improvement in the outlet boundary condition. Improving mesh quality, that is, reducing aspect ratio of the mesh at the inlet and outlet was also able to mitigate this problem. The problem was not significant in KVLCC2 case presumably due to the relatively slow speed, whereas it was obvious in KCS.

Forces and moments on the hull surface can be calculated by following formulas:

$$F_{pressure} = \int p \cdot \vec{n} dS \quad (3.4)$$

$$F_{viscous} = \int \tau \cdot \vec{n} dS \quad (3.5)$$

$$M_{pressure} = \int (x_f - x_o) \times p \cdot \vec{n} dS \quad (3.6)$$

$$M_{viscous} = \int (x_f - x_o) \times \tau \cdot \vec{n} dS \quad (3.7)$$

, where p is pressure, τ is wall shear stress, x_o is origin of measurement and x_f is the distance from origin to the face centre of the force component.

3.4.1 KVLCC2

Table 3-1 shows the principal dimension of KVLCC2 model ship that is geometrically scaled down by ratio of 58.0.

Table 3-1: Principal dimensions of KVLCC2 model

Length between perpendiculars (m)	5.5172
Breadth (m)	1.0000
Draught (m)	0.3586
Block coefficient	0.8098
Speed (m/s)	1.047
Froude number	0.142

Figure 3-2: Profile and front view of KVLCC2

Figure 3-2 is the profile and the front view of KVLCC2 hullform. OpenFOAM 2.1.0 and LTSInterFoam solver was used for this computation. Since the resistance at the steady velocity where flow is fully developed is important, transient solver has no advantage to predict the resistance. Flow analysis around KVLCC2 was carried out with grid convergence study (Table 3-2).

Table 3-2: Grid convergence study
(The experiment result: Kim et al. [38])

No. of mesh (K)	Pressure Force (N)	Viscous Force (N)	Total Force (N)	Limiter
690	11.16	10.45	21.61	Not used
1100	9.3	13.39	22.69	Not used
1470	7.17	12.08	19.25	100% -> 50%
1742	6.87	11.94	18.82	100% -> 50%
2434	6.78	11.74	18.52	100% -> 50%
3556	6.95	11.50	18.45	100% -> 50%
EXP	-	-	18.12	-

The number of mesh was tested up to 3.5 million and longitudinal force that is resistance was monitored. The force value did not change much after the mesh is getting finer to some extent, i.e. 1.7 million mesh. The mesh with 3.5 million cells showed 1.8% deviation from the experiment result. When the mesh was very fine, a limiter on the gradient scheme was necessary for the convergence. Otherwise the computation diverged before the iteration number 2000. In order to obtain better accuracy, 100% limiter was used first, and reduced to 50% when the force value is stabilised. Figure 3-3 shows the convergence history of resistance when the number of mesh was

3.5 million. With the iteration of 15000 (750 in the horizontal axis of Figure 3-3), the limiter has been reduced to 50%.

Figure 3-3: Convergence history of the resistance of KVLCC2

Wave elevation along the hull is compared with experiment result (Kim et al. [37]) in Figure 3-4 and it shows very good agreement. The horizontal axis stands for the distance from the forward perpendicular to aft direction, which means zero is at the forward perpendicular of the ship. Due to the relatively slow speed of tanker ship, the amount of changing water level was not significant.

Figure 3-4: Comparison of wave elevation along the hull of KVLCC2 (Experiment result: Kim et al. [38])

Wave pattern was compared to the experiment measurement in Figure 3-5. When limiter was used for very fine mesh, it resulted in oscillating behaviour in free surface pattern. For this reason, contour line of wave elevation became messy and it was hard to compare with experiment result, thus, the contour line was not presented in this study. This oscillation can be observed near the stern area of wave pattern in Figure 3-5. However, predicted wave pattern is in good tendency compared to the experiment. More comparisons were made in KCS container ship with the newer version of OpenFOAM.

Figure 3-5: Comparison of wave pattern of KVLCC2 of CFD (upper) and the experiment (Kim et al. [38]) (bottom)

3.4.2 KCS

Table 3-3 is the principal dimension of KCS model which is scaled down by the ratio of 31.599 from the full scale.

Table 3-3: Principal dimensions of KCS model

Length between perpendiculars (m)	7.2786
Breadth (m)	1.0190
Draught (m)	0.3418
Block coefficient	0.651
Speed (m/s)	2.196
Froude number	0.260

Figure 3-6 is the profile and the front view of KCS hull.

Figure 3-6: Profile and front view of KCS

The number of mesh used for the simulation was about 1.7 million. Figure 3-7 shows generated mesh around and on KCS hull. Trimmed mesh was generated for the mesh near the ship to be fine and for the outer region to be coarse. The grid convergence study was not carried out again for KCS as it has been done for KVLCC2, but reasonable number of mesh size was used based on the results from KVLCC2. One problem from the very fine mesh is y^+ value. When the mesh at the hull surface is too small, it was difficult to keep y^+ value all over the surface above 30. This would cause problem with wall function as discussed in Chapter 2.2.6 and the solution would not be correct.

Figure 3-7: Mesh of KCS - (a) whole domain, (b) forward and (c) stern area

Figure 3-8 shows y^+ distributions on the hull surface when the flow is fully developed. Due to the bulbous bow area where y^+ appeared low, it was hard to use finer mesh than that was used at the hull surface. Additional mesh refinement was applied near the stern boss (Mesh refinement in (c) of Figure 3-7) where y^+ appeared higher than other under water area. By this, it was possible to model the mesh with reasonable y^+ value. Moreover, finer mesh was also used around the free surface region for better free surface capturing because relatively high velocity would create more wave elevation.

Figure 3-8: y^+ distributions on the hull

For KCS analysis in open water, the problem of rising free surface level was quite significant due to the relatively higher velocity. After testing different settings, it was decided to use OpenFOAM version 2.3.0 that was released during that time. In the newer version of OpenFOAM, the problem did not appear anymore. In this version, there was significant improvement of

the outlet boundary condition to maintain the conservation of volume fraction. Another problem was over-prediction of force in LTSInterFoam solver although free surface elevation and the pattern were well predicted with this solver. It is presumed that the different time step throughout the domain leads to wrong solution. Several different grid types were tested but the problem was not solved. When the transient solver, interFoam was used, the obtained force value showed about 3.5% deviation from the experiment results. For the comparison purpose, the result of single-phase steady-state solver, simpleFoam has been added. From the result of the steady-state solver, it is assumed that free surface affects about 7.5% of the resistance in this KCS case. The comparison between the solvers and the experiment is presented in Table 3-4. The total resistance has been non-dimensionalised as below:

$$C_T = \frac{R_T}{\frac{1}{2}\rho S U^2}$$

R_T : Total resistance (N)

ρ : density of Water (kg/m³)

S : Wetted surface area (m²)

U : Velocity (m/s)

Table 3-4: The comparison of KCS resistance by different solvers (Experiment results: Kim et al. [38])

Solver	Resistance coefficient	Deviation	Gradient scheme
simpleFoam	3.415e-3	-3.99%	gauss linear
interFoam	3.681e-3	3.49%	cellLimited gauss linear 1.0
LTSinterFoam	5.108e-3	43.60%	gauss linear
EXP	3.557e-3	-	-

Although high maximum Courant number (Max. 25 and 10 for the interface) used for the simulation due to the semi-implicit algorithm adopted in OpenFOAM 2.3.0, the convergence of the force value was quite stable. Figure 3-9 is the convergence history of the resistance value of interFoam solver for the different gradient schemes.

A limiter was needed for the stability of the calculation. Because of the relatively big oscillation of Gauss linear gradient scheme, the limiter was removed after the computation became stable at 50 second. This suppressed the oscillation of the gradient scheme and also improved the accuracy. Least squares method showed less oscillating behaviour but slightly less accuracy of the resistance value. Gauss linear scheme resulted in 31.492 N for

the viscous drag force while Least squares scheme 32.157 N. Gauss Linear showed slightly better result by 2% when the limiter was removed. In Figure 3-9, it is found that the pressure force gets lower after 50 seconds when the limiter is removed and this reduces over-prediction. The solution using Least squares method without the limiter was not converged. Hereafter, the result with Gauss linear scheme is used for other validations in this chapter.

In Figure 3-10, wave elevation of OpenFOAM calculation was compared with experiment (Kim et al. [38]). As for the horizontal axis that shows x/L_{pp} , 0 stands for forward perpendicular, 0.5 for midship and 1 for aft perpendicular. Except for the crest at the bow wave, it shows very good agreement with the experiment.

In Figure 3-11, wave pattern also shows good agreement with the experiment result (Kim et al. [38]). 0 stands for the forward perpendicular and 1 for the aft perpendicular. Some strange shapes at the side of the divergent wave behind the stern are appeared due to the mesh refinement.

Figure 3-9: Convergence time history of the resistance (N) of KCS by (a) Gauss linear and (b) Least square gradient scheme

Figure 3-10: Comparison of wave elevation along the KCS hull (Experiment results: Kim et al. [38])

Figure 3-11: Comparison of wave pattern of KCS of OpenFOAM (upper) and the experiment (Kim et al., [38]) (bottom)

Figure 3-12: Wave pattern at the forward (a) and the stern (b)

In Figure 3-12, wave pattern at the forward and the stern are presented. It illustrates the maximum (red) and the minimum (blue) of the wave elevations. Kelvin wave which consists of diverging wave and transverse wave is well described in the wave pattern.

Stern flow, which is wake distribution, becomes important information for propeller design. Due to the aft hull shape, wake velocities entering into the propeller plane become inhomogeneous on the propeller plane and they cause pressure fluctuation on the propeller blades while they are rotating in high RPM. This fluctuating pressure causes cavitation on the propeller

blades and on the rudder, and leads to many problems. Cavitation also affects the propeller efficiency, vibration and erosion on the propeller blades and rudder. Therefore, it is important to design the aft hull in such a way that the wake velocity becomes evenly distributed at the same blade radius to minimise pressure fluctuation especially for the container ship which has bigger propeller loading with powerful main engine.

Wake velocities that are calculated by OpenFOAM are visualised at the couple of sections in Figure 3-13. In the experiment, a dummy hub is attached on the stern boss while it is absent in the CFD calculation. Therefore, difference is unavoidable near the stern boss on the propeller plane. Overall tendencies are well predicted. One reason for the difference is the mesh fineness. Behind the ship stern, the flow separation, drag and turbulence make the flow very complex, hence, very fine mesh is needed to predict the flow accurately. In author's opinion and from the experience with commercial marine CFD codes SHIPFLOW and WAVIS, these are reasonable results due to the complexity of flow.

Figure 3-13: Wake distribution (axial velocity [m/s]) of KCS at (a) the propeller plane and (b) at the station no. 1 (left : experiment (Kim et al. [38]), right : OpenFOAM result)

3.4.3 Summary of the evaluations

OpenFOAM was able to solve multi-phase problem in open water and calculate resistance force value with reasonable size of mesh. The converged force value showed good accuracy with the wall function. For the gradient scheme, Gauss linear scheme without a limiter gave slightly better accuracy than Least squares

method, however, the latter had advantage of less oscillation in the converged force value. The free surface pattern and the elevation were well captured too. The wake distribution also showed reasonable accuracy in spite of medium fineness of mesh. Several problems in the OpenFOAM 2.1.0 such as conservation of volume fraction, oscillations on the free surface level, etc. were all fixed in the new version.

The local time stepping solver computed a wrong force value for KCS whereas the problem did not occur with KVLCC. This will be further tested in the next chapter on confined water analyses.

Chapter 4

Analysis in Confined Water

4.1 Computational set-up and mesh

Numerical setup for confined water analyses was established according to the experimental setup done by FHR (Lataire et al. [5]). The test cases with several bank arrangements and different under keel clearance were chosen for numerical analysis. As found in the experimental research, a 8000TEU container ship in model scale was used for the simulations with a scale ratio of 80. Table 4-1 is the principle dimension of the model ship.

Table 4-1: Principal dimension of model ship

Length overall (m)	4.332
Breadth (m)	0.530
Draught (m)	0.149 ~ 0.180
Block Coefficient	0.65 ~ 0.66 (depending on the draught)

Figure 4-1 shows the coordinate system of the cases.

Figure 4-1: The coordinate system

Figure 4-2 shows the bank arrangements of the test case.

Figure 4-2: Schematic view of the bank arrangement (looking from the aft)

As can be seen in Figure 4-2, three types of the bank were used. The arrangement (a) has a straight vertical bank wall. The bank type (b) has an inclined bank wall and type (c) has a bank wall that is inclined and submerged. The coordinate system used for

the simulation is defined as body fixed and right handed system and that is different from the experiment. The origin is located at the amidships, centreline and keel for x, y and z, respectively. Longitudinal axis (x) is towards the bow, transverse axis (y) towards port side and vertical axis (z) upwards. Therefore, positive sway force directs to the outside the bank and yaw moment is positive when the bow is away and the stern towards the bank. The negative heading angle of the case A in Table 4-2 means that bow of the ship is toward the bank and the stern is away from the bank.

Four cases were chosen among the experiment results for CFD simulations. Table 4-2 includes the details of the case arrangements. Under keel clearance is 10% or 100% of the draught and various lateral position of the ship from the bank are included in the test cases. The test cases without propeller thrust have been chosen firstly to see the bank effect merely from asymmetric flow field. If propeller is rotating, it may affect the force and the moment values. In these test cases, propeller was basically prevented from the rotations, therefore, measured RPM is -1 and propeller thrusts are all slight negative because the propeller worked like a 'parachute'. The case A that has an heading angle toward the inflow and the measured negative propeller torque was relatively big.

Table 4-2: The test cases without propeller thrust

Test title		A	C	F	H
Under keel clearance	[%]	100	10	100	10
Draught	[m]	0.180	0.180	0.180	0.149
Water depth	[m]	0.360	0.198	0.360	0.163
Forward component of speed vector	[m/s]	0.6842	0.4578	0.8012	0.4578
Transverse component of speed vector	[m/s]	-0.0599	0.0	0.0	0.0
Lateral separation	[m]	0.265	0.265	1.435	1.965
Heading angle	[°]	-5.0	0.0	0.0	0.0
Propeller rate	[rpm]	-1	-1	-1	-1
Propeller thrust	[N]	-1.10	-0.20	-0.93	-0.37
Propeller torque	[Nmm]	-27.09	-3.05	0.84	7.28
Type of bank	-	(b)	(c)	(b)	(a)
Bank inclination		8:1	5:1	3:1	1

The case A has slight heading angle with small separation from the bank wall. The case C is the most severe case where both under keel clearance and lateral separation from the bank wall are very small. The minimum distance from the hull to the sloped bank wall is very small. The case F does not have extreme clearance from the bank wall or the bank bottom, however, the velocity of the ship is relatively high. The case H has very small under keel clearance but moderate distance to the bank wall.

Figure 4-3 is the graphical view of the test cases as modelled in OpenFOAM. It shows the bank shape, the location of the ship and water depth.

Figure 4-3: Graphical view of the case C and the case H

There is a need to find out how much propeller contributes to the bank effect. Although propeller rotation is not modelled in this study, another set of simulations has been done for the test cases with propeller thrust but without modelling the propeller. The purpose of this set of simulations is to compare the CFD results without propellers to the experiment results that have propeller thrust. This will indicate the significance of propeller contribution to the bank effect. They will show the difference as

much as propeller rotation contributes to the bank. Table 4-3 is the detailed experiment conditions for this set of test cases. The only difference in the CFD simulations is that the propeller is not modelled.

Table 4-3: The test cases with propeller thrust

Test title		D	I	J
Under keel clearance	[%]	100	35	100
Draught	[m]	0.149	0.149	0.149
Water depth	[m]	0.297	0.200	0.297
Forward component of speed vector	[m/s]	0.6868	0.5723	0.8012
Transverse component of speed vector	[m/s]	0.0	0.0	0.0
Lateral separation	[m]	1.340	0.565	0.530
Heading angle	[°]	0.0	0.0	0.0
Propeller rate	[rpm]	540	359	539
Propeller thrust	[N]	3.30	1.46	3.16
Propeller torque	[Nmm]	53.78	20.65	62.35
Type of bank	-	(b)	(c)	(c)
Bank inclination		8:1	8:1	5:1

For the confined water analysis, the mesh size and settings were determined from the open water analysis results. Efforts were made to model the mesh finer or prism boundary layers properly because a certain number of mesh layers would be important for

the narrow gap between the bank and the ship as well as the under keel clearance. At least, two boundary layers have been modelled. The OpenFOAM mesher was prone to fail to generate more than three layers even boundary layers at curvature of the hull. Nevertheless, the total number of meshes becomes less than the open water analysis due to the confined geometry of the computational domain. The size of the mesh are 0.8~1.4 million. Figure 4-4 is the mesh of whole domain, near the hull and hull surface mesh of case H. The computation domain has been modelled up to the deck height of the ship so that the bottom can be seen from the top. Freeboard height is enough to cover air domain because the confined water cases have relatively slow speed.

Too small size of the mesh at the hull surface was not appropriate because it caused y^+ value too small. A compromise was important between the cell sizes next to the hull surface and the y^+ value to model the narrow gap properly. The meshes were modelled in such a way that y^+ values ranged between 30 and 300 on all the area of the hull surface except the case H. This will be discussed in the next chapter. For the single-phase simulations, number of mesh was about 50~70% of the multi-phase case depending on the under keel clearance and water depth because the domain was modelled up to the flat free

surface level. Under water meshes are basically equal to those used for two-phase solver.

Figure 4-4: Mesh of (a) whole domain and (b) near the hull (looking from the top), and (c) mesh on the hull surface of case H

Turbulence model was determined as $k-\omega$ SST after simple comparison between $k-\epsilon$ and $k-\omega$ SST. Both gave similar force

value but $k-\epsilon$ showed a bit larger oscillation especially for sway force value though showed little difference in the converged force values. Since bigger oscillation makes it difficult to judge convergence, $k-\omega$ SST model was chosen for the analysis.

Just as the KCS open water analysis, three solvers – interFoam, LTSInterFoam and simpleFoam, are used for the confined water analysis. The comparison of the single-phase and the two-phase solvers will show how free surface contributes to the ship-to-bank interaction. As it is normally known that the VoF solver is much more sensitive and instable, this will also help finding out the stability of the VoF solver of OpenFOAM and how reliable results are in comparison to the single-phase solver.

4.2 Boundary conditions

As for the boundary conditions, a fixed value is applied at the inlet for velocity and turbulence properties. At the outlet, zero gradient is applied for velocity and turbulence properties. No slip condition is applied to the hull. For the velocity at the bank boundary, the same fixed value with the inlet velocity was used so that no-slip condition can be applied on the bank wall. This is reasonable for no-slip condition on the bank wall of body fixed coordinate system because the point of view is on the ship and undisturbed water and the bank should have the same velocity.

For the single-phase solver, a slip condition is used for the velocity on the top which is free surface level. Zero gradient can be used to the walls, top and outlet of the pressure boundary. Fixed value is applied at the inlet pressure. Standard wall function of OpenFOAM was used for turbulence properties. Table 4-4 is the summarised boundary conditions applied for the single-phase solver.

**Table 4-4: Boundary conditions for confined water analysis
of the single-phase solver**

Flow properties	inlet	outlet	top	bank	ship
U [m/s]	fixed value (u 0 0)	zero gradient	zero gradient	fixed value (u 0 0)	no slip (0 0 0)
p [m ² /s ²]	zero gradient	fixed value 0	slip	zero gradient	zero gradient
k [m ² /s ²]	fixed value 0.00015	zero gradient	zero gradient	kqRWallFunction 0.00015	
ω [1/s]	fixed value 2	zero gradient	Zero gradient	omegaWallFunction 2	

In the two-phase solver in which air and water exist together, top boundary plane becomes the boundary for atmosphere and some boundary conditions are different from the single-phase solver. The main differences are in the pressure boundary conditions and for the atmospheric boundary. The boundary conditions used for the multi-phase solver are listed in Table 4-5.

The alpha boundary conditions are needed for the volume fraction. In OpenFOAM 2.3.0, new alpha boundary condition has been implemented at the outlet for the conservation of the volume fraction.

Table 4-5: Boundary conditions for confined water analysis of the multi-phase solver

Flow properties	inlet	outlet	atmosphere	bank	ship
U [m/s]	fixed value (u 0 0)	zero gradient	pressureInlet- Outlet- Velocity	Fixed value (u 0 0)	Fixed value (0 0 0)
p_rgh (p _d) [kg / ms ²]	zero gradient	fixedFlux- pressure 0	Total pressure 0	fixedFluxPressure 0	
k [m ² /s ²]	fixed value 0.00015	zero gradient	inletOutlet inletValue 0.00015	kqRWallFunction 0.00015	
ω [1/s]	fixed value 2	zero gradient	inletOutlet inletValue 2	omegaWallFunction n 2	
Alpha [-]	calculated	Variable Height FlowRate	inletOutlet	zero gradient	

For the pressure boundary conditions of two phase solver, dynamic pressure is set at the boundary as discussed in the governing equation of the multi-phase solver. The pressure used for multi-phase solver has different unit from single-phase solver as density is being multiplied. ‘fixedFluxPressure’ is an improved boundary condition that sets pressure gradient based on the fixed flux, hence, determined from the velocity condition at the boundary. The flux equation (2.40) is used for this boundary condition to set pressure gradient. For instance, pressure condition becomes no gradient to apply zero flux on the no-slip wall boundary. Therefore, it is equivalent to zero gradient at the

no-slip walls. This was used for the pressure boundary conditions at the inlet and the wall boundaries. Zero gradient is used at the outlet and total pressure is used at the top. Total pressure adjust pressure based on U:

$$p_0 = p + \frac{1}{2}\rho|U|^2 \quad (4.1)$$

Chapter 5

Results and Discussion

5.1 Computation results

For the most of calculations in this study, High Performance Computing (HPC) system in Nanyang Technological University was used for parallel computation. It took about 3 hours for a 100 second simulation of about one million-mesh case with 32 processors. Because the semi-implicit method was implemented to the two-phase solver of OpenFOAM 2.3.0, the computation time was affordable with maximum Courant number 25 and interface Courant number 10. When both Courant numbers were set to be 0.4, it took about one day. For the single-phase steady-state calculations of the steady-state solver, a MacBook Pro with 2.3GHz Intel Core i7 CPU (8 cores) was used. Computation time was all less than 10 minutes.

5.1.1 Force and moment

Figure 5-1 and Figure 5-2 are the convergence time history of forces and moment by two different pressure gradient schemes.

Figure 5-1: Convergence history of (a) X and (b) Y forces (N), and (c) Z moment (Nm) from pressure of case F (Gradient scheme: Gauss linear, Max Co. reduced from 70s)

Figure 5-2: Convergence history of (a) X and (b) Y forces (N), and (c) Z moment (Nm) from pressure of case F (Gradient scheme: Least square)

The computation results of two-phase solver show that pressure component dominates over the viscosity with large oscillations for ship-to-bank interaction. It is presumed that the oscillation is caused from the interaction between the bank and the ship because the oscillation indicates quite different behaviour from the open water analysis in spite of relatively slow speed. Even the longitudinal force showed different oscillating behaviour from the open water analysis that was discussed in the previous chapter. Gauss linear and Least squares schemes that have 2nd order accuracy were evaluated as the gradient scheme. As Gauss Linear method showed big oscillation, Courant number was reduced to 0.4 when the pressure value is stabilised at 70 second. This oppressed the oscillation and helped in averaging values. However, the computation took about one day for the additional 50 seconds simulation with the low Courant number. The force values were less oscillating with Least Square method and with high Courant number. The under relaxation factors were also used for stable solutions as follows:

p	0.3
U	0.7
k	0.7
omega	0.7

Because the values showed oscillating behaviour, the computation had to be monitored over the sufficient time duration and the converged force values were obtained from the average over the time where the oscillation is stabilised.

Three different solvers have been used for the simulations. As discussed in the Chapter 3.3, `simpleFoam` is a single-phase steady-state solver, `interFoam` is a VoF transient solver and `LTSInterFoam` is a local time stepping VoF solver. Table 5-1 is the comparison table of forces and moments between the CFD results and the experiment (Lataire et al., 2009). From the comparison results, `LTSInterFoam` gave wrong values just as shown for KCS analysis in open water. Different mesh type and mesh fineness were tested to rectify this, however, all did not improve the results much although other numerical settings were almost the same with `interFoam`. As it gave wrong results, Least square scheme was not tested for the LTS solver, moreover, the LTS solver was excluded from the evaluation hereafter. `interFoam` gave reasonable accuracy. Comparing the different gradient schemes, least square generally showed better results compared to Gauss linear. `simpleFoam` also allowed good accuracy in spite of single-phase simulation without free surface. Least square method also showed better accuracy for single-phase calculations.

**Table 5-1: Comparison of the cases without thrust to the experiment results
(Lataire et al. [5])**

Case		A		C		F		H	
Solver	Value	Gauss linear	Least Squares	Gauss linear	Least Squares	Gauss linear	Least Squares	Gauss linear	Least Squares
LTSInter-Foam	Fx (N)	-6.960	-	Diverged		-9.345	-	-3.211	-
	Fy (N)	-8.190				-1.029		0.539	
	Mz (Nm)	-11.652				1.259		4.600	
Inter-Foam	Fx (N)	-4.799	-4.866			-5.465	-5.512	-1.559	-2.239
	Fy (N)	-7.122	-5.646			-1.111	-1.008	0.581	1.963
	Mz (Nm)	-8.112	-9.067			1.982	1.305	2.159	5.561
Simple-Foam	Fx (N)	-4.193	-3.922	-4.062	-4.094	-4.395	-3.794	-1.839	-1.880
	Fy (N)	-4.518	-5.343	-0.789	0.709	-1.106	-1.300	0.807	1.245
	Mz (Nm)	-9.948	-8.665	4.145	6.810	0.629	1.053	3.296	4.017
EXP	Fx (N)	-4.130		-2.618		-5.354		-2.935	
	Fy (N)	-5.139		0.962		-1.648		2.545	
	Mz (Nm)	-8.939		4.300		1.804		6.959	

The followings are the comparison results of each case.

Case A The two-phase solver over-predicted longitudinal force slightly, but sway force and yaw moment showed reasonable accuracy with Least square method. The single-phase solver showed good accuracy compared to the experiment. Least Square scheme gave better result of sway force and yaw moment for the both solvers with the reasonable accuracy.

Case C This case did not converge for the VoF solvers in spite of many trials and efforts. However, the single-phase solution was able to converge. Single-phase solver also did not show good accuracy. Even Gauss linear gradient scheme predicted sway force to the opposite direction.

Case F The two-phase solution with Gauss linear gradient scheme gave the good accuracy for longitudinal force and yaw moment, however, sway force was under-predicted. Single-phase solver predicted better sway force but not for other values. The reason for this is not clear.

Case H This case has small under keel clearance which is 10% of water draught, but the separation from the bank wall is however, not small. When same meshing type (2 layers of boundary layer) was used from other cases, all the values were under-predicted. The details will be discussed in the next Section 5.1.2. The presented values in Table 5-1 were when mesh was further refined to have more boundary layers.

5.1.2 Under keel space modelling of the case H

Because of small under keel clearance of the case H, it was difficult to model the mesh in the narrow space properly. If many layers are modelled in the under keel space, y^+ value becomes too low, i.e. below 30. In that case, the wall function may not be valid. To keep y^+ value bigger than 30, the optimum number of layer in the under keel clearance had to be only about two in the case H. The small under keel clearance is known to affect the bank effect significantly, therefore, too few number of layers would not be enough to capture the effect. In this case, the best thing is to model y^+ value less than 1 throughout the hull surface and resolve the whole boundary layer without using wall functions. In the OpenFOAM mesher by which meshing is very slow compared to commercial meshers, this was not efficient because number of mesh becomes too huge.

To compromise between the number of layers and y^+ value in the under keel space, a comparison has been made. Four different meshes are tested for the case H. They have same mesh size in other area but different boundary layers or different mesh in the under keel space. The tested meshes are shown in Figure 5-3 that illustrates the midship section near the bilge radius.

**Figure 5-3: Different meshing of case H -
no layer (a), three layers (b), eight layers (c), fine mesh (d)**

Figure 5-4 shows y^+ distribution for each of the four meshing types.

**Figure 5-4: y^+ of the case H with different meshing
no layer (a), three layers (b), eight layers (c), fine mesh (d)**

(a) has no specific modelling in the boundary layer, (b) has two prism boundary layers and total three layers in the under keel, (c) has eight prism boundary layers and total nine in the under keel, and (d) has very fine refinement in the under keel clearance without modelling boundary layer. (b) type meshing is a general boundary modelling for all other test cases used in this study.

(a) and (b) have y^+ values above 30 and all in good range. OpenFOAM mesher has problem when many prism layers have to be created at the boundary layer especially at large curvature. For this reason, prism layers have not been generated as much as amidships at fore and aft area of (c) where curvature appears. Instead, many mesh layers have been created in the boundary layer at the parallel middle body area where under keel clearance is most significant. In (d), very fine mesh was used in under keel space, hence, very low y^+ distribution can be seen on the bottom hull surface.

The computed results for the different meshes of the case H are tabulated in Table 5-2. Least Square methods are used for the gradient scheme. From the comparison results, (c) type meshing showed the best results. Therefore, it is considered that number of layers in the under keel clearance are important to predict

bank effect. Many prism layers at the boundary layer are more effective than the (d) type meshing.

Table 5-2: Comparison between different meshing in the under keel space of the case H

	(a)	(b)	(c)	(d)	EXP
No. of mesh (mil.)	1.11	1.13	1.15	5.21	-
F _x (N)	1.484	2.239	2.236	2.247	2.935
F _y (N)	0.368	1.104	1.963	1.037	2.545
M _z (Nm)	4.004	3.734	5.561	3.647	6.959

More comparisons are made for the meshing (b) and (c). Figure 5-5 shows pressure distribution in the water at the centreline of the ship.

Figure 5-5: Pressure [N] distribution in the water at the centreline of the case H with (b) three layers (c) eight layers

The pressure distribution on the hull was compared in Figure 5-6. For better visualisation, the range of the pressure applied became different for the two cases.

Figure 5-6: Pressure [N] distribution on the hull (looking from the bottom) of the case H with (b) three layers (c) eight layers

When eight layers are used (c), the overall pressure is higher both in the water (Figure 5-5) and on the hull (Figure 5-6). In Figure 5-5, the pressure difference between the low pressure region (under keel space) and the high pressure region (bow) of the case (c) (ranged in 1380~1475N) is bigger compared to the case (b) (ranged in 1350~1425 N).

5.1.3 Pressure and velocity asymmetry

The computation shows clear pressure asymmetry that causes the bank effect. Figure 5-7 shows the computed pressure distributions of the cases on the hull surface and the bank surface. The deck is not shown as the domain is modelled up to the freeboard height.

Figure 5-7: Pressure [N] distribution on the hull and bank surface of the cases F (a) and H (b) (looking from the top)

Because the incoming velocity is much faster in the case F (0.8012 m/s) than in the case H (0.4578 m/s), the pressure range and its variation are much higher in the case F. However,

the measured sway force and yaw moment from the experiment appeared much bigger in the case H than in the case F. Therefore, it is obvious that under keel clearance affects more significantly although bank effect is known to be almost proportional to velocity square.

Next, three sectional planes were cut into the water to investigate the pressure distributions in the flow. Sections were cut at amidships, near the bow (station 17.5) and near the stern (station 2.5) were cut and presented in Figure 5-8. As force is directed from high to low pressure, the direction of yaw moment can be guessed from this sectional views. High pressure is observed near the bow and this would contribute to the yaw moment. The calculated negative yaw moment of the case A can be observed from the very high pressure at the port side of the bow. On the other hand, the cases F and H where positive yaw moments were calculated have strong high pressure at the starboard side.

Figure 5-9, Figure 5-10 and Figure 5-11 show the velocity magnitude of the cases F, H and I at various water levels. The case I that has small clearance between the ship and the bank was added for better comparison and this case illustrates the most asymmetry of velocity field. The case F that has less

extreme condition than the cases H and I shows the least asymmetry among the three cases. The case F reveals biggest asymmetric velocity at the 1/3 draught level, and the cases H and I reveal at the 2/3 draught level.

Figure 5-8: Dynamic pressure [N] distribution of OpenFOAM calculation of case A (a), F (b) and H (c) at the different sections (station 2.5, amidships and station 17.5)

Figure 5-9: Velocity magnitude [m/s] of the case F at (a) free surface, (b) 2/3 of draught from baseline, (c) 1/3 of draught from baseline and (d) baseline

Figure 5-10: Velocity magnitude [m/s] of the case H at (a) free surface, (b) 2/3 of draught from baseline, (c) 1/3 of draught from baseline and (d) baseline

Figure 5-11: Velocity magnitude [m/s] of the case I at (a) free surface, (b) 2/3 of draught from baseline, (c) 1/3 of draught from baseline and (d) baseline

5.1.4 Results of the cases with propeller rotation

Table 5-3 is the comparison results of the cases with propeller thrust. Sway forces and yaw moments are compared with the experiment results. Since local time stepping solver showed wrong results for most of the cases, it has not been included in these cases. Although the case D and the case I had small clearance of the sloped bank and the hull surface, however the solution converged well with reasonable accuracy of yaw moment.

Table 5-3: Comparison of the cases with thrust to the experiment results (Lataire et al. [5])

Solver	Case	D		I		J	
	Value	Gauss Linear	Least Square	Gauss Linear	Least Square	Gauss Linear	Least Square
inter-Foam	Fy (N)	-2.011	-0.754	-0.201	0.137	-0.894	-0.696
	Mz (Nm)	19.28	15.932	9.750	12.60	2.169	1.838
simple-Foam	Fy (N)	-2.683	-2.354	-1.080	-0.699	-1.427	Diverged
	Mz (Nm)	4.692	5.108	1.960	2.117	1.688	
EXP	Fy (N)	-8.056		-3.682		-2.264	
	Mz (Nm)	16.407		8.312		2.065	

As can be seen in the comparison, when the propeller was rotating, all the sway forces showed under prediction compared

to the experiment. In these cases, Gauss linear method showed relatively better results than least square method for yaw moment. The direction of the sway forces was mostly correct but the magnitudes were all wrong. When the propeller is rotating, sway forces were all in suction direction to the bank wall. However, yaw moment values showed reasonable accuracy. This implies the propeller rotation significantly affects the magnitude of sway forces. Single-phase solver showed poor accuracy even for yaw moments. Further comparison for these cases are not reliable but will become speculation, hence, it is not meaningful.

The single-phase calculations with Least Squares scheme for the case J diverged.

5.1.5 Comparison between single-phase and two-phase solutions

It was expected that the difference between the calculated values and the experiment single-phase solver and two-phase solver would show how free surface affects on the bank effect. However, the single-phase solution of the cases A and F showed even better accuracy than two-phase solution. In other cases, single-phase solution showed under prediction compared to the two-phase solutions. Especially the cases with propulsion showed

poor accuracy. From the case C, single-phase solver is more stable than two-phase solver.

5.1.6 Free surface elevation

Free surface elevations were compared with experiment results. The wave measurements were done in three points of the cases C, F, I and J. The comparison was done for these available cases except the case C that was failed to converge. Table 5-4 is the location of the wave gauges from centreline of the ship according to the experiment. Wave gauges measure the wave elevation at the defined transverse distances from the centreline of the ship.

Table 5-4: The locations (m) of the wave gauges with reference to the centreline of the ship (Lataire et al. [5])

	Wave gauge 1	Wave gauge 2	Wave gauge 3
Case F	0.475	1.005	1.535
Case I	0.875	1.215	2.115
Case J	0.920	1.320	1.785

Figure 5-12 is the graphical views of the measuring locations of the wave gauges and also gives a brief idea about the water depth at each wave gauge.

Figure 5-12: The locations of the wave gauges 1, 2 and 3 of the cases (a) F, (b) I and (c) J

Figure 5-13, Figure 5-14 and Figure 5-15 show the comparison of wave elevations with the experiment measurements. In the horizontal scale, 0 stands for amidships and positive direction is toward the stern.

Figure 5-13: Wave elevation of the case F

at (a) gauge 1, (b) gauge 2 and (c) gauge 3 (Lataire et al. [5])

Figure 5-14: Wave elevation of the case I

at (a) gauge 1, (b) gauge 2 and (c) gauge 3 (Lataire et al. [5])

Figure 5-15: Wave elevation of the case J

at (a) gauge 1, (b) gauge 2 and (c) gauge 3 (Lataire et al. [5])

The further wave gauges from the ship showed less accuracy of the wave elevation. A steep wave is observed in the case I and it is continued from the hull to the bank wall. This was also found in the experiment but over-predicted in the computation result at the gauges 1 and 2. In addition, phase delay was observed, especially at the furthest wave gauge from the ship, i.e. wave gauge 3. This phase delay was also found in the case F. The phase of wave trough was deviated by about a quarter of wave length. The case J showed relatively good accuracy. As can be seen in Figure 5-12, the wave elevation is considered less accurate at shallow water region – at the wave gauges 3 of the case F, and at the wave gauge 2 and 3 of the case I. Lastly, the computations did not capture the oscillation of the wave elevation correctly. The experiment showed more oscillation but this was not captured or captured differently in the computation results.

5.2 Discussion

OpenFOAM was able to predict ship-to-bank interaction with the different bank types in extremely shallow water. Two-phase solver failed to converge from finding the solution of one severe case that has extreme under keel clearance and small separation from the sloped bank wall.

For the stable calculations and convergence, good quality of meshing was particularly important in OpenFOAM compared to other commercial CFD codes. Another reason for this is pressure becomes sensitive in the multi-phase solver compared to the single-phase solver. Special care is required for the aspect ratio and high skewness of the grid.

Another challenge is a compromise between the mesh fineness and y^+ value to model the narrow gap between the hull and the bank. Boundary layer and the narrow gap modelling are important although y^+ values have to go slightly below 30 that falls to buffer layer. For the case with small under keel clearance, good meshing improves sway force significantly.

The pressure values of the calculations in confined water often show more oscillating behaviour as compared to the open water

analysis when converging. It is considered to occur from the reflections and interactions between the bank and the ship. Reducing Courant number was helpful to suppress the oscillation. Least Square method of gradient scheme showed less oscillating behaviour than Gauss linear. Overall, Gauss linear scheme showed better accuracy, however Courant Number had to be reduced to very small, e.g. 0.4 in order to capture the force and moment correctly due to the oscillation.

The computation tends to under-predict forces and moment. This is most obvious in sway force. There would be several reasons for this. Firstly, there is limitation of meshing in the narrow gap. One thing can try is resolving whole boundary layer by modelling y^+ below 1. However, this is not easy with OpenFOAM mesher and a commercial mesher should be employed for efficient boundary layer modelling. Secondly, fixed heave and pitch motion of CFD calculations could have caused this under-prediction because the two motions were free in the model tests. This assumption of fixed motion neglects squat effect, therefore, more accuracy can be expected if the 2 DOF motions are set to be free. However, the calculated sway forces and yaw moments from the bank effect are all in correct directions for all the types of bank shapes and arrangement although ship-to-bank interaction is very complex and difficult to predict the directions.

The under-prediction was more obvious in the cases with propeller rotation while yaw moments were predicted with reasonable accuracy. This explains how propeller rotation affects the sway force. More study is needed whether propeller intervention cause additional sway force to specific direction. In this set of experiments, all the cases with propeller rotation resulted in suction forces and the computation without propeller modelling computed the sway forces to the correct direction but in smaller magnitude. However, it is hard to conclude this at this stage. From this comparison, it is noted that propeller rotation seems to affect the magnitude of sway forces but does not give significant contribution to yaw moment.

Free surface elevation was less accurate than the open water analysis. In addition, difference was found in wave phase. Such a phase delay was more obvious in the case where the water depth is shallow. Less oscillation on the free surface level was observed in the computation than the experiment result.

Chapter 6

Conclusion

6.1 Conclusion

In this study, CFD theory and numerical techniques were discussed. In order to use open source CFD code, the discretisation of governing equations, solution algorithm, stability of the computation, boundary condition and boundary layer treatment were explained. Several features and the solvers of OpenFOAM were also investigated.

In this study, flow around ships in open water and in confined water with several bank arrangements was simulated by incompressible RANS solver of OpenFOAM. The results from single-phase and two-phase solver were compared. The hydrodynamic forces and moment due to the bank, wave elevation were compared to the experiment results.

Under keel clearance modelling was evaluated to compromise between mesh fineness and y^+ distribution. The gradient schemes of OpenFOAM were compared in terms of accuracy and stable computation. OpenFOAM was able to predict the bank

effect in confined water that causes the bank effect as well as the open water analysis. Asymmetric pressure and velocity fields were well presented in connection to the calculated forces and moment. In order to find the propeller contribution to the bank effect, the cases with propeller thrust were compared to the computations without propeller.

The reason for under prediction of sway force has to be further investigated with careful meshing and numerical schemes. Unfortunately, the experiment results open to the public were only available to those particular cases, therefore, systematic parametric study was not possible.

6.2 Future work

As for the CFD analysis of the bank effect, only one case which is the vertical bank arrangement was successful in this report. Analysis for other cases with sloped banks and with different under keel clearances have to be tested, too. There are two types of sloped bank in the experiments. One type is surface piercing sloped bank and the other is where the slope ends below the waterline.

Since good meshing affects the calculations of multiphase solver of OpenFOAM significantly, there will be intensive study in meshing options especially for the refinement at near wall region. Otherwise, commercial software can be used for the meshing. Meshing software that can export compatible file type to OpenFOAM can be used for this. In addition, it will be valuable to compare the results of OpenFOAM to that of a commercial code.

OpenFOAM has strong dynamic mesh options with adaptive mesh refinement capability. This can add pitch and heave motion of ship, hence, it is expected that the inclusion of shallow water effect will give better accuracy.

Appendix

The source codes of interFoam

interFoam.C: The main codes of the solver. It handles PIMPLE loop and connects all the related header files to solve the governing equations.

```
1      #include "fvCFD.H"
2      #include "CMULES.H"
3      #include "subCycle.H"
4      #include "interfaceProperties.H"
5      #include "incompressibleTwoPhaseMixture.H"
6      #include "turbulenceModel.H"
7      #include "pimpleControl.H"
8      #include "fvIOoptionList.H"
9      #include "fixedFluxPressureFvPatchScalarField.H"
10
11     // * * * * *
12
13     int main(int argc, char *argv[])
14     {
15         #include "setRootCase.H"
16         #include "createTime.H"
17         #include "createMesh.H"
18
19         pimpleControl pimple(mesh);
20
21         #include "initContinuityErrs.H"
22         #include "createFields.H"
23         #include "readTimeControls.H"
24         #include "createPrghCorrTypes.H"
25         #include "correctPhi.H"
26         #include "CourantNo.H"
27         #include "setInitialDeltaT.H"
28
29     // * * * * *
30
```

```

31 Info<< "\nStarting time loop\n" << endl;
32
33 while (runTime.run())
34 {
35     #include "readTimeControls.H"
36     #include "CourantNo.H"
37     #include "alphaCourantNo.H"
38     #include "setDeltaT.H"
39
40     runTime++;
41
42     Info<< "Time = " << runTime.timeName() << nl << endl;
43
44     // --- Pressure-velocity PIMPLE corrector loop
45     while (pimple.loop())
46     {
47         #include "alphaControls.H"
48
49         if (pimple.firstIter() || alphaOuterCorrectors)
50         {
51             twoPhaseProperties.correct();
52
53             #include "alphaEqnSubCycle.H"
54             interface.correct();
55         }
56
57         #include "UEqn.H"
58
59         // --- Pressure corrector loop
60         while (pimple.correct())
61         {
62             #include "pEqn.H"
63         }
64
65         if (pimple.turbCorr())
66         {
67             turbulence->correct();
68         }
69     }
70
71     runTime.write();
72
73     Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
74         << " ClockTime = " << runTime.elapsedClockTime() << " s"
75         << nl << endl;

```

```

76     }
77
78     Info<< "End\n" << endl;
79
80     return 0;
81 }

```

AlphaEqn.H: It contains equations for the volume fraction α including MULES algorithm (Refer to Chapter 3.3)

```

1  {
2      word alphaScheme("div(phi,alpha)");
3      word alphasScheme("div(phirb,alpha)");
4
5      // Standard face-flux compression coefficient
6      surfaceScalarField phic(interface.cAlpha()*mag(phi/mesh.magSf()));
7
8      // Add the optional isotropic compression contribution
9      if (icAlpha > 0)
10     {
11         phic *= (1.0 - icAlpha);
12         phic += (interface.cAlpha()*icAlpha)*fvc::interpolate(mag(U));
13     }
14
15     // Do not compress interface at non-coupled boundary faces
16     // (inlets, outlets etc.)
17     forAll(phic.boundaryField(), patchi)
18     {
19         fvsPatchScalarField& phicp = phic.boundaryField()[patchi];
20
21         if (!phicp.coupled())
22         {
23             phicp == 0;
24         }
25     }
26
27     tmp<surfaceScalarField> tphiAlpha;
28
29     if (MULESCorr)

```

```

30  {
31      fvScalarMatrix alpha1Eqn
32      (
33          #ifdef LTSSOLVE
34      fv::localEulerDdtScheme<scalar>(mesh,rDeltaT.name()).fvmDdt(alpha1)
35          #else
36      fv::EulerDdtScheme<scalar>(mesh).fvmDdt(alpha1)
37          #endif
38      + fv::gaussConvectionScheme<scalar>
39      (
40          mesh,
41          phi,
42          upwind<scalar>(mesh, phi)
43      ).fvmDiv(phi, alpha1)
44      );
45
46      alpha1Eqn.solve();
47
48      Info<< "Phase-1 volume fraction = "
49          << alpha1.weightedAverage(mesh.Vsc()).value()
50          << " Min(alpha1) = " << min(alpha1).value()
51          << " Max(alpha1) = " << max(alpha1).value()
52          << endl;
53
54      tmp<surfaceScalarField> tphiAlphaUD(alpha1Eqn.flux());
55      tphiAlpha = tmp<surfaceScalarField>
56      (
57          new surfaceScalarField(tphiAlphaUD())
58      );
59
60      if (alphaApplyPrevCorr && tphiAlphaCorr0.valid())
61      {
62          Info<< "Applying the previous iteration compression flux" << endl;
63          #ifdef LTSSOLVE
64          MULES::LTScorrect(alpha1, tphiAlpha(), tphiAlphaCorr0(), 1, 0);
65          #else
66          MULES::correct(alpha1, tphiAlpha(), tphiAlphaCorr0(), 1, 0);
67          #endif
68
69          tphiAlpha() += tphiAlphaCorr0();
70      }
71
72      // Cache the upwind-flux
73      tphiAlphaCorr0 = tphiAlphaUD;
74

```

```

75     alpha2 = 1.0 - alpha1;
76
77     interface.correct();
78 }
79
80 for (int aCorr=0; aCorr<nAlphaCorr; aCorr++)
81 {
82     surfaceScalarField phir(phic*interface.nHatf());
83
84     tmp<surfaceScalarField> tphiAlphaUn
85     (
86         fvc::flux
87         (
88             phi,
89             alpha1,
90             alphaScheme
91         )
92     + fvc::flux
93     (
94         -fvc::flux(-phir, alpha2, alphasScheme),
95         alpha1,
96         alphasScheme
97     )
98 );
99
100 if (MULESCorr)
101 {
102     tmp<surfaceScalarField> tphiAlphaCorr(tphiAlphaUn() - tphiAlpha());
103     volScalarField alpha10(alpha1);
104
105     #ifdef LTSSOLVE
106     MULES::LTScorrect(alpha1, tphiAlphaUn(), tphiAlphaCorr(), 1, 0);
107     #else
108     MULES::correct(alpha1, tphiAlphaUn(), tphiAlphaCorr(), 1, 0);
109     #endif
110
111     // Under-relax the correction for all but the 1st corrector
112     if (aCorr == 0)
113     {
114         tphiAlpha() += tphiAlphaCorr();
115     }
116     else
117     {
118         alpha1 = 0.5*alpha1 + 0.5*alpha10;
119         tphiAlpha() += 0.5*tphiAlphaCorr();

```

```

120     }
121 }
122 else
123 {
124     tphiAlpha = tphiAlphaUn;
125
126     #ifdef LTSSOLVE
127     MULES::explicitLTSSolve(alpha1, phi, tphiAlpha(), 1, 0);
128     #else
129     MULES::explicitSolve(alpha1, phi, tphiAlpha(), 1, 0);
130     #endif
131 }
132
133     alpha2 = 1.0 - alpha1;
134
135     interface.correct();
136 }
137
138     rhoPhi = tphiAlpha()*(rho1 - rho2) + phi*rho2;
139
140     if (alphaApplyPrevCorr && MULESCorr)
141     {
142         tphiAlphaCorr0 = tphiAlpha() - tphiAlphaCorr0;
143     }
144
145     Info<< "Phase-1 volume fraction = "
146         << alpha1.weightedAverage(mesh.Vsc()).value()
147         << " Min(alpha1) = " << min(alpha1).value()
148         << " Max(alpha1) = " << max(alpha1).value()
149         << endl;
150 }

```

AlphaEqnSubCycle.H: This is for additional sub-cycles in the given time step in order to compute stable α field.

```

1     if (nAlphaSubCycles > 1)
2     {
3         dimensionedScalar totalDeltaT = runTime.deltaT();
4         surfaceScalarField rhoPhiSum

```

```

5      (
6      IOobject
7      (
8      "rhoPhiSum",
9      runTime.timeName(),
10     mesh
11     ),
12     mesh,
13     dimensionedScalar("0", rhoPhi.dimensions(), 0)
14     );
15
16     for
17     (
18         subCycle<volScalarField> alphaSubCycle(alpha1, nAlphaSubCycles);
19         !(++alphaSubCycle).end();
20     )
21     {
22         #include "alphaEqn.H"
23         rhoPhiSum += (runTime.deltaT()/totalDeltaT)*rhoPhi;
24     }
25
26     rhoPhi = rhoPhiSum;
27 }
28 else
29 {
30     #include "alphaEqn.H"
31 }
32
33 rho == alpha1*rho1 + alpha2*rho2;

```

UEqn.H: The velocity equation of the governing equation. This is coupled with pressure equation in “PEqn.H”.

```

1      fvVectorMatrix UEqn
2      (
3          fvm::ddt(rho, U)
4          + fvm::div(rhoPhi, U)
5          + turbulence->divDevRhoReff(rho, U)
6      );
7

```



```

8      UEqn.relax();
9
10     if (pimple.momentumPredictor())
11     {
12         solve
13         (
14             UEqn
15             ==
16             fvc::reconstruct
17             (
18                 (
19                     fvc::interpolate(interface.sigmaK())*fvc::snGrad(alpha1)
20                     - ghf*fvc::snGrad(rho)
21                     - fvc::snGrad(p_rgh)
22                 ) * mesh.magSf()
23             )
24         );
25     }

```

PEqn.H: The pressure equation of the governing equation as discussed in Chapter 2.2.2. “phiHbyA” in the source codes stands for $\frac{H(U)}{a_p}$. See the equations (2.35) ~ (2.39).

```

1      {
2          volScalarField rAU("rAU", 1.0/UEqn.A());
3          surfaceScalarField rAUf("rAUf", fvc::interpolate(rAU));
4
5          volVectorField HbyA("HbyA", U);
6          HbyA = rAU*UEqn.H();
7
8          surfaceScalarField phiHbyA
9          (
10             "phiHbyA",
11             (fvc::interpolate(HbyA) & mesh.Sf())
12             + fvc::interpolate(rho*rAU)*fvc::ddtCorr(U, phi)
13         );
14
15         adjustPhi(phiHbyA, U, p_rgh);

```

```

16
17     surfaceScalarField phig
18     (
19         (
20             fvc::interpolate(interface.sigmaK())*fvc::snGrad(alpha1)
21             - ghf*fvc::snGrad(rho)
22         )*rAUf*mesh.magSf()
23     );
24
25     phiHbyA += phig;
26
27     // Update the fixedFluxPressure BCs to ensure flux consistency
28     setSnGrad<fixedFluxPressureFvPatchScalarField>
29     (
30         p_rgh.boundaryField(),
31         (
32             phiHbyA.boundaryField()
33             - (mesh.Sf()).boundaryField() & U.boundaryField())
34         )/(mesh.magSf()).boundaryField()*rAUf.boundaryField()
35     );
36
37     while (pimple.correctNonOrthogonal())
38     {
39         fvScalarMatrix p_rghEqn
40         (
41             fvm::laplacian(rAUf, p_rgh) == fvc::div(phiHbyA)
42         );
43
44         p_rghEqn.setReference(pRefCell, getRefCellValue(p_rgh, pRefCell));
45
46         p_rghEqn.solve(mesh.solver(p_rgh.select(pimple.finalInnerIter())));
47
48         if (pimple.finalNonOrthogonalIter())
49         {
50             phi = phiHbyA - p_rghEqn.flux();
51
52             p_rgh.relax();
53
54             U = HbyA + rAU*fvc::reconstruct((phig - p_rghEqn.flux())/rAUf);
55             U.correctBoundaryConditions();
56             fvOptions.correct(U);
57         }
58     }
59
60     #include "continuityErrs.H"

```

```

61
62     p == p_rgh + rho*gh;
63
64     if (p_rgh.needReference())
65     {
66         p += dimensionedScalar
67         (
68             "p",
69             p.dimensions(),
70             pRefValue - getRefCellValue(p, pRefCell)
71         );
72         p_rgh = p - rho*gh;
73     }
74 }

```

References

- [1] Alberto Ghione, Development and Validation of a two-phase CFD model using OpenFOAM, Master Thesis, Royal Institute of Technology, Stockholm, Sweden, 2012
- [2] C. Lo, Dong-Taur Su and I-Fu Lin, Applying Computational Fluid Dynamics to Simulate Bank Effects, 19th International Offshore and Polar Engineering Conference, page 466~471, Osaka, Japan, 2009
- [3] D. W. Park, H. J. Choi and K. J. Paik, Bank Effect of a Ship Operating in a Shallow Water and Channel, Journal of Navigation and Port, Volume 38, No. 1, page 19~27, 2014
- [4] E. Lataire, M. Vantorre, E. Laforce, K. Eloot and G. Delefortrie, Navigation in Confined Waters: Influence of Bank Characteristics on Ship-Bank Interaction, International Conference on Ship Manoeuvring in Shallow and Confined Water, page 135~143, Abtwerp, Belgium, 2007
- [5] E. Lataire, M. Vantorre and K. Eloot, Systematic Model Tests on Ship-Bank Interaction Effects, International Conference on Ship Manoeuvring in Shallow and Confined Water: Bank Effect, Abtwerp, Belgium, 2009
- [6] Eric Furbo, Evaluation of RANS turbulence models for flow problems with significant impact of boundary layers, Master thesis, Uppsala University, Sweden, 2010
- [7] H. K. Versteeg and W. Malalasekera, An Introduction to Computational Fluid Dynamics, 2nd Edition, Pearson Education Limited, UK, 2007

- [8] Hirt C. W., Nichols B. D, Volume of Fluid (VoF) method for the dynamics of free boundaries, J. of Computer Physics, 1981, 39(1), page 201-225
- [9] Hrvoje Jasak, Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows, Thesis for Doctor of Philosophy, University of London and Imperial College, UK, 1996
- [10] Huaming Wang, Zaojian Zou, Yonghe Xie and Wanlun Kong, Numerical Study of Viscous Hydrodynamic Forces on a Ship Navigating near Bank in Shallow Water, Proceedings of the 20th International Offshore and Polar Engineering Conference, page 523~528, Beijing, China, 2010
- [11] Iranzo, J. Domingo, I. Trejo, M. Terceno and J. Valle, Calculation of the resistance and the profile of a 3600 TEU container ship, Canal de Experiencias Hidrodinamicas, El Pardo, Ministerio de Defensa, Barcelona, Spain, 2007
- [12] Issa, R.I., Solution of the implicitly discretized fluid flow equations by operator-splitting, J. Comp. Physics, 62:40-65, 1986
- [13] John D. Anderson, Computational Fluid Dynamics, The Basics with Applications, McGraw-Hill Inc., New York, USA, 1995
- [14] K. Lee and S. G. Lee, Investigation of ship maneuvering with hydrodynamic effects between ship and bank, Journal of Mechanical Science and Technology, No. 22, page 1230~1236, 2008
- [15] Kevin Maki, Ship Resistance Simulation with OpenFOAM, 6thOpenFOAM workshop, PA, USA, 2011

- [16] Kongsberg Maritime, Hydrodynamic Modelling Tool Report SU-0689, Mathematical Model of Ship Dynamics, Norway, 2012
- [17] Knut Inge Engelbreth, Flow Around a Circular Cylinder Near a Plate Wall, Master Thesis, Norwegian University of Science and Technology, Norway, 2011
- [18] L. Larsson, B. Regnstrom, L. Broberg, D. Li, C. Janson, Features, Fantasies, and Feats in the Theoretical/Numerical Prediction of Ship Performance, Proceedings of the 22nd Symposium on Naval Hydrodynamics, page 11-32, Washington D.C, USA, 1998
- [19] L. Zou, L. Larsson, G. Delefortrie and E. Lataire, CFD Prediction and Validation of Ship-bank Interaction in a Canal, 2nd International Conference on Ship Manoeuvring in Shallow and Confined Water: Ship to Ship Interaction, May 18-20, Trondheim, Norway, 2011
- [20] Liu Jing, Investigation of multiphase liquid system in microchannels, PhD thesis, School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore, 2014
- [21] M. Vantorre, and G. Delefortrie, Experimental Investigation of Ship-Bank Interaction Forces, Proceeding of International Conference on Marine Simulation and Ship Manoeuvrability, MARSIM'03, page RC-31-1~9, Kanazawa, Japan, 2003
- [22] Mehdi Nakisa, Adi Maimun, Yasser M. Ahmed, A.Y.Sian, A. Priyanto, Jaswaramd F. Behrouzi, RANS Simulation of the Viscous Flow around Hull of LNG Ship in Confined Water, 15th Mathematical and Computational Methods in Science and Engineering, page 133~137, Kuala Lumpur, Malaysia, 2013

- [23] Norrbin, Bank Effect on a Ship Moving through a Short Dredged Channel, Proceedings of 10th Symposium on Naval Hydrodynamics, page 71~88, Cambridge, MA, USA, 1974
- [24] OpenFOAM®, The Open source CFD toolbox, User Guide 2.3.0
- [25] OpenFOAM®, The Open source CFD toolbox, Programmers Guide 2.3.0
- [26] OpenFOAM® v2.0.0 official website about Steady-State VoF solver, <http://www.openfoam.org/version2.0.0/steady-VoF.php> (Last accessed in Jan 2015).
- [27] OpenFOAM® v2.3.0 official website about Multiphase modelling,
<http://www.openfoam.org/version2.3.0/mutiphase.php>
(Last accessed in Mar 2015).
- [28] OpenFOAM wiki website, <http://openfoamwiki.net> (Last accessed in May 2014).
- [29] Patankar, S.V., Numerical heat transfer and fluid flow, Hemisphere Publishing Corporation, Taylor & Francis Group, New York, USA, 1980
- [30] Pedro Miguel Borges Lopes, Free-surface flow interface and air-entrainment modelling using OpenFOAM, Doctorial Program Report, Universidade de Coimbra, Portugal, 2013
- [31] Q. Li, M. Leer-Andersen, P. Ottosson, P. Trägårdh, Experimental Investigation of Bank Effects Under Extreme Conditions, Proceeding of PRADS 2001, page 541-546, Shanghai, China, 2001

- [32] Quanming Miao and Jinzhu Xia, Numerical Study of Bank Effect on a Ship Traveling in a Channel, The 8th International Conference on Numerical Ship Hydrodynamics, Busan, Korea, 2003
- [33] Santiago Marquez Damian, An Extended Mixture Model for the Simultaneous Treatment of Short and Long Scale Interfaces, PhD Thesis, Universidad Nacional Del Litoral, Argentina, 2013
- [34] Schlichting H., Boundary Layer Theory, 7th edition, McGraw-Hill, USA, 1979
- [35] S. G. Lee, Ship motion and manoeuvring, Lecture Note, page 116~119, Department of Naval Architecture and Ocean Engineering, Pusan National University, Korea, 1998
- [36] The manoeuvring Committee of ITTC, Final Report and Recommendations to the 23rd ITTC, Proceedings of the 23rd International Towing Tank Conference - Volume I, page 201~234, Venice, Italy, 2002
- [37] The manoeuvring Committee of ITTC, Final Report and Recommendations to the 26th ITTC, Proceedings of the 26th International Towing Tank Conference - Volume I, page 129~154, Rio de Janeiro, Brazil, 2011
- [38] W. J. Kim, S. H. Van and D. H. Kim, Measurements of Flows Around Modern Commercial Ship Models, Experiments in Fluids, page 567~578, 2001

Publication from the present work

Youngkook Kim, E. Y. K. Ng and Gopala Krishnan, CFD Prediction of Ship-to-bank Interaction for Shiphandling Simulator, MARTECH 2014, Singapore, 29~30 April, 2014

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

CFD STUDY OF SHIP-TO-BANK INTERACTION

KIM YOUNGKOOK

SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING

2015

CFD Study of Ship-To-Bank Interaction

KIM YOUNGKOOK

School of Mechanical & Aerospace Engineering

A thesis submitted to the Nanyang Technological University

in fulfillment of the requirement for the degree of

Master of Engineering

2015

Acknowledgement

I would like to express my gratitude to Associate Professor Eddie Ng Yin Kwee who is my supervisor for his excellent guidance and keen support. Prof. Ng was always patient and supportive for my problems and questions during the research period.

I also would like to extend my gratitude to NTU HPC support team, especially Soh Hwee Jin (Melvin) and Bong Zoebir who used to fix the problems and kindly updated the newer version of OpenFOAM in the HPC server so that I could use OpenFOAM 2.3.0 for my research.

Furthermore, I would like to appreciate to my wife, Cho Eunjin who supported my study in all.

Mar 2015

Kim Youngkook

Table of Contents

ACKNOWLEDGEMENT	2
TABLE OF CONTENTS	3
LIST OF FIGURES	5
LIST OF TABLES	8
ABSTRACT.....	9
NOMENCLATURE	10
CHAPTER 1 INTRODUCTION.....	15
1.1 BACKGROUND	15
1.2 OBJECTIVES	18
1.3 SCOPE OF RESEARCH	19
1.4 LITERATURE REVIEW.....	20
1.4.1 <i>Experimental approaches of ship-to-bank interaction</i>	<i>20</i>
1.4.2 <i>Numerical approaches of ship-to-bank interaction.....</i>	<i>22</i>
1.5 ORDER OF REPORT	26
CHAPTER 2 NUMERICAL METHODS.....	28
2.1 CFD THEORY	28
2.1.1 <i>Potential flow.....</i>	<i>28</i>
2.1.2 <i>RANS (Reynolds Averaged Navier-Stokes) equations</i>	<i>29</i>
2.1.3 <i>Finite Volume Method (FVM).....</i>	<i>33</i>
2.1.4 <i>Volume of fluid method.....</i>	<i>36</i>
2.2 CFD ALGORITHM	38
2.2.1 <i>Numerical discretisation and solution algorithm.....</i>	<i>38</i>
2.2.2 <i>Pressure corrections</i>	<i>44</i>
2.2.3 <i>Convergence and residuals.....</i>	<i>48</i>
2.2.4 <i>Courant number</i>	<i>50</i>
2.2.5 <i>Boundary conditions.....</i>	<i>52</i>
2.2.6 <i>Boundary layer treatment and y plus.....</i>	<i>54</i>
2.2.7 <i>Good practice of boundary layer treatment.....</i>	<i>58</i>
2.3 MATHEMATICAL MODEL OF SHIP-HANDLING SIMULATOR	61

CHAPTER 3	OPEN SOURCE CFD CODE	66
3.1	OPEN SOURCE CFD CODES AND OPENFOAM	66
3.2	OPENFOAM SOFTWARE STRUCTURES	68
3.2.1	<i>Architecture</i>	68
3.2.2	<i>Solvers</i>	69
3.2.3	<i>Numerical algorithm</i>	70
3.3	MULTI-PHASE SOLVER INTERFOAM AND LTSINTERFOAM	73
3.4	OPENFOAM EVALUATIONS IN OPEN WATER	76
3.4.1	<i>KVLCC2</i>	79
3.4.2	<i>KCS</i>	83
3.4.3	<i>Summary of the evaluations</i>	94
CHAPTER 4	ANALYSIS IN CONFINED WATER	96
4.1	COMPUTATIONAL SET-UP AND MESH	96
4.2	BOUNDARY CONDITIONS	105
CHAPTER 5	RESULTS AND DISCUSSION	109
5.1	COMPUTATION RESULTS	109
5.1.1	<i>Force and moment</i>	109
5.1.2	<i>Under keel space modelling of the case H</i>	116
5.1.3	<i>Pressure and velocity asymmetry</i>	122
5.1.4	<i>Results of the cases with propeller rotation</i>	128
5.1.5	<i>Comparison between single-phase and two-phase solutions</i>	129
5.1.6	<i>Free surface elevation</i>	130
5.2	DISCUSSION	136
CHAPTER 6	CONCLUSION	139
6.1	CONCLUSION	139
6.2	FUTURE WORK	141
APPENDIX	THE SOURCE CODES OF INTERFOAM	142
REFERENCES		152
PUBLICATION FROM THE PRESENT WORK		157

List of Figures

Figure 2-1: Finite volume discretisation [21].....	35
Figure 2-2: Explicit (upper) and implicit (upper) method [13]	43
Figure 2-3: Comparison of implicit and explicit solutions for same time step [7]	51
Figure 2-4: Velocity distribution near a solid wall [34]	57
Figure 2-5: Coordinate system of manoeuvring mathematical model	62
Figure 2-6: Six degrees of freedom of motion	63
Figure 2-7: The trajectory in ship-handling simulator with ship- to-bank interaction x and y axis: distance(m) from the initial point.....	65
Figure 3-1: OpenFOAM case structure.....	68
Figure 3-2: Profile and front view of KVLCC2.....	79
Figure 3-3: Convergence history of the resistance of KVLCC2	81
Figure 3-4: Comparison of wave elevation along the hull of KVLCC2 (Experiment result: Kim et al. [38])	82
Figure 3-5: Comparison of wave pattern of KVLCC2 of CFD (upper) and the experiment (Kim et al. [38]) (bottom)	83
Figure 3-6: Profile and front view of KCS.....	84
Figure 3-7: Mesh of KCS - (a) whole domain, (b) forward and (c) stern area	85
Figure 3-8: y^+ distributions on the hull.....	86
Figure 3-9: Convergence time history of the resistance (N) of KCS by (a) Gauss linear and (b) Least square gradient scheme	90
Figure 3-10: Comparison of wave elevation along the KCS hull (Experiment results: Kim et al. [38])	91

Figure 3-11: Comparison of wave pattern of KCS of OpenFOAM (upper) and the experiment (Kim et al., [38]) (bottom)	91
Figure 3-12: Wave pattern at the forward (a) and the stern (b)....	92
Figure 3-13: Wake distribution (axial velocity [m/s]) of KCS at (a) the propeller plane and (b) at the station no. 1 (left : experiment (Kim et al. [38]), right : OpenFOAM result)	94
Figure 4-1: The coordinate system.....	97
Figure 4-2: Schematic view of the bank arrangement (looking from the aft).....	97
Figure 4-3: Graphical view of the case C and the case H.....	100
Figure 4-4: Mesh of (a) whole domain and (b) near the hull (looking from the top), and (c) mesh on the hull surface of case H	103
Figure 5-1: Convergence history of (a) X and (b) Y forces (N), and (c) Z moment (Nm) from pressure of case F (Gradient scheme: Gauss linear, Max Co. reduced from 70s)	110
Figure 5-2: Convergence history of (a) X and (b) Y forces (N), and (c) Z moment (Nm) from pressure of case F (Gradient scheme: Least square)	111
Figure 5-3: Different meshing of case H - no layer (a), three layers (b), eight layers (c), fine mesh (d)	117
Figure 5-4: y^+ of the case H with different meshing no layer (a), three layers (b), eight layers (c), fine mesh (d)	118
Figure 5-5: Pressure [N] distribution in the water at the centreline of the case H with (b) three layers (c) eight layers	120
Figure 5-6: Pressure [N] distribution on the hull (looking from the bottom) of the case H with (b) three layers (c) eight layers	121

Figure 5-7: Pressure [N] distribution on the hull and bank surface of the cases F (a) and H (b) (looking from the top)	122
Figure 5-8: Dynamic pressure [N] distribution of OpenFOAM calculation of case A (a), F (b) and H (c) at the different sections (station 2.5, amidships and station 17.5)	124
Figure 5-9: Velocity magnitude (m/s) of the case F at (a) free surface, (b) 2/3 of draught from baseline, (c) 1/3 of draught from baseline and (d) baseline	125
Figure 5-10: Velocity magnitude (m/s) of the case H at (a) free surface, (b) 2/3 of draught from baseline, (c) 1/3 of draught from baseline and (d) baseline	126
Figure 5-11: Velocity magnitude (m/s) of the case I at (a) free surface, (b) 2/3 of draught from baseline, (c) 1/3 of draught from baseline and (d) baseline	127
Figure 5-12: The locations of the wave gauges 1, 2 and 3 of the cases (a) F, (b) I and (c) J	131
Figure 5-13: Wave elevation of the case F at (a) gauge 1, (b) gauge 2 and (c) gauge 3 (Lataire et al. [5])	132
Figure 5-14: Wave elevation of the case I at (a) gauge 1, (b) gauge 2 and (c) gauge 3 (Lataire et al. [5])	133
Figure 5-15: Wave elevation of the case J at (a) gauge 1, (b) gauge 2 and (c) gauge 3 (Lataire et al. [5])	134

List of Tables

Table 3-1: Principal dimensions of KVLCC2 model.....	79
Table 3-2: Grid convergence study (The experiment result: Kim et al. [38])	80
Table 3-3: Principal dimensions of KCS model.....	83
Table 3-4: The comparison of KCS resistance by different solvers (Experiment results: Kim et al. [38]).....	88
Table 4-1: Principal dimension of model ship	96
Table 4-2: The test cases without propeller thrust.....	99
Table 4-3: The test cases with propeller thrust	101
Table 4-4: Boundary conditions for confined water analysis of the single-phase solver	106
Table 4-5: Boundary conditions for confined water analysis of the multi-phase solver	107
Table 5-1: Comparison of the cases without thrust to the experiment results (Lataire et al. [5])	114
Table 5-2: Comparison between different meshing in the under keel space of the case H.....	120
Table 5-3: Comparison of the cases with thrust to the experiment results (Lataire et al. [4]).....	128
Table 5-4: The locations (m) of the wave gauges with referenrece to the centreline of the ship (Lataire et al. [5])	130

Abstract

Ship-to-bank interaction is a complex physical phenomenon that involves not only in the asymmetric pressure field near banks or channels but also shallow water effect. Traditionally many experimental studies were carried out in this field. As numerical method is getting popular, there were various computational approaches as well. In this study, CFD (Computational Fluid Dynamics) analysis is carried out to investigate ship-to-bank interaction. CFD analysis is also useful for finding hydrodynamic coefficients in manoeuvring mathematical model of the ship-handling simulator. For the numerical method, RANS based solvers with free surface capturing feature are used.

Flow around a container ship in confined water is investigated with the open source C++ toolbox, OpenFOAM. Computations with several bank arrangements and different settings are performed. The OpenFOAM results are also compared with experiment results for validation. Meshing, numerical scheme, numerical settings and calculation results are evaluated and discussed in this study.

Nomenclature

E	Integration constant of the roughness of the wall	-
Fn	Froude number ($Fn = U/\sqrt{gL_{pp}}$)	-
$F (F_i)$	Total force (in i coordinate)	N
g	Acceleration of gravity	m/s ²
H	H Operator (See Section 2.2.2)	-
I	Tensor matrix of inertia of body	kg·m ²
K	Roll moment	Nm
k	Sum of the mean kinetic energy of turbulence	m ² /s ²
L_{pp}	Length between perpendiculars	m
$M (M_i)$	Total moment (in i coordinate)	Nm
M	Pitch moment	Nm
$\dot{M}$	Mass flux	kg/(m ² ·s)
N	Yaw moment	Nm
p	Pressure	m ² /s ²
p_{rgh} (p_d)	Dynamic pressure	kg/(m·s ²)
S_f	Surface area of the cell face	m ²
t	time	s
U	Speed vector (u, v, w)	m/s

u_τ	Friction velocity	m/s
u^+	Non-dimensional velocity near the wall	-
w	Artificial velocity normal to the interface of volume fraction	m/s
X	Resistance / Longitudinal force	N
x_0	Origin of measurement	m
x_f	Distance from origin to the face centre	m
Y	Transverse force / Sway force	N
y_B	Distance from bank wall to the centreline of ship	m
y_p	Distance from the wall to the first grid point	m
y^+	Non-dimensional y_p	-
Z	Vertical force / Heave force	N
Γ	Diffusion coefficient	m ² /s
α	Scalar field of volume fraction	-
γ	Under-relaxation factor	-
ε	Rate of viscous dissipation of turbulent kinetic energy	m ² /s ³
κ	Surface curvature	1/m
κ	Von Karman's constant	-
μ	Dynamic viscosity	kg/(m·s)
ν	Kinematic viscosity	m ² /s

ρ	Density of fluid	kg/m^3
σ	Surface tension coefficient	N/m
τ_{ij}	Reynolds stress	$\text{kg}/(\text{m}\cdot\text{s}^2)$
τ_w	Wall shear stress	$\text{kg}/(\text{m}\cdot\text{s}^2)$
ϕ	Velocity potential	m^2/s
Ω	Angular velocity vector (p, q, r)	rad/s
ω	Turbulent frequency	$1/\text{s}$

Abbreviations

CFD	Computational Fluid Dynamics
CFL	Courant Friedrichs Lewy
CPU	Central Processing Unit
CV	Control Volume
DOF	Degree(s) of Freedom
EXP	Experiment
FDM	Finite Difference Method
FHR	Flanders Hydraulics Research
FVM	Finite Volume Method
HPMM	Horizontal Planar Motion Mechanism
KCS	KRISO Container Ship
KRISO	Korea Research Institute of Ship and Ocean engineering
KVLCC	KRISO Very Large Crude-oil Carrier
LES	Large Eddy Simulation
LNG	Liquefied Natural Gas
LTS	Local Time Stepping
MOERI	Maritime & Ocean Engineering Research Institute
MULES	Multi-dimensional Universal Limiter for Explicit Solution
ODE	Ordinary Differential Equation

OpenFOAM	Open Field Operation And Manipulation
OS	Operating System
PDE	Partial Differential Equation
PIMPLE	(Merged method of) PISO and SIMPLE
PISO	Pressure Implicit with Splitting of Operators
RANS	Reynolds Averaged Navier-Stokes (equation)
RPM	Revolution Per Minute
SIMPLE	Semi-Implicit Method for Pressure-Linked Equations
SST	Shear Stress Transport
STL	STereo-Lithography
TEU	Twenty-foot Equivalent Unit
VoF	Volume of Fluid

Chapter 1

Introduction

1.1 Background

When a ship navigates in the proximity of banks or harbours, the pressure field near the ship becomes asymmetric and ship experiences repelling or attracting forces. It makes hard for the ship to manoeuvre and can lead to an accident. In such cases, it is advised to travel at the middle of water to minimise any interaction from the bank or the harbour. However, as this is not always possible, it is important to understand the phenomenon to prevent any dangerous situation. Basically ship-to-bank interaction causes sway force and yaw moment on the ship when separation from the bank wall is small. Moreover, small under keel clearance in confined water makes the interaction even stronger due to the shallow water effect. In addition, there are several other factors that contribute such as propeller effect and bank shape. Because ships have been increased in size and length in recent years, the importance of prediction and understanding of ship-to-bank interaction are thus getting more significant.

Nowadays, ship-handling simulators play an important role in training captains and crew for safe operation and preventing from accidents. In the mathematical model of the ship-handling simulator, various forces and moments that arise from hull, rudder, propeller, wind, wave, bank effect, etc have to be carefully modelled for realistic simulations. The best way to obtain those forces and moments is to carry out model tests. Captive model test such as HPMM (Horizontal Planar Motion Mechanism) test enables measuring of each force and moment components. However, the test is costly and time consuming. Therefore, it is normal to use empirical methods that are based on the regression formulas from database to estimate these forces and moments. When the accuracy is concerned, CFD (Computational Fluid Dynamics) analysis could give better accuracy. Nowadays, it is not much to say that CFD technology became essential in marine industry. As CFD technology is developing, the accuracy and reliability of CFD analysis are getting higher. There were many studies to calculate hydrodynamic forces and moments of mathematical model of ship-handling simulator using CFD analysis.

As for ship-to-bank interaction, it is quite straightforward to use CFD calculations and model them into ship-handling simulator. However, careful approach is needed to calculate and model the

complex phenomenon due to the interaction with banks and shallow water effect. Recently, open source CFD codes have become popular due to their strong capability. They are totally free of charge to use and free to modify the source code. On the other hands, it is a challenge to evaluate the accuracy of the code despite the steep learning curve of open source codes. In this study, it is decided to use an open source CFD code to analyse ship-to-bank interaction from various bank arrangements and compare them to the experiment results.

1.2 Objectives

This study aims to predict ship-to-bank interaction with an open source CFD code. The numerical analysis of multiphase RANS solver will be evaluated so that the effect from free surface, turbulence and viscosity can be taken into account. In addition, the results will be compared with single-phase RANS solver. Making use of the open source CFD code, it is expected to know capability and limitation of the open source code. Moreover, it would be helpful to understand numerical algorithms of CFD codes.

By performing numerical simulation, ship-to-bank interaction with different bank arrangements will be investigated. This would also identify the benefit from CFD analysis compared to the experimental approach.

1.3 Scope of research

The open source CFD code, OpenFOAM will be used for numerical analysis. Among numerous solvers in OpenFOAM package, a VoF multiphase solver will be used to predict ship-to-bank interaction. The solver will be evaluated for open water cases before analysis is carried out in confined water. Resistance of a container ship and a large tanker ship will be calculated and compared to the experiment results.

The OpenFOAM solver will be used to calculate ship-to-bank interaction in confined water and the results will be compared to the experiment. CFD calculations will cover from the comparisons of forces and moment to the evaluation of meshing, modelling and wave elevation.

1.4 Literature Review

1.4.1 Experimental approaches of ship-to-bank interaction

From 1970s to these days, many experimental studies have been carried out to investigate ship-to-bank interaction and researchers developed empirical formulas that could predict ship-to-bank interaction. During 1970s, SSPA (Swedish maritime research centre) carried out various tank tests for different bank types and showed how the forces and moments vary over transient period. It was found that sway force and yaw moment vary in proportion to the square of ship speed. Using this test results, empirical formulas were also suggested from model test results (Norrbin [23]).

However, some studies that were carried out later showed that ship-to-bank effect occurs in a more complex manner. A continued study done about 30 years later in SSPA (Li et al [31]) proved that there is a point where the sway force is switched from suction to repulsive force when under keel clearance is extremely small, approximately at about 10% of water draught. Moreover, unlike the study in early times, it turns out that the sway force tends to increase in a higher order than two with respect to the ship speed when under keel clearance is very small.

Even in some cases, it became more than order of two at 40% of under keel clearance. At a sloped bank with 30 degrees angle, it tends to reduce force and moments at low speeds but bank effects increase at high speeds as compared to a vertical bank at a low under keel clearance. Moreover, a suction force arises from a rotating propeller compared to the cases without propeller according to the experiments. It was found the more propeller loading caused the larger contribution of suction force.

Expansive shallow water tank tests were carried out at FHR (Flanders Hydraulics Research) in Belgium (Vantorre et al. [21]) (Lataire et al. [4] [5]). More than 10,000 model tests of the bank effect were performed and many factors that influence the bank effect were investigated. Through the experiments, it was confirmed that there is a transition point from attraction sway force to repulsion force in shallow water. Small difference from the test results of SSPA is that it happened at slightly higher under keel clearance which is in range of 15% to 20%. Therefore, it is understood that the transition from attraction to repulsion force appears at the under keel clearance of 10~20% water draught. The test results also showed that propeller action increased the bank effect. Similarly to the results of SSPA, propeller action changed repulsion force into suction force in shallow water where under keel clearance is less than 35% of

water draught. These tests results are also developed into empirical formulas. Several sets of test data are made available for the bench marking purpose of simulation model (Lataire et al. [5]). In this study, these open test results are used for the comparison to the CFD calculations.

1.4.2 Numerical approaches of ship-to-bank interaction

As numerical technique was developing, there were many computational studies on the bank effect. In early times, numerical method based on potential theory was popular in marine industry.

In many studies, panel method was used to evaluate the bank effect. When the free surface condition is linearised by double model solution, it was good enough to show qualitative predictions only when the draught and water depth ratio is not too small, such that the ratio is greater than 1.5. (Q. Miao and J. Xia [32] and Lee and Lee [14]). When non-linear boundary condition is applied, panel method was also able to calculate ship-to-bank interaction in extreme shallow water with qualitative agreement (Park et al. [38]). With this non-linear potential method, even the transition point from suction force to repulsion force was found at 40% of the draught-water depth

ratio which is slightly higher than the experiment results reported by Li et al. [31] and Lataire et al. [4].

As computational capability is rapidly improving, RANS based finite volume methods were also getting popular. However, it was quite recent that RANS based method started to be used in this topic.

Lo et al. [2] carried out transient analysis on the bank effect. Trajectory of KCS container travelling near the bank was illustrated. Sway force and yaw moment with time variations at various velocities were calculated at about 30% of the draught-water depth ratio. However, there was no comparison with experiment result.

Wang et al. [10] studied viscous analysis around series 60 hull with SST $k-\omega$ turbulence model from 50% to 1000% of the water depth – draught ratio and showed good agreement with experiments in sway forces but over prediction with the very short distance to the bank wall.

Zou et al. [19] used validation and verification method including a grid convergence study to evaluate RANS based CFD prediction of ship-to-bank interaction. The bank effect between KVLCC hull and a sloped bank was analysed with an overlapping grid of

structured mesh and $k-\omega$ SST turbulence model. CFD calculations in the comparison with model tests were done at 50%, 35% and even 10% of the draught-water depth ratio. Furthermore, not only sway force and yaw moment but also rolling, sinkage and trim moment were calculated and compared with potential flow analysis. Because sinkage and trim moment did not show big difference between the potential method and CFD calculation result, free surface effect at slow speed is considered to be insignificant for sinkage and trim. In the study, slip boundary condition was applied on the bank wall instead of no slip condition. The rationale for this boundary condition is not clear and the author also stated that the boundary condition may not be the accurate. The bank effect analysis gave good agreement in tendencies of yaw and roll moments but with under prediction of sway force.

Mehdi et al. [22] carried out simulation of the viscous flow around LNG ship using RANS method with unstructured hybrid mesh and $k-\omega$ turbulence model. Yaw moment was compared with the experiment results at various velocities and distance from the bank wall but sway force results were not presented in their paper. Calculated yaw moments at different distance from the bank wall and 60% of the draught-water depth ratio were

compared to the tank test results and showed qualitative agreement with some under prediction.

From the various studies done in the past, it is still a big challenge to use CFD for ship-to-bank interaction analysis with extreme shallow water. Many studies failed to calculate extreme conditions such as 10% of under keel clearance or did not predict ship-to-bank correctly.

During the literature review about numerical studies of the bank effect, viscous analysis including free surface effect on the bank effect was not found so far. When a ship travels at slow speed, it is no doubt that free surface elevation is not so high and the importance from free surface might be relatively low. Furthermore, it was found that free surface also does not give significant contribution on the sinkage and the trim at low speed (Zou et al. [19]). Nevertheless, it is believed to be worthwhile to carry out viscous flow analysis with free surface for more accurate prediction of the flow near the proximity of banks. It is also expected to see the difference between the analysis with and without free surface.

1.5 Order of report

The structure of this report is as follows:

Chapter 1 introduces overview of this research and ship-to-bank interaction. It also includes literature review that covers ship-to-bank interaction study done both in experimental and numerical approaches.

Chapter 2 covers numerical methods of CFD including basic theory. It discusses numerical algorithms, boundary condition and boundary layer theory. Moreover, mathematical model of ship-handling simulator will be explained.

Chapter 3 introduces open source CFD codes and the solver used for this study. Moreover, the evaluation of the solver in open water is included.

Chapter 4 outlines numerical setup of CFD analysis in confined water and this covers computational setup and boundary conditions of CFD calculation.

Chapter 5 deals with the results of CFD calculations in confined water and discussions. Various comparisons and evaluations will be carried out.

Chapter 6 includes the summary of current work and discusses the future plan.

Appendix contains the main source codes of Open source CFD tool that was used for this study.

Lastly, References and Publication of this study are listed.

Chapter 2

Numerical Methods

2.1 CFD theory

2.1.1 Potential flow

In marine industry, traditional panel method based on potential theory as assumptions was popular and it is still used by many people due to its simplicity and quick computation in spite of inviscid and irrotational property of flow. High Reynolds number of many marine applications also enables potential flow remains useful and efficient for the area where viscous effect is not predominant. Nevertheless, potential flow is not applicable where the flow is dominated by viscous drag and turbulence, e.g. near the aft hull. Basic theory of potential flow will be briefly explained here.

In potential flow, velocity potential (ϕ) exists and governing equation is Laplace equation of continuity,

$$\nabla^2 \phi = 0 \quad (2.1)$$

This method must satisfy several boundary conditions.

Firstly, no flux on the body and the wall as such,

$$\nabla\phi \cdot \vec{\eta} = 0 \quad (2.2)$$

Secondly, fluid particle should not leave the free surface which is kinetic boundary condition. Thirdly, pressure on the free surface should be the same as atmospheric pressure, which is dynamic boundary condition.

Normally, the last two boundary conditions have non-linear terms and they are often linearised for simplification. One of the traditional methods is double model solution, which reflects hull form at waterline of calm water. With double model approximation, flow with rigid free surface can be calculated. Nowadays, non-linear potential method is also available to describe free surface elevation. Non-linear potential method is very useful and fast, hence, popular in minimising wave breaking resistance and ship hullform optimisations.

2.1.2 RANS (Reynolds Averaged Navier-Stokes) equations

When viscosity and turbulence cannot be ignored in the flow, potential method is no more accurate. The most popular method

is RANS equations with turbulence model and it gives better accuracy than potential flow that is irrotational and inviscid.

Basically turbulence has fluctuating property and needs very fine mesh with huge computing resources to solve it directly. RANS method takes average of fluctuating properties of turbulent flow and can save computational effort, memory and time. Therefore, RANS method is most appropriate for industrial cases which require huge computation mesh or domain.

The governing equations generally come from the continuity equation (2.3) and the momentum equations (2.4) ~ (2.5) of incompressible flow.

$$\text{div } \mathbf{u} = 0 \quad (2.3)$$

$$\frac{\partial u}{\partial t} + \text{div}(\mathbf{u}\mathbf{u}) = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \text{div}(\text{grad}(\mathbf{u})) \quad (2.4)$$

$$\frac{\partial v}{\partial t} + \text{div}(\mathbf{v}\mathbf{u}) = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \text{div}(\text{grad}(\mathbf{v})) \quad (2.5)$$

$$\frac{\partial w}{\partial t} + \text{div}(\mathbf{w}\mathbf{u}) = -\frac{1}{\rho} \frac{\partial p}{\partial z} + \nu \text{div}(\text{grad}(\mathbf{w})) \quad (2.6)$$

The first term on the left hand side of the momentum equations represents the time derivatives and is also called as unsteady

term. The second term is the convection term. On the right hand side, the first and the second terms are pressure term and viscous term, respectively.

RANS equations can be derived from the momentum equations by taking the time average.

Velocity vector and pressure vector in turbulent flow can be expressed as sum of steady mean component (U, V, W and P) and fluctuating component (u', v', w' and p').

$$u = U + u' \quad v = V + v' \quad w = W + w' \quad p = P + p' \quad (2.7)$$

As integration or differentiation of fluctuating flow is similar to time averaging operation, so the followings can be achieved.

$$\begin{aligned} \overline{\frac{\partial u}{\partial t}} &= \frac{\partial U}{\partial t} & \overline{\text{div}(\mathbf{u}\mathbf{u})} &= \text{div}(U\mathbf{U}) + \overline{\text{div}(\mathbf{u}'\mathbf{u}')} \\ -\frac{1}{\rho} \frac{\partial \overline{p}}{\partial x} &= -\frac{1}{\rho} \frac{\partial P}{\partial x} & \overline{\nu \text{div}(\text{grad}(\mathbf{u}))} &= \nu \text{div}(\text{grad}(U)) \end{aligned} \quad (2.8)$$

Therefore, momentum equations can be written as time averaged momentum equations. e.g. x-momentum equation becomes,

$$\frac{\partial U}{\partial t} + \text{div}(U\mathbf{U}) + \text{div}(\overline{u'\mathbf{u}'}) = -\frac{1}{\rho} \frac{\partial P}{\partial x} + \nu \text{div}(\text{grad}(U)) \quad (2.9)$$

Move the third term on the left hand side to the right hand side as it plays a role as turbulent stresses, then we have:

$$\frac{\partial U}{\partial t} + \text{div}(U\mathbf{U}) = -\frac{1}{\rho} \frac{\partial P}{\partial x} + \nu \text{div}(\text{grad}(U)) + \frac{1}{\rho} \left[\frac{\partial(-\rho \overline{u'^2})}{\partial x} + \frac{\partial(-\rho \overline{u'v'})}{\partial y} + \frac{\partial(-\rho \overline{u'w'})}{\partial z} \right] \quad (2.10)$$

$$\frac{\partial V}{\partial t} + \text{div}(V\mathbf{U}) = -\frac{1}{\rho} \frac{\partial P}{\partial y} + \nu \text{div}(\text{grad}(V)) + \frac{1}{\rho} \left[\frac{\partial(-\rho \overline{u'v'})}{\partial x} + \frac{\partial(-\rho \overline{v'^2})}{\partial y} + \frac{\partial(-\rho \overline{v'w'})}{\partial z} \right] \quad (2.11)$$

$$\frac{\partial W}{\partial t} + \text{div}(W\mathbf{U}) = -\frac{1}{\rho} \frac{\partial P}{\partial z} + \nu \text{div}(\text{grad}(W)) + \frac{1}{\rho} \left[\frac{\partial(-\rho \overline{u'w'})}{\partial x} + \frac{\partial(-\rho \overline{v'w'})}{\partial y} + \frac{\partial(-\rho \overline{w'^2})}{\partial z} \right] \quad (2.12)$$

These are called Reynolds Averaged Navier-Stoke (RANS) equations of incompressible flow.

The additional stress terms on the right hand side are named as Reynolds stresses, and they have three normal stresses

$$\tau_{xx} = -\rho \overline{u'^2} \quad \tau_{yy} = -\rho \overline{v'^2} \quad \tau_{zz} = -\rho \overline{w'^2} \quad (2.13)$$

and three shear stresses.

$$\tau_{xy} = \tau_{yx} = -\rho \overline{u'v'} \quad \tau_{xz} = \tau_{zx} = -\rho \overline{u'w'} \quad \tau_{yz} = \tau_{zy} = -\rho \overline{v'w'} \quad (2.14)$$

Turbulence fluctuations require extra terms in the time averaged or Reynolds averaged equations, and they have to be properly modelled with turbulence models. Therefore, RANS equations focus on the properties of the mean flow rather than computing fluctuating turbulent velocity directly. This can save numeral cost significantly.

2.1.3 Finite Volume Method (FVM)

Finite Volume Method (FVM) is one of the discretisation methods that complex equations can be solved with finite space using control volumes. Governing equations such as RANS equations include PDEs (Partial Differential Equations) and a non-linear term, therefore, it is very hard to solve them directly. Hence, it is important to apply appropriate discretization method to solve those equations for complex geometry. The discretisation process includes discretisation of the calculation domain and equations. The solution domain, which is generally physical space, is divided into a finite number of cells which are termed as control

volumes. Equation discretisation transforms partial differential equations into sets of linear algebraic equations and they are applied on the computational mesh. The equations should be also discretised for a finite number of time steps, if transient simulation is required.

The control volumes fill the computational domain without overlapping each other. In finite volume method, by principal, all the numerical properties are computed at the centroid of cells although they use the values on the vertices or faces during the computations. Cells are connected to each other by faces which are the boundary surfaces of each cell. The faces also form the boundaries of the computational space domain and called boundary faces. Others inside the domain are called internal faces. Figure 2-1 shows parameters in finite volume discretisation. Here f stands for the cell face, P the cell centre, N centre of the neighbouring cell, d is orthogonal direction through the cell face and S_f is the surface area of the cell face.

Figure 2-1: Finite volume discretisation [21]

The advantages of FVM are firstly, mass, momentum and energy are well conserved. Secondly, it can also handle unstructured mesh compared to traditional FDM (Finite Difference Method). The unstructured mesh is getting popular because it can handle various shapes more easily and efficiently. Thirdly, the solvers are efficient for iterative methods.

Disadvantage is the occurrence of false diffusion when numerical scheme is simple or mesh is not fine enough. For this reason, large computational resources should be applied to industrial cases, however, high computational cost of FVM is getting affordable due to the development of computer CPU and parallel computing nowadays.

2.1.4 Volume of fluid method

In many of marine CFD applications, prediction of free surface is very important. Traditionally, potential flow was popular for the free surface flow. Nowadays, viscous flow analysis with free surface is common as the computer performance is getting powerful.

There are several numerical modelling techniques to describe the interface of multiphase systems which deal with water and air together. As for free surface capturing techniques in finite volume method, VoF (Volume of Fluid) and level set method are most popular – even the two methods can be combined. VoF method that was introduced by Hirt and Nichols [8] is a free surface modelling technique applying volume fraction to convection schemes. It uses a scalar field to express the volume fraction (α) in whole domain, which represents water when it is 1 and air when it is 0, hence, the region where the volume fraction is between 0 and 1 becomes the free surface. It satisfies Eulerian material derivative.

$$\frac{D\alpha}{Dt} = 0 \quad (2.15)$$

, and this is

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot \mathbf{u} \alpha = 0 \quad (2.16)$$

VoF method is known to show high accuracy in the three dimensional applications. However, the mesh has to be very fine in order to capture the free surface properly as α is transitional from 0 to 1. Otherwise the interface will show smearing results. As a remedy, OpenFOAM uses modified governing equation for α by adding an artificial compression velocity which is a vector normal to the interface.

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot \mathbf{u} \alpha + \nabla \cdot \mathbf{w}(\alpha(1 - \alpha)) = 0 \quad (2.17)$$

As shown in the formula, the artificial term goes to zero if α goes to zero or one, which means that the artificial term does not affect to the result where the phase is completely water or air. The term only works within the interface zone to make it possible to describe sharp free surface by working to the normal direction to the interface. Magnitude of the compression, $\mathbf{w}$ (*cAlpha* in OpenFOAM) can be used to turn on or off the compression term.

2.2 CFD algorithm

In this chapter, brief algorithm and numerical techniques are described. All the CFD theory in this chapter may be limited to the followings:

- Finite Volume Method
- Incompressible Navier-Stokes equations with Mach number < 1
- General fluid flow simulations

2.2.1 Numerical discretisation and solution algorithm

The generic form of the transport equation for a fluid property ϕ is like the following:

$$\frac{\partial}{\partial t}(\rho\phi) + \text{div}(\rho u\phi) = \text{div}(\Gamma \text{grad } \phi) + S_\phi \quad (2.18)$$

Here, u is the velocity vector, ρ is the density, Γ is the diffusion coefficient and S_ϕ is a source term. For the simplicity, a one-dimensional case with a uniform grid is explained here. For the finite volume discretization, integrating this equation over the control volume (CV) and time,

$$\begin{aligned}
& \int_t^{t+\Delta t} \int_{CV} \frac{\partial}{\partial t} (\rho \phi) dV dt + \int_t^{t+\Delta t} \int_{CV} \text{div}(\rho u \phi) dV dt \\
& = \int_t^{t+\Delta t} \left[\int_{CV} \text{div}(\Gamma \text{grad } \phi) dV + \int_{CV} S_\phi dV \right] dt \quad (2.19)
\end{aligned}$$

Using the Gauss' theorem (2.20), the volume integrals of divergent terms can be expressed as surface integrals.

$$\int_V \nabla \phi dV = \oint_A \phi dS \quad (2.20)$$

Re-arranging time integration and volume integration, this can be written as follows:

$$\begin{aligned}
& \int_{CV} \int_t^{t+\Delta t} \frac{\partial}{\partial t} (\rho \phi) dt dV + \int_t^{t+\Delta t} \int_A n \cdot (\rho u \phi) dA dt \\
& = \int_t^{t+\Delta t} \left(\int_A n \cdot \left(\Gamma \frac{d\phi}{dx} \right) dA \right) dt + \int_t^{t+\Delta t} \int_{CV} S_\phi dV dt \quad (2.21)
\end{aligned}$$

Here, n is the unit vector normal to the face and A stands for the area of the cell face.

All the terms in the equation have to be approximated by various schemes in finite space and time to discretise the equations. The first term on the left hand side in the unsteady term becomes,

$$\int_{CV} \int_t^{t+\Delta t} \frac{\partial}{\partial t} (\rho\phi) dt dV = \int_{CV} \rho(\phi^n - \phi^0) dV = \rho(\phi_P^n - \phi_P^0) V_P \quad (2.22)$$

where 0 denotes current time step, n new value in the next time step, p cell point, V_P is the control volume and

$$\begin{aligned} \phi^n &= \phi(t + \Delta t) \\ \phi^0 &= \phi(t) \end{aligned} \quad (2.23)$$

The second term which is convection term and can be discretised as,

$$\int_t^{t+\Delta t} \int_A n \cdot (\rho u \phi) dA dt = \int_t^{t+\Delta t} (F\phi_E - F\phi_W) dt \quad (2.24)$$

F stands for the mass flux passes through the face and defined as $F = A(\rho u)$. E and W stand for the nodal point to the east and the west, respectively.

Depending on the time integrations whether defining $\int_{\Delta t} \phi_P dt \approx \phi_P^0$ or ϕ_P^n , this becomes,

$$[F\phi_E^n - F\phi_W^n]\Delta t \quad \text{or} \quad [F\phi_E^0 - F\phi_W^0]\Delta t \quad (2.25)$$

Similarly, the third term which is diffusion term becomes:

$$\int_t^{t+\Delta t} \left(\int_A n \cdot \left(\Gamma \frac{d\phi}{dx} \right) dA \right) dt = \int_t^{t+\Delta t} \left[\Gamma A \left(\frac{\phi_E - \phi_P}{\delta x} \right) - \Gamma A \left(\frac{\phi_P - \phi_W}{\delta x} \right) \right] dt \quad (2.26)$$

Then integrate this in time gives:

$$\begin{aligned} & \left[\Gamma A \left(\frac{\phi_E^0 - \phi_P^0}{\delta x} \right) - \Gamma A \left(\frac{\phi_P^0 - \phi_W^0}{\delta x} \right) \right] \Delta t \\ & \text{or} \\ & \left[\Gamma A \left(\frac{\phi_E^n - \phi_P^n}{\delta x} \right) - \Gamma A \left(\frac{\phi_P^n - \phi_W^n}{\delta x} \right) \right] \Delta t \end{aligned} \quad (2.27)$$

The fourth term is the source term and becomes:

$$\begin{aligned} \int_t^{t+\Delta t} \int_{CV} S_\phi dV dt &= \int_t^{t+\Delta t} \bar{S} V_P dt = \int_t^{t+\Delta t} (S_u V_P + S_P \phi_P V_P) dt \\ &= (S_u V_P + S_P \phi_P^0 V_P) \Delta t \quad \text{or} \quad (S_u V_P + S_P \phi_P^n V_P) \Delta t \end{aligned} \quad (2.28)$$

Finally, discretization in time can express the equations in two different ways depending on how we choose unknowns from the different time step.

The convection and diffusion terms are simplified by taking surface integrals over the cell and by using Gauss theorem.

These surface integrals can be expressed by the summation of the mass flux through the faces to describe general form.

$$\int_V \nabla \phi dV = \sum_f \left(\int_f \phi dS \right) \approx \sum_f A \phi \quad (2.29)$$

First method is explicit approach which can be written as:

$$\frac{\rho(\phi_P^n - \phi_P^0)V_P}{\Delta t} + \sum_f F \phi_f^0 = \sum_f \Gamma A \text{grad } \phi_f^0 + S_u V_P + S_P \phi_P^0 V_P \quad (2.30)$$

Another method is implicit approach that involves more complex calculations:

$$\frac{\rho(\phi_P^n - \phi_P^0)V_P}{\Delta t} + \sum_f F \phi_f^n = \sum_f \Gamma A \text{grad } \phi_f^n + S_u V_P + S_P \phi_P^n V_P \quad (2.31)$$

Here, A represents the area of the cell face and V the control volume. The surface integrals can be expressed as face summation where f stands for cell faces. For the surface integrals, the implicit approach involves neighbouring faces in the new time step. This makes calculations more complex. In the explicit form, there is only one unknown nodal point which is straightforward, while the implicit form has three unknown points.

Then, what is the reason for using the implicit method in spite of the complex algorithm? The implicit approach allows larger time step than the explicit method. More details will be discussed in Ch 2.2.4.

Figure 2-2 illustrates how grid points are deployed in two methods for time marching solution.

Figure 2-2: Explicit (upper) and implicit (upper) method [13]

Therefore, the momentum equations are discretised into linear algebraic equations that normally becomes,

$$a_P \phi_P = \sum a_{nb} \phi_{nb} + b \quad (2.32)$$

a is a function of velocity and $_{nb}$ represents neighbouring cells.

Then it can be arranged into the form of

$$[A] [\phi] = [B] \quad (2.33)$$

$[A]$ is a square matrix and $[B]$ is the source vector in column. Exact linear algebraic equations and forms in the matrix depend on the governing equation and the numerical scheme used for the discretisation. Applying appropriate matrix solving algorithm, the matrix equation can be solved to obtain the solution.

2.2.2 Pressure corrections

For incompressible flow, the continuity equation itself does not contribute to the pressure solution, as density is constant. Thus, the pressure gradient is unknown and has to be solved separately while solving the momentum equations.

Adding the pressure term in the momentum equation, the equation (2.32) can be written as,

$$a_p U_p = H(U) - \nabla p \quad (2.34)$$

where $H(U)$ term contains all the terms for the neighbours and all the source parts of transient terms. Writing this for the velocity U becomes:

$$U_p = \frac{H(U)}{a_p} - \frac{1}{a_p} \nabla p \quad (2.35)$$

The continuity equation is also discretised as,

$$\nabla \cdot U = \sum_f A \cdot U_f = 0 \quad (2.36)$$

where A denotes area of the cell face.

The velocity of the equation (2.35) can be expressed on the cell face:

$$U_f = \left(\frac{H(U)}{a_p} \right)_f - \left(\frac{1}{a_p} \right)_f (\nabla p)_f \quad (2.37)$$

Substitute the equation (2.37) into discretised continuity equation (2.36):

$$\nabla \cdot \left(\frac{1}{a_p} \nabla p \right) = \nabla \cdot \left(\frac{H(U)}{a_p} \right) = \sum_f A \cdot \left(\frac{H(U)}{a_p} \right)_f = 0 \quad (2.38)$$

The equation (2.38) has to be satisfied to guarantee the conservation of the face fluxes.

By expressing the gradient into the summation of mass flux at the cell face, the final form of Navier-Stokes equation becomes,

$$a_p U_p = H(U) - \sum_f A \cdot (p)_f \quad (2.39)$$

The face flux F is the velocity at the face (2.36) multiplied by the area of the cell face:

$$F = S \cdot U_f = S \cdot \left[\left(\frac{H(U)}{a_p} \right)_f - \left(\frac{1}{a_p} \right)_f (\nabla p)_f \right] \quad (2.40)$$

Normally pressure is solved by the pressure-velocity coupling algorithms such as SIMPLE (Patankar [29]) and PISO (Issa [12]).

They are iterative methods of guess and correction.

SIMPLE (Semi-Implicit Method for Pressure Linked Equations) process is in the following sequences:

- 1) Discretised momentum equations are solved with the initial guessed pressure which is denoted by p^* to find velocity components (u^* , v^* and w^*). This is a momentum predictor step.
- 2) Discretised equations including the $H(U)$ operator are used to solve pressure correction equation.

- 3) Then the pressure correction is defined from the difference between the guessed pressure and correct pressure field. The corrected pressure (p) is obtained by the guessed pressure (p^*) and pressure correction (p_{corr}).

$$p = p^* + p_{\text{corr}} \quad (2.41)$$

The same applies to the velocity corrections from the pressure correction:

$$u = u^* + u_{\text{corr}} \quad (2.42)$$

$$v = v^* + v_{\text{corr}}$$

$$w = w^* + w_{\text{corr}}$$

- 4) Solve other scalar equations (e.g. turbulence properties)
- 5) Repeat this process until the velocity field satisfies the continuity equation.

PISO (Pressure Implicit with Splitting of Operators) algorithm is also similar to SIMPLE but has one more corrector step. First, in PISO, the same steps with SIMPLE are performed and there is one more correction for pressure and velocities to be carried out. These steps are repeated until it reaches convergence. In spite of the additional step for pressure correction, PISO is known to be efficient and fast. However, this also depends on the case. It is not clear which method is the best for general flow case.

For this iterative method, under-relaxation factors help stability of the calculations. Especially for the pressure correction

methods, the equations are prone to diverge easily during the iterations. Generally, when a new value of variable (ϕ) is calculated, under-relaxation is applied as follows:

$$\phi_P^{new} = \phi_P^{old} + \gamma(\phi_P^{predicted} - \phi_P^{old}) \quad (2.43)$$

The under-relaxation γ is normally applied between 0 and 1 so that the iterative solution can be changed slowly and it increases stability of the calculation. Basically it does not affect the converged values but helps reduce the oscillation. However, this may slow down the convergence. Under-relaxation factors for all the variables are case dependent and there is no general recommendation.

If the factor equals to 1, it means no relaxation applied. Predicted value will be used without under-relaxation.

Sometimes the factor can be more than 1 to accelerate the convergence. In this case, the stability should be ensured during the calculations.

2.2.3 Convergence and residuals

The discretised governing equations are solved by iterative algorithm because the initial solution will not satisfy them. The left hand side and the right hand side of the matrix form of linear algebraic equations will not be equal with the initial values in

the first iteration that are assumed from the boundary conditions. The difference between the left and the right hand side is called momentum residual. During the iterations, the absolute values of residuals should be reduced so that the left and right hand sides of the equation can get closer. By this guessing and correction procedure, the variables get nearer to the actual solutions. Therefore, the residuals are important measures on how the solution satisfies the governing equations and can be used to estimate the error. In other words, the solution would not be correct and diverge if the absolute values of residuals are becomes bigger during the iterations.

There are many things that can affect the convergence of the solution. They are relaxation factors, quality of the mesh, numerical schemes, diagonal dominance of the matrix, etc. Wrong selection of mesh, numerical settings or boundary conditions would cause the solution to diverge. Nevertheless, this is always case dependent and there are no optimum settings. In some cases, convergence should be judged by the force values rather than the momentum residuals. The oscillation of force values will be stabilised or will not change any more when the solution converges. In this case, however, monitoring the residuals still can give a good indication of whether the solution diverges or not.

2.2.4 Courant number

The Courant number is a stability criterion of computation and defined as follows.

$$Co = \frac{\delta t |U|}{\delta x} \quad (2.44)$$

δt is the time step, U is the magnitude of velocity at the cell and δx is the size of the cell in the direction of the velocity.

In explicit approach, the Courant number must be less than one to ensure the numerical stability while running the computation. This is also called as CFL (Courant-Friedrichs-Lewy) condition. As can be seen in the formula, the time step of transient calculations has to be small enough if the velocity is high or the cell size in the direction of the velocity is very small in order to make sure CFL condition is met. CFL condition must be ensured in all the cells throughout the computation domain. Especially Courant number has to be far smaller at the interface region of multi-phase flow for the stability of the solution.

However, the Courant number can easily exceed one in the implicit approach. In some of the commercial codes, the Courant number is allowed even beyond hundreds. This means much larger time step can be used in the implicit approach. Although

implicit method allows larger time step, an appropriate Courant number depends on the type of the calculations and should be selected carefully for the accurate solution. Figure 2-3 is an example that shows the comparison between the implicit and explicit method. When the equal time step is used for the implicit and explicit algorithms, the solution of implicit approach gives more stable and accurate results.

Figure 2-3: Comparison of implicit and explicit solutions for same time step [7]

2.2.5 Boundary conditions

Boundary conditions are essential to find the unknowns in the governing equations. Wrong or poor boundary conditions will lead the computation to wrong solution or divergence. There are several types of boundary conditions and they are mainly divided into fixed value and gradient types. The fixed value type boundary conditions set the specific value of ϕ on the cell face, and this is called Dirichlet type. The fixed gradient boundary condition applies gradient of ϕ to the normal direction pointing outward on the boundary faces and is known as Von Neumann type.

For the general fluid flow simulation, boundary conditions are normally set for inlets, outlets, walls and/or symmetry planes.

Inlet: A fixed value is set for the velocity and zero gradient for the pressure at the inlet boundary.

Outlet: It is important for an outlet boundary to be set in such a way that it satisfies the overall mass conservation. Therefore, the outlet boundary condition is set in conjunction with the inlet boundary condition and velocity boundary condition.

For the mass conservation, the total mass flux which are going out (M_{out}) and coming in (M_{in}) must be the equal. In order to achieve this, the flux ratio (M_{in}/M_{out}) is multiplied to outlet velocity to ensure the continuity and subsequently multiplied to the East neighboring velocities (Versteeg and Malalasekera, 2007). Therefore, normally zero gradient is set for the velocity and fixed value for the pressure at the outlet boundary.

It is also important to locate the outlet boundary far enough from the region where the flow is expected to be developed so that the flow gradients become zero in the direction normal to the outlet boundary.

Wall: The wall boundary conditions are used on the solid objects and on the walls. On the wall boundaries, no-slip condition has to be met so that the velocity on the wall becomes the equal to that of the wall. For instance, velocity should be zero on the wall of the static object. When the object is moving, many CFD codes provide moving wall velocity boundary condition to be able to apply no-slip wall. On the wall boundaries, the mesh should be defined carefully to resolve the viscous flow characteristics in the boundary layer. More details for this will be discussed in the next sections.

Slip wall: When this boundary condition is set, the flow property normal to the wall becomes zero and tangential components zero gradient. This is normally applied to the wall where flow is just passing along.

Symmetry plane: The boundary conditions for symmetry planes give no flow gradient or flux across this boundary. This can be applied to save computational cost where the geometry and the flow field are exactly symmetric across the boundary.

2.2.6 Boundary layer treatment and y plus

y plus (y^+) is a non-dimensionalised distance by the friction velocity and kinematic viscosity and represents the distance from the solid wall to the first grid point from the wall.

$$y^+ = \frac{\Delta y_p}{\nu} \sqrt{\frac{\tau_w}{\rho}} = \frac{\Delta y_p \cdot u_\tau}{\nu} \quad (2.45)$$

, where Δy_p is distance from the wall to the first grid point, τ_w is wall shear stress, u_τ is friction velocity, ρ is the density of fluid and ν is the velocity component perpendicular to the wall.

It is known that the flow is laminar near the wall and is transitional from laminar to turbulence at about $y^+ = 11.225$. In

order to calculate viscous flow properties in the boundary layer correctly, mesh needs to be carefully designed to locate y^+ in the proper range.

On the surface of a wall, no slip condition is applicable which means the fluid on the wall has the same velocity with the wall itself. For instance, if a wall is stationary, the fluid on the wall is also stationary. At the very next to the wall, there is a very thin layer where viscous effect dominates and the flow is laminar. It is named viscous sub-layer or laminar sub-layer. It is also called as linear sub-layer because of the linear relation between y^+ and u^+ . u^+ is non-dimensional velocity with friction velocity, u_τ .

$$u^+ = \frac{U}{u_\tau} \quad (2.46)$$

$$u_\tau = \sqrt{\tau_w/\rho} \quad (2.47)$$

, where τ_w is wall shear stress

In this region, the shear stress is approximately equal to the wall shear stress and $y^+ = u^+$ is satisfied. When wall bounded effects are highly significant in a flow, viscous sub-layer has to be resolved, hence, y^+ must be in order of 1 to have at least one point in the viscous sub-layer. For a good result, y^+ should be less than 1 in all the area. In other words, mesh near the wall

has to be well refined to resolve the whole boundary layer, and turbulence model also needs to be of low Reynolds number model such as SST (Shear Stress Transport) type that can resolve viscous sub-layer properly. Epsilon based turbulence models are not designed for the viscous sub-layer. Generally, cases with low Reynolds number should have mesh with the first grid point from the wall within this sub-layer.

Outside this sub-layer, there is a turbulence region where u^+ and y^+ vary in a logarithmic relation as such,

$$u^+ = \frac{1}{\kappa} \ln(Ey^+) \quad (2.48)$$

where κ is von Karman's constant and E is an integration constant from the roughness of the wall.

This is the region where $30 < y^+ < 500$ and called log-law layer. In this region, the shear stress is no more constant and varies with the distance from the wall. When wall bounded effects are not priority and Reynolds number is high, the first grid point can be put in this log-law layer region to apply wall functions. Wall function is a semi-empirical method that can predict the velocity profile near-wall region. In other words, y^+ has to be in the range of 30 to 500 in order to use wall function. However, it is

important to make sure that the first point is not placed outside of the boundary layer when using wall function. By using wall functions, mesh resolution can be reduced, hence, saving computational costs and efforts.

If y^+ is between 5 and 30, it is a region where laminar flow transits to turbulent flow and calculated shear stress is neither constant nor logarithmic. This region is called buffer layer and has to be avoided when generating mesh with certain y^+ value. Therefore, the user should determine whether viscous sub-layer would be resolved or a wall function would be used to calculate near-wall region for the grid design near the wall. Figure 2-4 shows velocity distribution near a solid wall with respect to $\log y^+$.

Figure 2-4: Velocity distribution near a solid wall [34]

2.2.7 Good practice of boundary layer treatment

As discussed, for the grid design, special care should be taken at the boundary layer region in which flow properties vary rapidly. Wrong design of boundary mesh layer will give wrong calculation results. When one designs the grid to target certain range of y^+ value, valid y^+ value can be known only after calculation is carried out with enough iterations. It is because y^+ value is different between the initial field and after flow is developed sufficiently. Therefore, it is recommended to monitor y^+ values during the computations.

Normally marine CFD applications are high Reynolds number flow and wall functions can be used to calculate the near wall region. In this case, y^+ value is recommended to be below 300 and above 30 when flow is developed.

If the user wants to resolve whole boundary layer, enough number of mesh layers (i.e. at least 20 layers) needs to be designed in this laminar region without sudden increase in the size of mesh between the layers. Several marine CFD codes do not use wall function but resolve whole boundary layer from the wall where the velocity is set to zero although the marine applications has high Reynolds number. In order to achieve this,

it is recommended that the mesh near the wall needs to be very fine with $y^+ \leq 1$ and turbulence model must be valid in all regions of the boundary layer. This is known to be very accurate in calculating ship resistance (Larsson et al. [18])

This practice is also important for full scale analysis. When it comes to the experiment, it is well known that there is a big discrepancy of Reynolds number between model scale and full scale when scaling down with geometrical similarity. This causes relatively thicker boundary layer in the model scale compared to full scale due to the different Reynolds number and brings forth different results. For this reason, there were many efforts to carry out full scale CFD analysis. Traditionally CFD technology was limited to the model scale not only because of limitation of computational resources but also because the first cell from the hull surface gets to have very large aspect ratio in full scale. In order to have y^+ within 1, too large aspect ratio of grid causes solvers to be broken down easily. Nowadays, many commercial solvers are able to handle high aspect ratio. Another problem comes from full scale analysis is when turbulence model is used with wall function. General wall functions are mainly developed and validated with the relatively small scale but not robust enough for full scale ships with slender hullform where Reynolds number easily goes above 10^9 . However, improvement of CFD

techniques as well as increasing hardware capability enables full scale analysis nowadays (Larsson et al. [18]).

2.3 Mathematical model of ship-handling simulator

This chapter will cover mathematical model of ship-handling simulator and discuss how to make use of CFD results into ship-handling simulator.

According to the Euler's equations of rigid body dynamics, the motion of a rigid body can be written as follows,

$$\vec{F} = m\dot{U} + m\Omega \times U \quad (2.49)$$

$$\vec{M} = I\dot{\Omega} + \Omega \times I\Omega \quad (2.50)$$

Here, ship's centre of gravity is assumed to be at the midship.

Figure 2-5 shows coordinate system of mathematical model. Considering three dimensions, force vector and moment vector are,

$$\vec{F} = (X, Y, Z), \quad \vec{M} = (K, M, N)$$

Here, m is mass of the ship, I is moment of inertia of the body, linear speed vector is $U(u, v, w)$ and angular velocity vector is $\Omega(p, q, r)$.

Figure 2-5: Coordinate system of manoeuvring mathematical model

Substituting all these vectors into the equation (2.50),

$$I_{xx}\dot{p} - (I_{zz} - I_{yy})qr + I_{xz}(\dot{r} - qp) + I_{yz}(r^2 - q^2) + I_{xy}(pr + \dot{q}) = K \quad (2.51)$$

$$I_{yy}\dot{q} - (I_{xx} - I_{zz})rp + I_{xy}(\dot{p} - rq) - I_{xz}(p^2 - r^2) + I_{yz}(qp + \dot{r}) = M \quad (2.52)$$

$$I_{zz}\dot{r} - (I_{yy} - I_{xx})pq - I_{yz}(\dot{q} - pr) + I_{xy}(q^2 - p^2) - I_{xz}(rq + \dot{p}) = N \quad (2.53)$$

Assuming to be $(I_{xy} = I_{yz} = I_{xz} = 0)$ from the tensor matrix,

$$I = \begin{pmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{pmatrix}$$

Hence, the equations (2.49) and (2.50) can be expanded as follows:

$$m(\dot{u} - vr + wq) = X \quad (2.54)$$

$$m(\dot{v} - wp + ur) = Y \quad (2.55)$$

$$m(\dot{w} - uq - vp) = Z \quad (2.56)$$

$$I_{xx}\dot{p} + (I_{zz} - I_{yy})qr = K \quad (2.57)$$

$$I_{yy}\dot{q} + (I_{xx} - I_{zz})rp = M \quad (2.58)$$

$$I_{zz}\dot{r} + (I_{yy} - I_{xx})pq = N \quad (2.59)$$

These are the motion equations for 6 degrees of freedom (DOF) of motion in the general ship-handling simulator when the centre of mass is at the midship. Figure 2-6 illustrates 6 DOF of motions.

Figure 2-6: Six degrees of freedom of motion

The right hand side represents forces (X, Y, Z) and moments (K, M, N) in 6 DOF of motions which are surge, sway, heave, roll, pitch and yaw, respectively. Each of them represents the sum of

all the external forces and moments from hull, propeller, rudder, aerodynamic, wave, mooring lines, etc.

Likewise, the forces and moments from ship-to-bank interaction can be added into the additional terms in the right hand side, too. For this reason, sway force and yaw moment from the bank effect can be non-dimensionalised and modelled for different water depth and separation from the bank wall.

$$C_{BY} = \frac{F_Y}{mgFn^2} \quad : \quad \begin{array}{l} \text{non-dimensionalised sway force from the} \\ \text{bank effect} \end{array} \quad (2.60)$$

$$C_{BN} = \frac{M_Z}{mgLFn^2} \quad : \quad \begin{array}{l} \text{non-dimensionalised yaw moment from the} \\ \text{bank effect} \end{array} \quad (2.61)$$

Even if CFD calculations are carried out for steady state without mesh motion, the results can be easily implemented in the ship-handling simulator for time variant simulations. When the ship moves near the bank during the real time simulation, modelled sway force and yaw moment of the ship at the specific separation from the bank and water depth can be added in the right hand side of motion equations of simulator (Equation 2.55 and 2.59) as an additional force and a moment components. As the ship is repelled from or attracted to the bank due to the added force and moment, force and moment can be updated by interpolating at a new distance that varies at each time step. By this, simulations

for the irregular bank and the bathymetry will also be possible, hence, training of crew in confined water can be more realistic.

Figure 2-7 shows a simulation result of a vertical bank case in ship-handling simulator after modelling the forces and moments at various distances but at a fixed water depth. Kongsberg system [15] has been used for the testing of ship-handling simulator.

**Figure 2-7: The trajectory in ship-handling simulator with
ship-to-bank interaction
x and y axis: distance(m) from the initial point**

Chapter 3

Open source CFD Code

3.1 Open Source CFD Codes and OpenFOAM

CFD technology has become essential in many fields including marine industry. There are popular commercial CFD codes with huge user groups due to the convenient and strong functions. Meanwhile, open source CFD codes are also improving in functions and it is also becoming popular in many applications these days. They are free to use and even free for modification under the GNU General Public License. The users need not worry about the license and can run it anywhere. They also have good pre- and post- processors that are also free to use, and parallel computing interface is useful to speed up the computations. Although they are still not as convenient as commercial codes and it takes some time to leverage, it is decided to use an open source CFD code to build its capability in this study.

Among many open source CFD codes, OpenFOAM has been chosen for this study due to the massive user group and numerous tutorials.

OpenFOAM (Open Field Operation And Manipulation) is a C++ library toolbox based on finite volume method. It is first released by Prof. Hrvoje Jasak and Mr. Henry G. Weller in 2004 after several studies and projects. Various solvers with different governing equations and numerical schemes can be used to solve different problems. The user can even create a new solver because the libraries in OpenFOAM toolbox are designed to solve ODE (ordinary differential equations) or PDE (partial differential equations). The latest version 2.3.0 was released in Feb 2014. There is another group called OpenFOAM extend-project who develops modified OpenFOAM solvers by themselves and discusses related issues. There are many seminars and conferences in many countries to discuss about the applications. The capability and applications are known to be comparable to commercial codes. Initially OpenFOAM 2.1.0 was used in this study, and then changed to the version 2.3.0 in the middle of the study when it was released.

3.2 OpenFOAM software structures

3.2.1 Architecture

Basically, an OpenFOAM case is composed of various files in several folders. Figure 3-1 shows the structure of a general fluid flow case of OpenFOAM by the folders and the major input files.

Figure 3-1: OpenFOAM case structure

The 'system' folder has numerical schemes and their options. Users can set discretisation schemes, pressure solvers, matrix

solver, time step, the numerical settings, etc. to be used for governing equations.

The 'constant' folder has mesh and physical properties such as density, kinematic viscosity, acceleration of gravity, turbulence model information. The mesh information are saved individually as points, faces, cell, boundary, etc. in the sub-folder, 'PolyMesh'.

The time directories are the folders where all the variables are stored. Initial time folder, normally '0', has initial and boundary conditions in it. As computation is run, folders for other time steps with output files are saved by the solver according to the time step set by the user.

3.2.2 Solvers

There are several solver groups in the package such as incompressible flow, compressible flow, multiphase flow, combustion, heat transfer and buoyancy-driven flow, electromagnetics, stress analysis of solids, etc. Under each group, there are several solvers too. This study will focus on the multiphase solvers to include free surface effect. Single-phase solver has also been used to compare with multi-phase solutions.

3.2.3 Numerical algorithm

OpenFOAM is a toolbox that can solve partial differential equations. For example, the follow momentum equation,

$$\frac{\partial \rho U}{\partial t} + \nabla \cdot \phi U - \nabla \cdot \mu \nabla U = -\nabla p$$

is expressed as:

```
solve
(
    fvm::ddt(rho, U)
    + fvm::div(phi, U)
    - fvm::laplacian(mu, U)
    ==
    - fvc::grad(p)
);
```

Here ‘fvm’ and ‘fvc’ are the classes to discretise differential operators such as ddt ($\partial / \partial t$), div ($\nabla \cdot$) and laplacian (∇^2). The class ‘fvm’ involves implicit derivatives to generate matrix. ‘fvc’ stands for Finite Volume Calculus and calculates explicit operations to return geometric field data.

OpenFOAM gives options for each numerical scheme. For the gradient scheme, 2nd order schemes (Gauss linear and Least square) are used in this study to maintain the accuracy of the solution. Gradient scheme is the numerical scheme that interpolates the value to the cell faces since the variables are basically stored at the cell centre. Gauss linear gradient scheme means Gaussian integration to interpolate the value from the cell centre to the face centre and the interpolation is done linearly.

A limiter can be used for gradient calculations especially when high order scheme is applied. There are cell type and face type limiters. Basically cell type limiter is applied to the gradient between the cell centres by clipping the gradient equally, whereas face limiter applies limiting gradient on the face of the cells. MD stands for multi-dimensional which clips the flux in the direction normal to the cell faces. In the simulation of this study, *cellMDLimited* was applied to maintain certain accuracy of numerical scheme. Available limiters are followings.

- *cellMDLimited* : Least diffusive and more accuracy
- *cellLimited* : Less dissipative
- *faceMDLimited* : More dissipative
- *faceLimited* : Most diffusive but robust scheme

Another pressure-velocity solving algorithm, PIMPLE is provided by OpenFOAM. It is a hybrid method of PISO (Pressure Implicit with Splitting of Operators) and SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm. It is, in fact, not different from the PISO procedure except for the outer correctors. If the outer corrector parameter is set to be 1 or not set, it is simply same as the PISO algorithm. Outer iteration option is known to give more stable and accurate calculations.

3.3 Multi-phase solver interFOAM and LTSInterFOAM

In many marine applications of CFD, free surface cannot be neglected, hence, multi-phase solvers are very useful for interface capturing. For this reason, interFoam and LTSInterFoam were chosen for this study. interFoam is a solver for transient solutions of two incompressible, isothermal and immiscible fluids using a VoF (volume of fluid) phase-fraction. LTSInterFoam adopts local time stepping (LTS) that uses different time steps for each cell depending on the different velocity and cell size, and enables to reach steady state quickly. Since this study will focus on capturing forces and moments at steady state, LTSInterFoam is useful to obtain them in relatively shorter time. The governing equations used in the solver are the continuity equation (3.1) and momentum equation (3.2) for incompressible flow:

$$\nabla \cdot \mathbf{U} = 0 \quad (3.1)$$

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = -\nabla p^* - \mathbf{g} \cdot h \nabla \rho + \mathbf{F}_S + \nabla \cdot (\mu \nabla \mathbf{U}) + (\nabla \mathbf{U}) \cdot \nabla \mu \quad (3.2)$$

where μ is dynamic viscosity, p^* is the dynamic pressure, $\mathbf{g}$ is acceleration of gravity, ρ is density, $\mathbf{U}$ is velocity vector, h is hydrostatic height of the fluid, hence, first term in the right hand

side represents dynamic pressure and second term hydrostatic pressure. F_S is the source momentum of surface tension, which is expressed as;

$$F_S = \sigma \kappa \nabla \alpha \quad (3.3)$$

, where σ represents the surface tension coefficient, κ is the surface curvature and α is volume fraction which is bounded in $0 \leq \alpha \leq 1$ (0 for air and 1 for water). Surface curvature can be modelled as the boundary condition of wall surfaces setting contact angle. Surface tension can be neglected in the simulation in which huge mass of fluid develops the flow fast in large domain and the effect from surface tension is not important. Therefore, in this simulation, surface tension between the interface and walls are neglected.

To make sure scalar fields are bounded in multi-phase flow, OpenFOAM uses MULES (Multi-dimensional Universal Limiter for Explicit Solution) algorithm. This handles free surface developments by limiting the face fluxes multi-directionally for the boundedness of hyperbolic equations. Although Euler implicit unsteady method is used in OpenFOAM, this explicit limiter requires Courant number below 1. In OpenFOAM 2.3.0, semi-implicit version of MULES has been introduced. This

executes an implicit predictor first and followed by explicit correction step. This allows quite bigger time step with larger Courant number and reduces computational time from older versions of OpenFOAM.

The main source codes of interFoam are included in the Appendix. They include most of the CFD algorithm and techniques discussed in this study.

3.4 OpenFOAM evaluations in open water

Starting from the tutorial on open water, the mesh, numerical settings, boundary conditions, etc. are tested to determine the most suitable settings.

Prior to studying the bank effect using OpenFOAM, it would be beneficial to evaluate the solver with open water analysis to know its accuracy and find the robust computational settings. Compared to flow in confined water, open water analysis is relatively simpler, therefore, it is a good start before moving on to the confined water analysis that involves with shallow water effect as well as interaction with banks. KVLCC2 (KRISO Very Large Crude-oil Carrier 2) and KCS (KRISO Container Ship) that are popular for the benchmarking are chosen to compare with CFD analysis. The test results of these two ships were released by MOERI (Maritime & Ocean Engineering Research Institute), formerly known as KRISO (Korea Research Institute Ship and Ocean engineering) for validation purpose.

Considering the numerical cost and the accuracy, the minimum size of the mesh near the hull was determined. The mesh was created by *snappyHexMesh* combining several blocks to make graded mesh efficiently. As the ship and the domain are

symmetric, half hull was modelled and calculated. To save the computational cost, the mesh is created to be finer near the hull and getting coarse towards the outside of the domain. Turbulence model used for simulation was $k-\omega$ SST with wall function and the average y^+ was kept below 200 on the most of the surface area.

The boundary conditions are set as fixed value at the velocity inlet and pressure outlet, and zero gradient at the velocity outlet and pressure inlet. No slip condition was applied on the ship hull and symmetry plane on the side, bottom and middle symmetry plane.

In OpenFOAM 2.1.0, it turns out that the free surface level tends to be changed as the computation develops. This means water fraction is not well conserved. The problem started to occur after some iterations when the solution is getting stabilised. This probably happens because different inlet and outlet velocity could cause continuity correction as discussed in the chapter 2.2.5. This could affect total water fraction and make the water level changes. This is nearly certain because the problem does not appear if the outlet velocity is set to be equal to the inlet velocity. Therefore, when the inlet velocity is high, the phenomenon is more obvious. This problem has been reported

many times in the OpenFOAM forum (www.cfd-online.com), however, the problem disappeared in the newer version of OpenFOAM. In version 2.3.0, there was an improvement in the outlet boundary condition. Improving mesh quality, that is, reducing aspect ratio of the mesh at the inlet and outlet was also able to mitigate this problem. The problem was not significant in KVLCC2 case presumably due to the relatively slow speed, whereas it was obvious in KCS.

Forces and moments on the hull surface can be calculated by following formulas:

$$F_{pressure} = \int p \cdot \vec{n} dS \quad (3.4)$$

$$F_{viscous} = \int \tau \cdot \vec{n} dS \quad (3.5)$$

$$M_{pressure} = \int (x_f - x_o) \times p \cdot \vec{n} dS \quad (3.6)$$

$$M_{viscous} = \int (x_f - x_o) \times \tau \cdot \vec{n} dS \quad (3.7)$$

, where p is pressure, τ is wall shear stress, x_o is origin of measurement and x_f is the distance from origin to the face centre of the force component.

3.4.1 KVLCC2

Table 3-1 shows the principal dimension of KVLCC2 model ship that is geometrically scaled down by ratio of 58.0.

Table 3-1: Principal dimensions of KVLCC2 model

Length between perpendiculars (m)	5.5172
Breadth (m)	1.0000
Draught (m)	0.3586
Block coefficient	0.8098
Speed (m/s)	1.047
Froude number	0.142

Figure 3-2: Profile and front view of KVLCC2

Figure 3-2 is the profile and the front view of KVLCC2 hullform. OpenFOAM 2.1.0 and LTSInterFoam solver was used for this computation. Since the resistance at the steady velocity where flow is fully developed is important, transient solver has no advantage to predict the resistance. Flow analysis around KVLCC2 was carried out with grid convergence study (Table 3-2).

Table 3-2: Grid convergence study
(The experiment result: Kim et al. [38])

No. of mesh (K)	Pressure Force (N)	Viscous Force (N)	Total Force (N)	Limiter
690	11.16	10.45	21.61	Not used
1100	9.3	13.39	22.69	Not used
1470	7.17	12.08	19.25	100% -> 50%
1742	6.87	11.94	18.82	100% -> 50%
2434	6.78	11.74	18.52	100% -> 50%
3556	6.95	11.50	18.45	100% -> 50%
EXP	-	-	18.12	-

The number of mesh was tested up to 3.5 million and longitudinal force that is resistance was monitored. The force value did not change much after the mesh is getting finer to some extent, i.e. 1.7 million mesh. The mesh with 3.5 million cells showed 1.8% deviation from the experiment result. When the mesh was very fine, a limiter on the gradient scheme was necessary for the convergence. Otherwise the computation diverged before the iteration number 2000. In order to obtain better accuracy, 100% limiter was used first, and reduced to 50% when the force value is stabilised. Figure 3-3 shows the convergence history of resistance when the number of mesh was

3.5 million. With the iteration of 15000 (750 in the horizontal axis of Figure 3-3), the limiter has been reduced to 50%.

Figure 3-3: Convergence history of the resistance of KVLCC2

Wave elevation along the hull is compared with experiment result (Kim et al. [37]) in Figure 3-4 and it shows very good agreement. The horizontal axis stands for the distance from the forward perpendicular to aft direction, which means zero is at the forward perpendicular of the ship. Due to the relatively slow speed of tanker ship, the amount of changing water level was not significant.

Figure 3-4: Comparison of wave elevation along the hull of KVLCC2 (Experiment result: Kim et al. [38])

Wave pattern was compared to the experiment measurement in Figure 3-5. When limiter was used for very fine mesh, it resulted in oscillating behaviour in free surface pattern. For this reason, contour line of wave elevation became messy and it was hard to compare with experiment result, thus, the contour line was not presented in this study. This oscillation can be observed near the stern area of wave pattern in Figure 3-5. However, predicted wave pattern is in good tendency compared to the experiment. More comparisons were made in KCS container ship with the newer version of OpenFOAM.

Figure 3-5: Comparison of wave pattern of KVLCC2 of CFD (upper) and the experiment (Kim et al. [38]) (bottom)

3.4.2 KCS

Table 3-3 is the principal dimension of KCS model which is scaled down by the ratio of 31.599 from the full scale.

Table 3-3: Principal dimensions of KCS model

Length between perpendiculars (m)	7.2786
Breadth (m)	1.0190
Draught (m)	0.3418
Block coefficient	0.651
Speed (m/s)	2.196
Froude number	0.260

Figure 3-6 is the profile and the front view of KCS hull.

Figure 3-6: Profile and front view of KCS

The number of mesh used for the simulation was about 1.7 million. Figure 3-7 shows generated mesh around and on KCS hull. Trimmed mesh was generated for the mesh near the ship to be fine and for the outer region to be coarse. The grid convergence study was not carried out again for KCS as it has been done for KVLCC2, but reasonable number of mesh size was used based on the results from KVLCC2. One problem from the very fine mesh is y^+ value. When the mesh at the hull surface is too small, it was difficult to keep y^+ value all over the surface above 30. This would cause problem with wall function as discussed in Chapter 2.2.6 and the solution would not be correct.

Figure 3-7: Mesh of KCS - (a) whole domain, (b) forward and (c) stern area

Figure 3-8 shows y^+ distributions on the hull surface when the flow is fully developed. Due to the bulbous bow area where y^+ appeared low, it was hard to use finer mesh than that was used at the hull surface. Additional mesh refinement was applied near the stern boss (Mesh refinement in (c) of Figure 3-7) where y^+ appeared higher than other under water area. By this, it was possible to model the mesh with reasonable y^+ value. Moreover, finer mesh was also used around the free surface region for better free surface capturing because relatively high velocity would create more wave elevation.

Figure 3-8: y^+ distributions on the hull

For KCS analysis in open water, the problem of rising free surface level was quite significant due to the relatively higher velocity. After testing different settings, it was decided to use OpenFOAM version 2.3.0 that was released during that time. In the newer version of OpenFOAM, the problem did not appear anymore. In this version, there was significant improvement of

the outlet boundary condition to maintain the conservation of volume fraction. Another problem was over-prediction of force in LTSInterFoam solver although free surface elevation and the pattern were well predicted with this solver. It is presumed that the different time step throughout the domain leads to wrong solution. Several different grid types were tested but the problem was not solved. When the transient solver, interFoam was used, the obtained force value showed about 3.5% deviation from the experiment results. For the comparison purpose, the result of single-phase steady-state solver, simpleFoam has been added. From the result of the steady-state solver, it is assumed that free surface affects about 7.5% of the resistance in this KCS case. The comparison between the solvers and the experiment is presented in Table 3-4. The total resistance has been non-dimensionalised as below:

$$C_T = \frac{R_T}{\frac{1}{2}\rho S U^2}$$

R_T : Total resistance (N)

ρ : density of Water (kg/m³)

S : Wetted surface area (m²)

U : Velocity (m/s)

Table 3-4: The comparison of KCS resistance by different solvers (Experiment results: Kim et al. [38])

Solver	Resistance coefficient	Deviation	Gradient scheme
simpleFoam	3.415e-3	-3.99%	gauss linear
interFoam	3.681e-3	3.49%	cellLimited gauss linear 1.0
LTSinterFoam	5.108e-3	43.60%	gauss linear
EXP	3.557e-3	-	-

Although high maximum Courant number (Max. 25 and 10 for the interface) used for the simulation due to the semi-implicit algorithm adopted in OpenFOAM 2.3.0, the convergence of the force value was quite stable. Figure 3-9 is the convergence history of the resistance value of interFoam solver for the different gradient schemes.

A limiter was needed for the stability of the calculation. Because of the relatively big oscillation of Gauss linear gradient scheme, the limiter was removed after the computation became stable at 50 second. This suppressed the oscillation of the gradient scheme and also improved the accuracy. Least squares method showed less oscillating behaviour but slightly less accuracy of the resistance value. Gauss linear scheme resulted in 31.492 N for

the viscous drag force while Least squares scheme 32.157 N. Gauss Linear showed slightly better result by 2% when the limiter was removed. In Figure 3-9, it is found that the pressure force gets lower after 50 seconds when the limiter is removed and this reduces over-prediction. The solution using Least squares method without the limiter was not converged. Hereafter, the result with Gauss linear scheme is used for other validations in this chapter.

In Figure 3-10, wave elevation of OpenFOAM calculation was compared with experiment (Kim et al. [38]). As for the horizontal axis that shows x/L_{pp} , 0 stands for forward perpendicular, 0.5 for midship and 1 for aft perpendicular. Except for the crest at the bow wave, it shows very good agreement with the experiment.

In Figure 3-11, wave pattern also shows good agreement with the experiment result (Kim et al. [38]). 0 stands for the forward perpendicular and 1 for the aft perpendicular. Some strange shapes at the side of the divergent wave behind the stern are appeared due to the mesh refinement.

Figure 3-9: Convergence time history of the resistance (N) of KCS by (a) Gauss linear and (b) Least square gradient scheme

Figure 3-10: Comparison of wave elevation along the KCS hull (Experiment results: Kim et al. [38])

Figure 3-11: Comparison of wave pattern of KCS of OpenFOAM (upper) and the experiment (Kim et al., [38]) (bottom)

Figure 3-12: Wave pattern at the forward (a) and the stern (b)

In Figure 3-12, wave pattern at the forward and the stern are presented. It illustrates the maximum (red) and the minimum (blue) of the wave elevations. Kelvin wave which consists of diverging wave and transverse wave is well described in the wave pattern.

Stern flow, which is wake distribution, becomes important information for propeller design. Due to the aft hull shape, wake velocities entering into the propeller plane become inhomogeneous on the propeller plane and they cause pressure fluctuation on the propeller blades while they are rotating in high RPM. This fluctuating pressure causes cavitation on the propeller

blades and on the rudder, and leads to many problems. Cavitation also affects the propeller efficiency, vibration and erosion on the propeller blades and rudder. Therefore, it is important to design the aft hull in such a way that the wake velocity becomes evenly distributed at the same blade radius to minimise pressure fluctuation especially for the container ship which has bigger propeller loading with powerful main engine.

Wake velocities that are calculated by OpenFOAM are visualised at the couple of sections in Figure 3-13. In the experiment, a dummy hub is attached on the stern boss while it is absent in the CFD calculation. Therefore, difference is unavoidable near the stern boss on the propeller plane. Overall tendencies are well predicted. One reason for the difference is the mesh fineness. Behind the ship stern, the flow separation, drag and turbulence make the flow very complex, hence, very fine mesh is needed to predict the flow accurately. In author's opinion and from the experience with commercial marine CFD codes SHIPFLOW and WAVIS, these are reasonable results due to the complexity of flow.

Figure 3-13: Wake distribution (axial velocity [m/s]) of KCS at (a) the propeller plane and (b) at the station no. 1 (left : experiment (Kim et al. [38]), right : OpenFOAM result)

3.4.3 Summary of the evaluations

OpenFOAM was able to solve multi-phase problem in open water and calculate resistance force value with reasonable size of mesh. The converged force value showed good accuracy with the wall function. For the gradient scheme, Gauss linear scheme without a limiter gave slightly better accuracy than Least squares

method, however, the latter had advantage of less oscillation in the converged force value. The free surface pattern and the elevation were well captured too. The wake distribution also showed reasonable accuracy in spite of medium fineness of mesh. Several problems in the OpenFOAM 2.1.0 such as conservation of volume fraction, oscillations on the free surface level, etc. were all fixed in the new version.

The local time stepping solver computed a wrong force value for KCS whereas the problem did not occur with KVLCC. This will be further tested in the next chapter on confined water analyses.

Chapter 4

Analysis in Confined Water

4.1 Computational set-up and mesh

Numerical setup for confined water analyses was established according to the experimental setup done by FHR (Lataire et al. [5]). The test cases with several bank arrangements and different under keel clearance were chosen for numerical analysis. As found in the experimental research, a 8000TEU container ship in model scale was used for the simulations with a scale ratio of 80. Table 4-1 is the principle dimension of the model ship.

Table 4-1: Principal dimension of model ship

Length overall (m)	4.332
Breadth (m)	0.530
Draught (m)	0.149 ~ 0.180
Block Coefficient	0.65 ~ 0.66 (depending on the draught)

Figure 4-1 shows the coordinate system of the cases.

Figure 4-1: The coordinate system

Figure 4-2 shows the bank arrangements of the test case.

Figure 4-2: Schematic view of the bank arrangement (looking from the aft)

As can be seen in Figure 4-2, three types of the bank were used. The arrangement (a) has a straight vertical bank wall. The bank type (b) has an inclined bank wall and type (c) has a bank wall that is inclined and submerged. The coordinate system used for

the simulation is defined as body fixed and right handed system and that is different from the experiment. The origin is located at the amidships, centreline and keel for x, y and z, respectively. Longitudinal axis (x) is towards the bow, transverse axis (y) towards port side and vertical axis (z) upwards. Therefore, positive sway force directs to the outside the bank and yaw moment is positive when the bow is away and the stern towards the bank. The negative heading angle of the case A in Table 4-2 means that bow of the ship is toward the bank and the stern is away from the bank.

Four cases were chosen among the experiment results for CFD simulations. Table 4-2 includes the details of the case arrangements. Under keel clearance is 10% or 100% of the draught and various lateral position of the ship from the bank are included in the test cases. The test cases without propeller thrust have been chosen firstly to see the bank effect merely from asymmetric flow field. If propeller is rotating, it may affect the force and the moment values. In these test cases, propeller was basically prevented from the rotations, therefore, measured RPM is -1 and propeller thrusts are all slight negative because the propeller worked like a 'parachute'. The case A that has an heading angle toward the inflow and the measured negative propeller torque was relatively big.

Table 4-2: The test cases without propeller thrust

Test title		A	C	F	H
Under keel clearance	[%]	100	10	100	10
Draught	[m]	0.180	0.180	0.180	0.149
Water depth	[m]	0.360	0.198	0.360	0.163
Forward component of speed vector	[m/s]	0.6842	0.4578	0.8012	0.4578
Transverse component of speed vector	[m/s]	-0.0599	0.0	0.0	0.0
Lateral separation	[m]	0.265	0.265	1.435	1.965
Heading angle	[°]	-5.0	0.0	0.0	0.0
Propeller rate	[rpm]	-1	-1	-1	-1
Propeller thrust	[N]	-1.10	-0.20	-0.93	-0.37
Propeller torque	[Nmm]	-27.09	-3.05	0.84	7.28
Type of bank	-	(b)	(c)	(b)	(a)
Bank inclination		8:1	5:1	3:1	1

The case A has slight heading angle with small separation from the bank wall. The case C is the most severe case where both under keel clearance and lateral separation from the bank wall are very small. The minimum distance from the hull to the sloped bank wall is very small. The case F does not have extreme clearance from the bank wall or the bank bottom, however, the velocity of the ship is relatively high. The case H has very small under keel clearance but moderate distance to the bank wall.

Figure 4-3 is the graphical view of the test cases as modelled in OpenFOAM. It shows the bank shape, the location of the ship and water depth.

Figure 4-3: Graphical view of the case C and the case H

There is a need to find out how much propeller contributes to the bank effect. Although propeller rotation is not modelled in this study, another set of simulations has been done for the test cases with propeller thrust but without modelling the propeller. The purpose of this set of simulations is to compare the CFD results without propellers to the experiment results that have propeller thrust. This will indicate the significance of propeller contribution to the bank effect. They will show the difference as

much as propeller rotation contributes to the bank. Table 4-3 is the detailed experiment conditions for this set of test cases. The only difference in the CFD simulations is that the propeller is not modelled.

Table 4-3: The test cases with propeller thrust

Test title		D	I	J
Under keel clearance	[%]	100	35	100
Draught	[m]	0.149	0.149	0.149
Water depth	[m]	0.297	0.200	0.297
Forward component of speed vector	[m/s]	0.6868	0.5723	0.8012
Transverse component of speed vector	[m/s]	0.0	0.0	0.0
Lateral separation	[m]	1.340	0.565	0.530
Heading angle	[°]	0.0	0.0	0.0
Propeller rate	[rpm]	540	359	539
Propeller thrust	[N]	3.30	1.46	3.16
Propeller torque	[Nmm]	53.78	20.65	62.35
Type of bank	-	(b)	(c)	(c)
Bank inclination		8:1	8:1	5:1

For the confined water analysis, the mesh size and settings were determined from the open water analysis results. Efforts were made to model the mesh finer or prism boundary layers properly because a certain number of mesh layers would be important for

the narrow gap between the bank and the ship as well as the under keel clearance. At least, two boundary layers have been modelled. The OpenFOAM mesher was prone to fail to generate more than three layers even boundary layers at curvature of the hull. Nevertheless, the total number of meshes becomes less than the open water analysis due to the confined geometry of the computational domain. The size of the mesh are 0.8~1.4 million. Figure 4-4 is the mesh of whole domain, near the hull and hull surface mesh of case H. The computation domain has been modelled up to the deck height of the ship so that the bottom can be seen from the top. Freeboard height is enough to cover air domain because the confined water cases have relatively slow speed.

Too small size of the mesh at the hull surface was not appropriate because it caused y^+ value too small. A compromise was important between the cell sizes next to the hull surface and the y^+ value to model the narrow gap properly. The meshes were modelled in such a way that y^+ values ranged between 30 and 300 on all the area of the hull surface except the case H. This will be discussed in the next chapter. For the single-phase simulations, number of mesh was about 50~70% of the multi-phase case depending on the under keel clearance and water depth because the domain was modelled up to the flat free

surface level. Under water meshes are basically equal to those used for two-phase solver.

Figure 4-4: Mesh of (a) whole domain and (b) near the hull (looking from the top), and (c) mesh on the hull surface of case H

Turbulence model was determined as $k-\omega$ SST after simple comparison between $k-\epsilon$ and $k-\omega$ SST. Both gave similar force

value but $k-\epsilon$ showed a bit larger oscillation especially for sway force value though showed little difference in the converged force values. Since bigger oscillation makes it difficult to judge convergence, $k-\omega$ SST model was chosen for the analysis.

Just as the KCS open water analysis, three solvers – interFoam, LTSInterFoam and simpleFoam, are used for the confined water analysis. The comparison of the single-phase and the two-phase solvers will show how free surface contributes to the ship-to-bank interaction. As it is normally known that the VoF solver is much more sensitive and instable, this will also help finding out the stability of the VoF solver of OpenFOAM and how reliable results are in comparison to the single-phase solver.

4.2 Boundary conditions

As for the boundary conditions, a fixed value is applied at the inlet for velocity and turbulence properties. At the outlet, zero gradient is applied for velocity and turbulence properties. No slip condition is applied to the hull. For the velocity at the bank boundary, the same fixed value with the inlet velocity was used so that no-slip condition can be applied on the bank wall. This is reasonable for no-slip condition on the bank wall of body fixed coordinate system because the point of view is on the ship and undisturbed water and the bank should have the same velocity.

For the single-phase solver, a slip condition is used for the velocity on the top which is free surface level. Zero gradient can be used to the walls, top and outlet of the pressure boundary. Fixed value is applied at the inlet pressure. Standard wall function of OpenFOAM was used for turbulence properties. Table 4-4 is the summarised boundary conditions applied for the single-phase solver.

**Table 4-4: Boundary conditions for confined water analysis
of the single-phase solver**

Flow properties	inlet	outlet	top	bank	ship
U [m/s]	fixed value (u 0 0)	zero gradient	zero gradient	fixed value (u 0 0)	no slip (0 0 0)
p [m ² /s ²]	zero gradient	fixed value 0	slip	zero gradient	zero gradient
k [m ² /s ²]	fixed value 0.00015	zero gradient	zero gradient	kqRWallFunction 0.00015	
ω [1/s]	fixed value 2	zero gradient	Zero gradient	omegaWallFunction 2	

In the two-phase solver in which air and water exist together, top boundary plane becomes the boundary for atmosphere and some boundary conditions are different from the single-phase solver. The main differences are in the pressure boundary conditions and for the atmospheric boundary. The boundary conditions used for the multi-phase solver are listed in Table 4-5.

The alpha boundary conditions are needed for the volume fraction. In OpenFOAM 2.3.0, new alpha boundary condition has been implemented at the outlet for the conservation of the volume fraction.

Table 4-5: Boundary conditions for confined water analysis of the multi-phase solver

Flow properties	inlet	outlet	atmosphere	bank	ship
U [m/s]	fixed value (u 0 0)	zero gradient	pressureInlet- Outlet- Velocity	Fixed value (u 0 0)	Fixed value (0 0 0)
p_rgh (p _d) [kg / ms ²]	zero gradient	fixedFlux- pressure 0	Total pressure 0	fixedFluxPressure 0	
k [m ² /s ²]	fixed value 0.00015	zero gradient	inletOutlet inletValue 0.00015	kqRWallFunction 0.00015	
ω [1/s]	fixed value 2	zero gradient	inletOutlet inletValue 2	omegaWallFunction 2	
Alpha [-]	calculated	Variable Height FlowRate	inletOutlet	zero gradient	

For the pressure boundary conditions of two phase solver, dynamic pressure is set at the boundary as discussed in the governing equation of the multi-phase solver. The pressure used for multi-phase solver has different unit from single-phase solver as density is being multiplied. ‘fixedFluxPressure’ is an improved boundary condition that sets pressure gradient based on the fixed flux, hence, determined from the velocity condition at the boundary. The flux equation (2.40) is used for this boundary condition to set pressure gradient. For instance, pressure condition becomes no gradient to apply zero flux on the no-slip wall boundary. Therefore, it is equivalent to zero gradient at the

no-slip walls. This was used for the pressure boundary conditions at the inlet and the wall boundaries. Zero gradient is used at the outlet and total pressure is used at the top. Total pressure adjust pressure based on U:

$$p_0 = p + \frac{1}{2}\rho|U|^2 \quad (4.1)$$

Chapter 5

Results and Discussion

5.1 Computation results

For the most of calculations in this study, High Performance Computing (HPC) system in Nanyang Technological University was used for parallel computation. It took about 3 hours for a 100 second simulation of about one million-mesh case with 32 processors. Because the semi-implicit method was implemented to the two-phase solver of OpenFOAM 2.3.0, the computation time was affordable with maximum Courant number 25 and interface Courant number 10. When both Courant numbers were set to be 0.4, it took about one day. For the single-phase steady-state calculations of the steady-state solver, a MacBook Pro with 2.3GHz Intel Core i7 CPU (8 cores) was used. Computation time was all less than 10 minutes.

5.1.1 Force and moment

Figure 5-1 and Figure 5-2 are the convergence time history of forces and moment by two different pressure gradient schemes.

Figure 5-1: Convergence history of (a) X and (b) Y forces (N), and (c) Z moment (Nm) from pressure of case F (Gradient scheme: Gauss linear, Max Co. reduced from 70s)

Figure 5-2: Convergence history of (a) X and (b) Y forces (N), and (c) Z moment (Nm) from pressure of case F (Gradient scheme: Least square)

The computation results of two-phase solver show that pressure component dominates over the viscosity with large oscillations for ship-to-bank interaction. It is presumed that the oscillation is caused from the interaction between the bank and the ship because the oscillation indicates quite different behaviour from the open water analysis in spite of relatively slow speed. Even the longitudinal force showed different oscillating behaviour from the open water analysis that was discussed in the previous chapter. Gauss linear and Least squares schemes that have 2nd order accuracy were evaluated as the gradient scheme. As Gauss Linear method showed big oscillation, Courant number was reduced to 0.4 when the pressure value is stabilised at 70 second. This oppressed the oscillation and helped in averaging values. However, the computation took about one day for the additional 50 seconds simulation with the low Courant number. The force values were less oscillating with Least Square method and with high Courant number. The under relaxation factors were also used for stable solutions as follows:

p	0.3
U	0.7
k	0.7
omega	0.7

Because the values showed oscillating behaviour, the computation had to be monitored over the sufficient time duration and the converged force values were obtained from the average over the time where the oscillation is stabilised.

Three different solvers have been used for the simulations. As discussed in the Chapter 3.3, `simpleFoam` is a single-phase steady-state solver, `interFoam` is a VoF transient solver and `LTSInterFoam` is a local time stepping VoF solver. Table 5-1 is the comparison table of forces and moments between the CFD results and the experiment (Lataire et al., 2009). From the comparison results, `LTSInterFoam` gave wrong values just as shown for KCS analysis in open water. Different mesh type and mesh fineness were tested to rectify this, however, all did not improve the results much although other numerical settings were almost the same with `interFoam`. As it gave wrong results, Least square scheme was not tested for the LTS solver, moreover, the LTS solver was excluded from the evaluation hereafter. `interFoam` gave reasonable accuracy. Comparing the different gradient schemes, least square generally showed better results compared to Gauss linear. `simpleFoam` also allowed good accuracy in spite of single-phase simulation without free surface. Least square method also showed better accuracy for single-phase calculations.

**Table 5-1: Comparison of the cases without thrust to the experiment results
(Lataire et al. [5])**

Case		A		C		F		H	
Solver	Value	Gauss linear	Least Squares	Gauss linear	Least Squares	Gauss linear	Least Squares	Gauss linear	Least Squares
LTSInter-Foam	Fx (N)	-6.960	-	Diverged		-9.345	-	-3.211	-
	Fy (N)	-8.190				-1.029		0.539	
	Mz (Nm)	-11.652				1.259		4.600	
Inter-Foam	Fx (N)	-4.799	-4.866			-5.465	-5.512	-1.559	-2.239
	Fy (N)	-7.122	-5.646			-1.111	-1.008	0.581	1.963
	Mz (Nm)	-8.112	-9.067			1.982	1.305	2.159	5.561
Simple-Foam	Fx (N)	-4.193	-3.922	-4.062	-4.094	-4.395	-3.794	-1.839	-1.880
	Fy (N)	-4.518	-5.343	-0.789	0.709	-1.106	-1.300	0.807	1.245
	Mz (Nm)	-9.948	-8.665	4.145	6.810	0.629	1.053	3.296	4.017
EXP	Fx (N)	-4.130		-2.618		-5.354		-2.935	
	Fy (N)	-5.139		0.962		-1.648		2.545	
	Mz (Nm)	-8.939		4.300		1.804		6.959	

The followings are the comparison results of each case.

Case A The two-phase solver over-predicted longitudinal force slightly, but sway force and yaw moment showed reasonable accuracy with Least square method. The single-phase solver showed good accuracy compared to the experiment. Least Square scheme gave better result of sway force and yaw moment for the both solvers with the reasonable accuracy.

Case C This case did not converge for the VoF solvers in spite of many trials and efforts. However, the single-phase solution was able to converge. Single-phase solver also did not show good accuracy. Even Gauss linear gradient scheme predicted sway force to the opposite direction.

Case F The two-phase solution with Gauss linear gradient scheme gave the good accuracy for longitudinal force and yaw moment, however, sway force was under-predicted. Single-phase solver predicted better sway force but not for other values. The reason for this is not clear.

Case H This case has small under keel clearance which is 10% of water draught, but the separation from the bank wall is however, not small. When same meshing type (2 layers of boundary layer) was used from other cases, all the values were under-predicted. The details will be discussed in the next Section 5.1.2. The presented values in Table 5-1 were when mesh was further refined to have more boundary layers.

5.1.2 Under keel space modelling of the case H

Because of small under keel clearance of the case H, it was difficult to model the mesh in the narrow space properly. If many layers are modelled in the under keel space, y^+ value becomes too low, i.e. below 30. In that case, the wall function may not be valid. To keep y^+ value bigger than 30, the optimum number of layer in the under keel clearance had to be only about two in the case H. The small under keel clearance is known to affect the bank effect significantly, therefore, too few number of layers would not be enough to capture the effect. In this case, the best thing is to model y^+ value less than 1 throughout the hull surface and resolve the whole boundary layer without using wall functions. In the OpenFOAM mesher by which meshing is very slow compared to commercial meshers, this was not efficient because number of mesh becomes too huge.

To compromise between the number of layers and y^+ value in the under keel space, a comparison has been made. Four different meshes are tested for the case H. They have same mesh size in other area but different boundary layers or different mesh in the under keel space. The tested meshes are shown in Figure 5-3 that illustrates the midship section near the bilge radius.

**Figure 5-3: Different meshing of case H -
no layer (a), three layers (b), eight layers (c), fine mesh (d)**

Figure 5-4 shows y^+ distribution for each of the four meshing types.

**Figure 5-4: y^+ of the case H with different meshing
no layer (a), three layers (b), eight layers (c), fine mesh (d)**

(a) has no specific modelling in the boundary layer, (b) has two prism boundary layers and total three layers in the under keel, (c) has eight prism boundary layers and total nine in the under keel, and (d) has very fine refinement in the under keel clearance without modelling boundary layer. (b) type meshing is a general boundary modelling for all other test cases used in this study.

(a) and (b) have y^+ values above 30 and all in good range. OpenFOAM mesher has problem when many prism layers have to be created at the boundary layer especially at large curvature. For this reason, prism layers have not been generated as much as amidships at fore and aft area of (c) where curvature appears. Instead, many mesh layers have been created in the boundary layer at the parallel middle body area where under keel clearance is most significant. In (d), very fine mesh was used in under keel space, hence, very low y^+ distribution can be seen on the bottom hull surface.

The computed results for the different meshes of the case H are tabulated in Table 5-2. Least Square methods are used for the gradient scheme. From the comparison results, (c) type meshing showed the best results. Therefore, it is considered that number of layers in the under keel clearance are important to predict

bank effect. Many prism layers at the boundary layer are more effective than the (d) type meshing.

Table 5-2: Comparison between different meshing in the under keel space of the case H

	(a)	(b)	(c)	(d)	EXP
No. of mesh (mil.)	1.11	1.13	1.15	5.21	-
F _x (N)	1.484	2.239	2.236	2.247	2.935
F _y (N)	0.368	1.104	1.963	1.037	2.545
M _z (Nm)	4.004	3.734	5.561	3.647	6.959

More comparisons are made for the meshing (b) and (c). Figure 5-5 shows pressure distribution in the water at the centreline of the ship.

Figure 5-5: Pressure [N] distribution in the water at the centreline of the case H with (b) three layers (c) eight layers

The pressure distribution on the hull was compared in Figure 5-6. For better visualisation, the range of the pressure applied became different for the two cases.

Figure 5-6: Pressure [N] distribution on the hull (looking from the bottom) of the case H with (b) three layers (c) eight layers

When eight layers are used (c), the overall pressure is higher both in the water (Figure 5-5) and on the hull (Figure 5-6). In Figure 5-5, the pressure difference between the low pressure region (under keel space) and the high pressure region (bow) of the case (c) (ranged in 1380~1475N) is bigger compared to the case (b) (ranged in 1350~1425 N).

5.1.3 Pressure and velocity asymmetry

The computation shows clear pressure asymmetry that causes the bank effect. Figure 5-7 shows the computed pressure distributions of the cases on the hull surface and the bank surface. The deck is not shown as the domain is modelled up to the freeboard height.

Figure 5-7: Pressure [N] distribution on the hull and bank surface of the cases F (a) and H (b) (looking from the top)

Because the incoming velocity is much faster in the case F (0.8012 m/s) than in the case H (0.4578 m/s), the pressure range and its variation are much higher in the case F. However,

the measured sway force and yaw moment from the experiment appeared much bigger in the case H than in the case F. Therefore, it is obvious that under keel clearance affects more significantly although bank effect is known to be almost proportional to velocity square.

Next, three sectional planes were cut into the water to investigate the pressure distributions in the flow. Sections were cut at amidships, near the bow (station 17.5) and near the stern (station 2.5) were cut and presented in Figure 5-8. As force is directed from high to low pressure, the direction of yaw moment can be guessed from this sectional views. High pressure is observed near the bow and this would contribute to the yaw moment. The calculated negative yaw moment of the case A can be observed from the very high pressure at the port side of the bow. On the other hand, the cases F and H where positive yaw moments were calculated have strong high pressure at the starboard side.

Figure 5-9, Figure 5-10 and Figure 5-11 show the velocity magnitude of the cases F, H and I at various water levels. The case I that has small clearance between the ship and the bank was added for better comparison and this case illustrates the most asymmetry of velocity field. The case F that has less

extreme condition than the cases H and I shows the least asymmetry among the three cases. The case F reveals biggest asymmetric velocity at the 1/3 draught level, and the cases H and I reveal at the 2/3 draught level.

Figure 5-8: Dynamic pressure [N] distribution of OpenFOAM calculation of case A (a), F (b) and H (c) at the different sections (station 2.5, amidships and station 17.5)

Figure 5-9: Velocity magnitude [m/s] of the case F at (a) free surface, (b) 2/3 of draught from baseline, (c) 1/3 of draught from baseline and (d) baseline

Figure 5-10: Velocity magnitude [m/s] of the case H at (a) free surface, (b) 2/3 of draught from baseline, (c) 1/3 of draught from baseline and (d) baseline

Figure 5-11: Velocity magnitude [m/s] of the case I at (a) free surface, (b) 2/3 of draught from baseline, (c) 1/3 of draught from baseline and (d) baseline

5.1.4 Results of the cases with propeller rotation

Table 5-3 is the comparison results of the cases with propeller thrust. Sway forces and yaw moments are compared with the experiment results. Since local time stepping solver showed wrong results for most of the cases, it has not been included in these cases. Although the case D and the case I had small clearance of the sloped bank and the hull surface, however the solution converged well with reasonable accuracy of yaw moment.

Table 5-3: Comparison of the cases with thrust to the experiment results (Lataire et al. [5])

Solver	Case	D		I		J	
	Value	Gauss Linear	Least Square	Gauss Linear	Least Square	Gauss Linear	Least Square
inter-Foam	Fy (N)	-2.011	-0.754	-0.201	0.137	-0.894	-0.696
	Mz (Nm)	19.28	15.932	9.750	12.60	2.169	1.838
simple-Foam	Fy (N)	-2.683	-2.354	-1.080	-0.699	-1.427	Diverged
	Mz (Nm)	4.692	5.108	1.960	2.117	1.688	
EXP	Fy (N)	-8.056		-3.682		-2.264	
	Mz (Nm)	16.407		8.312		2.065	

As can be seen in the comparison, when the propeller was rotating, all the sway forces showed under prediction compared

to the experiment. In these cases, Gauss linear method showed relatively better results than least square method for yaw moment. The direction of the sway forces was mostly correct but the magnitudes were all wrong. When the propeller is rotating, sway forces were all in suction direction to the bank wall. However, yaw moment values showed reasonable accuracy. This implies the propeller rotation significantly affects the magnitude of sway forces. Single-phase solver showed poor accuracy even for yaw moments. Further comparison for these cases are not reliable but will become speculation, hence, it is not meaningful.

The single-phase calculations with Least Squares scheme for the case J diverged.

5.1.5 Comparison between single-phase and two-phase solutions

It was expected that the difference between the calculated values and the experiment single-phase solver and two-phase solver would show how free surface affects on the bank effect. However, the single-phase solution of the cases A and F showed even better accuracy than two-phase solution. In other cases, single-phase solution showed under prediction compared to the two-phase solutions. Especially the cases with propulsion showed

poor accuracy. From the case C, single-phase solver is more stable than two-phase solver.

5.1.6 Free surface elevation

Free surface elevations were compared with experiment results. The wave measurements were done in three points of the cases C, F, I and J. The comparison was done for these available cases except the case C that was failed to converge. Table 5-4 is the location of the wave gauges from centreline of the ship according to the experiment. Wave gauges measure the wave elevation at the defined transverse distances from the centreline of the ship.

Table 5-4: The locations (m) of the wave gauges with reference to the centreline of the ship (Lataire et al. [5])

	Wave gauge 1	Wave gauge 2	Wave gauge 3
Case F	0.475	1.005	1.535
Case I	0.875	1.215	2.115
Case J	0.920	1.320	1.785

Figure 5-12 is the graphical views of the measuring locations of the wave gauges and also gives a brief idea about the water depth at each wave gauge.

Figure 5-12: The locations of the wave gauges 1, 2 and 3 of the cases (a) F, (b) I and (c) J

Figure 5-13, Figure 5-14 and Figure 5-15 show the comparison of wave elevations with the experiment measurements. In the horizontal scale, 0 stands for amidships and positive direction is toward the stern.

Figure 5-13: Wave elevation of the case F

at (a) gauge 1, (b) gauge 2 and (c) gauge 3 (Lataire et al. [5])

Figure 5-14: Wave elevation of the case I

at (a) gauge 1, (b) gauge 2 and (c) gauge 3 (Lataire et al. [5])

Figure 5-15: Wave elevation of the case J

at (a) gauge 1, (b) gauge 2 and (c) gauge 3 (Lataire et al. [5])

The further wave gauges from the ship showed less accuracy of the wave elevation. A steep wave is observed in the case I and it is continued from the hull to the bank wall. This was also found in the experiment but over-predicted in the computation result at the gauges 1 and 2. In addition, phase delay was observed, especially at the furthest wave gauge from the ship, i.e. wave gauge 3. This phase delay was also found in the case F. The phase of wave trough was deviated by about a quarter of wave length. The case J showed relatively good accuracy. As can be seen in Figure 5-12, the wave elevation is considered less accurate at shallow water region – at the wave gauges 3 of the case F, and at the wave gauge 2 and 3 of the case I. Lastly, the computations did not capture the oscillation of the wave elevation correctly. The experiment showed more oscillation but this was not captured or captured differently in the computation results.

5.2 Discussion

OpenFOAM was able to predict ship-to-bank interaction with the different bank types in extremely shallow water. Two-phase solver failed to converge from finding the solution of one severe case that has extreme under keel clearance and small separation from the sloped bank wall.

For the stable calculations and convergence, good quality of meshing was particularly important in OpenFOAM compared to other commercial CFD codes. Another reason for this is pressure becomes sensitive in the multi-phase solver compared to the single-phase solver. Special care is required for the aspect ratio and high skewness of the grid.

Another challenge is a compromise between the mesh fineness and y^+ value to model the narrow gap between the hull and the bank. Boundary layer and the narrow gap modelling are important although y^+ values have to go slightly below 30 that falls to buffer layer. For the case with small under keel clearance, good meshing improves sway force significantly.

The pressure values of the calculations in confined water often show more oscillating behaviour as compared to the open water

analysis when converging. It is considered to occur from the reflections and interactions between the bank and the ship. Reducing Courant number was helpful to suppress the oscillation. Least Square method of gradient scheme showed less oscillating behaviour than Gauss linear. Overall, Gauss linear scheme showed better accuracy, however Courant Number had to be reduced to very small, e.g. 0.4 in order to capture the force and moment correctly due to the oscillation.

The computation tends to under-predict forces and moment. This is most obvious in sway force. There would be several reasons for this. Firstly, there is limitation of meshing in the narrow gap. One thing can try is resolving whole boundary layer by modelling y^+ below 1. However, this is not easy with OpenFOAM mesher and a commercial mesher should be employed for efficient boundary layer modelling. Secondly, fixed heave and pitch motion of CFD calculations could have caused this under-prediction because the two motions were free in the model tests. This assumption of fixed motion neglects squat effect, therefore, more accuracy can be expected if the 2 DOF motions are set to be free. However, the calculated sway forces and yaw moments from the bank effect are all in correct directions for all the types of bank shapes and arrangement although ship-to-bank interaction is very complex and difficult to predict the directions.

The under-prediction was more obvious in the cases with propeller rotation while yaw moments were predicted with reasonable accuracy. This explains how propeller rotation affects the sway force. More study is needed whether propeller intervention cause additional sway force to specific direction. In this set of experiments, all the cases with propeller rotation resulted in suction forces and the computation without propeller modelling computed the sway forces to the correct direction but in smaller magnitude. However, it is hard to conclude this at this stage. From this comparison, it is noted that propeller rotation seems to affect the magnitude of sway forces but does not give significant contribution to yaw moment.

Free surface elevation was less accurate than the open water analysis. In addition, difference was found in wave phase. Such a phase delay was more obvious in the case where the water depth is shallow. Less oscillation on the free surface level was observed in the computation than the experiment result.

Chapter 6

Conclusion

6.1 Conclusion

In this study, CFD theory and numerical techniques were discussed. In order to use open source CFD code, the discretisation of governing equations, solution algorithm, stability of the computation, boundary condition and boundary layer treatment were explained. Several features and the solvers of OpenFOAM were also investigated.

In this study, flow around ships in open water and in confined water with several bank arrangements was simulated by incompressible RANS solver of OpenFOAM. The results from single-phase and two-phase solver were compared. The hydrodynamic forces and moment due to the bank, wave elevation were compared to the experiment results.

Under keel clearance modelling was evaluated to compromise between mesh fineness and y^+ distribution. The gradient schemes of OpenFOAM were compared in terms of accuracy and stable computation. OpenFOAM was able to predict the bank

effect in confined water that causes the bank effect as well as the open water analysis. Asymmetric pressure and velocity fields were well presented in connection to the calculated forces and moment. In order to find the propeller contribution to the bank effect, the cases with propeller thrust were compared to the computations without propeller.

The reason for under prediction of sway force has to be further investigated with careful meshing and numerical schemes. Unfortunately, the experiment results open to the public were only available to those particular cases, therefore, systematic parametric study was not possible.

6.2 Future work

As for the CFD analysis of the bank effect, only one case which is the vertical bank arrangement was successful in this report. Analysis for other cases with sloped banks and with different under keel clearances have to be tested, too. There are two types of sloped bank in the experiments. One type is surface piercing sloped bank and the other is where the slope ends below the waterline.

Since good meshing affects the calculations of multiphase solver of OpenFOAM significantly, there will be intensive study in meshing options especially for the refinement at near wall region. Otherwise, commercial software can be used for the meshing. Meshing software that can export compatible file type to OpenFOAM can be used for this. In addition, it will be valuable to compare the results of OpenFOAM to that of a commercial code.

OpenFOAM has strong dynamic mesh options with adaptive mesh refinement capability. This can add pitch and heave motion of ship, hence, it is expected that the inclusion of shallow water effect will give better accuracy.

Appendix

The source codes of interFoam

interFoam.C: The main codes of the solver. It handles PIMPLE loop and connects all the related header files to solve the governing equations.

```
1      #include "fvCFD.H"
2      #include "CMULES.H"
3      #include "subCycle.H"
4      #include "interfaceProperties.H"
5      #include "incompressibleTwoPhaseMixture.H"
6      #include "turbulenceModel.H"
7      #include "pimpleControl.H"
8      #include "fvIOoptionList.H"
9      #include "fixedFluxPressureFvPatchScalarField.H"
10
11     // * * * * *
12
13     int main(int argc, char *argv[])
14     {
15         #include "setRootCase.H"
16         #include "createTime.H"
17         #include "createMesh.H"
18
19         pimpleControl pimple(mesh);
20
21         #include "initContinuityErrs.H"
22         #include "createFields.H"
23         #include "readTimeControls.H"
24         #include "createPrghCorrTypes.H"
25         #include "correctPhi.H"
26         #include "CourantNo.H"
27         #include "setInitialDeltaT.H"
28
29         // * * * * *
30
```



```

31 Info<< "\nStarting time loop\n" << endl;
32
33 while (runTime.run())
34 {
35     #include "readTimeControls.H"
36     #include "CourantNo.H"
37     #include "alphaCourantNo.H"
38     #include "setDeltaT.H"
39
40     runTime++;
41
42     Info<< "Time = " << runTime.timeName() << nl << endl;
43
44     // --- Pressure-velocity PIMPLE corrector loop
45     while (pimple.loop())
46     {
47         #include "alphaControls.H"
48
49         if (pimple.firstIter() || alphaOuterCorrectors)
50         {
51             twoPhaseProperties.correct();
52
53             #include "alphaEqnSubCycle.H"
54             interface.correct();
55         }
56
57         #include "UEqn.H"
58
59         // --- Pressure corrector loop
60         while (pimple.correct())
61         {
62             #include "pEqn.H"
63         }
64
65         if (pimple.turbCorr())
66         {
67             turbulence->correct();
68         }
69     }
70
71     runTime.write();
72
73     Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
74         << " ClockTime = " << runTime.elapsedClockTime() << " s"
75         << nl << endl;

```

```

76     }
77
78     Info<< "End\n" << endl;
79
80     return 0;
81 }

```

AlphaEqn.H: It contains equations for the volume fraction α including MULES algorithm (Refer to Chapter 3.3)

```

1  {
2    word alphaScheme("div(phi,alpha)");
3    word alphasScheme("div(phirb,alpha)");
4
5    // Standard face-flux compression coefficient
6    surfaceScalarField phic(interface.cAlpha()*mag(phi/mesh.magSf()));
7
8    // Add the optional isotropic compression contribution
9    if (icAlpha > 0)
10   {
11     phic *= (1.0 - icAlpha);
12     phic += (interface.cAlpha()*icAlpha)*fvc::interpolate(mag(U));
13   }
14
15   // Do not compress interface at non-coupled boundary faces
16   // (inlets, outlets etc.)
17   forAll(phic.boundaryField(), patchi)
18   {
19     fvsPatchScalarField& phicp = phic.boundaryField()[patchi];
20
21     if (!phicp.coupled())
22     {
23       phicp == 0;
24     }
25   }
26
27   tmp<surfaceScalarField> tphiAlpha;
28
29   if (MULESCorr)

```

```

30  {
31      fvScalarMatrix alpha1Eqn
32      (
33          #ifdef LTSSOLVE
34      fv::localEulerDdtScheme<scalar>(mesh,rDeltaT.name()).fvmDdt(alpha1)
35          #else
36      fv::EulerDdtScheme<scalar>(mesh).fvmDdt(alpha1)
37          #endif
38      + fv::gaussConvectionScheme<scalar>
39      (
40          mesh,
41          phi,
42          upwind<scalar>(mesh, phi)
43      ).fvmDiv(phi, alpha1)
44      );
45
46      alpha1Eqn.solve();
47
48      Info<< "Phase-1 volume fraction = "
49          << alpha1.weightedAverage(mesh.Vsc()).value()
50          << " Min(alpha1) = " << min(alpha1).value()
51          << " Max(alpha1) = " << max(alpha1).value()
52          << endl;
53
54      tmp<surfaceScalarField> tphiAlphaUD(alpha1Eqn.flux());
55      tphiAlpha = tmp<surfaceScalarField>
56      (
57          new surfaceScalarField(tphiAlphaUD())
58      );
59
60      if (alphaApplyPrevCorr && tphiAlphaCorr0.valid())
61      {
62          Info<< "Applying the previous iteration compression flux" << endl;
63          #ifdef LTSSOLVE
64          MULES::LTScorrect(alpha1, tphiAlpha(), tphiAlphaCorr0(), 1, 0);
65          #else
66          MULES::correct(alpha1, tphiAlpha(), tphiAlphaCorr0(), 1, 0);
67          #endif
68
69          tphiAlpha() += tphiAlphaCorr0();
70      }
71
72      // Cache the upwind-flux
73      tphiAlphaCorr0 = tphiAlphaUD;
74

```

```

75     alpha2 = 1.0 - alpha1;
76
77     interface.correct();
78 }
79
80 for (int aCorr=0; aCorr<nAlphaCorr; aCorr++)
81 {
82     surfaceScalarField phir(phic*interface.nHatf());
83
84     tmp<surfaceScalarField> tphiAlphaUn
85     (
86         fvc::flux
87         (
88             phi,
89             alpha1,
90             alphaScheme
91         )
92     + fvc::flux
93     (
94         -fvc::flux(-phir, alpha2, alphasScheme),
95         alpha1,
96         alphasScheme
97     )
98 );
99
100 if (MULESCorr)
101 {
102     tmp<surfaceScalarField> tphiAlphaCorr(tphiAlphaUn() - tphiAlpha());
103     volScalarField alpha10(alpha1);
104
105     #ifdef LTSSOLVE
106     MULES::LTScorrect(alpha1, tphiAlphaUn(), tphiAlphaCorr(), 1, 0);
107     #else
108     MULES::correct(alpha1, tphiAlphaUn(), tphiAlphaCorr(), 1, 0);
109     #endif
110
111     // Under-relax the correction for all but the 1st corrector
112     if (aCorr == 0)
113     {
114         tphiAlpha() += tphiAlphaCorr();
115     }
116     else
117     {
118         alpha1 = 0.5*alpha1 + 0.5*alpha10;
119         tphiAlpha() += 0.5*tphiAlphaCorr();

```

```

120     }
121 }
122 else
123 {
124     tphiAlpha = tphiAlphaUn;
125
126     #ifdef LTSSOLVE
127     MULES::explicitLTSSolve(alpha1, phi, tphiAlpha(), 1, 0);
128     #else
129     MULES::explicitSolve(alpha1, phi, tphiAlpha(), 1, 0);
130     #endif
131 }
132
133     alpha2 = 1.0 - alpha1;
134
135     interface.correct();
136 }
137
138     rhoPhi = tphiAlpha()*(rho1 - rho2) + phi*rho2;
139
140     if (alphaApplyPrevCorr && MULESCorr)
141     {
142         tphiAlphaCorr0 = tphiAlpha() - tphiAlphaCorr0;
143     }
144
145     Info<< "Phase-1 volume fraction = "
146         << alpha1.weightedAverage(mesh.Vsc()).value()
147         << " Min(alpha1) = " << min(alpha1).value()
148         << " Max(alpha1) = " << max(alpha1).value()
149         << endl;
150 }

```

AlphaEqnSubCycle.H: This is for additional sub-cycles in the given time step in order to compute stable α field.

```

1     if (nAlphaSubCycles > 1)
2     {
3         dimensionedScalar totalDeltaT = runTime.deltaT();
4         surfaceScalarField rhoPhiSum

```

```

5      (
6      IOobject
7      (
8      "rhoPhiSum",
9      runTime.timeName(),
10     mesh
11     ),
12     mesh,
13     dimensionedScalar("0", rhoPhi.dimensions(), 0)
14     );
15
16     for
17     (
18         subCycle<volScalarField> alphaSubCycle(alpha1, nAlphaSubCycles);
19         !(++alphaSubCycle).end();
20     )
21     {
22         #include "alphaEqn.H"
23         rhoPhiSum += (runTime.deltaT()/totalDeltaT)*rhoPhi;
24     }
25
26     rhoPhi = rhoPhiSum;
27 }
28 else
29 {
30     #include "alphaEqn.H"
31 }
32
33 rho == alpha1*rho1 + alpha2*rho2;

```

UEqn.H: The velocity equation of the governing equation. This is coupled with pressure equation in “PEqn.H”.

```

1      fvVectorMatrix UEqn
2      (
3          fvm::ddt(rho, U)
4          + fvm::div(rhoPhi, U)
5          + turbulence->divDevRhoReff(rho, U)
6      );
7

```

```

8      UEqn.relax();
9
10     if (pimple.momentumPredictor())
11     {
12         solve
13         (
14             UEqn
15             ==
16             fvc::reconstruct
17             (
18                 (
19                     fvc::interpolate(interface.sigmaK())*fvc::snGrad(alpha1)
20                     - ghf*fvc::snGrad(rho)
21                     - fvc::snGrad(p_rgh)
22                 ) * mesh.magSf()
23             )
24         );
25     }

```

PEqn.H: The pressure equation of the governing equation as discussed in Chapter 2.2.2. “phiHbyA” in the source codes stands for $\frac{H(U)}{a_p}$. See the equations (2.35) ~ (2.39).

```

1      {
2          volScalarField rAU("rAU", 1.0/UEqn.A());
3          surfaceScalarField rAUf("rAUf", fvc::interpolate(rAU));
4
5          volVectorField HbyA("HbyA", U);
6          HbyA = rAU*UEqn.H();
7
8          surfaceScalarField phiHbyA
9          (
10             "phiHbyA",
11             (fvc::interpolate(HbyA) & mesh.Sf())
12             + fvc::interpolate(rho*rAU)*fvc::ddtCorr(U, phi)
13          );
14
15          adjustPhi(phiHbyA, U, p_rgh);

```

```

16
17     surfaceScalarField phig
18     (
19         (
20             fvc::interpolate(interface.sigmaK())*fvc::snGrad(alpha1)
21             - ghf*fvc::snGrad(rho)
22         )*rAUf*mesh.magSf()
23     );
24
25     phiHbyA += phig;
26
27     // Update the fixedFluxPressure BCs to ensure flux consistency
28     setSnGrad<fixedFluxPressureFvPatchScalarField>
29     (
30         p_rgh.boundaryField(),
31         (
32             phiHbyA.boundaryField()
33             - (mesh.Sf()).boundaryField() & U.boundaryField())
34         )/(mesh.magSf()).boundaryField()*rAUf.boundaryField()
35     );
36
37     while (pimple.correctNonOrthogonal())
38     {
39         fvScalarMatrix p_rghEqn
40         (
41             fvm::laplacian(rAUf, p_rgh) == fvc::div(phiHbyA)
42         );
43
44         p_rghEqn.setReference(pRefCell, getRefCellValue(p_rgh, pRefCell));
45
46         p_rghEqn.solve(mesh.solver(p_rgh.select(pimple.finalInnerIter())));
47
48         if (pimple.finalNonOrthogonalIter())
49         {
50             phi = phiHbyA - p_rghEqn.flux();
51
52             p_rgh.relax();
53
54             U = HbyA + rAU*fvc::reconstruct((phig - p_rghEqn.flux())/rAUf);
55             U.correctBoundaryConditions();
56             fvOptions.correct(U);
57         }
58     }
59
60     #include "continuityErrs.H"

```



```

61
62     p == p_rgh + rho*gh;
63
64     if (p_rgh.needReference())
65     {
66         p += dimensionedScalar
67         (
68             "p",
69             p.dimensions(),
70             pRefValue - getRefCellValue(p, pRefCell)
71         );
72         p_rgh = p - rho*gh;
73     }
74 }

```

References

- [1] Alberto Ghione, Development and Validation of a two-phase CFD model using OpenFOAM, Master Thesis, Royal Institute of Technology, Stockholm, Sweden, 2012
- [2] C. Lo, Dong-Taur Su and I-Fu Lin, Applying Computational Fluid Dynamics to Simulate Bank Effects, 19th International Offshore and Polar Engineering Conference, page 466~471, Osaka, Japan, 2009
- [3] D. W. Park, H. J. Choi and K. J. Paik, Bank Effect of a Ship Operating in a Shallow Water and Channel, Journal of Navigation and Port, Volume 38, No. 1, page 19~27, 2014
- [4] E. Lataire, M. Vantorre, E. Laforce, K. Eloot and G. Delefortrie, Navigation in Confined Waters: Influence of Bank Characteristics on Ship-Bank Interaction, International Conference on Ship Manoeuvring in Shallow and Confined Water, page 135~143, Abtwerp, Belgium, 2007
- [5] E. Lataire, M. Vantorre and K. Eloot, Systematic Model Tests on Ship-Bank Interaction Effects, International Conference on Ship Manoeuvring in Shallow and Confined Water: Bank Effect, Abtwerp, Belgium, 2009
- [6] Eric Furbo, Evaluation of RANS turbulence models for flow problems with significant impact of boundary layers, Master thesis, Uppsala University, Sweden, 2010
- [7] H. K. Versteeg and W. Malalasekera, An Introduction to Computational Fluid Dynamics, 2nd Edition, Pearson Education Limited, UK, 2007

- [8] Hirt C. W., Nichols B. D, Volume of Fluid (VoF) method for the dynamics of free boundaries, J. of Computer Physics, 1981, 39(1), page 201-225
- [9] Hrvoje Jasak, Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows, Thesis for Doctor of Philosophy, University of London and Imperial College, UK, 1996
- [10] Huaming Wang, Zaojian Zou, Yonghe Xie and Wanlun Kong, Numerical Study of Viscous Hydrodynamic Forces on a Ship Navigating near Bank in Shallow Water, Proceedings of the 20th International Offshore and Polar Engineering Conference, page 523~528, Beijing, China, 2010
- [11] Iranzo, J. Domingo, I. Trejo, M. Terceno and J. Valle, Calculation of the resistance and the profile of a 3600 TEU container ship, Canal de Experiencias Hidrodinamicas, El Pardo, Ministerio de Defensa, Barcelona, Spain, 2007
- [12] Issa, R.I., Solution of the implicitly discretized fluid flow equations by operator-splitting, J. Comp. Physics, 62:40-65, 1986
- [13] John D. Anderson, Computational Fluid Dynamics, The Basics with Applications, McGraw-Hill Inc., New York, USA, 1995
- [14] K. Lee and S. G. Lee, Investigation of ship maneuvering with hydrodynamic effects between ship and bank, Journal of Mechanical Science and Technology, No. 22, page 1230~1236, 2008
- [15] Kevin Maki, Ship Resistance Simulation with OpenFOAM, 6thOpenFOAM workshop, PA, USA, 2011

- [16] Kongsberg Maritime, Hydrodynamic Modelling Tool Report SU-0689, Mathematical Model of Ship Dynamics, Norway, 2012
- [17] Knut Inge Engelbreth, Flow Around a Circular Cylinder Near a Plate Wall, Master Thesis, Norwegian University of Science and Technology, Norway, 2011
- [18] L. Larsson, B. Regnstrom, L. Broberg, D. Li, C. Janson, Features, Fantasies, and Feats in the Theoretical/Numerical Prediction of Ship Performance, Proceedings of the 22nd Symposium on Naval Hydrodynamics, page 11-32, Washington D.C, USA, 1998
- [19] L. Zou, L. Larsson, G. Delefortrie and E. Lataire, CFD Prediction and Validation of Ship-bank Interaction in a Canal, 2nd International Conference on Ship Manoeuvring in Shallow and Confined Water: Ship to Ship Interaction, May 18-20, Trondheim, Norway, 2011
- [20] Liu Jing, Investigation of multiphase liquid system in microchannels, PhD thesis, School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore, 2014
- [21] M. Vantorre, and G. Delefortrie, Experimental Investigation of Ship-Bank Interaction Forces, Proceeding of International Conference on Marine Simulation and Ship Manoeuvrability, MARSIM'03, page RC-31-1~9, Kanazawa, Japan, 2003
- [22] Mehdi Nakisa, Adi Maimun, Yasser M. Ahmed, A.Y.Sian, A. Priyanto, Jaswaramd F. Behrouzi, RANS Simulation of the Viscous Flow around Hull of LNG Ship in Confined Water, 15th Mathematical and Computational Methods in Science and Engineering, page 133~137, Kuala Lumpur, Malaysia, 2013

- [23] Norrbin, Bank Effect on a Ship Moving through a Short Dredged Channel, Proceedings of 10th Symposium on Naval Hydrodynamics, page 71~88, Cambridge, MA, USA, 1974
- [24] OpenFOAM®, The Open source CFD toolbox, User Guide 2.3.0
- [25] OpenFOAM®, The Open source CFD toolbox, Programmers Guide 2.3.0
- [26] OpenFOAM® v2.0.0 official website about Steady-State VoF solver, <http://www.openfoam.org/version2.0.0/steady-VoF.php> (Last accessed in Jan 2015).
- [27] OpenFOAM® v2.3.0 official website about Multiphase modelling,
<http://www.openfoam.org/version2.3.0/mutiphase.php>
(Last accessed in Mar 2015).
- [28] OpenFOAM wiki website, <http://openfoamwiki.net> (Last accessed in May 2014).
- [29] Patankar, S.V., Numerical heat transfer and fluid flow, Hemisphere Publishing Corporation, Taylor & Francis Group, New York, USA, 1980
- [30] Pedro Miguel Borges Lopes, Free-surface flow interface and air-entrainment modelling using OpenFOAM, Doctorial Program Report, Universidade de Coimbra, Portugal, 2013
- [31] Q. Li, M. Leer-Andersen, P. Ottosson, P. Trägårdh, Experimental Investigation of Bank Effects Under Extreme Conditions, Proceeding of PRADS 2001, page 541-546, Shanghai, China, 2001

- [32] Quanming Miao and Jinzhu Xia, Numerical Study of Bank Effect on a Ship Traveling in a Channel, The 8th International Conference on Numerical Ship Hydrodynamics, Busan, Korea, 2003
- [33] Santiago Marquez Damian, An Extended Mixture Model for the Simultaneous Treatment of Short and Long Scale Interfaces, PhD Thesis, Universidad Nacional Del Litoral, Argentina, 2013
- [34] Schlichting H., Boundary Layer Theory, 7th edition, McGraw-Hill, USA, 1979
- [35] S. G. Lee, Ship motion and manoeuvring, Lecture Note, page 116~119, Department of Naval Architecture and Ocean Engineering, Pusan National University, Korea, 1998
- [36] The manoeuvring Committee of ITTC, Final Report and Recommendations to the 23rd ITTC, Proceedings of the 23rd International Towing Tank Conference - Volume I, page 201~234, Venice, Italy, 2002
- [37] The manoeuvring Committee of ITTC, Final Report and Recommendations to the 26th ITTC, Proceedings of the 26th International Towing Tank Conference - Volume I, page 129~154, Rio de Janeiro, Brazil, 2011
- [38] W. J. Kim, S. H. Van and D. H. Kim, Measurements of Flows Around Modern Commercial Ship Models, Experiments in Fluids, page 567~578, 2001

Publication from the present work

Youngkook Kim, E. Y. K. Ng and Gopala Krishnan, CFD Prediction of Ship-to-bank Interaction for Shiphandling Simulator, MARTECH 2014, Singapore, 29~30 April, 2014