VERTICAL MOTION SIMULATION OF A DTMB 5415M SHIP IN HEAD SEA USING THE VARIATIONAL MULTI-SCALE METHOD

ZHONGWEI SHEN

Supervisor: Ido Akkerman
Co-supervisor: Marco ten Eikelder

Ship Hydrodynamics
3mE
TU Delft

November 2019

Zhongwei Shen: *Vertical Motion Simulation Of A DTMB 5415M Ship In Head Sea Using The Variational Multi-scale Method,* © November 2019 TU Delft

Father’s love like mountains, and mother’s love like water.

— Mao Dun

Dedicated to my parents.

ABSTRACT

This master thesis presents the computation of free-surface flow phenomena with emphasis on ship hydromechanics. We perform a simulation of the heave and the pitch motion of the DTMB 5415M ship moving in head sea. To this purpose we employ an existing Arbitrary-Lagrangian-Eulerian residual-based variational multiscale method (ALE-RBVMS) to discretize the incompressible Navier-Stokes equations and use the level-set method to capture the interface. We validate the numerical results obtained from the ALE-RBVMS computation with experimental data. We show that these results significantly improve upon those acquired from other CFD codes. This demonstrates the viability of the framework for simulations in ship hydromechanics.

"Obvious" is the most dangerous word in mathematics

— E.T.Bell

ACKNOWLEDGEMENTS

Many people helped me during the process of my thesis. However, I am not good at the expression of my gratitude. Here, I will give my best thankfulness to all of you. First of all, I want to thank my supervisor assistant professor Ido Akkerman. Thank you for your instruction, patience and suggestions. Thank you for teaching me how to perform scientific research and introducing me into the Linux world, which is initially painful but also beautiful. I never thought I could come to the world of programming. Although there is still a long way to be an expert, you are my first guidance counselor.

In addition, many thanks to my co-supervisor, and also my dear friend Marco who is studying variational multi-scale method as a PhD. If anyone is interested in this topic, please contact him. I think he made a break through recently. As my co-supervisor, you give my a lot of useful suggestions to my thesis writing and a lot of guidance in the formula derivation. Without your supervision and your help, I will struggle longer! It is a great honour to work with you.

At the end, I want to thank my parents. Family is everything, without my parents' support and encouragement I could not be here. Also thank the people who helped me, or encouraged me, in the process of my master degree. Especially thank my friend Boyao Wang. Many times we discussed about FEM and FVM. I think both of us got a lot of improvements.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

SYMBOLS

Operator

$\frac{D}{Dt}$	material derivative
∂	partial differential
∇	gradient operator
$\nabla\cdot$	divergence
∇^s	symmetric gradient
$\|\cdot\|$	normal of a vector
$\dot{\mathbf{d}}$	discrete time-derivative of $\mathbf{d}$
$()^T$	transpose vector
$\otimes$	dyadic products
$\propto$	proportional to

Roman symbols

A	area
$B(,)$	bi-linear form
$B(,,)$	tri-linear form
C	constant
$\mathbf{c}$	convective velocity
$\mathbf{d}$	displacement
$\mathbf{e}$	element vector
F	force on the ship
$\mathbf{G}$	second rank metric tensor
$\mathbf{g}$	gravity vector
H	Heaviside function
H_ε	regularized Heaviside function
H_n	Hessenberg matrix
$\underline{\underline{\mathbf{I}}}$	identity matrix
$\mathbf{J}$	Jacobean symbol
K, G, D, L	matrices
K_n	Krylov spaces
L	random material property
M	momentum
N	shape function
$\mathbf{n}$	normal vector
$\mathbf{R}$	residual

$\mathbf{r}_M$	momentum residual
r_C	continuity residual
r_ϕ	level set residual
$\mathbb{O}$	knot vector space
$\mathbb{P}$	projector
p	pressure
Q	source or sink
$\mathbf{Q}$	angular velocity matrix
$\underline{\underline{\mathbf{T}}}$	deviatoric stress tensor
t	time
$\triangle t$	time step
$\tilde{t}$	pseudo time
$\mathbf{X}$	material domain
$\mathbf{x}$	spatial domain
x, y, z	Cartesian coordinate system
$\mathbf{U}$	unknown variables set
$\mathbb{V}$	shape function space
$\mathbf{u}$	fluid velocity vector
$\mathbb{W}$	weighting function space
$\boldsymbol{w}(q)$	weighting function

Greek character

ρ	density
Ω	problem domain
μ	dynamic viscosity
ν	kinematic viscosity
$\boldsymbol{\tau}$	shear stress
σ	Cauchy stress tensor
ϕ	level set factor
ϕ_d	level set factor after re-distancing
ξ, η, ψ	reference coordinate system (η in the code represents the weighting function of level set)
$\boldsymbol{\omega}$	angular velocity vector
χ	referential domain
α, γ	time derivative parameter
Γ or $\partial\Omega$	boundary
ϕ, Φ, Ψ	transfer functions
ε, ϵ	regularization parameter
λ_{pen}	penalty constant
$\boldsymbol{\kappa}$	discontinuity-capturing viscosity

ACRONYMS

ALE	Arbitrary Lagrangian Eulerian
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CFD	Computational Fluid Dynamics
CFL	Courant–Friedrichs–Lewy number
COG	Center Of Gravity
DOF	Degree of Freedom
FDM	Finite Difference Method
FEM	Finite Element Method
FFT	Fast Fourier Transform
FVM	Finite Volume Method
GMRES	Generalized Minimum Residual
IGA	Isogeometric Analysis
LES	Large Eddy Simulation
NURBS	Non Uniform Rational B Spline
RANS	Reynolds Averaged Navier-Stokes
RBVMS	Residual-Based Variational Multi-scale
SUPG	Streamline Upwind Petrov Galerkin
VMS	Variational Multi-scale

1 INTRODUCTION

The first numerical simulation was made by Richardson in 1922 for weather prediction. However, it is made by hand rather than computers. Although the result of this first try diverged, it is still a mile stone in the CFD history. Several years later, computers are applied in CFD simulations which leads to a break through.

To discretize the Navier-Stokes equations, many methods have been proposed. At the beginning, Direct Numerical Simulation (DNS) is chosen based on people's tuition. It proved impossible to simulate high Reynolds number problems, since the mesh size is scaled with $1/Re^4$ for unsteady flow. Here Re is the symbol of the Reynolds Number. To solve this problem, many solutions are created including the two main branches, Reynolds Averaged Navier-Stokes (RANS) and Large Eddy Simulation (LES). This is because the turbulence occurs cascade and the smaller class turbulence behave isotropic, which can be expressed by models based on the understanding of turbulence behavior. Although the formulations are quite similar in these two branches, the physical meanings are different. For RANS, it consumes less computational resources than LES, however it wipes many details out of the fluid through time average method. Since it is faster than LES, the engineers prefer to use it. Some famous models such as $k-\varepsilon$ model (*Jones & Launder 1972*) and $k-\omega$ (*Wilcox 1988*) are developed to fulfill the need of the industry. LES nowadays are mainly used in the academic domain, which gives more accurate results but consume more resources. The first LES simulation was made by Smagorinsky 1963 [19] with an eddy viscosity model.

1.1 RBVMS AND FEM

RBVMS was first introduced in [11]. Comparing to those traditional LES methods, projectors are used in RBVMS to substitute LES filters and this approach saved a lot of troubles at the same time. The concept is to separate the governing equations into two scales through a projector. Only the coarse-scale is solved while the fine scale will be used to solve the residual based variables. In this way, it retains numerical consistency in the coarse-scale equations and thus permits full rate-of-convergence comparing to the usual method.

In this thesis and also in our code, the discrete algorithm is Finite Element Method (FEM). FEM is one of the most prominent numerical simulation techniques in continuum mechanics. It has successfully been applied in many engineering problems, especially in solid mechanics. For fluid problems, Finite Volume Method (FVM) is mostly used in the previous history. However, in this thesis, since a ship will be studied, which means extremely complicated geometry, FEM becomes our

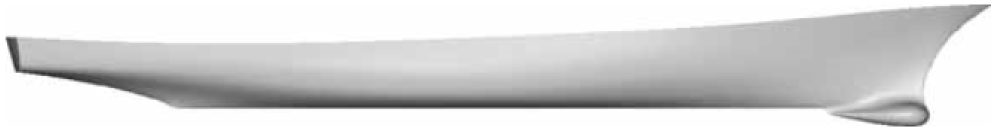


Figure 1.1: US Navy Combatant, DTMB 5415M

first choice based on its advantage of the description of geometry.

1.2 DTMB

In this thesis, benchmark DTMB 5415M is used to for our validation. DTMB 5415M is a famous standard US Navy combatant. Its geometry and public data can be downloaded from website [1]. Figure 1.1 is the picture from its website and it is clear to find that DTMB 5415M has a bulb bow, which means a very complex geometry. This is the reason that we choose FEM. The experimental data and some famous hydrodynamic and CFD software results are reported in [21] and [22].

1.3 SOFTWARE

Usually people would like to use some commercial software to do a CFD simulation. However, there is no such commercial software using RBVMS theory in the market now. And also it is quite expensive to use an LES method in an engineering problem. Therefore, we decided to use the RBVMS code which was built by University of California, San Diego (UCSD) several years ago. It allows large MPI separation which hugely decreases the time consuming in simulations. The code is compiled with the Fortran language. And there are other options such as Phython and C++.

Besides the RBVMS code, many other software are used to accomplish this thesis. The geometry is recorded in a Rhino file with a .igs format. A software called Gmsh is used for generating the mesh and another software named Metis is used to partition our mesh. Paraview is applied in this thesis as the post-processing software.

1.4 CHALLENGES IN RBVMS

In traditional LES methods, an eddy viscosity is applied in the resolved fine scales to represent turbulent dissipation, which introduces a consistency error. This makes the consistency in the coarse scales not fully retained. Additionally, the energy transfer to unresolved modes is too small when the discretization is very coarse. To

[1] US Navy Combatant, DTMB 5415:
http://www.simman2008.dk/5415/5415_geometry.htm

overcome these two mainly shortcomings, this RBVMS method is created to capture all scales consistently and to avoid use of any ad hoc devices, such as eddy viscosity.

However, some other challenges appear in the process of derivation. The only approximation is made in the expression of the fine scale solution when the decomposition is performed in time and space. To make the fine scale approximation analyzed in a framework, an assumption of infinite perturbation series expansion is proposed to treat the fine scale nonlinear term in the fine scale equation. According to this assumption, with the smaller of fine scales, the expansion should converge rapidly. With combining a matrix Green's function[8], this theory is expected to to give an exact formal solution of the fine scale Navier Stokes equations. Nevertheless, it is not easy to solve NS equations analytically in a such way. Some approximations will be employed in practical problem solving. Reader can consult this content in [3] for more detail.

1.5 OBJECTIVE AND OUTLINE

The final objective of this master thesis is performing the validation of a DTMB 5415M ship in vertical motion in a real scale using the RBVMS code. Before this, a wave simulation should be done to decide the grid size and the parameters in the input file. Based on these parameters, many optimization method will be tested to speed up our simulation time and ensure a good simulation result.

There are three parts with seven chapters in total and the work of this master thesis will be introduced step by step.

In the first part Chapter 2, the governing equations of free-surface air-water modeling will be introduced. At the beginning, the derivation will start with Navier-Stokes equations in section 2.1. Then the Arbitrary Lagrangian-Eulerian method is applied to our governing equations to make the ship move in a fix domain. In section 2.3, the continuous level set method is discussed to separate the whole domain into two parts, the air and the water, and create a surface flow. Till now, the whole chapter is in a continuous state which means our governing equations are continuous. In Chapter 3, the governing equations will be discretized in space. Therefore, in section 3.1 the RBVMS method will be introduced and some assumptions are made here to approximate the exact solution. In section 3.2, after discretizing the Navier-Stokes equations, level set function will also be discretized in the same way. In section 3.3, to make smooth the change from the water to the air, a regularized Heaviside function will be introduced. Later, in section 3.4, a technology call re-distancing is presented to optimize the level set method. In section 3.5, with the above derivation, the mass is not balanced because of the re-distancing method. Therefore, another formulation is added to correct the mass balance. In section 3.6, an additional term called discontinuity capturing term will be added in our governing equation to avoid the appearance of wiggles. At the end, in section 3.7, the ship motion equation will be written to close our formulation group. After spacial discretization, our time integration method will be introduced in Chapter 4. Here, General-α method is chosen in section 4.1. In section 4.2 the level set time integration method will be introduced.

Part 2 is about the simulation results calculated by our code in comparison with the benchmark DTMB 5415M ship. In Chapter 5, a wave simulation is demonstrated first to give the feeling of parameter set and test our code. In section 5.1, an introduction of the wave simulation will be given. In section 5.2, The boundary conditions and test method will be introduced. In section 5.3 some methods are used to optimize the simulation and improve our results. Based on this simulation, it is much easier to forward to ship simulation and save a lot of computational time. In section 5.4, the final parameter setting will be given to end this chapter. In chapter 6, the simulation of DTMB 5415M ship is performed. At the beginning in section 6.1, some detail information of our study case is introduced. In section 6.2, the mesh will be decided based on wave simulation. Later in section 6.3, the simulation results of the DTMB 5415M ship are shown. The results will be compared with experimental results and another CFD software results in heave and pitch motion. To make our simulation more reliable, in section 6.4, the mesh convergence will be studied. In section 6.5, the error estimation will be shown based on the study of section 6.4. For Chapter 7, an conclusion will be made according to our simulation results and some recommendations are given for the future study.

Part 3 is the appendix about some concepts, formulations and technologies used in our code. Appendix A gives the shape function of our code and a small introduction of IGA will be given as additional background knowledge which is considered good for future study. Appendix B and appendix C will introduce GMRES technology and Newton iteration separately.

Part I

NUMERICAL METHODOLOGY

This part is divided into three chapters. Chapter 2 is about the governing equations in this thesis in a continuous form. The Navier-Stokes equations are introduced in the beginning and then the ALE method is applied to the NS equations. At the end, the level set method is discussed to capture the free surface. The formulas in this chapter are all in a continuous form. In Chapter 3, the spatial discrete formulations of the governing equations are performed using the RBVMS method. Chapter 4 presents the time integration method for which the Generalized-α is employed.

2 GOVERNING EQUATIONS OF FREE-SURFACE FLOW

In this chapter, three sections will be introduced step by step. At the very beginning, the derivation of Navier-Stokes is written to give readers our initial governing equations of this thesis. Although many books have already shown the process of derivation, section 2.1 just gives its definition based on author's physical understanding. In section 2.2, ALE approach is written to make the ship moves in a fixed domain. In section 2.3 a method of tracking free surface, called Level Set method, is illustrated. One thing should be mentioned is that all of the equations in this chapter is in a continuous form. No assumptions and discrete technology is used here.

2.1 NAVIER-STOKES EQUATION

To derive Navier-Stokes equation, material derivative should be introduced first :

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla. \tag{2.1}$$

Here $\frac{D}{Dt}$ is the material derivative and $\mathbf{u}$ is fluid velocity. The blue color indicates that it is convective velocity. ∇ represents gradient operator and t means time.

For structure study, Lagrangian description method is often chosen based on its convenience for study. However, in fluid mechanics, people usually do not concern each particles movement and meanwhile it is not convenient to track the material in fluid experiments. Therefore, people normally use Eulerian description to study fluid. The material derivative can transfer the coordinate system from Lagrangian description to Eulerian description.

The NS equations are derived from conservation principles. For incompressible flow, mass conservation and momentum conservation rules are used. While for compressible flow, the energy conservation equation will be considered, where the density is not a constant. One more unknown variable means the need of one more equation. The formula below is the Reynolds Transport Theorem. From a physical understanding, it means the sum of changes of some extensive property equals the lost or gained through boundaries plus source or sink,

$$\frac{d}{dt}\int_{\Omega} L dV = -\int_{\partial\Omega} L\mathbf{u} \cdot n dA - \int_{\Omega} Q dV. \tag{2.2}$$

Here L is some extensive property defined over a control volume Ω and Q represents sources and sinks. $\partial\Omega$ represents domain boundary. n is the normal vector that is outside oriented.

Applying divergence theorem to the first term on the right side of Equation 2.2, then,

$$\frac{d}{dt}\int_{\Omega} L dV = -\int_{\Omega} \nabla \cdot (L\mathbf{u}) dV - \int_{\Omega} Q dV,$$
$$\int_{\Omega} (\frac{\partial L}{\partial t} + \nabla \cdot (L\mathbf{u}) + Q) dV = 0,$$
$$\frac{\partial L}{\partial t} + \nabla \cdot (L\mathbf{u}) + Q = 0.$$

From the first equation to the second equation , Leibniz's rule is used to shift total derivative to partial derivative.

For mass conservation, L will be substituted by ρ and Q will be zero since there is no source or sink term,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\mathbf{u}) = \frac{\partial \rho}{\partial t} + \rho\nabla \cdot \mathbf{u} + \mathbf{u} \cdot \nabla\rho = 0. \tag{2.3}$$

In incompressible flow, the density is a constant, which means its time derivative is zero, then the mass conservation formulation (also known as continuity equation) will be,

$$\nabla \cdot \mathbf{u} = 0. \tag{2.4}$$

Momentum conservation equation is much more complicated, where force will be calculated as sorce term. Then, L will be substituted by $\rho\mathbf{u}$, and this $\mathbf{u}$ represent momentum velocity.

$$\frac{\partial \rho\mathbf{u}}{\partial t} + \nabla \cdot (\rho\mathbf{u}\mathbf{u}) + (-\nabla \cdot \sigma - \rho\mathbf{g}) = 0. \tag{2.5}$$

Here, σ is Cauchy stress tensor and $\mathbf{g}$ is gravity vector. For the first two terms, it can be expressed in another way since it is incompressible flow:

$$\begin{aligned}\frac{\partial \rho\mathbf{u}}{\partial t} + \nabla \cdot (\rho\mathbf{u}\mathbf{u}) &= \rho\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\frac{\partial \rho}{\partial t} + \rho\mathbf{u}\nabla \cdot \mathbf{u} + \rho\mathbf{u} \cdot \nabla\mathbf{u} + \mathbf{u}\mathbf{u} \cdot \nabla\rho \\ &= \mathbf{u}(\frac{\partial \rho}{\partial t} + \rho\nabla \cdot \mathbf{u} + \mathbf{u} \cdot \nabla\rho) + \rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla\mathbf{u}).\end{aligned}$$

The first term on the right hand side equals to zero according to mass conservation Equation 2.3. For the force term it can be express in another way:

$$\begin{aligned}\sigma = \begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{bmatrix} &= -\begin{bmatrix} p & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & p \end{bmatrix} + \begin{bmatrix} \sigma_{xx}+p & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy}+p & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz}+p \end{bmatrix} \\ &= -p\underline{\underline{\mathbf{I}}} + \underline{\underline{\mathbf{T}}}.\end{aligned}$$

Here $\underline{\underline{\mathbf{I}}}$ is identity matrix, and $\underline{\underline{\mathbf{T}}}$ is the deviatoric stress tensor, p is the mean normal strss. The stress tensor can be expressed as the sum of two stress tensors, namely: the hydrostatic stress tensor and the deviatoric stress tensor.

$$p = -\frac{1}{3}(\sigma_{xx} + \sigma_{yy} + \sigma_{zz}).$$

Then it becomes,

$$\rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}) = \rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \nabla \cdot \underline{\underline{\mathbf{T}}} + \rho \mathbf{g}. \tag{2.6}$$

Here, readers may notice that material derivative is used here. If Newton second law is used here to derive momentum equation, this material derivative is just the right hand side of equation $F = ma$ in an Eulerian form. For Newtonian fluid, $\tau \propto \frac{\partial u}{\partial y}$. There are three assumptions made here: 1) The stress tensor is a linear function of the strain rates. 2) The fluid is isotropic. 3) For a fuild at rest, $\nabla \cdot \underline{\underline{\mathbf{T}}} = 0$. This results in,

$$T_{ij} = \mu(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}) + \delta_{ij}\lambda\frac{\partial \mathbf{u}}{\partial x_k}.$$

Here δ_{ij} is the Kronecker delta, when $i = j$ $\delta_{ij} = 1$, otherwise if $i \neq j$ $\delta_{ij} = 0$. μ and λ are proportionality constants associated with the assumption that stress depends on strain linearly, known as viscosity. And λ is called bulk viscosity, which most common approximation is $\lambda \approx -2/3\mu$. For incompressibe flow, the last term equals to zero. Then the Navier-Stokes equation will be written as below,

A fluid normally can be defined as a Newtonian fluid or a non-Newtonian fluid. The difference is that in the Newtonian fluid, the viscous stresses are linearly proportional to the local strain rate, $\tau = \mu\frac{du}{dy}$. Which means if μ is a constant, the fluid is a Newtonian fluid. However, no real fluid fits the definition perfectly, many common liquids and gases, such as water and air, can be assumed to be Newtonian for practical calculations under ordinary conditions.

$$\rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \nabla \cdot \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathbf{T}}) + \rho \mathbf{g}, \tag{2.7}$$

or,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = -\nabla P + \nabla \cdot \nu(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathbf{T}}) + \mathbf{g}, \tag{2.8}$$

where $\otimes$ is the dyadic product symbol, and $P = p/\rho$.

This is non-conservation format of Navier-Stokes equation. In some other Navier-Stokes equation derivation, conservation form is used. If your solution is expected to be smooth, then non-conservative may work. For fluids, this is shock-free flows. If you have shocks, or chemical reactions, or any other sharp interfaces, then the conservative form can be used. It shows at below:

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathbf{T}}) + \rho \mathbf{g}.$$

Here ν is kinetic viscosity and $\nu = \mu/\rho$. In this thesis, non-conservation form is used to derive our formulation, and also the same in the code. It is quite similar in conservation derivation, reader can derive it by yourself. In Equation 2.7, which is used in our code, although ρ is a constant in single fluid study, it will be still kept in the governing equation. Because this is a two phase flow with a free surface which means the density will change from water to air through this surface. Without this density, the whole equation is not correct.

2.2 ARBITRARY LAGRANGIAN-EULERIAN (ALE) METHOD

Although Eulerian description usually used in a fluid study, it has its own weakness to present the flow in our problem. For purely Eulerian description, precise

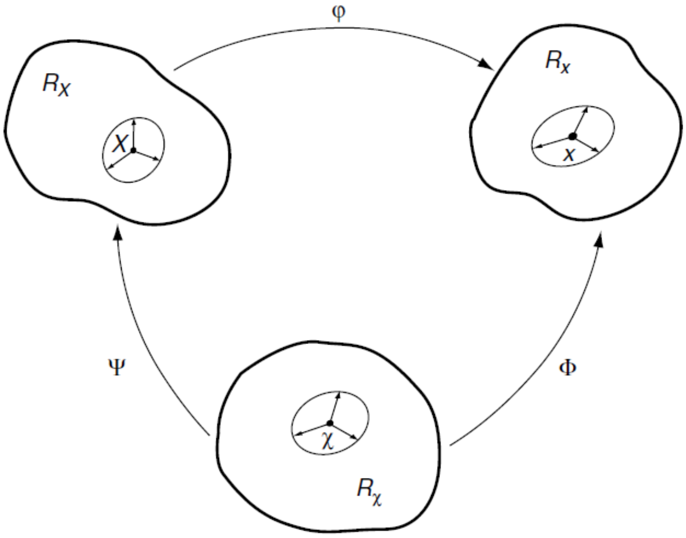


Figure 2.1: Eulerian domian, Lagrangian domian and ALE domian

interface definition and high resolution of flow details are not possible. However, this method can handle large distortion problems easily. This is totally the opposite of purely Lagrangian description. It is quite easy to track the free surfaces and interfaces between different materials for Lagrangian description. And "It also facilitates the treatment of materials with history-dependent constitutive relations", quote from [7]. However, for large distortion problems, Lagrangian description is not powerful anymore.

Therefore, people created a new method to describe the flow, namely Arbitrary Lagrangian Eulerian method. It combines the best features of both the Lagrangian and the Eulerian approaches. Using this ALE method can solve large distortions problems while allowing for a clear delineation of free surfaces and fluid-fluid interfaces easily. The formula below is introduced in paper [7].

Here a short introduction is written to give readers a basic feeling of ALE. Three symbols (φ, Φ and Ψ) are used to express the applications among three different coordinate systems, φ is the application function from material domain (Eulerian domain) to spatial domain (Lagrangian domain), Φ is the application function from referential domain (ALE domain) to material domain and Ψ is the application function from referential domain to spatial domain. This is illustrated by Figure 2.1. Their expressions are shown below,

$$\frac{\partial \varphi}{\partial(\mathbf{X},t)} = \begin{pmatrix} \frac{\partial x}{\partial \mathbf{X}} & \mathbf{u} \\ 0^T & 1 \end{pmatrix}, \quad \frac{\partial \Phi}{\partial(\chi,t)} = \begin{pmatrix} \frac{\partial x}{\partial \chi} & \hat{\mathbf{u}} \\ 0^T & 1 \end{pmatrix}, \quad \frac{\partial \Psi^{-1}}{\partial(\mathbf{X},t)} = \begin{pmatrix} \frac{\partial \chi}{\partial \mathbf{X}} & \boldsymbol{w} \\ 0^T & 1 \end{pmatrix}. \tag{2.9}$$

The velocity in each domain can be expressed as,

$$\begin{aligned}
\mathbf{u}(\mathbf{X},t) &= \left.\frac{\partial x}{\partial t}\right|_{X}, \\
\hat{\mathbf{u}}(\chi,t) &= \left.\frac{\partial x}{\partial t}\right|_{\chi}, \\
\boldsymbol{w} &= \left.\frac{\partial \chi}{\partial t}\right|_{X}.
\end{aligned} \tag{2.10}$$

The relation among velocities $\mathbf{u}, \hat{\mathbf{u}}$ and $\boldsymbol{w}$ can be obtained by differentiation $\varphi = \Phi \cdot \Psi^{-1}$, therefore,

$$\mathbf{u} = \hat{\mathbf{u}} + \frac{\partial x}{\partial \chi} \cdot \boldsymbol{w}. \tag{2.11}$$

Hence, convective velocity is

$$\mathbf{c} = \mathbf{u} - \hat{\mathbf{u}} = \frac{\partial x}{\partial \chi} \cdot \boldsymbol{w}. \tag{2.12}$$

This will be the new convective velocity of fluid. At the beginning, the fluid convective velocity is expressed as $\mathbf{u}$, now it can be substitute by this $\mathbf{c}$ or $\mathbf{u} - \hat{\mathbf{u}}$. After applying ALE, Equation 2.7 becomes:

$$\rho(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla \mathbf{u}) = -\nabla p + \nabla \cdot \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^{T}) + \rho \mathbf{g}. \tag{2.13}$$

In the above equation, $\hat{\mathbf{u}}$ is the domain velocity. For example, in our simulation, the Froude number is 0.248, which means ship velocity is $9.256 m/s$. Then the domain velocity is also $9.256 m/s$, and ship will move in the domain and it will not step out of the domain gradually.

2.3 CONTINUOUS LEVEL SET METHOD

For simulating a free surface flow, a scalar, named ϕ, is added into the governing equation. For this part, reader can refer paper [1], [2] and [17]. It is assumed that the air-water interface is moving with the fluid material particles, which gives us an additional convection equation for the level set ϕ:

$$\frac{\partial \phi}{\partial t} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla \phi = 0. \tag{2.14}$$

Reader should notice that in Equation 2.14, ALE method is applied into the equation.

The scalar ϕ is used to separate the water domain and air domain. Let's set the domain to $\Omega \in \mathbb{R}$, and $\partial\Omega$ denotes its boundary, $\partial\Omega_d$ denotes essential boundary condition, also known as Dirichlet boundary condition, and $\partial\Omega_n$ denotes natural boundary condition. $\mathbf{u} : \Omega \in \mathbb{R}^3$ is the flow velocity, and let's define the inflow and outflow parts at boundary as below,

$$\begin{aligned}
\partial\Omega^{-} &= \{x \in \partial\Omega | \mathbf{u} \cdot \mathbf{n} < 0\}, \\
\partial\Omega^{+} &= \{x \in \partial\Omega | \mathbf{u} \cdot \mathbf{n} \geqslant 0\}.
\end{aligned} \tag{2.15}$$

Here $\mathbf{n}$ is the normal vector pointed outside. And $p : \Omega \in \mathbb{R}$. As we discussed before, for a free surface flow studied in this thesis, a scalar ϕ is built to define air domain and water domain, $\phi : \Omega \in \mathbb{R}$ moving with the fluid, and also define its surface as below,

$$\begin{aligned}
\Omega_w &= \{\mathbf{x}|\phi(\mathbf{x},t) > 0\},\\
\Omega_a &= \{\mathbf{x}|\phi(\mathbf{x},t) < 0\},\\
\Gamma_{aw} &= \{\mathbf{x}|\phi(\mathbf{x},t) = 0\}.
\end{aligned}$$

For the air and the water, the density and viscosity function can be expressed in the following way separately,

$$\begin{aligned}
\rho &= \rho_w H(\phi) + \rho_w(1 - H(\phi)),\\
\mu &= \mu_a H(\phi) + \mu_a(1 - H(\phi)),
\end{aligned}$$

where,

$$H(\phi) = \left\{\begin{array}{ll} 0 & \text{if } \phi > 0 \\ \frac{1}{2} & \text{if } \phi = 0 \\ 1 & \text{if } \phi < 0 \end{array}\right\}. \tag{2.16}$$

Equation 2.16 describes the Heaviside function.

Some reader may find that the above function has a problem, when the code simulates the free surface, there is a jump value of density and viscosity from the water to the air, which makes the solution discontinuous and it is not able to solve. To fix this problem, a regularized Heaviside function is developed to avoid this problem. It will be introduced in Chapter 3. Therefore, our governing equations become,

$$\begin{aligned}
&\rho(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla \mathbf{u}) + \nabla p - \nabla \cdot \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) = \rho \mathbf{g},\\
&\nabla \cdot \mathbf{u} = 0,\\
&\frac{\partial \phi}{\partial t} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla \phi = 0.
\end{aligned} \tag{2.17}$$

This is the final form of our continuous governing equations, however, it is not the final form of our discrete governing equations. After applying RBVMS method to discrete our governing equation, it will become much more longer, because of the existence of additional terms generated by RBVMS. Furthermore, additional discontinuity capturing terms will be added into the governing equations to prevent the appearance of wiggles. Therefore, there is still a long way to get the end.

3 SPATIAL DISCRETE FORMULATION

This chapter is the main chapter of our first part. It consists of 7 different sections which will present the spatial discritization of our governing equation. In section 3.1, RBVMS method is introduced through 3 subsections. The weak form of our governing equation is introduced first, then the VMS method and its assumptions are followed. In section 3.2, the VMS form of level set function is written. In section 3.3, to solve the problem in section 2.3, regularized Heaviside function is introduced here to smooth the free surface. Then, in section 3.4, re-distancing method is introduced to re-define the free surface. And a penalty term is added to maintain the free surface. Since the re-distancing method is used here, a new problem will come up. The mass conservation equation will not be balanced. Therefore, in section 3.5 a new equation is added to re-balance our mass conservation equation. In section 3.6, the concept of discontinuity capturing will be introduced, the reason of creating this term in equation is to avoid wiggles in the results. At last, in section 3.7, since a ship is simulated in waves, the ship motion equation will be shown to close our equation group.

3.1 RBVMS

RBVMS is proposed by *Hughes, Thomas JR and Mazzei, Luca and Jansen, Kenneth E* in year 2000 [11], and other important papers are *Hughes, Thomas JR 1995* [9], *Hughes, Thomas JR et al 1998* [14] and *Y Bazilevs 2007* [3]. It is a kind of large eddy simulation but using a projector not a filter. Furthermore, it does not employ any ad hoc devices, such as eddy viscosity. After many experts' tests, it appears that it has an advantage of approximating advective and diffusive process than the other methods.

Traditionally, spatial filtering has been used to separate the scales in a turbulent simulation. The VMS method has several advantages over classical large-eddy simulation.The variational projection and a priori scale separation provide a solid mathematical framework for the development of sub grid-scale models. Furthermore, the method can be used to apply different modeling assumptions to different scales in the simulation. In this section, a detailed derivation process will be introduced.

3.1.1 *Weak Form Equation*

Before changing our equations into a weak form formulation, let's define solution space and weighting function space first. Let $\mathbb{V}$ denotes the discrete solution space for the velocity-pressure-level set $\{\mathbf{u}, p, \phi\}$, and let $\mathbb{M}$ denotes the discrete weighting function space for linear momentum, continuity and level set equations $\{w, q, \eta\}$.

For simplification of the formulation, the governing formulation will be express in a variational form, like $(A, B)_\Omega = \int_\Omega AB d\Omega$, which means it can be square integral, then the governing equations in a integral format can be written as below,

$$\begin{aligned}
&\left(w, \rho\frac{\partial u}{\partial t}\right)_\Omega + (w, \rho(u - \hat{u}) \cdot \nabla u)_\Omega + (w, \nabla p)_\Omega - \left(w, \nabla \cdot \mu \left(\nabla u + (\nabla u)^T\right)\right)_\Omega \\
&\qquad = (w, \rho g)_\Omega, \\
&(q, \nabla \cdot u)_\Omega = 0.
\end{aligned} \tag{3.1}$$

Here, level set function is not considered. It will be introduced later. To simplify the diffusion term, $\left(\nabla u + (\nabla u)^T\right)/2$ will be substituted by $\nabla^s u$, then it becomes,

$$\begin{aligned}
&\left(w, \rho\frac{\partial u}{\partial t}\right)_\Omega + (w, \rho(u - \hat{u}) \cdot \nabla u)_\Omega + (w, \nabla p)_\Omega - (w, \nabla \cdot 2\mu\nabla^s u)_\Omega = (w, \rho g)_\Omega, \\
&(q, \nabla \cdot u)_\Omega = 0.
\end{aligned} \tag{3.2}$$

After using integration by parts, $\int_\Omega f(\nabla \cdot g) d\Omega = \int_\Gamma (fg) \cdot n d\Gamma - \int_\Omega \nabla f \cdot g d\Omega$, Equation 3.2 becomes,

$$\begin{aligned}
&\left(w, \rho\frac{\partial u}{\partial t}\right)_\Omega + (w, \rho(u - \hat{u}) \cdot \nabla u)_\Omega - (\nabla \cdot w, p)_\Omega + (\nabla^s w, 2\mu\nabla^s u)_\Omega \\
&- (w, \rho g)_\Omega - (w, 2\mu\nabla^s u \cdot n)_\Gamma + (w, p \cdot n)_\Gamma = 0, \\
&(q, \nabla \cdot u)_\Omega = 0.
\end{aligned} \tag{3.3}$$

Here, only diffusion term and pressure term used integration by parts. This is because the unknown variable p is a constant, which does not have C^1 continuity.[1] And the unknown variable velocity is linear, according to our linear shape function, which means it does not have C^2 continuity. Through this integration by parts, the continuity of these two terms are decreased.

In the weak form of FEM, there are many branches. Normally people use the classical method called Garlerkin method, which means the shape function of unknown variables and weighting functions are the same. For more information of our shape function, reader can refer Appendix A. For the other methods, like least square method, is frequently used in energy calculation. Garlerkin method is very successful in symmetric matrix simulation. However, for asymmetric matrix, it is not so powerful, therefore SUPG method is invited as the "upwind scheme" in FEM analysis, which will be introduced in the next section.

[1] The smoothness of a function can be expressed by the symbol C, their meanings are shown below:
C^0: Curves are continuous.
C^1: First derivatives are continuous.
C^2: First and second derivatives are continuous.
C^n: First through nth derivatives are continuous

3.1.2 *Variational Multiscale Method*

For saving the formulation hand writing, the equation will be changed to another format. $B_1(\cdot,\cdot)$ express a bilinear form and $B_2(\cdot,\cdot,\cdot)$ represent a trilinear form, and the continuity equation and momentum equation can be written into one equation, as it shows below,

$$B(\boldsymbol{W},\boldsymbol{U}) = B_1(\boldsymbol{W},\boldsymbol{U}) + B_2(\boldsymbol{W},\boldsymbol{U},\boldsymbol{U}) = L(\boldsymbol{W}). \tag{3.4}$$

Where,

$$\begin{aligned}
&B_1(\boldsymbol{W},\boldsymbol{U}) = \left(\boldsymbol{w},\rho\frac{\partial \boldsymbol{u}}{\partial t}\right)_\Omega + (q,\nabla\cdot\boldsymbol{u})_\Omega - (\nabla\cdot\boldsymbol{w},p)_\Omega + (\nabla^s\boldsymbol{w},2\mu\nabla^s\boldsymbol{u})_\Omega \\
&B_2(\boldsymbol{W},\boldsymbol{U},\boldsymbol{U}) = (\boldsymbol{w},\rho(\boldsymbol{u}-\hat{\boldsymbol{u}})\cdot\nabla\boldsymbol{u})_\Omega\,, \\
&L(\boldsymbol{W}) = (\boldsymbol{w},\rho\boldsymbol{g})_\Omega\,.
\end{aligned}$$

Here we put the continuity equation and momentum equation into one equation for simplification. To simplify the derivation, in this thesis we assume $\boldsymbol{u} = 0$ on the boundary, which means essential boundary condition $\partial\Omega_d$ and for pressure $\int_\Omega p(t)d\Omega = 0$. Therefore the boundary terms are disappeared. However, in the wave simulation and ship simulation, the velocity has its value on boundaries, for example at inflow boundary condition, the velocity equals wave x-direction velocity and for back and forth domain boundary, the velocity equal to domain velocity in y direction. This will be introduced in Part 2.

Then projector $\mathbb{P} : \mathbb{V} \to \overline{\mathbb{V}}$ will be introduced , for more detail of projector please refer [12]. Usually there are two types of projector, $\mathbb{H}$ and $\mathbb{P}$, $\mathbb{H}$ is good for nonlinear and $\mathbb{P}$ is good for this thesis. Then the mesh can be separated into two parts, coarse mesh and fine mesh by using this projector:

$$\overline{\boldsymbol{U}} = \mathbb{P}\boldsymbol{U},\ \boldsymbol{U} = \overline{\boldsymbol{U}} + \boldsymbol{U}', \tag{3.5}$$

$$\begin{aligned}
&B(\overline{\boldsymbol{W}},\overline{\boldsymbol{U}}+\boldsymbol{U}') = L(\overline{\boldsymbol{W}}) - - - - \text{coarse scale}, \\
&B(\boldsymbol{W}',\overline{\boldsymbol{U}}+\boldsymbol{U}') = L(\boldsymbol{W}') - - - - \text{fine scale}.
\end{aligned} \tag{3.6}$$

In the code only the coarse mesh will be solved, while the fine mesh formulation will be used to calculate $\boldsymbol{U}'$ and then substitute the $\boldsymbol{U}'$ in the coarse mesh to solve the whole equation. The detailed functions of these two meshes will be shown below,

$$\begin{aligned}
B(\overline{\boldsymbol{W}},\overline{\boldsymbol{U}}+\boldsymbol{U}') =& B_1(\overline{\boldsymbol{W}},\overline{\boldsymbol{U}}) + B_1(\overline{\boldsymbol{W}},\boldsymbol{U}') + B_2(\overline{\boldsymbol{W}},\overline{\boldsymbol{U}},\overline{\boldsymbol{U}}) + B_2(\overline{\boldsymbol{W}},\overline{\boldsymbol{U}},\boldsymbol{U}') + \\
& B_2(\overline{\boldsymbol{W}},\boldsymbol{U}',\overline{\boldsymbol{U}}) + B_2(\overline{\boldsymbol{W}},\boldsymbol{U}',\boldsymbol{U}') = L(\overline{\boldsymbol{W}}), \\
B(\boldsymbol{W}',\overline{\boldsymbol{U}}+\boldsymbol{U}') =& B_1(\boldsymbol{W}',\overline{\boldsymbol{U}}) + B_1(\boldsymbol{W}',\boldsymbol{U}') + B_2(\boldsymbol{W}',\overline{\boldsymbol{U}},\overline{\boldsymbol{U}}) + B_2(\boldsymbol{W}',\overline{\boldsymbol{U}},\boldsymbol{U}') + \\
& B_2(\boldsymbol{W}',\boldsymbol{U}',\overline{\boldsymbol{U}}) + B_2(\boldsymbol{W}',\boldsymbol{U}',\boldsymbol{U}') = L(\boldsymbol{W}').
\end{aligned} \tag{3.7}$$

For the fine mesh it can be rewritten as below,

$$B_{\boldsymbol{U}}(\boldsymbol{W}',\boldsymbol{U}') + B_2(\boldsymbol{W}',\boldsymbol{U}',\boldsymbol{U}') = \left\{\boldsymbol{W}',\mathrm{Res}(\overline{\boldsymbol{U}})\right\}_{\mathbb{V}',\mathbb{V}'^*}, \tag{3.8}$$

whereas,

$$\begin{aligned}
&B_U(\mathbf{W}',\mathbf{U}') = B_1(\mathbf{W}',\mathbf{U}') + B_2(\mathbf{W}',\overline{\mathbf{U}},\mathbf{U}') + B_2(\mathbf{W}',\mathbf{U}',\overline{\mathbf{U}}),\\
&\left\{\mathbf{W}', \mathrm{Res}(\overline{\mathbf{U}})\right\}_{\mathbb{V}',\mathbb{V}'^*} = L(\mathbf{W}') - B_1(\mathbf{W}',\overline{\mathbf{U}}) - B_2(\mathbf{W}',\overline{\mathbf{U}},\overline{\mathbf{U}}).
\end{aligned}$$

$\mathbf{Res}(\overline{\mathbf{U}}) \in \mathbb{V}'^*$.

3.1.3 *Residual Based Assumption*

For this part, reader can find detail in [3]. Physically, it seems reasonable to give an assumption that the larger the space $\overline{\mathbb{V}}$, the better the approximation of the fine scale solution, and the smaller the coarse scale residual $\mathbf{Res}(\overline{\mathbf{U}}) \in \mathbb{V}'^*$. However, it is quite difficult to give a rigorous mathematical justification. Here, based on this consideration, people suggest a perturbation series expansion to describe the approximation:

$$\mathbf{U}' = \epsilon\mathbf{U}'_1 + \epsilon^2\mathbf{U}'_2 + \epsilon^3\mathbf{U}'_3 + \cdots = \sum_{k=1}^{n} \epsilon^k\mathbf{U}'_k. \tag{3.9}$$

Here ϵ represent the residual, $\epsilon = ||\mathbf{Res}(\overline{\mathbf{U}})||_{\mathbb{V}'^*}$. After test, it is satisfied to work on the most simple and basic end of the approximation spectrum. The idea is to compute $\boldsymbol{\tau}$, the element-wise stabilization operators. This is why it is called Residual based simulation, the approximation is based on the residual.

$$\mathbf{U}' \approx \widetilde{\mathbf{U}}' = \begin{Bmatrix} \widetilde{\mathbf{u}}' \\ \widetilde{p}' \end{Bmatrix} \approx -\boldsymbol{\tau}\mathbf{Res}(\overline{\mathbf{U}}^h), \tag{3.10}$$

$$\begin{aligned}
&\boldsymbol{\tau} = \begin{bmatrix} \tau_M \mathbf{I}_{3\times3} & 0 \\ 0 & \tau_C \end{bmatrix}, \ \mathbf{Res}(\mathbf{U}^h) = \begin{Bmatrix} \mathbf{r}_M(\mathbf{u}^h, p^h) \\ r_C(\mathbf{u}^h) \end{Bmatrix},\\
&\widetilde{\mathbf{u}}' = \tau_M \mathbf{r}_M(\mathbf{u}^h, p^h),\\
&\widetilde{p}' = \tau_C r_C(\mathbf{u}^h),\\
&\mathbf{r}_M(\mathbf{u}^h, p^h) = \rho\frac{\partial \mathbf{u}^h}{\partial t} + \rho(\mathbf{u}^h - \hat{\mathbf{u}}^h)\cdot\nabla\mathbf{u}^h + \nabla p^h - \nabla\cdot 2\mu\nabla^s\mathbf{u}^h - \rho\mathbf{g},\\
&r_C(\mathbf{u}^h) = \nabla\cdot\mathbf{u}^h,\\
&\tau_M = \left(\frac{4}{\triangle t^2} + (\mathbf{u}^h - \hat{\mathbf{u}}^h)\cdot\mathbf{G}(\mathbf{u}^h - \hat{\mathbf{u}}^h) + C_I\nu^2\mathbf{G}:\mathbf{G}\right)^{-1/2},\\
&\tau_C = (\tau_M\mathbf{g}\cdot\mathbf{g})^{-1}.
\end{aligned}$$

$\overline{\mathbf{U}}^h$ is an assumption of $\overline{\mathbf{U}}$, where h is related to the mesh size. For the coarse mesh, the unknown variables $\overline{\mathbf{U}}$ will be approximated by shape functions. In this thesis, linear shape function is only considered. The derivation of FEM shape function can be found in Appendix A. The higher order we use, the higher convergence speed will be, however, it also costs much more time to compute. In this thesis it is believed that the linear basic function is enough[2].

In above equations, $\mathbf{G}$ is a second rank metric tensor, defined as $\mathbf{G} = \frac{\partial \xi^T}{\partial x}\frac{\partial \xi}{\partial x}$, and ξ is reference coordinate. Since the mesh is built by different shape of tetrahedron and they need to be mapped into the standard tetrahedron to make the simulation simpler, and this standard tetrahedron is demonstrated in a reference coordinate system using ξ to express. This $\mathbf{G}$ formulation express the shift between two coordinates. This will be specified more clearly in the Appendix A. $\mathbf{g} \cdot \mathbf{g} = \sum_{i=1}^{3} g_i g_i$ and $g_i = \sum_{j=1}^{3} \frac{\partial \xi_j}{\partial x}$. C_I is a constant, in the code it equals to 144. For the coarse mesh, it is assumed that term $\left(\boldsymbol{w}^h, \frac{\partial \mathbf{u}'}{\partial t}\right)_\Omega$ and term $\left(\nabla^s \boldsymbol{w}^h, 2\mu\nabla^s \mathbf{u}'\right)_\Omega$ are neglected. The last assumption follows from the orthogonality conditions induced by the projector emanating from the bilinear form describing the viscous term. Therefore, the equation becomes,

$$\begin{aligned} B(\overline{\boldsymbol{W}}, \overline{\mathbf{U}} + \mathbf{U}') \approx B_1^h(\overline{\boldsymbol{W}}^h, \overline{\mathbf{U}}^h) + \widetilde{B}_1^h(\overline{\boldsymbol{W}}^h, \mathbf{U}') + B_2^h(\overline{\boldsymbol{W}}^h, \overline{\mathbf{U}}^h, \overline{\mathbf{U}}^h) + \\ B_2^h(\overline{\boldsymbol{W}}^h, \overline{\mathbf{U}}^h, \mathbf{U}') + B_2^h(\overline{\boldsymbol{W}}^h, \mathbf{U}', \overline{\mathbf{U}}^h) + B_2^h(\overline{\boldsymbol{W}}^h, \mathbf{U}', \mathbf{U}') = L(\overline{\boldsymbol{W}}^h). \end{aligned} \tag{3.11}$$

$\widetilde{B}_1^h(\overline{\boldsymbol{W}}^h, \mathbf{U}')$ applied above assumptions. When the equation is expended, it becomes,

$$\begin{aligned} &\left(\boldsymbol{w}^h, \rho\frac{\partial \mathbf{u}^h}{\partial t}\right)_\Omega + \left(\boldsymbol{w}^h, \rho(\mathbf{u}^h - \hat{\mathbf{u}}^h)\cdot\nabla\mathbf{u}^h\right)_\Omega - (\nabla\cdot\boldsymbol{w}^h, p^h)_\Omega + (\nabla^s\boldsymbol{w}^h, 2\mu\nabla^s\mathbf{u}^h)_\Omega + \\ &(q^h, \nabla\cdot\mathbf{u}^h)_\Omega - \left(\nabla\cdot\boldsymbol{w}^h, p'\right)_\Omega + \left(q^h, \nabla\cdot\mathbf{u}'\right)_\Omega + \left(\boldsymbol{w}^h, \rho(\mathbf{u}' - \hat{\mathbf{u}}^h)\cdot\nabla\mathbf{u}^h\right)_\Omega + \\ &\left(\boldsymbol{w}^h, \rho(\mathbf{u}^h - \hat{\mathbf{u}}^h)\cdot\nabla\mathbf{u}'\right)_\Omega + \left(\boldsymbol{w}^h, \rho(\mathbf{u}' - \hat{\mathbf{u}}^h)\cdot\nabla\mathbf{u}'\right)_\Omega = (\boldsymbol{w}^h, \rho\mathbf{g})_\Omega. \end{aligned}$$

Then, put the residual equation into above function, it becomes,

$$\begin{aligned} &\left(\boldsymbol{w}^h, \rho\frac{\partial \mathbf{u}^h}{\partial t}\right)_\Omega + \left(\boldsymbol{w}^h, \rho(\mathbf{u}^h - \hat{\mathbf{u}}^h)\cdot\nabla\mathbf{u}^h\right)_\Omega - (\nabla\cdot\boldsymbol{w}^h, p^h)_\Omega + (\nabla^s\boldsymbol{w}^h, 2\mu\nabla^s\mathbf{u}^h)_\Omega \\ &+ (q^h, \nabla\cdot\mathbf{u}^h)_\Omega + (\nabla\cdot\boldsymbol{w}^h, \tau_C r_C(\mathbf{u}^h))_\Omega - (q^h, \nabla\cdot\tau_M \mathbf{r}_M(\mathbf{u}^h, p^h))_\Omega \\ &- \left(\boldsymbol{w}^h, \rho(\tau_M \mathbf{r}_M(\mathbf{u}^h, p^h) + \hat{\mathbf{u}}^h)\cdot\nabla\mathbf{u}^h\right)_\Omega - \left(\boldsymbol{w}^h, \rho(\mathbf{u}^h - \hat{\mathbf{u}}^h)\cdot\nabla\tau_M \mathbf{r}_M(\mathbf{u}^h, p^h)\right)_\Omega \\ &+ \left(\boldsymbol{w}^h, \rho(\tau_M \mathbf{r}_M(\mathbf{u}^h, p^h) + \hat{\mathbf{u}}^h)\cdot\nabla\tau_M \mathbf{r}_M(\mathbf{u}^h, p^h)\right)_\Omega = (\boldsymbol{w}^h, \rho\mathbf{g})_\Omega. \end{aligned}$$

Now, we are quite close to our final discrete form of governing equation. The level set function in a VMS form will be introduced in the next section. And also, when the above equation is applied, there appears wiggles in the results in some sharp layers. Then some other artificial terms should be added into the equation to wipe the wiggles from the solution. In section 3.6, discontinuity capturing terms are introduced.

[2] One more thing, there is a new technology called IGA, which is developed several years ago. It has many kinds and NURBS is frequently used in CAE. It is quite similar with FEM, however, it can save a lot of time from CAD to CAE with respect to refinement. The basic and simple derivation is also given in Appendix A

3.2 LEVEL SET IN VMS FORM

For level set function, there is no diffusion term in the equation. It is only a convection equation, therefore, its VMS form is the same as SUPG form. However, in Navier-Stokes equations, the VMS form is not the same as its SUPG form. VMS form is "bigger" than the SUPG form normally. Reader should notice this.

$$\frac{\partial \phi}{\partial t} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla \phi = 0. \tag{3.12}$$

Equation 3.12 is known from Chapter 2. After applying VMS, it becomes,

$$\left(\eta^h, \frac{\partial \phi^h}{\partial t} + (\mathbf{u}^h - \hat{\mathbf{u}}^h)\nabla \phi^h\right)_\Omega + \left(\tau_\phi (\mathbf{u}^h - \hat{\mathbf{u}}^h)\nabla \eta^h, \frac{\partial \phi^h}{\partial \tilde{t}} + (\mathbf{u}^h - \hat{\mathbf{u}}^h)\nabla \phi^h\right)_\Omega . \tag{3.13}$$

For this part readers can consult [4]. The second term on the left-hand side results from the VMS method (which is here an SUPG method). The VMS formulation parameter $\boldsymbol{\tau}_\phi$ in Equation 3.13 is defined as,

$$\tau_\phi = \left(\frac{4}{\triangle \tilde{t}^2} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \mathbf{G}(\mathbf{u} - \hat{\mathbf{u}})\right)^{-1/2} .$$

After discretizing the level set function, the majority job of our spacial discretization is done. However, we still can not use above equations into our computer code since other problems will occur. In the next section, we will modify the original Heaviside function to adapt our free surface problem by creating a regularized Heaviside function. Our level set factor ϕ will be used to separate the domain smoothly.

3.3 REGULARIZED HEAVISIDE FUNCTION

In Chapter 2, the density and viscosity are given by:

$$\begin{aligned} \rho &= \rho_w H(\phi) + \rho_w (1 - H(\phi)), \\ \mu &= \mu_a H(\phi) + \mu_a (1 - H(\phi)), \end{aligned} \tag{3.14}$$

We now approximate Equation 3.14 by regularizing the Heaviside function.

$$\begin{aligned} \rho &= \rho_w H_\varepsilon(\phi) + \rho_w (1 - H_\varepsilon(\phi)), \\ \mu &= \mu_a H_\varepsilon(\phi) + \mu_a (1 - H_\varepsilon(\phi)). \end{aligned} \tag{3.15}$$

Here, H_ε is the regularized Heaviside function. The reason for using such a function is to change the density smoothly near the surface. ε is a small value depending on the mesh size. Figure 3.1 shows the detail with a random $\varepsilon = 0.5$ and the function shows below,

$$\begin{aligned} H_\varepsilon(\phi) &= \left\{ \begin{array}{ll} 0 & \text{if} \phi \leqslant -\varepsilon \\ \frac{1}{2}(1 + \frac{\phi}{\varepsilon} + \frac{1}{\pi}\sin(\frac{\phi\pi}{\varepsilon})) & \text{if} |\phi| < \varepsilon \\ 1 & \text{if} \phi \geqslant \varepsilon \end{array} \right\}, \\ \varepsilon &= \alpha \left(\frac{\nabla \phi^h}{||\nabla \phi^h||} \cdot \mathbf{G} \frac{\nabla \phi^h}{||\nabla \phi^h||}\right)^{-1/2} . \end{aligned} \tag{3.16}$$

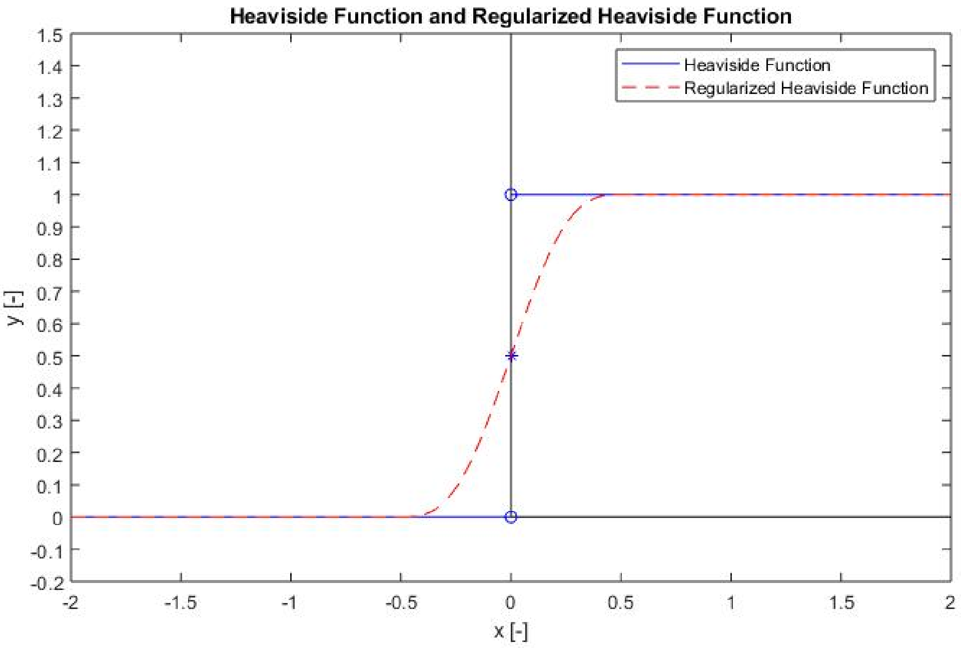


Figure 3.1: Heaviside function plot and regularized Heaviside function plot

The bold character **G** is introduced before as a sencond order rank metric tensor. From above equation it is easy to find that ε is inversely proportional to the magnitude of the gradient of level set.

3.4 RE-DISTANCING

In this section, a new numerical technology will be introduced to overcome the shortcoming associated with the direct application of the above discrete procedures, namely, excessive interface distortion. In order to maintain long-time integration accuracy and overall stability of the formulation, it needs to insist that the level set function satisfies the signed-distance property, which controls the width of the interface layer. To prevent excessive motions of the air-water interface during the re-distancing procedure, it is recommended to add a penalty term, which will be introduced in subsection 3.4.2.

3.4.1 *Re-distancing Method*

For this part, reader can consult paper [1], [2] and [16]. To ensure the level set function satisfies the signed-distance property, it is needed to make use of the

Eikonal partial differential equation[3] in the air and the water sub-domains subject to zero boundary conditions at the air-water interface, that is,

$$\begin{aligned} ||\nabla\phi_d|| &= 1 \text{ in } \Omega_a, \\ ||\nabla\phi_d|| &= 1 \text{ in } \Omega_w, \\ \phi_d &= 0 \text{ on } \Gamma. \end{aligned}$$

In the water domain, the distance has a positive value, while in the air domain the distance has a negative value. In another word, if $||\nabla\phi|| = 1$, then it means it is in the fluid domain, $\nabla\phi_d = \phi$, nothing will be changed. If $||\nabla\phi|| \neq 1$, which means it is on the free surface, then $\nabla\phi_d \neq \phi$. Before discretizing in space, we need to introduce a "pseudo-time" variable $\tilde{t}$ to re-state the above problem and find ϕ_d:

$$\begin{aligned} &\frac{\partial\phi_d}{\partial\tilde{t}} + sign(\phi)(||\nabla\phi_d|| - 1) = 0, \\ &\phi_d(x, 0) = \phi(x, t). \end{aligned} \tag{3.17}$$

Whereas,

$$sign(\phi) = S_\varepsilon(\phi_d) = 2H_\varepsilon(\phi_d) - 1. \tag{3.18}$$

From above equation, readers can find our sign function is built by Heaviside function. It is just some small modifications applied on Heaviside function.

When it is written in a SUPG form, it becomes,

$$\left(\eta_d^h, \frac{\partial\phi_d^h}{\partial\tilde{t}} + S_\varepsilon(\phi_d^h)(||\nabla\phi_d^h|| - 1)\right)_\Omega + \left(\tau_{\phi_d} \mathbf{a}\cdot\nabla\eta_d^h, \frac{\partial\phi_d^h}{\partial\tilde{t}} + S_\varepsilon(\phi_d^h)(||\nabla\phi_d^h|| - 1)\right)_\Omega . \tag{3.19}$$

This is semi-discrete, which means only spatial discrete and no time discrete. It is assumed time derivative of ϕ is zero. Then its residual and τ_ϕ will becomes,

$$\tau_{\phi_d} = \left(\frac{4}{\triangle\tilde{t}^2} + \mathbf{a}\cdot\mathbf{G}\mathbf{a}\right)^{-1/2} .$$

In the above equation, $\mathbf{a}$ is expressed by the following equation:

$$\mathbf{a} = S_\varepsilon(\phi)\frac{\nabla\phi_d^h}{||\nabla\phi_d^h||}.$$

Now the discretized governing equation becomes,

$$\begin{aligned} &\left(\boldsymbol{w}^h, \rho\frac{\partial\mathbf{u}^h}{\partial t}\right)_\Omega + \left(\boldsymbol{w}^h, \rho(\mathbf{u}^h - \hat{\mathbf{u}}^h)\cdot\nabla\mathbf{u}^h\right)_\Omega - (\nabla\cdot\boldsymbol{w}^h, p^h)_\Omega + (\nabla^s\boldsymbol{w}^h, 2\mu\nabla^s\mathbf{u}^h)_\Omega + \\ &(q^h, \nabla\cdot\mathbf{u}^h)_\Omega - \left(\nabla\cdot\boldsymbol{w}^h, p^{'}\right)_\Omega + \left(q^h, \nabla\cdot\mathbf{u}^{'}\right)_\Omega + \left(\boldsymbol{w}^h, \rho(\mathbf{u}^{'} - \hat{\mathbf{u}}^h)\cdot\nabla\mathbf{u}^h\right)_\Omega + \\ &\left(\boldsymbol{w}^h, \rho(\mathbf{u}^h - \hat{\mathbf{u}}^h)\cdot\nabla\mathbf{u}^{'}\right)_\Omega + \left(\boldsymbol{w}^h, \rho(\mathbf{u}^{'} - \hat{\mathbf{u}}^h)\cdot\nabla\mathbf{u}^{'}\right)_\Omega + \left(\eta_d^h, \frac{\partial\phi_d^h}{\partial\tilde{t}} + S_\varepsilon(\phi_d^h)(||\nabla\phi_d^h|| - 1)\right)_\Omega \\ &+ \left(\tau_{\phi_d}\mathbf{a}\cdot\nabla\eta_d^h, \frac{\partial\phi_d^h}{\partial\tilde{t}} + S_\varepsilon(\phi_d^h)(||\nabla\phi_d^h|| - 1)\right)_\Omega = (\boldsymbol{w}^h, \rho\mathbf{g})_\Omega . \end{aligned}$$

[3] The eikonal equation is $||\nabla\phi_d|| = 1$, a non-linear partial differential equation subject to the interior constraint given by $\phi_d = 0$. For more information, readers can consult paper [24].

This re-distancing approach is first proposed in [20] to solve incompressible two phase flow problem. Then this technology is used in many applications of simulation. After the introduction of re-distancing method, another term call penalty will be added to maintain the interface in the next section.

3.4.2 *Penalty Term*

After applying re-distancing method, there is till one term, namely penalty term, should be added in the equation to prevent excessive motions of the air-water interface. This penalty term formula, in paper [2] and also the same in the code, shows below,

$$+(\eta_d^h, \lambda_{pen} H_\varepsilon^{'}(\phi^h)(\phi_d^h - \phi^h))_\Omega. \tag{3.20}$$

In Equation 3.20, λ_{pen} is a constant which will be changed in our parameter file. $H_\varepsilon^{'}(\phi^h)$ is our regularized Heaviside function scales as $1/h$ in the interface layer and it equals to zero outside of the interface. Hence the term is only active where it is necessary.

Then, finally our governing equations become:

$$\begin{aligned}
&\left(\boldsymbol{w}^h, \rho\frac{\partial \mathbf{u}^h}{\partial t}\right)_\Omega + \left(\boldsymbol{w}^h, \rho(\mathbf{u}^h - \hat{\mathbf{u}}^h)\cdot\nabla\mathbf{u}^h\right)_\Omega - (\nabla\cdot\boldsymbol{w}^h, p^h)_\Omega + (\nabla^s\boldsymbol{w}^h, 2\mu\nabla^s\mathbf{u}^h)_\Omega \\
&+ (q^h, \nabla\cdot\mathbf{u}^h)_\Omega + (\nabla\cdot\boldsymbol{w}^h, \tau_C r_C(\mathbf{u}^h))_\Omega - (q^h, \nabla\cdot\tau_M\mathbf{r}_M(\mathbf{u}^h, p^h))_\Omega \\
&- \left(\boldsymbol{w}^h, \rho(\tau_M\mathbf{r}_M(\mathbf{u}^h, p^h) + \hat{\mathbf{u}}^h)\cdot\nabla\mathbf{u}^h\right)_\Omega - \left(\boldsymbol{w}^h, \rho(\mathbf{u}^h - \hat{\mathbf{u}}^h)\cdot\nabla\tau_M\mathbf{r}_M(\mathbf{u}^h, p^h)\right)_\Omega \\
&+ \left(\boldsymbol{w}^h, \rho(\tau_M\mathbf{r}_M(\mathbf{u}^h, p^h) + \hat{\mathbf{u}}^h)\cdot\nabla\tau_M\mathbf{r}_M(\mathbf{u}^h, p^h)\right)_\Omega + \left(\eta_d^h, \frac{\partial\phi_d^h}{\partial\tilde{t}} + S_\varepsilon(\phi_d^h)(\|\nabla\phi_d^h\| - 1)\right)_\Omega \\
&+ \left(\tau_{\phi_d}\mathbf{a}\cdot\nabla\eta_d^h, \frac{\partial\phi_d^h}{\partial\tilde{t}} + S_\varepsilon(\phi_d^h)(\|\nabla\phi_d^h\| - 1)\right)_\Omega + (\eta_d^h, \lambda_{pen}H_\varepsilon^{'}(\phi^h)(\phi_d^h - \phi^h))_\Omega \\
&= (\boldsymbol{w}^h, \rho\mathbf{g})_\Omega\,.
\end{aligned}$$

This re-distancing method somehow will cause another problem, the unbalance of mass conservation equation. Although the deficit of mass is quite small, after many iterations it can not be ignored. And if the mesh is too coarse, the mass loss will be a serious problem. Therefore, a correction of mass should be considered which will be introduced in the next section.

3.5 MASS BALANCE

There are mainly two reasons to cause the mass loss. One reason is may be caused by both convection and re-distancing of the level set. The amount of mass loss depends on many factors. If the mesh is too coarse, then there will be large amount of mass loss. The other reason is that when the discrete governing equations are integrated for a long time period. Minor mass loss for each time step will accumulated for a large mass error. Therefore, a mass correction procedure is necessary. The global mass conservation for a moving domain is:

$$\frac{d}{dt}\int_{\Omega_t} \rho d\Omega + \int_{\Gamma_t} \rho(\mathbf{u} - \hat{\mathbf{u}}) \cdot \mathbf{n} d\Gamma = 0. \tag{3.21}$$

For the boundary integral approximation, the midpoint rule is used rule which yields:

$$\int_{\Omega_{n+1}} \rho_{n+1} d\Omega - \int_{\Omega_n} \rho_n d\Omega + \triangle t_n \int_{\Gamma_{n+1/2}} \rho_{n+1/2}(\mathbf{u}_{n+1/2} - \hat{\mathbf{u}}_{n+1/2}) \cdot \mathbf{n}_{n+1/2} d\Gamma = 0. \tag{3.22}$$

This procedure ensures the global mass balance at every time step and for more information readers can consult [2]. After this section, the governing equations should be applicable for numerical simulation. However, the wiggles appear along sharp layers which makes the solution not perfect. In the next section, we will find a way to fix the problem by using discontinuity capturing terms.

3.6 DISCONTINUITY CAPTURING

The wiggles in the solution of a simulation are found many years ago with FVM especially in convection dominated problem. People usually use the Upwind scheme with adding an artificial diffusion term to solve such a problem. However, there is no "Upwind scheme" in FEM until streamline upwind/Petrov-Galerkin (SUPG) method is developed by Brooks and Hughes [4]. This method gives robust results of a classical Upwind method. "Unfortunately, the SUPG method does not preclude spurious oscillations localized in narrow regions along sharp layers and hence various terms introducing artificial crosswind diffusion in the neighbourhood of layers have been proposed to be added to the SUPG formulation." quoted in [15]. In this paper, it introduced different kinds of discontinuity capturing methods. In this master thesis, the RBVMS method also uses the SUPG method. Then in the same case, two additional discontinuity-capturing terms, in the air and the water separately, should be added into our governing equations to prevent the appearance of the wiggles:

$$\begin{aligned} &+ \int_{\Omega} \nabla \boldsymbol{w}^h \kappa_{ns} \nabla^s \mathbf{u}^h d\Omega, \\ &+ \int_{\Omega} \nabla \eta^h \kappa_{ls} \nabla \phi^h d\Omega. \end{aligned} \tag{3.23}$$

where κ_{ns} and κ_{ls} are tensor-valued, residual-based, discontinuity-capturing viscosities. The derivation of these two terms can be found in paper [6], which gives another way to derive these discontinuity capturing terms.

More specifically,

$$\begin{aligned}\kappa_{ns} &= C_{ns}\frac{\|\mathbf{r}_M(\mathbf{u}^h,p^h)\|}{u_{ref}\sqrt{\mathbf{G}:\mathbf{G}}}\mathbf{I},\\ \kappa_{ls} &= C_{ls}\frac{\|\frac{\partial\phi}{\partial t}+(\mathbf{u}^h-\hat{\mathbf{u}}^h)\cdot\nabla\phi^h\|}{\sqrt{\nabla\phi^h\cdot\mathbf{G}\nabla\phi^h}}\mathbf{I}.\end{aligned} \tag{3.24}$$

However, in the code, there is a small difference. There is no u_{ref} in κ_{ns} and an extremely small number is added in κ_{ns} denominator to avoid NaN,

$$\begin{aligned}\kappa_{ns} &= C_{ns}\frac{\|\mathbf{r}_M(\mathbf{u}^h,p^h)\|}{\sqrt{\mathbf{G}:\mathbf{G}}}\mathbf{I},\\ \kappa_{ls} &= C_{ls}\frac{\|\frac{\partial\phi}{\partial t}+(\mathbf{u}^h-\hat{\mathbf{u}}^h)\cdot\nabla\phi^h\|}{\sqrt{\nabla\phi^h\cdot\mathbf{G}\nabla\phi^h+1\times10^{-15}}}\mathbf{I}.\end{aligned} \tag{3.25}$$

Whereas,

$$C_{ns} = H_\varepsilon(\phi)C_{ns\_w} + (1-H_\varepsilon(\phi))C_{ns\_a}.$$

C_{ns_w} and C_{ns_a} are the parameters in our input file which represent the air and water constant separately. In the level set function, after applying SUPG, a discontinuity capturing term is also added. In the code it shows that its discontinuity capturing term looks like below,

$$+\int_\Omega \nabla\eta_d^h\kappa_{LSRD}\nabla\phi_d^h d\Omega$$

$$\kappa_{LSRD} = C_{LSRD}\frac{|\mathrm{sign}(\phi)(\|\nabla\phi_d\|-1)|}{\sqrt{\nabla\phi\cdot\mathbf{G}\nabla\phi+1\times10^{-15}}}.$$

These constants above, like C_{ns_w} and C_{ns_a} and C_{LSRD}, are the parameters in our parameter input file. Their value will be studied in Chapter 5.

3.7 SHIP MOTION EQUATION

Firstly, let's define Ω_0 as the reference configuration. Let Ω_t denote its current configuration. Let $\mathbf{Z}$ denote the reference configuration coordinates of the rigid body and let $\mathbf{z}$ denote the current configuration coordinates of the rigid body. $\mathbf{d}$ is the displacement. The rigid body kinematics is composed of translation and rotational motions and can be summarized as,

$$\mathbf{z} = \mathbf{Q}(\mathbf{Z}-\mathbf{Z}_0)+\mathbf{Z}_0+\mathbf{d}_0. \tag{3.26}$$

For more reference and how to get $\mathbf{Q}$ reader can consult paper [13]. The displacement can be written as,

$$\mathbf{d} = \mathbf{z}-\mathbf{Z} = \mathbf{Q}(\mathbf{Z}-\mathbf{Z}_0)+\mathbf{Z}_0-\mathbf{Z}+\mathbf{d}_0 = (\mathbf{Q}-\mathbf{I})(\mathbf{Z}-\mathbf{Z}_0)+\mathbf{d}_0. \tag{3.27}$$

Then its velocity can be written as,

$$\dot{\mathbf{d}} = \dot{\mathbf{Q}}(\mathbf{Z}-\mathbf{Z}_0)+\dot{\mathbf{d}}_0. \tag{3.28}$$

Let's take the first equation into this equation, then it becomes,

$$\dot{\mathbf{d}} = \dot{\mathbf{Q}}\left(\mathbf{Q}^{-1}(z - \mathbf{Z}_0 - \mathbf{d}_0)\right) + \dot{\mathbf{d}}_0. \tag{3.29}$$

Here another symbol z_0 will be used to substitute $\mathbf{Z}_0 + \mathbf{d}_0$, therefore,

$$\begin{aligned} &\dot{\mathbf{d}} = \dot{\mathbf{Q}}\mathbf{Q}^{-1}(z - z_0) + \dot{\mathbf{d}}_0 = \Omega(z - z_0) + \dot{\mathbf{d}}_0, \\ &\Omega = \dot{\mathbf{Q}}\mathbf{Q}^{-1} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \end{aligned} \tag{3.30}$$

Therefore, $\dot{\mathbf{Q}} = \Omega\mathbf{Q}$. Then a vector $\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]^T$ can be used to express the velocity,

$$\dot{\mathbf{d}} = \boldsymbol{\omega} \times (z - z_0) + \dot{\mathbf{d}}_0. \tag{3.31}$$

These variables represent the translation velocity and rotational velocity of the ship, which means the 6 DOF (degree of freedom) of the ship. The dynamics of the rigid body is governed by the equations of balance of global linear and angular momentum, they are,

$$\begin{aligned} &\frac{d}{dt}\int_{\Omega} \rho\dot{\mathbf{d}} d\Omega = F, \\ &\frac{d}{dt}\int_{\Omega} (z - z_0) \times \rho\dot{\mathbf{d}} d\Omega = M. \end{aligned} \tag{3.32}$$

In Equation 3.32, F and M are the global force and moment vectors acting on a rigid body. They can be expressed by:

$$\begin{aligned} &F = mg + \int_{\Gamma i} \mathbf{h} d\Gamma \\ &M = \int_{\Gamma i} (z - z_0) \times \mathbf{h} d\Gamma \end{aligned} \tag{3.33}$$

Here, m is the mass of object and $\mathbf{h}$ is the traction vector. In our problem, $\mathbf{h}$ is the fluid traction. Introducing the rigid body kinematics into the linear and angular momentum balance equations, we obtain the following system of six ordinary differential equations,

$$\begin{aligned} &\frac{d}{dt}(m\dot{\mathbf{d}}_0) = F \\ &\frac{d}{dt}(J_t\omega) = M \end{aligned} \tag{3.34}$$

Whereas,

$$\begin{aligned} &J_t = QJ_0Q^T \\ &J_0 = \int_{\Omega_0} \rho(\mathbf{Z} - \mathbf{Z}_0) \cdot (\mathbf{Z} - \mathbf{Z}_0)\mathbf{I} d\Omega - \int_{\Omega_0} \rho(\mathbf{Z} - \mathbf{Z}_0) \otimes (\mathbf{Z} - \mathbf{Z}_0)\mathbf{I} d\Omega \end{aligned} \tag{3.35}$$

After introducing the ship motion equations, the whole process is clear now. Solving the governing equations is the first step, then the output results of velocity and pressure are used to calculate the force and momentum exerted on the ship. Based on these outside force and momentum, the displacement of mesh for each node is known. Using the new mesh information, another iteration step can be simulated. However, these formulations will not be illustrated in this thesis, reader can find it in [2].

Finally, the spacial discretization of the governing equations is finished. In the next chapter, time integration method will be introduced. The Generalized-α, a implicit and explicit mixed method is chosen.

4 TIME INTEGRATION OF THE FREE-SURFACE FLOW

After spatial discretization, the time discretization method should be introduced next. Here, the Generalized-α method is applied in this master thesis. In this chapter there are 2 sections in total. In section 4.1 the generalized-α method is discussed. The detail of derivation will be written in two stages. In section 4.2 the time integration method of re-distancing method will be introduced.

In CFD simulation, there are two categories of methods to do time discretization, namely explicit methods and implicit methods. An explicit solution results from a method that is independent of other values (for the same level), a single equation is used to evaluate new nodal variables for a single time step. An implicit solution contains information obtained from solving simultaneous equations for the full grid for each time step. This is computationally more demanding but allows for larger time steps and better stability. Generalized-α method is the method which combines these two methods.

4.1 GENERALIZED-α METHOD

Generalized-α method is unconditionally stable. At each time step, a method called Newton iteration is used to solve the nonlinear algebraic-equation. And at each Newton iteration step, another technology called GMRES is used for solving the linear system. In Appendix B and Appendix C, the reader can find two methods GMRES and Newton Iteration separately. For many years, it is really hard to solve nonlinear algebraic-equation. Only Newton iteration can solve such problems. With the creation of GMRES, it saves a lot of time to solve non-linear algebraic-equation when using Newton Iteration. Therefore, these two methods are introduced in Appendix B and Appendix C. Now come back to our governing equation, it is needed to use V_n and $\dot{V}_n$ P_n to calculate V_{n+1},$\dot{V}_{n+1}$,$V_{n+\alpha_f}$,$\dot{V}_{n+\alpha_m}$,P_{n+1}. V_n is the same with vector $\mathbf{u}$ and n denotes the time step. Although $\dot{V}_n$ is the time derivative of V_n, it is still be treated as a unknown variable. At the beginning, the following five equations are known,

$$\begin{aligned}
&\mathbf{R}^M\left(\dot{V}_{n+\alpha_m}, V_{n+\alpha_f}, P_{n+1}\right) = 0,\\
&\mathbf{R}^C\left(\dot{V}_{n+\alpha_m}, V_{n+\alpha_f}, P_{n+1}\right) = 0,\\
&V_{n+1} = V_n + \triangle t\dot{V}_n + \gamma \triangle t(\dot{V}_{n+1} - \dot{V}_n),\\
&\dot{V}_{n+\alpha_m} = \dot{V}_n + \alpha_m(\dot{V}_{n+1} - \dot{V}_n),\\
&V_{n+\alpha_f} = V_n + \alpha_f(V_{n+1} - V_n).
\end{aligned} \tag{4.1}$$

To get second order accuracy and stability, Equation 4.2 condition should be satisfied. To find the reason of this reader can consult paper [5] and [23],

$$\gamma = \frac{1}{2} + \alpha_m - \alpha_f. \tag{4.2}$$

While unconditional stability is attained if ,

$$\alpha_m \geqslant \alpha_f \geqslant \frac{1}{2}. \tag{4.3}$$

Whereas,

$$\alpha_m = \frac{1}{2}\left(\frac{3-\rho_\infty}{1+\rho_\infty}\right) \quad \text{and} \quad \alpha_f = \frac{1}{1+\rho_\infty}. \tag{4.4}$$

In our simulation, the ρ_∞ is equal to 0.5 and it is the spectral radius of the amplification matrix as $\triangle t \to \infty$. Therefore, α_m is 0.83 and α_f is 0.67. And there are two stages to do the time integration.

4.1.1 *Predictor Stage*

In this stage, some initial value will be decided through the following equations:

$$\begin{aligned} V_{n+1,(0)} &= V_n \\ \dot{V}_{n+1,(0)} &= \frac{\gamma-1}{\gamma}\dot{V}_n \\ P_{n+1,(0)} &= P_n \end{aligned} \tag{4.5}$$

Here 0 indicates the iteration number. With this initial guess, the calculation can be started.

4.1.2 *Multi-corrector Stage*

1) Calculate $V_{n+\alpha_f}$,$\dot{V}_{n+\alpha_m}$,P_{n+1},

$$\begin{aligned} \dot{V}_{n+\alpha_m,(l)} &= \dot{V}_n + \alpha_m(\dot{V}_{n+1,(l-1)} - \dot{V}_n), \\ V_{n+\alpha_f,(l)} &= V_n + \alpha_f(V_{n+1,(l-1)} - V_n), \\ P_{n+1,(l)} &= P_n + \alpha_f(P_{n+1,(l-1)} - P_n). \end{aligned} \tag{4.6}$$

2) After the first step, the following equations should be used,

$$\begin{aligned} \mathbf{R}^M\left(\dot{V}_{n+\alpha_m}, V_{n+\alpha_f}, P_{n+\alpha_f}\right) &= 0, \\ \mathbf{R}^C\left(\dot{V}_{n+\alpha_m}, V_{n+\alpha_f}, P_{n+\alpha_f}\right) &= 0. \end{aligned} \tag{4.7}$$

To get $\triangle\dot{V}_{n+1,(l)})$ and $\triangle P_{n+1,(l)})$ Newton's iteration should be used now. It is a very famous method to solve non-linear equation. Here we will apply Newton iteration method to Equation 4.1. For the process, the reader can consult Appendix C.

Then the above equation will be written as following,

$$\begin{aligned} \frac{\partial R^{mom}}{\partial \dot{u}_{n+1}} \triangle \dot{u} + \frac{\partial R^{mom}}{\partial p_{n+1}} \triangle p = -R^{mom}_{(l)}, \\ \frac{\partial R^{con}}{\partial \dot{u}_{n+1}} \triangle \dot{u} + \frac{\partial R^{con}}{\partial p_{n+1}} \triangle p = -R^{con}_{(l)}. \end{aligned} \tag{4.8}$$

If there is level set function, the above function will be changed as,

$$\begin{aligned}
&\frac{\partial R^{mom}}{\partial \dot{u}_{n+1}} \triangle \dot{u} + \frac{\partial R^{mom}}{\partial p_{n+1}} \triangle p + \frac{\partial R^{mom}}{\partial \dot{\phi}_{n+1}} \triangle \dot{\phi} = -R^{mom}_{(l)},\\
&\frac{\partial R^{con}}{\partial \dot{u}_{n+1}} \triangle \dot{u} + \frac{\partial R^{con}}{\partial p_{n+1}} \triangle p + \frac{\partial R^{con}}{\partial \dot{\phi}_{n+1}} \triangle \dot{\phi} = -R^{con}_{(l)},\\
&\frac{\partial R^{ls}}{\partial \dot{u}_{n+1}} \triangle \dot{u} + \frac{\partial R^{ls}}{\partial \dot{\phi}_{n+1}} \triangle \dot{\phi} = -R^{ls}_{(l)}.
\end{aligned} \tag{4.9}$$

Here, we only give the steps of time discretization with the absence of level set function. Our governing equations can be written as below,

$$\begin{aligned}
K_{(l)} \triangle \dot{V}_{n+1} + G_{(l)} \triangle P_{n+1} &= -R^{M}_{(l)},\\
D_{(l)} \triangle \dot{V}_{n+1} + L_{(l)} \triangle P_{n+1} &= -R^{C}_{(l)}.
\end{aligned} \tag{4.10}$$

In above equations, the matrices $K_{(l)}, G_{(l)}, D_{(l)}$ and $L_{(l)}$ are shown below.

$$\begin{aligned}
K_{(l)} &= \frac{\partial \mathbf{R}^{M}\left(\dot{V}_{n+\alpha_m}, V_{n+\alpha_f}, P_{n+\alpha_f}\right)}{\partial \dot{V}_{n+\alpha_m}} \cdot \frac{\partial \dot{V}_{n+\alpha_m}}{\partial \dot{V}_{n+1}} + \frac{\partial \mathbf{R}^{M}\left(\dot{V}_{n+\alpha_m}, V_{n+\alpha_f}, P_{n+\alpha_f}\right)}{\partial V_{n+\alpha_f}} \cdot \frac{\partial V_{n+\alpha_f}}{\partial \dot{V}_{n+1}},\\
G_{(l)} &= \frac{\partial \mathbf{R}^{M}\left(\dot{V}_{n+\alpha_m}, V_{n+\alpha_f}, P_{n+\alpha_f}\right)}{\partial P_{n+\alpha_f}} \cdot \frac{\partial P_{n+\alpha_f}}{\partial \dot{P}_{n+1}},\\
D_{(l)} &= \frac{\partial \mathbf{R}^{C}\left(\dot{V}_{n+\alpha_m}, V_{n+\alpha_f}, P_{n+\alpha_f}\right)}{\partial \dot{V}_{n+\alpha_m}} \cdot \frac{\partial \dot{V}_{n+\alpha_m}}{\partial \dot{V}_{n+1}} + \frac{\partial \mathbf{R}^{C}\left(\dot{V}_{n+\alpha_m}, V_{n+\alpha_f}, P_{n+\alpha_f}\right)}{\partial V_{n+\alpha_f}} \cdot \frac{\partial V_{n+\alpha_f}}{\partial \dot{V}_{n+1}},\\
L_{(l)} &= \frac{\partial \mathbf{R}^{C}\left(\dot{V}_{n+\alpha_m}, V_{n+\alpha_f}, P_{n+\alpha_f}\right)}{\partial P_{n+\alpha_f}} \cdot \frac{\partial P_{n+\alpha_f}}{\partial \dot{P}_{n+1}}.
\end{aligned} \tag{4.11}$$

Update the fluid unknown variables:

$$\begin{aligned}
\dot{u}_{n+1,(l)} &= \dot{u}_{n+1,(l)} + \triangle\dot{u},\\
u_{n+1,(l)} &= u_{n+1,(l)} + \gamma \triangle t \triangle \dot{u},\\
\dot{p}_{n+1,(l)} &= \dot{p}_{n+1,(l)} + \triangle\dot{p},\\
p_{n+1,(l)} &= p_{n+1,(l)} + \gamma \triangle t \triangle \dot{p},\\
\dot{\phi}_{n+1,(l)} &= \dot{\phi}_{n+1,(l)} + \triangle\dot{\phi},\\
\phi_{n+1,(l)} &= \phi_{n+1,(l)} + \gamma \triangle t \triangle \dot{\phi}.
\end{aligned} \tag{4.12}$$

Since the ship is moving in head seas, the mesh should also move with the ship. Therefore, there will be more equations to solve the mesh dynamics equations. And these equations should be similar as above derivation. Hence, for the rigid body and mesh movement, it will not be introduced here. If the reader is interested please read [2].

4.2 RE-DISTANCING TIME INTEGRATION

The formulation of the pseudo time in our code is given here in a bilinear form:

$$\triangle\tilde{t} = \left(\eta^h_d, \frac{\partial \phi^h_d}{\partial \tilde{t}}\right)_{\Omega}. \tag{4.13}$$

This $\frac{\partial \phi_d^h}{\partial \tilde{t}}$ is the input parameter which is set in our input file. The pseudo-time step is increased with a decreasing nonlinear residual of the re-distancing equations. And because of the penalty terms present, we are able to take large pseudo-time steps without loosing the interface position predicted by the convection step. The level set is then re-distanced in pseudo-time using the backward Euler method. The level set is re-distanced at every physical time step using a small fixed number of pseudo-time steps. The choice of the pseudo-time step can also be based on the convective CFL for the Eikonal equation, and is computed as,

$$\triangle \tilde{t} = 2\gamma \min \left(S(\phi) \frac{\nabla \phi^h}{||\nabla \phi^h||} \cdot \mathbf{G} S(\phi) \frac{\nabla \phi^h}{||\nabla \phi^h||} \right)^{-1/2} . \tag{4.14}$$

The parameter $\gamma = 1$ corresponds to the CFL of unity. In the computations, usually it is computed as $\gamma = 6$. Based on this equation, the pseudo time step can be decided.

Here, the first part of this master thesis is finished. In the next part, the real numerical simulation is introduced. The simulation results will be compared with a paper which is published in International Shipbuilding Progress journal. The experiment is taken in MARIN, a Dutch institution.

Part II

NUMERICAL RESULTS AND VALIDATION

In this part, two results of simulations will be presented. For the wave simulation, in Chapter 5, it is used to decide the parameter setting which will be used in the ship simulation. Then, Chapter 6 is the main part of this master thesis, the validation of DTMB 5415M heave and pitch motion in head sea. In Chapter 7, an conclusion will be made to end this thesis and some recommendations will be given for the future study.

5 WAVE SIMULATION

Before doing the validation of DTMB 5415M, a simulation of wave will be performed to find out how fine of the mesh should be and meanwhile find the right parameter setting for the following DTMB case. In this regard, we systematically determine the domain size, time step, and input parameters linked to the RBVMS method.

5.1 PROBLEM DESCRIPTION

In this chapter, a single harmonic wave will be simulated in a box shape domain. The wave comes from domain left to domain right which is the same condition in the future ship case. Then compare our results to the airy wave theory with checking the wave amplitude. Since Generalized-α method is converged anyway, the time step selection will be important for saving our simulation time.

5.1.1 *Domain Set*

The original idea is to use the same size of the domain of the DTMB 5415M ship to simulate the wave. However, to save the simulation time, the domain of wave is a little bit different with DTMB 5415M ship. In DTMB 5415M, the domain is box shaped extending from $-0.5L < x < 1.8L, -1.1L < y < 1.1L, -1.0L < z < 0.25L$. Here L is the length of ship, $142m$. In the wave simulation, the width of domain shrinks to $-0.1L < y < 0.1L$. The mesh image shows in Figure 5.1 gives the left and the front mesh image of wave domain in the software Gmsh.

From the mesh image it can be found that the surface area of waves is more dense than the other area. This is because the wave motion will effect only the surface area, as for the deep water, the fluid rarely moves and therefore no turbulence will appear in the bottom of domain. In our case, it is a deep water condition, therefore the grid of the domain bottom can be coarser to save the computational time.

5.1.2 *Wave Condition*

The wave amplitude and some other constants in the code are recorded in Table 5.1. This is the same wave condition in DTMB 5415M case.

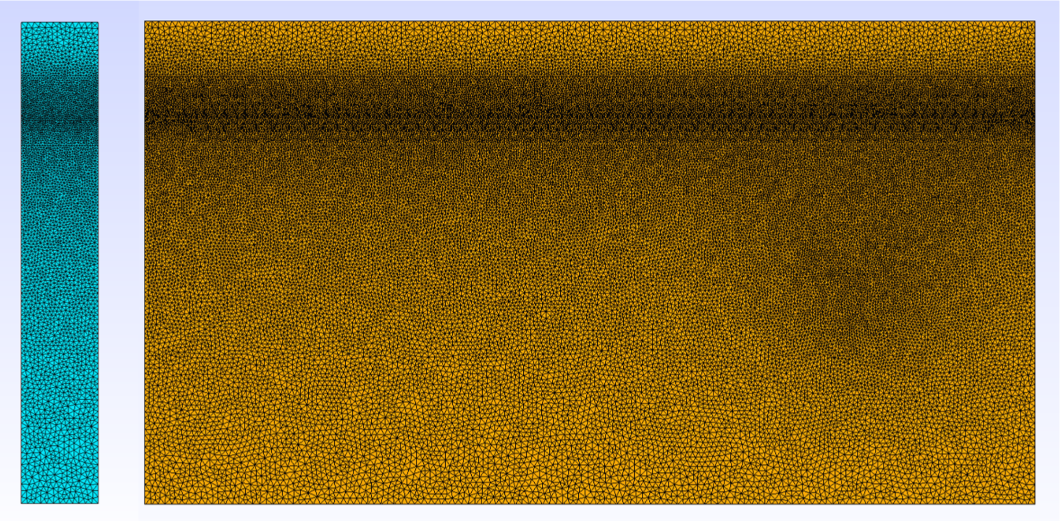


Figure 5.1: The mesh of the wave simulation

Constants in the RBVMS code

Wave amplitude:A	$1.03m$	Water density: ρ_w	$1000kg/m^3$
Wave length:λ	$1.206L$	Air density: ρ_a	$1kg/m^3$
Wave frequency:f	$0.6rad/s$	Water viscosity: μ_w	2e-5$N \cdot s/m$
Wave slope:Ak	0.0378	Air viscosity: μ_a	1e-3$N \cdot s/m$
Gravity acceleration:g	$9.81m/s^2$		

Table 5.1: Constants in the RBVMS code

5.2 BOUNDARY CONDITIONS AND TEST METHOD

The inflow boundary condition is set as Direcret boundary condition with an inflow velocity:

$$u_x = \omega a \frac{\cosh(k(d+z))}{\sinh(kd)} \cos(\omega t - kx) + U_{in}. \tag{5.1}$$

Here, d is the depth of the domain and z is the location of the wave. And ω is the wave frequency and k is the wave number. U_{in} is the ship velocity calculated by the Froude number. The wave x-direction velocity is 0.62 m/s and the ship velocity is 9.26 m/s, which means the ship velocity is faster than the wave. Hence, there is no reflection happened on the right boundary of the domain. This inflow velocity is based on the airy wave theory using the information of our wave condition. No traction boundary condition is set for out flow. As for the back and forth, top and bottom, no slip boundary conditions are set. The bottom of fluid is not moved therefore $u = 0$. For back and forth, y direction velocity is the same with domain, for top domain, x direction velocity is the same with domain.

Traction free boundary condition (also known as no traction B.C.)means that the the surface is free from external stress. It can be mathematically expressed as $t = \sigma \cdot n = 0$ *where, t is the surface traction; σ is the Cauchy stress tensor; n is the vector normal to the deformed surface.*

$$\begin{aligned} u_y &= \omega a \frac{\cosh(k(d+z))}{\sinh(kd)} \sin(\omega t - kx), \\ u_z &= \omega a \frac{\sinh(k(d+z))}{\sinh(kd)}. \end{aligned} \tag{5.2}$$

Two single points are picked as test points. The first point is located at $(-66, 0)$, and the other one is located at $(155.6, 0)$. Through their x and y coordinate, its free surface location should be found (where $\phi = 0$). To get the output file, a subroutine is written in the code to plot the wave height and write the information into a file with each time step. To keep the same condition as the validation paper, here 100 seconds is decided as our simulation time length.

We decide to use 0.1 for all discontinuity capturing terms, C_{ns_a}, C_{ns_w}, C_{ls} and C_{LSRD}, as our first try. This is because there is no large sharp layers exist in a wave simulation, and usually the code creator will make these coefficients of kdc in the same order. For example from 0.1 to 1. Therefore, our initial guess is 0.1. As for the penalty term λ_{pen}, 800 is used to maintain the free surface and 0.1 is set for pseudo time. For the CFL number, we try to keep it below 1 at first (0.3 is tried at the beginning), although it will also be stable above 1. The residual tolerance is set to be $1e-3$, reader should notice this residual value is not the absolute value of residual, it means the proportional reduce of initial residual. The author has already tested, it gives the same results with either $1e-3$ or $1e-8$.

One thing should be illustrated clearly again at the beginning. There is a mesh factor called scale will be used in generating the mesh, reader will see this in the following chapter. This scale factor does not mean the size of the geometry. It decides how fine the mesh is and how many element nodes of each mesh are created. The smaller of this scale factor, the finer of the mesh will be.

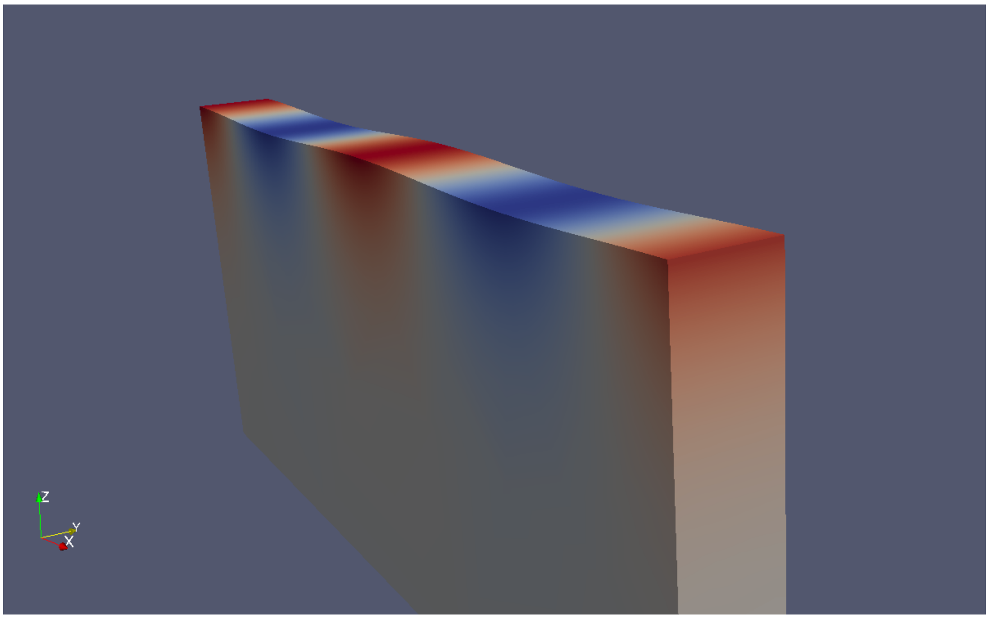


Figure 5.2: Wave velocity simulation result

5.3 RESULTS AND OPTIMIZATION

To quantify how fine the mesh is, a factor will be used to express this as we introduced. For example, wave 0.1 is much finer than wave 0.2. Wave 0.1 has 585474 element nodes and wave 0.2 has 83055 element nodes, which means nearly seven times larger. If wave 0.4 is used, then the same order of element nodes should be decreased and in our simulation wave 0.4 has 13255 element nodes.

5.3.1 *Wave Results*

Figure 5.2 gives a snapshot of our wave velocity simulation results. Three different meshes are tested in the simulation of wave. The numbers of nodes for each mesh are 13255, 83055 and 585474, and the factor changes from 0.4 to 0.2 to 0.1. For simplification, we can define them as course mesh, middle mesh and fine mesh. Their results are shown in the first plot of Figure 5.3. With the finer of the mesh, the wave amplitude will be closer to the set value $1.03m$. The test point in the first figure is $(155.6, 0)$. In the second plot of Figure 5.3, it shows the amplitude of the inflow point and out flow point in the very fine mesh. After traveling $221.6m$, the amplitude of this wave does not decrease in this simulation, which is nicely coincident with Airy wave theory.

For the fine mesh, comparing to the default amplitude value $1.03m$, the amplitude lose less than 5% and this error is acceptable. For wave 0.2, the amplitude error is about 8%. From this result, it is assured that wave simulation with factor 0.2 is acceptable but factor 0.1 is the best. And we should use at least factor 0.2 in ship simulation.

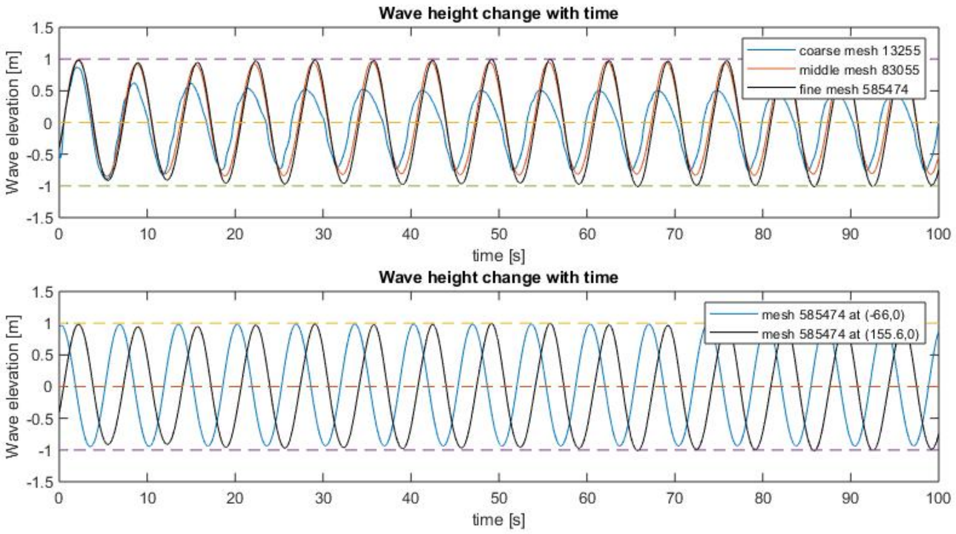


Figure 5.3: Wave mesh comparison

Since wave 0.2 gives a close result and with much less element nodes. Then in the next section, wave mesh with factor 0.2 will be used at least to optimize our grid and improve the results. This will save a lot of time and get a reasonable result.

5.3.2 *CFL number Check*

The Courant–Friedrichs–Lewy (CFL) condition is a necessary condition for convergence while solving certain partial differential equations (usually hyperbolic PDEs) numerically. It arises in the numerical analysis of explicit time integration schemes, when these are used for the numerical solution. As a consequence, the time step must be less than a certain time in many explicit time-marching computer simulations, otherwise the simulation produces incorrect results.

$$\mathrm{CFL} = \frac{u \triangle t}{\triangle x} \leqslant \mathrm{CFL}_{max}. \tag{5.3}$$

Whereas, u is the magnitude of the velocity (m/s), and $\triangle t$ is the time step (s), and $\triangle x$ is the length interval (m). The value of CFL_{max} changes with the method used to solve the discretised equation, especially depending on whether the method is explicit or implicit. If an explicit (time-marching) solver is used then typically $\mathrm{CFL}_{max} = 1$. Implicit (matrix) solvers are usually less sensitive to numerical instability and so larger values of CFL_{max} may be tolerated.

The General-α time integration, implicit and explicit combined method, is unconditionally converged based on our previous introduction in Chapter 4. It is useful to decide which CFL number is applied in the simulation. If increasing the CFL value does not effect the result, then it will save a lot of simulation time. In our code the CFL formulation looks like below,

$$\mathrm{CFL} = \triangle t \sqrt{(\mathbf{u} - \hat{\mathbf{u}})\mathbf{G}(\mathbf{u} - \hat{\mathbf{u}})}. \tag{5.4}$$

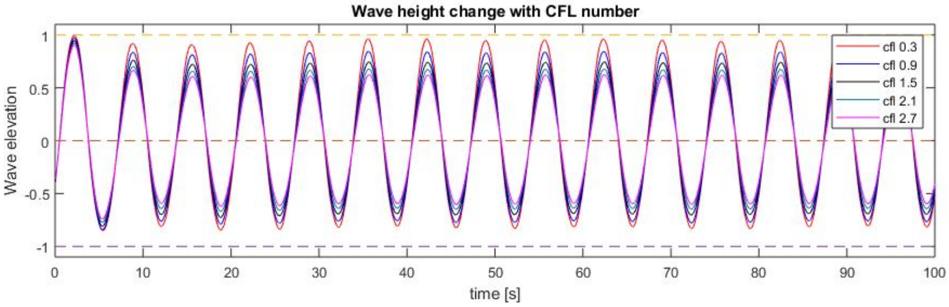


Figure 5.4: Wave height change with CFL number

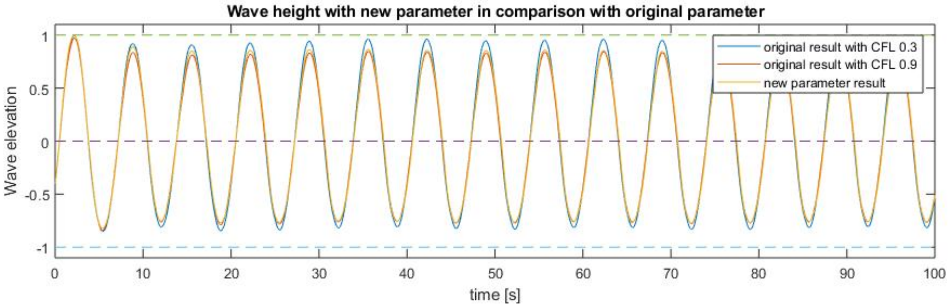


Figure 5.5: Wave height comparison among different parameter settings.

This equation is the same as usual CFL number but in a different form. **G** is metric tensor with unit $1/m^2$, then it is the same unit as normal. In Figure 5.4 the wave heights with different CFL number are shown.

From the results it can be found that the CFL definition will affect the output obviously. The reason of this is quite complicated. Comparing to CFL 0.3 case, CFL 0.9 lose 10% amplitude, which is quit large. However, this could be the reason of unsuitable setting of other parameter values. With some parameters increasing, the wave amplitude will decrease. For the other parameters, decreasing the value of parameter will cause an increment of the amplitude. Therefore, the amplitude error is a combination of these parameter results.

In Figure 5.5, it shows the difference between two different parameter setting. The yellow line shows our new parameter results with pseudo time equals to 0.2, penalty number equals to 50 and discontinuity capturing term for LSRD equals to 0.05. Although after changing the LSRD parameter, our simulation results with CFL 0.9 is closer original value. There is still a big gap between them. Therefore, the discontinuity capturing term of level set is tuned to see the change. In Figure 5.6, it shows the wave height with new parameter change with κ_{ls}. The new parameter with $\kappa_{ls} = 0.3$ gives a very close result to the original simulation. For higher κ_{ls}, the result is even better, but the simulation does not converge. Therefore, $\kappa_{ls} = 0.3$ is decided as our new parameter.

For higher CFL number, larger than 1, the wave height will still decrease a lot no matter how to change the parameter. Therefore, 0.9 will be our final CFL number.

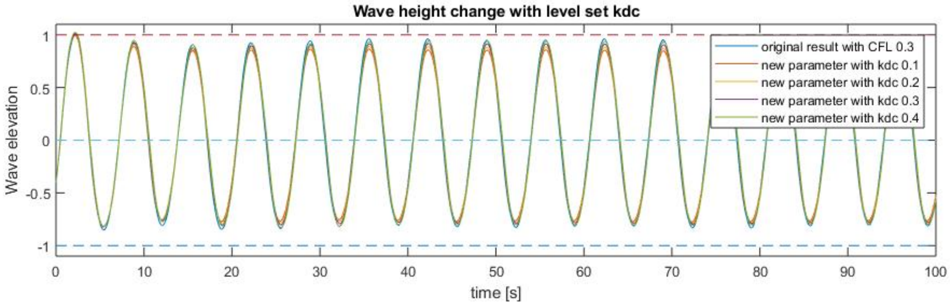


Figure 5.6: Wave height comparison among different κ_{ls}.

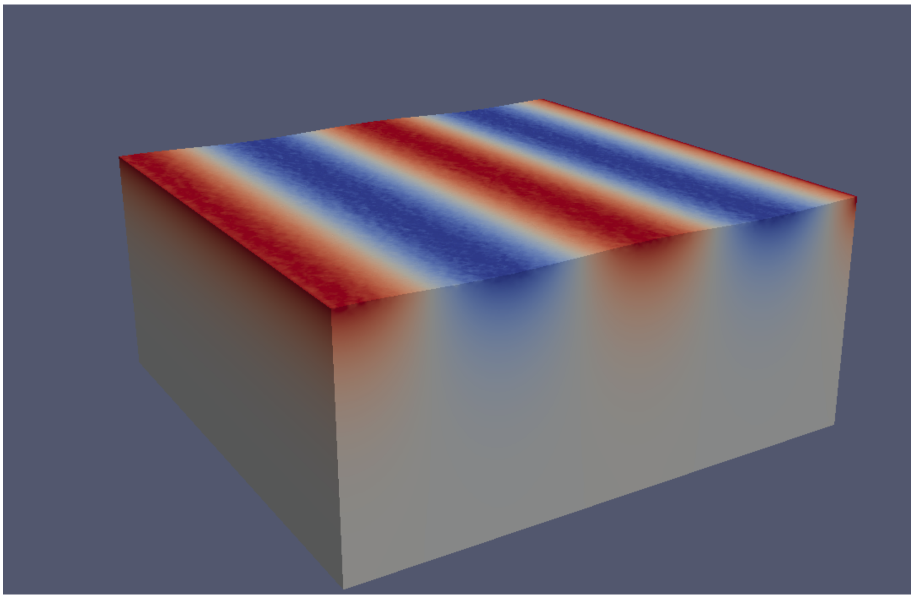


Figure 5.7: A snapshot of the wave velocity simulation result in enlarged domain

5.3.3 *Enlarged Domain Check*

After CFL number check, it is still needed to check domain effect. Since we shrink the domain width at the beginning, now we need to use the same domain as used in ship simulation and check if there is any difference of effect from boundary. In Figure 5.7, it gives a snapshot of enlarged domain wave simulation. And Figure 5.8 gives the comparison of two wave amplitude results.

From these results, it is assured that the domain width does not influence the wave amplitude. However, since the width is larger, the wave movement becomes more complicated, some non-linear effect could appear. This is why there are some disturbances in the wave amplitude

5.4 parameter setting

The decision of final parameter setting is half based on mathematical formulations and half based on intuitions. In the end with the new parameter setting, our CFL

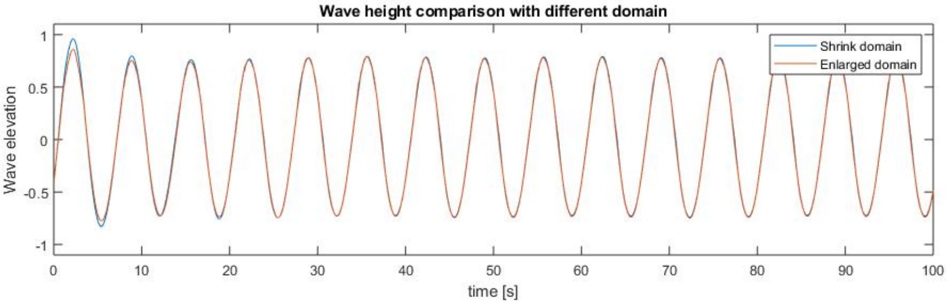


Figure 5.8: Results between enlarged domain and shrink domain

Original and new parameter setting comparison

Original parameter setting comparison		New parameter setting comparison	
C_{ns_a}	0.1	C_{ns_a}	0.1
C_{ns_w}	0.1	C_{ns_w}	0.1
C_{ls}	0.1	C_{ls}	0.3
C_{LSRD}	0.1	C_{LSRD}	0.05
λ_{pen}	800	λ_{pen}	50
$\triangle\widetilde{t}$	0.1	$\triangle\widetilde{t}$	0.2
$\triangle t\sqrt{(\mathbf{u}-\hat{\mathbf{u}})\cdot\mathbf{G}(\mathbf{u}-\hat{\mathbf{u}})}$	0.3	$\triangle t\sqrt{(\mathbf{u}-\hat{\mathbf{u}})\cdot\mathbf{G}(\mathbf{u}-\hat{\mathbf{u}})}$	0.9

Table 5.2: Original and new parameter setting comparison

number expressed as $\triangle t\sqrt{(\mathbf{u}-\hat{\mathbf{u}})\cdot\mathbf{G}(\mathbf{u}-\hat{\mathbf{u}})}$ is increased to 0.9 which is three times larger than before. This also means it will save three times shorter of the simulation time. Meanwhile, the wave amplitude result is almost the same as before which is believed good enough for the following simulation.

Table 5.2 gives the comparison between original parameter setting and new parameter setting. This new parameter setting will be used in the DTMB 5415M ship simulation in Chapter 6.

6 DTMB VALIDATION

6.1 CASE DESCRIPTION

In this chapter, a DTMB 5415M ship will be simulated. The ship will move in heading sea with the same condition in Chapter 5. The heave motion and pitch motion of the ship will be recorded for the validation.

As it is introduced before, the simulation domain is box shaped extending from $-0.5L < x < 1.8L, -1.1L < y < 1.1L, -1.0L < z < 0.25L$. Here L is the length of ship, $142m$. We will use the same boundary condition for ship simulation. The geometry of DTMB 5415M is shown in Figure 6.1. It is a benchmark in the ship simulation and many experimental tests are performed using this ship. It is a U.S. surface combatant. The main particular parameters are shown in Table 6.1. For more detail about fins, rudder, propeller and more experimental information, the reader can consult [18].

The six ship motions in the steadily translating system are defined by 3 translations, surge in the longitudinal x-direction with positive forwards, sway in lateral y-direction with positive to port side and heave in the vertical z-direction with positive upwards, and 3 rotations, roll about the x-axis with positive right turning, pitch about the y-axis with positive right turning and yaw about the z-axis with positive right turning. This is shown in Figure 6.2. The geometry of DTMB 5415M in Gmsh software is shown in Figure 6.3 Since our ship forward direction is heading to the negative x-direction, which means the longitudinal coordinate of gravity is changed to $0.683\%L$, which is equal to $71.48493m$. The wave angle is set to be 0.

Figure 6.4 shows a snapshot of our ship simulation results. Here, the result of velocity field is shown. The dark blue means high speed and the light blue means low speed, and with the color fated until white, the velocity will decrease to zero.

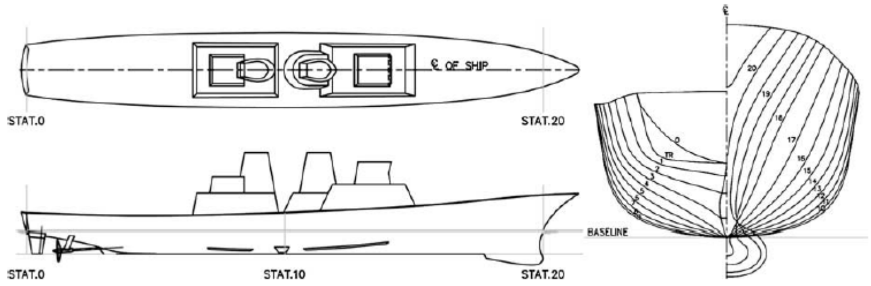


Figure 6.1: DTMB 5415M plan

DTMB 5415M main particulars

Length:L	142.0m	Natural period of roll	11.50s
Breadth:B	19.06m	Roll radii of gyration: k_{xx}	0.4B
Draft:T	6.15m	Pitch and yaw radius of gyration: k_{yy},k_{yy}	0.25L
Volume of displacement:∇	8432m^3	Propeller diameter:D_p	6.15m
Transverse metacentric height:GM	1.95m	Vertical position of CG:z_{CG}	7.51m
Block coefficient:C_b	0.507	Expanded blade area ratio:A_E/A_0	0.58
Longitudinal centre of gravity:x_{CG}	-0.683%L	Wetted surface area (bare hull)	2972.6m^2

Table 6.1: DTMB 5415M main particulars

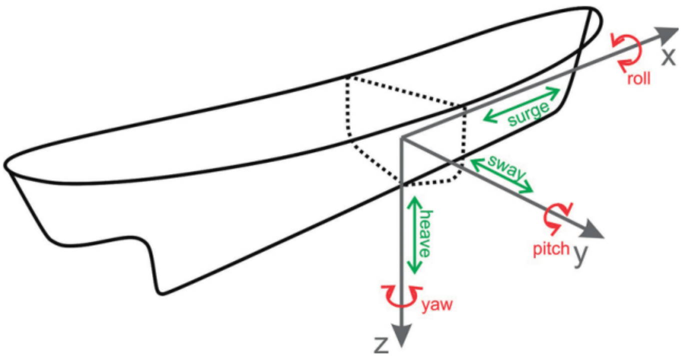


Figure 6.2: Six degree freedom

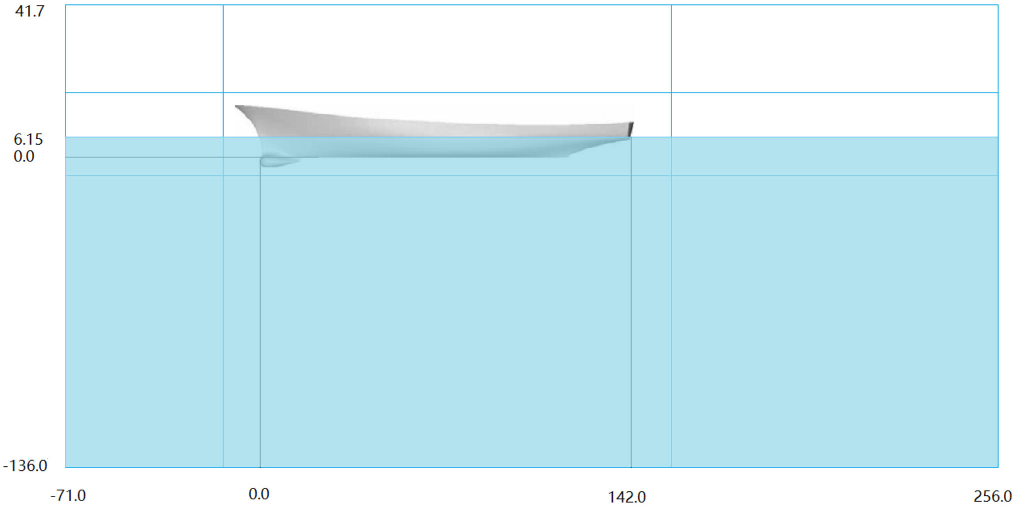


Figure 6.3: Gmsh geometry

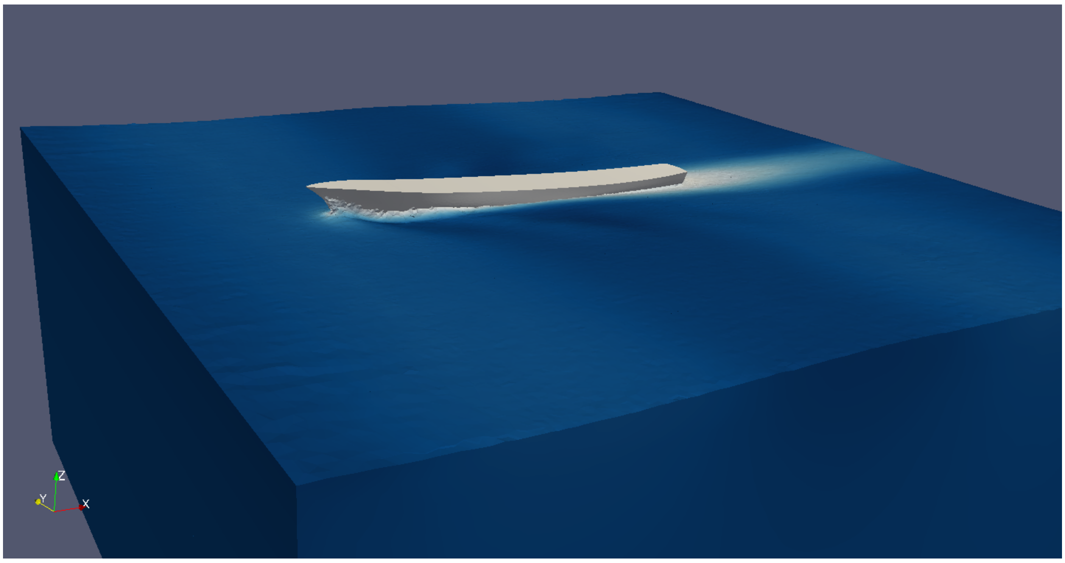


Figure 6.4: DTMB snapshot

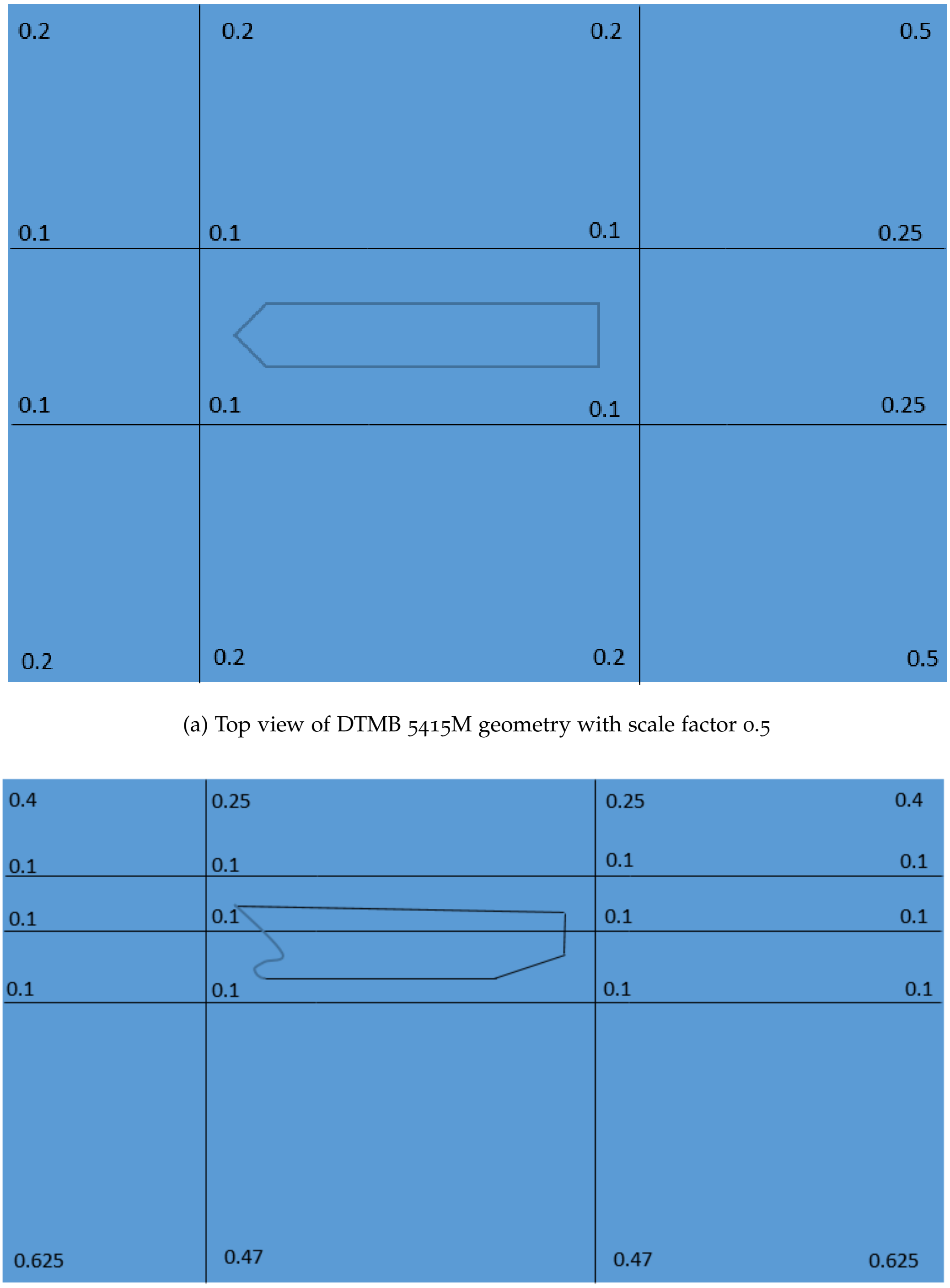
6.2 MESH GENERATION

In the wave simulation, factor 0.1 gives the most accurate result and theoretically the same factor should be used for the whole domain. However, in ship simulation, the domain is enlarged and the number of element nodes is also increased, which means more time will be taken to simulate. To save the time, only the middle of the domain is set to be scale 0.1. As for the rest, the factor is set to be 0.2. The reason to do this is because in head wave simulation, the far field area will not effect the middle domain too much. For the rear of the ship, since there is no need to study the detail of flow far behind of the ship, then there is no need to refine the mesh in the far rear. To generate such a mesh, two x-direction surface should be built and two y-direction surface should be built comparing to the wave simulation. It is shown in Figure 6.5. Therefore, there will be 36 domains divided in the entire domain comparing to 4 sub-domains in the wave simulation.

Meanwhile, since our problem is a deep water problem and no superstructure build in our geometry, the influence from the bottom and the influence from the top area of the air will not be important in the simulation. Therefore, these two area can be coarser than wave 0.1. In Figure 6.5, it shows the top view and front view of ship domain, the number in this figure means the factor with the same condition of wave simulation. This will give the reader more feeling of our domain mesh condition.

To get more accurate result, some specific area is refined again to capture the turbulence flow accurately. The bow area is refined. The second area is the rear of ship, there is a fold corner of the geometry and this area will occur turbulence. And the waterline area of the ship is also refined.

Figure 6.6 shows the ship mesh information from software CFDShip-Iowa, which is used in the validation paper. Figure 6.7 gives the detail of our mesh file in soft-



(a) Top view of DTMB 5415M geometry with scale factor 0.5

(b) Front view of mid-section of DTMB 5415M geometry with scale factor 0.5

Figure 6.5: DTMB 5415M mesh set.

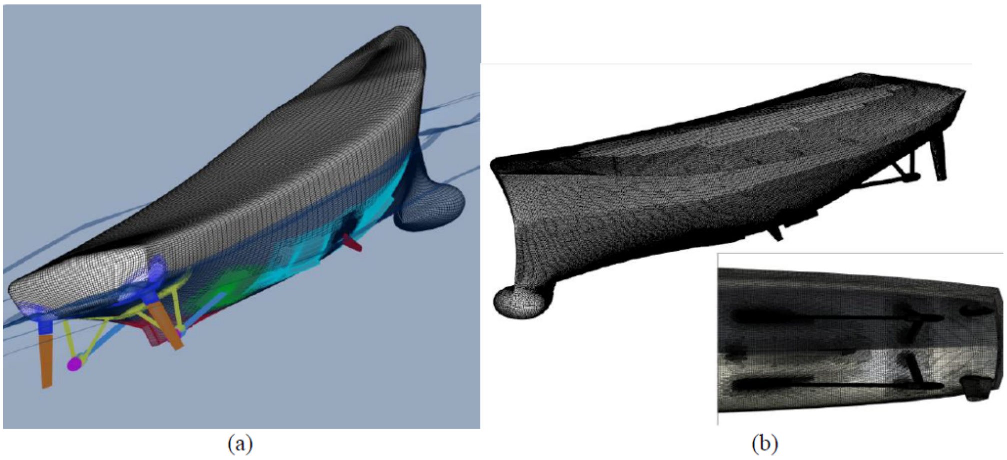


Figure 6.6: The mesh used in the software CFDShip-Iowa

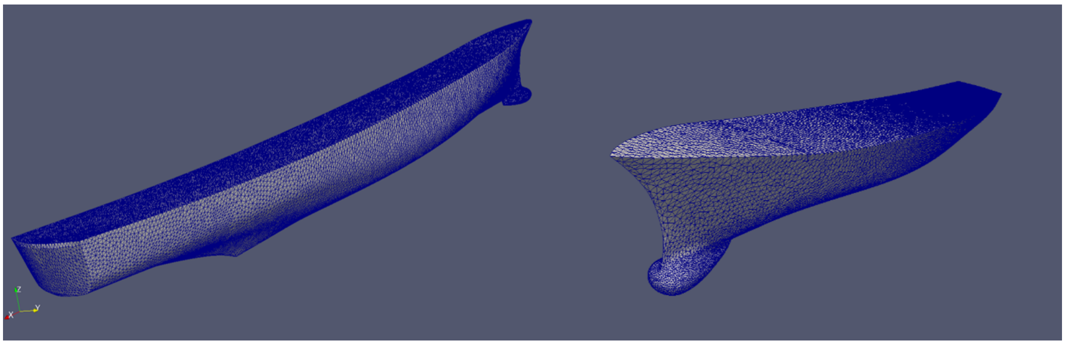


Figure 6.7: The mesh used in our RBVMS code

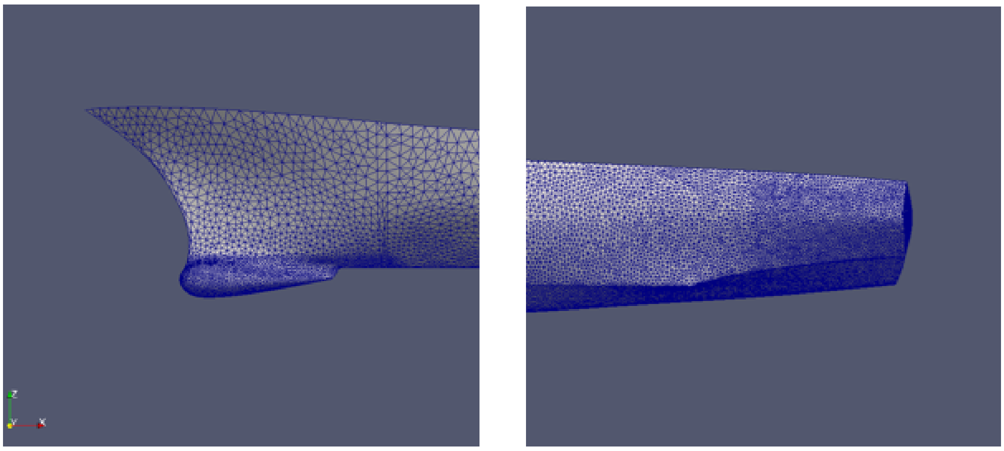


Figure 6.8: DTMB specific refine area

FFT of ship motions and rudder and fin angles for regular head waves

Value	EFD		CFD Ship-Iowa		RBVMS	
	a_1	a_2/a_1	a_1 E%D	a_2/a_1	a_1 E%D	a_2/a_1
z/A	0.753	0.008	-1.3	0	0.1	0.007
θ/Ak	0.808	0.008	-7.6	0	-0.38	0.003

Table 6.2: Results comparison among the experiment and CFD simulations

ware Gmsh. Some specific area mentioned before is also presented in Figure 6.8.

6.3 NUMERICAL RESULTS

The results of the experiment and another CFD software are shown in Table 6.2. EFD stands for experiments results and CFDShip-Iowa is the CFD software used in the validation paper. The last column is our RBVMS code results. For simplification, only heave motion and pitch motion are checked. The surge motion is not easy to check because of the absence of rudder in our geometry. In the table, a_1 and a_2 are the Fast Fourier Transform results. a_1 is the first harmonic wave amplitude result from FFT and a_2 is the second harmonic wave amplitude result from FFT. In the second harmonic wave, the wave frequency is twice larger than the first harmonic wave frequency. Since the phase value does not give any useful information for validation, we will not check it. The comparison error E between the experimental data D and the simulation value S is reported in a percentage form through the formulation: $E\%D = 100 \cdot (D - S)/D$.

There is one thing should be mentioned, in CFD Ship-Iowa the model is build with fin and rudder and also the same as experiment. However, for studying the

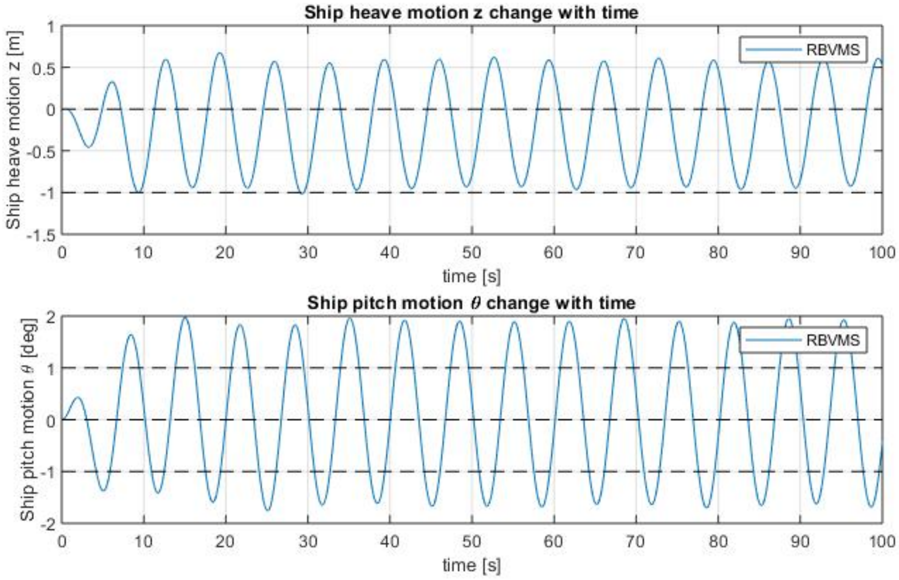


Figure 6.9: The results of heave and pitch motion of DTMB 5415M ship in the RBVMS code

heave motion and the pitch motion there is not too much effect from the existence of fin and rudder. Therefore, in our simulation, there is no fin and other appendix in our geometry.

Our results are shown in Figure 6.9. The first graph is the result of heave motion amplitude of the ship changing with time. The second graph is about the pitch motion amplitude. Comparing to the experimental data in the validation paper, our results are very close with only 0.1% error. Moreover, the software CFDShip-Iowa using a mesh with 18.6 million, our simulation with less element nodes 1.6 million. In the following contents, we will demonstrate the heave and pitch motion results together. To use FFT to get accurate value, we delete the first wave and calculate the rest entire waves. This is because the ship does not follow the wave motion at the beginning. Readers can imagine that the ship is set upright at the beginning without the consideration of the existence of the wave. The water level does not affect the result. Through a software called WebPlotDigitizer, the results in the validation paper are re-digitized. The comparison between the validation paper results and our code results are shown in Figure 6.10.

After applying FFT, our results looks like below in Figure 6.11 and Figure 6.12. In Table 6.2, the value of a_2 is not equal to zero after applying FFT. In the heave motion, a_2 equals to 0.00529, which means a_2/a_1 equals to 0.007. In the pitch motion, a_2 equals to $9.2e-5$, which means a_2/a_1 equals to 0.003. Our results are quit close to the experiment value, which means this RBVMS code can capture the nonlinear phenomena quite well. However, in the validation paper, no matter which method is used, there is no a_2 value available from the simulation, therefore, all a_2/a_1 equals to 0.

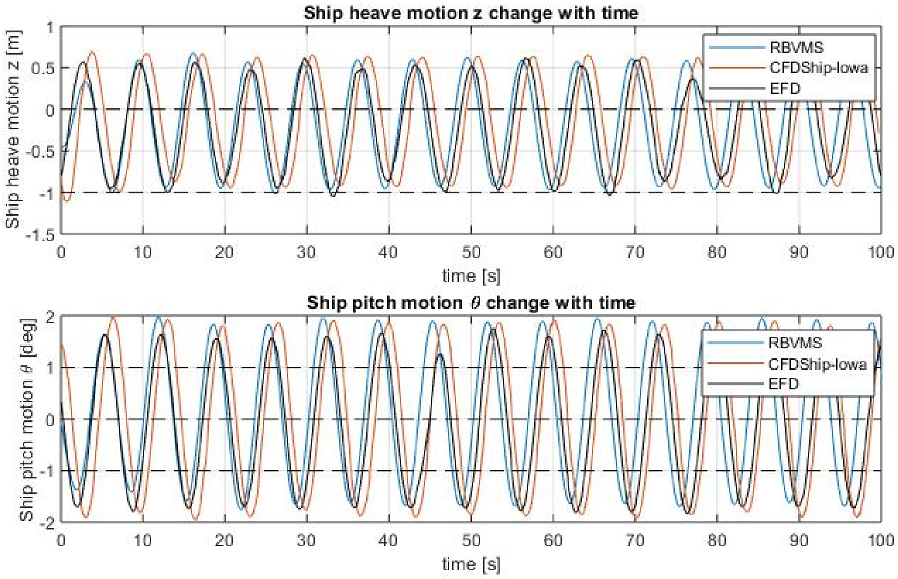


Figure 6.10: Ship simulation results comparison

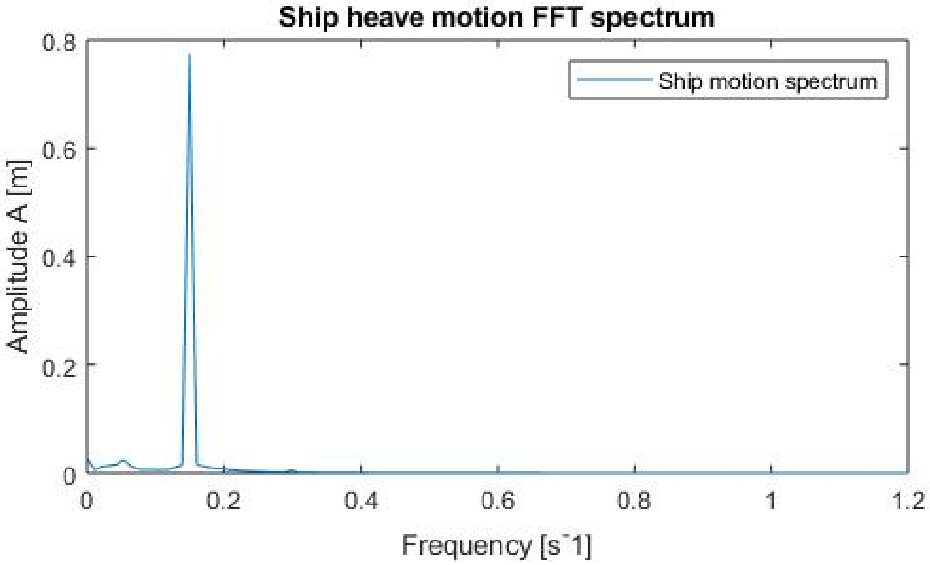


Figure 6.11: Heave motion FFT results

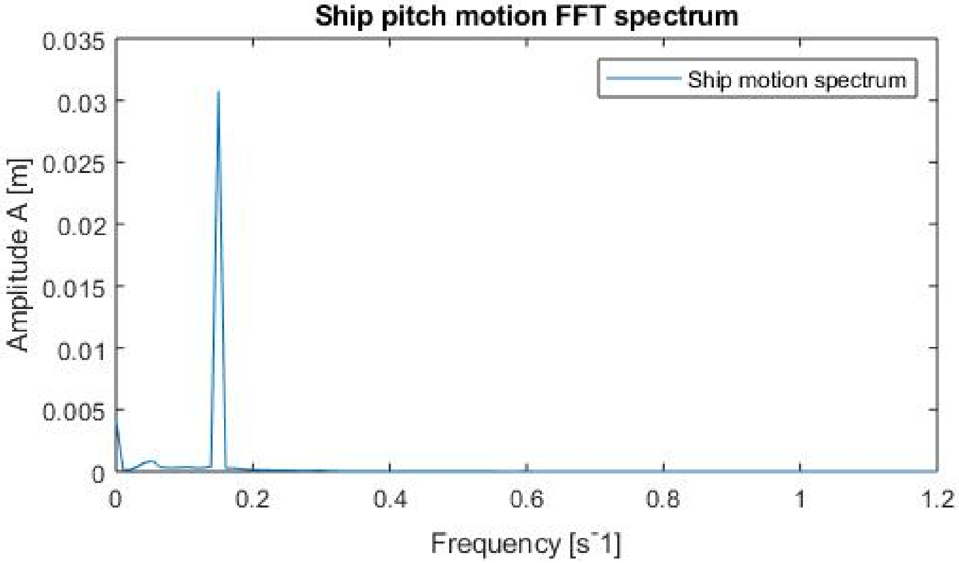


Figure 6.12: Pitch motion FFT results

FFT of ship motions for regular head waves

Value	EFD		Very fine mesh		Fine mesh		Middle mesh	
	a_1	a_2/a_1	a_1 E%D	a_2/a_1	a_1 E%D	a_2/a_1	a_1 E%D	a_2/a_1
z/A	0.753	0.008	0.1	0.007	2.35	0.008	5.23	0.009
θ/Ak	0.808	0.008	-0.38	0.003	3.82	0.003	6.54	0.004

Table 6.3: Mesh convergence results for DTMB 5415M ship

6.4 SHIP MESH CONVERGENCE STUDY

To do the mesh convergence study, here we will use reverse convergence method. Three meshes will be applied. The most fine mesh is the mesh we used in the last section with 1585100 element nodes. The fine mesh has 820939 element nodes. And the middle mesh has 416573 element nodes. This middle mesh is the coarsest mesh in the ship simulation. If there is a mesh with less element nodes than the middle mesh, it will give an incorrect result.

The results of the mesh convergence study is given in Table 6.3. According to this table we can find that our simulation results are reliable. The coarser of the mesh, the lower wave amplitude will be. And with the decrease of wave amplitude, the ship motion amplitude is also decreased. This is nicely coincide with the physical sense.

Moreover, there is an interesting phenomenon in Table 6.3. With decreasing the finite elements of the mesh, the value of a_2/a_1 is increasing. The reason could be the expression of $\mathbf{u}'$. In the RBVMS theory, we only use the simplest assumption and basic end of the approximation spectrum. To get more accurate results of a_2, a

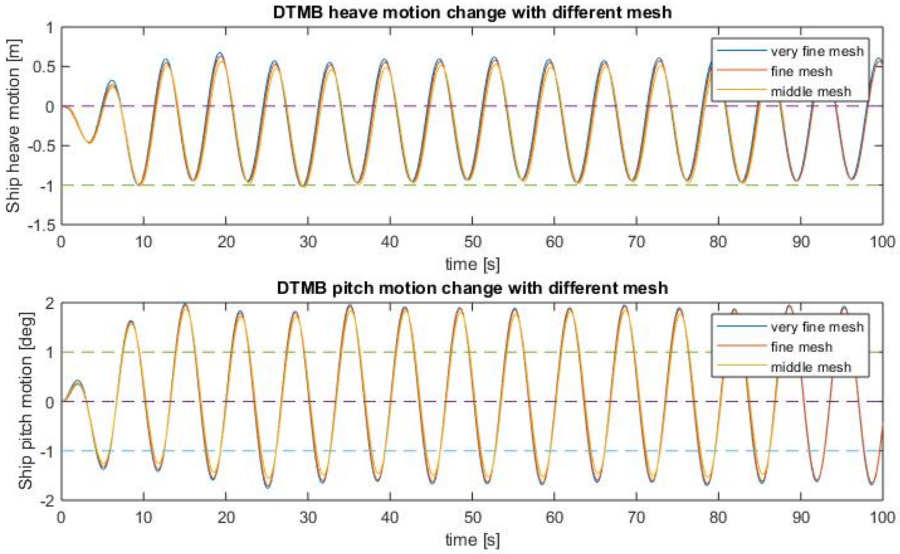


Figure 6.13: Mesh convergence

fully perturbation series expansion should be used. Reader can consult [3] for this part. Here, we believe the basic assumption is enough for the ship motion study since the value of a_2 is quite small comparing to a_1. And a conclusion can be made is that the VMS theory can capture the non-linear phenomenon very well.

6.5 ERROR ESTIMATION

In this section, Richard exploration method will be used to estimate the error and the uncertainty of our simulation results. Since our shape function is linear, a quadrature explanation should be used to calculate the error. The procedure is shown below.

$$\phi - \phi^h = Ch^2 \tag{6.1}$$

In Equation 6.1, the exact value is expressed as ϕ. The approximate value, also known as the simulation result, is expressed as ϕ^h. Whereas h is related to the size of grid. C is a constant. The estimation of exact solution will use the following equation,

$$\phi \approx \frac{\phi_2 - \frac{h_2}{h_1}\phi_1}{1 - \frac{h_2}{h_1}}. \tag{6.2}$$

The approximation of the exact heave motion amplitude is 0.758 and the approximation of the exact pitch motion amplitude is 0.824. These two exact value are drawn as a black line in Figure 6.14 and Figure 6.15 separately. To calculate the uncertainty, Equation 6.3 is applied,

$$\triangle\delta = \frac{||\phi - \phi_2||}{||\phi_2||}. \tag{6.3}$$

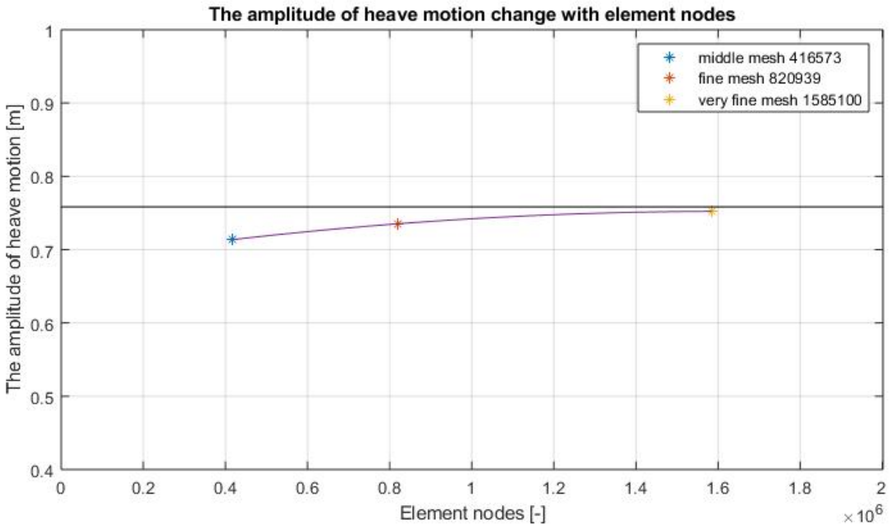


Figure 6.14: Heave amplitude error exploration

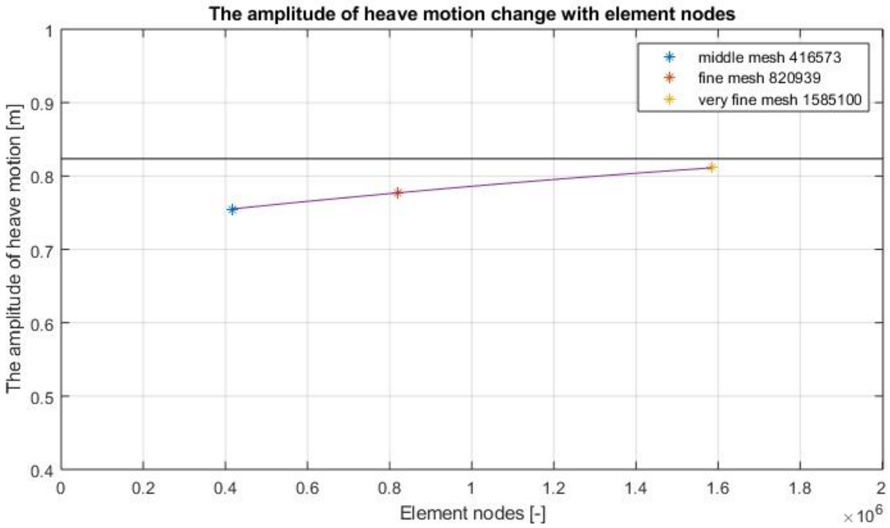


Figure 6.15: Pitch amplitude error exploration

The uncertainty, which is expressed by $\triangle\delta$, of the heave amplitude is about 0.82%. And the uncertainty of pitch amplitude is about 1.54%. The final error exploration plot of heave and pitch amplitude shows below in Figure 6.14 and Figure 6.15.

Until here, everything is done for this master thesis and this Richard error and uncertainty estimation gives us more confidence to our RBVMS code. The coarser of the mesh, the more divergence will be given. In the next chapter, our conclusions and recommendations will be given based on the whole thesis.

7 CONCLUSIONS AND RECOMMENDATIONS

Although RBVMS is developed for many years ago, few people use it. However, it is quite a solid numerical way for convection dominated problem based on this master thesis work. The conclusions and recommendations will be given in the following sections and some suggestions will be given for the future work.

7.1 CONCLUSIONS AND RECOMMENDATIONS

This master thesis starts with the introduction of RBVMS theory and gives the basic derivation of the governing equations. Then the spatial numerical discretizaion and time discretizaion method are presented. Based on above theory and techniques, a wave simulation is performed to test the RBVMS code and decide the parameter setting and the mesh grid size. Comparing to the airy wave theory, the wave amplitude is checked in this step and the results coincide with the theory very well. Afterwards, the real validation for this master thesis, DTMB 5415M ship, is performed using the information from the wave simulation. A very nice simulation result is obtained through this RBVMS code. Comparing to the experimental data, a 0.1% error with a 0.82% uncertainty is gotten at the end for the heave motion amplitude. For the pitch motion, a -0.38% error with a 1.54% uncertainty is obtained. Moreover, the second harmonic motion amplitude is captured using this code which gives us confidence for the simulation.

7.1.1 *Conclusions*

We can not get a conclusion that our RBVMS code is the best CFD code than the others since we could not make sure that the experiment gives 100% exact solution. However, comparing to the software CFDShip-Iowa the difference of between the heave motion error and the pitch motion error is much smaller. Then it gives us confidence to say that this RBVMS method is a good method to simulate ship motion in head sea. And the results are reliable with a small error comparing to the experimental results. Although some appendages are not meshed in our simulation, it does not give an obvious effect of the results.

Furthermore, this RBVMS code can capture the second harmonic motion amplitude very well comparing to the other methods. Although we only input a harmonic wave and the non-linear system does not play a major role in our simulation results, still a small value of a_2 is captured and very close to the experiment results. This gives us confidence to say that this RBVMS code can capture the non-linear system very well for the ship motion.

As we tested in Chapter 5, the CFL number should be lower than 1. Although General-α method is unconditionally converged, it is still needed to make sure the time step is not too large for the sake of accuracy. Another conclusion of this thesis is that the error is caused by different parameters setting. Changing some parameters sometimes will compensate the error which hides the real problem. Therefore, for a quick and accurate result, testing is needed before the real simulation.

It gives the author feeling that there is a lower limit requirement for the number of the mesh element nodes. When we were doing the wave simulation, if the mesh is coarser than wave0.4, the wave amplitude is not correct any more. As for the ship simulation, if the mesh is coarser than the middle mesh, the output will give a negative error percentage for the heave motion. These two phenomena suggest the number of element nodes should not be too small.

7.1.2 *Recommendations*

In this master thesis, linear shape function is used to approximate the exact solution. According to the results, it is enough accurate for a ship motion simulation using the linear shape functions. However, it is still quite interesting to use quadrature shape functions to see if a more accurate result could be attained. Furthermore, how much time will be spend for the quadrature shape function is also an interesting question. If it will consume double time with little accuracy, then it is not needed to do that for the future study.

Since the computer resources in my faculty is limited, a finer mesh is not tested in this master thesis. If it is possible, a finer mesh should be calculated to see if there is much more difference than the current mesh. This mesh is acceptable for the ship motion simulation, however, in the other tests, like ship rolling decay and wave added resistance test, the mesh should be different.

7.2 FUTURE WORK

It is mentioned many times in this master thesis, a new technology called IGA is developed, which will exactly describe the profile of a ship and even more complex geometry. Then it is quite interesting to study if a better result will get using this technology. The IGA has already been tested in many other people's work and their results seem quite good. However, an application in a real scale ship is not performed. Therefore, it is a quite interesting topic for the future study.

As for a scientific research, more parameters existed in a code means more uncertainty. Therefore, using only one single parameter to express all the other parameters is an interesting topic for the future study. This parameter should be decided based on the mesh information. However, this job is not easy to be done since it requires the student a strong mathematical background. One more thing, since few people are using this code, only a few types of boundary conditions are written in it. Therefore, more typed of boundary conditions can be added into this code, such as periodical boundary condition.

Part III

APPENDIX

A SHAPE FUNCTION AND IGA INTRODUCTION

A.1 FINITE ELEMENT METHODS

To approximate $\mathbf{u}^h$ and p^h, shape function should be introduced. For the Garlerkin method, which is used in this code, the weighting function use the same shape function as unknown variables,

$$\mathbf{u}^h = \sum_{i=1}^{4} N_i(\mathbf{x}) u_i, \qquad \boldsymbol{w}^h = \sum_{i=1}^{4} N_i(\mathbf{x}). \tag{A.1}$$

In this thesis, linear interpolation will be used to express shape function and tetrahedron is chosen as element shape, therefore i counts to four, which means four nodes for each element. This $N_i(\mathbf{x})$ must satisfied two properties: 1) $\sum_{k=1}^{n} N_k = 1$ 2) $N_i(x_k) = \delta_{ik}$. The second requirement is called Kronecker-delta property. The shape function shows below,

$$N_i(\mathbf{x}) = a_i + b_i x + c_i y + d_i z. \tag{A.2}$$

Here, x, y and z are global coordinate, we need to transform it into a local coordinate system $\{\xi, \eta, \zeta\}$ which is shown in Figure A.1.

The calculations can be significantly simplified by carrying out a coordinate transformation. Each point (x, y, z) of the tetrahedron in the original coordinate system can be mapped to a corresponding point (ξ, η, ζ) in the transformed coordinate system uding the following equation group:

$$\begin{aligned}
x &= x_1 + (x_2 - x_1)\xi + (x_3 - x_1)\eta + (x_4 - x_1)\zeta, \\
y &= y_1 + (y_2 - y_1)\xi + (y_3 - y_1)\eta + (y_4 - y_1)\zeta, \\
z &= z_1 + (z_2 - z_1)\xi + (z_3 - z_1)\eta + (z_4 - z_1)\zeta.
\end{aligned}$$

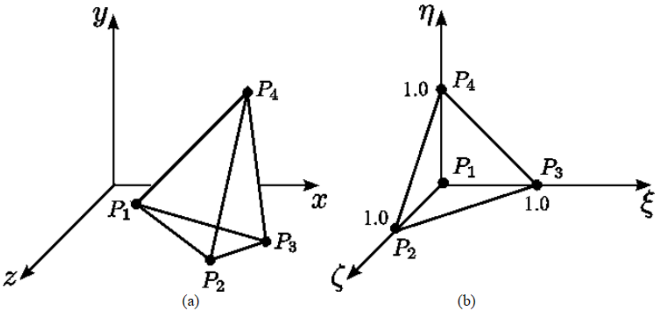


Figure A.1: Element coordinate transformation

Which in matrix form it leads to the Jacobian matrix :

$$J = \begin{bmatrix} x_2 - x_1 & x_3 - x_1 & x_4 - x_1 \\ y_2 - y_1 & y_3 - y_1 & y_4 - y_1 \\ z_2 - z_1 & z_3 - z_1 & z_4 - z_1. \end{bmatrix}$$

In this way, the nodal basis functions for the tetrahedron in the transformed coordinate system are given by ,

$$\begin{aligned} N_1^t(\xi,\eta,\zeta) &= 1 - \xi - \eta - \zeta, \\ N_2^t(\xi,\eta,\zeta) &= \xi, \\ N_3^t(\xi,\eta,\zeta) &= \eta, \\ N_4^t(\xi,\eta,\zeta) &= \zeta. \end{aligned} \tag{A.3}$$

And its gradient formulation becomes,

$$\nabla N(\xi,\eta,\zeta) = J^T \nabla N(x,y,z). \tag{A.4}$$

A.2 IGA INTRODUCTION

A new technology called isogeometric analysis is introduced here, which is NURBS based. The reason to develop it is trying to eliminate the gap between FEA and CAD. People found there are many similarities between finite element analysis and isogeometric analysis. Both of them have similar basic functions and h- and p- refinement characteristics. However, the IGA still has many difference with FEA and some important advantages. A key feature is to represent geometry exactly by NURBS elements and then invoke the isoparametric concept to define field variables, such as displacement, temperature, etc. The coarsest mesh encapsulates the exact geometry. This means that mesh refinement is simply accomplished by reindexing the parametric space. The refinement process can proceed without interaction with the CAD system, a distinct advantage over finite element procedures. In this part, the basic function of IGA (NURBS based) will be written, for more information, reader can reference [1] and [10].

In one dimension, a knot vector is a set of coordinates in the parametric space, $\mathbb{O} = \{\xi_1, \xi_2, \cdots, \xi_{n+p+1}\}$. p is the ploy-nominal order, and n is basics functions number which comprise the B-spline. For $p = 0$, B-spline basis functions shows below,

$$N_{i,0}(\xi) = \begin{cases} 0 & \text{if } \xi_i \leqslant \xi \leqslant_{i+1}, \\ 1 & \text{otherwise}, \end{cases} \tag{A.5}$$

for p=1,2,3...

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \tag{A.6}$$

NURBS is the short of non uniform rational B spline. The above functions are

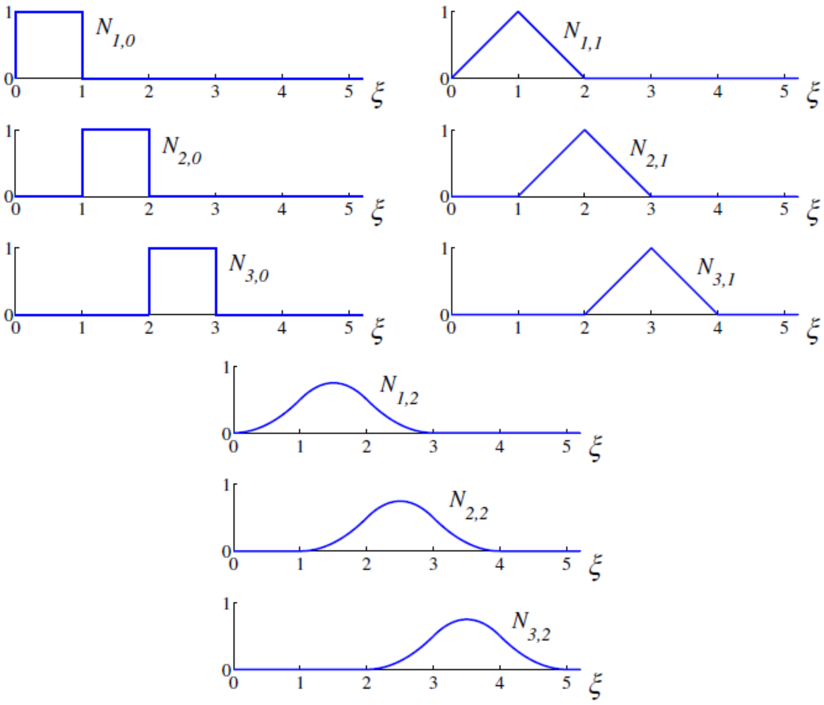


Figure A.2: NURBS basic functions of order 0, 1, 2

uniform. Therefore the weights should be consideration into our basis functions:

$$\begin{aligned} R_i^p(\xi) &= \frac{N_{i,p}(\xi)w_i}{W(\xi)} = \frac{N_{i,p}(\xi)w_i}{\sum_{i=1}^{n} N_{i,p}(\xi)w_i}, \\ W(\xi) &= \sum_{i=1}^{n} N_{i,p}(\xi)w_i. \end{aligned} \tag{A.7}$$

NURBS curve then can be represented as below:

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} R_i^p(\xi)\mathbf{B}_i. \tag{A.8}$$

$\mathbf{B}_i$ are the control points (NURBS object). Surface is two dimensional, and its NURBS-spline surface function is,

$$\mathbf{S}(\xi,\eta) = \sum_i \sum_j N_{i,p}(\xi)R_{j,q}(\eta)\mathbf{B}_{i,j}. \tag{A.9}$$

The pictures in Figure A.3 and Figure A.4 show the basic functions of NURBS and FEM. Which give more feelings to the reader.

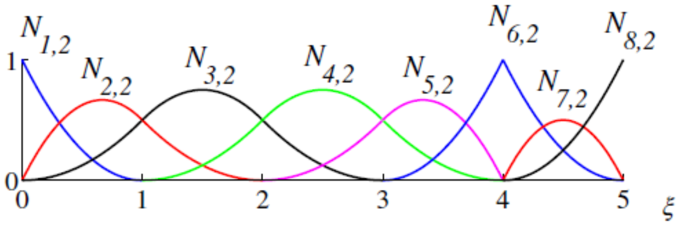


Figure A.3: NURBS one dimensional basic functions

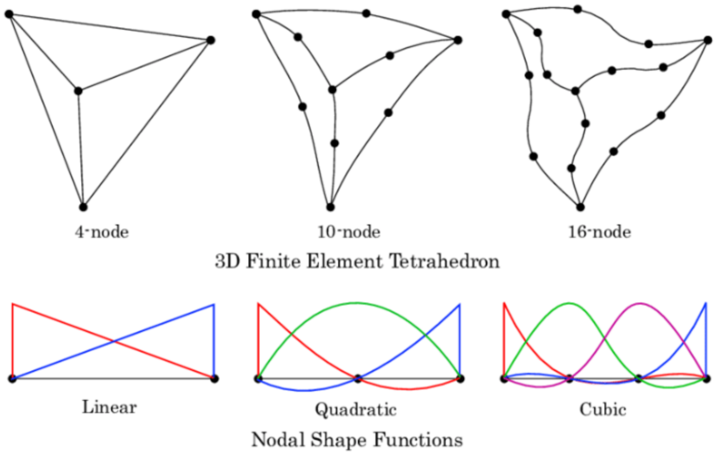


Figure A.4: Finite element basic functions

B GMRES

Direct methods based on Gaussian elimination can be used to solve A=bx, usually people do it like $x = A^{-1}b$. However, these methods quickly become too memory and CPU intensive to be of practical interest, especially in three dimensions. Consequently, the finite element community has turned towards iterative solvers based on the generation of Krylov spaces. A Krylov Space looks like below,

$$K_n = K_n(A, b) = span\{b, Ab, A^2b, \cdots, A^{m-1}b\}. \tag{B.1}$$

And, we can substitute A^{-1} as following,

$$A^{-1}b \approx \sum_{i=0}^{m-1} \beta_i A^i b = \beta_0 b + \beta_1 Ab + \beta_2 A^2 b + \cdots + \beta_{m-1} A^{m-1} b. \tag{B.2}$$

m smaller than the dimension of matrix of A, β_i is unknown vector, therefore if we know β_i, the equation will be solved.

However, the vectors $b, Ab, A^2b, \cdots, A^{m-1}b$ might be close to linearly dependent, and they are ill-defined for general non-symmetric systems may not result in robust algorithms. Therefore it is needed to use Arnoldi iteration to find orthogonal vectors $q_1, q_2, \cdots, q_n$ which from a basis of K_n.

The Arnoldi method is to start with an arbitrary vector q_1 with norm 1, and then use $q_k = q_{k-1}A$ to do iteration. Then use a loop to project the vector into each direction, after deduction each projection, a new vector h is found, now $q_k = q_k/h_{k,k-1}$. These h vectors consist upper Hessenberg matrix, $H_n = Q_n^* A Q_n$, where,

$$H_n = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & \cdots & h_{1,n} \\ h_{2,1} & h_{2,2} & h_{2,3} & \cdots & h_{2,n} \\ 0 & h_{3,2} & h_{3,3} & \cdots & h_{3,n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & h_{n-1,n} & h_{n,n} \end{bmatrix}. \tag{B.3}$$

The matrix H_n can be characterized by the following optimally condition. The characteristic polynomial of H_n minimizes $||p(A)q_1||_2$ among all monic polynomials of degree n. This optimally problem has a unique solution if and only if the Arnoldi iteration does not break down.

The relation between the Q matrices in subsequent iterations is given by $AQ_n = Q_{n+1}\tilde{H}$, where,

$$\tilde{H}_n = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & \cdots & h_{1,n} \\ h_{2,1} & h_{2,2} & h_{2,3} & \cdots & h_{2,n} \\ 0 & h_{3,2} & h_{3,3} & \cdots & h_{3,n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & 0 & h_{n-1,n} & h_{n,n} \\ 0 & \dots & \dots & 0 & h_{n+1,n} \end{bmatrix}. \tag{B.4}$$

Because of Q_n is orthogonal, we have,

$$||Ax_n - b|| = ||\tilde{H}_n y_n - Q_{n+1}^T b|| = ||\tilde{H}_n y_n - \beta e_1||. \tag{B.5}$$

Where, $e_1 = (1, 0, 0, \cdots, 0)^T$ and $\beta = ||b - Ax_0||$, therefore, x_n can be found by minimizing the Euclidean norm of the residual $r_n = \tilde{H}_n y_n - \beta e_1$ and $x_n = Q_n y_n$.

In the code, GMRES part reference paper [18] which gives the detail of Fortran code.

C NEWTON ITERATION

The idea is to start with an initial guess which is reasonably close to the true root, then to approximate the function by its tangent line using calculus, and finally to compute the x-intercept of this tangent line by elementary algebra. This x-intercept will typically be a better approximation to the original function's root than the first guess, and the method can be iterated,

$$y = f^{'}(x_n)(x - x_n) + f(x_n).$$

This is the tangent line function, x_n is a initial guess. Which means initial condition where $f^{'}$ denotes the derivative. The x-intercept of this line (the value of x which makes y = 0) is taken as the next approximation, x_{n+1}, to the root, so that the equation of the tangent line is satisfied when $(x, y) = (x_{n+1}, 0)$,

$$0 = f^{'}(x_n)(x_{n+1} - x_n) + f(x_n).$$

Solving for x_{n+1} gives :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

This method is used in time derivation, which will be introduced in the next section. Here, a short introduction is shown to give the reader a basic idea.

BIBLIOGRAPHY

[1] Ido Akkerman, Y Bazilevs, CE Kees, and MW Farthing. "Isogeometric Analysis of Free-Surface Flow." In: *Journal of Computational Physics* 230 (2011), pp. 4137–4152.

[2] Ido Akkerman, Y Bazilevs, DJ Benson, MW Farthing, and CE Kees. "Free-surface flow and fluid-object interaction modeling with emphasis on ship hydrodynamics." In: *Journal of Applied Mechanics* 79.1 (2012), p. 010905.

[3] Y Bazilevs, VM Calo, JA Cottrell, Thomas JR Hughes, A Reali, and G Scovazzi. "Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows." In: *Computer methods in applied mechanics and engineering* 197.1-4 (2007), pp. 173–201.

[4] Alexander N Brooks and Thomas JR Hughes. "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations." In: *Computer methods in applied mechanics and engineering* 32.1-3 (1982), pp. 199–259.

[5] Jintai Chung and GM Hulbert. "A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method." In: *Journal of applied mechanics* 60.2 (1993), pp. 371–375.

[6] PAB De Sampaio and Alvaro Luiz Gayoso de Azeredo Coutinho. "A natural derivation of discontinuity capturing operator for convection–diffusion problems." In: *Computer methods in applied mechanics and engineering* 190.46-47 (2001), pp. 6291–6308.

[7] Jean Donea, Antonio Huerta, Jean-Philippe Ponthot, and Antonio Rodríguez-Ferran. "Arbitrary L agrangian–E ulerian Methods." In: *Encyclopedia of Computational Mechanics Second Edition* (2017), pp. 1–23.

[8] Friedel Hartmann. *Green's functions and finite elements*. Springer Science & Business Media, 2012.

[9] Thomas JR Hughes. "Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods." In: *Computer methods in applied mechanics and engineering* 127.1-4 (1995), pp. 387–401.

[10] Thomas JR Hughes, John A Cottrell, and Y Bazilevs. "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement." In: *Computer methods in applied mechanics and engineering* 194.39-41 (2005), pp. 4135–4195.

[11] Thomas JR Hughes, Luca Mazzei, and Kenneth E Jansen. "Large eddy simulation and the variational multiscale method." In: *Computing and visualization in science* 3.1-2 (2000), pp. 47–59.

[12] Thomas JR Hughes and Assad A Oberai. "Calculation of shear stresses in the Fourier–Galerkin formulation of turbulent channel flows: projection, the Dirichlet filter and conservation." In: *Journal of Computational Physics* 188.1 (2003), pp. 281–295.

[13] Thomas JR Hughes and James Winget. "Finite rotation effects in numerical integration of rate constitutive equations arising in large-deformation analysis." In: *International journal for numerical methods in engineering* 15.12 (1980), pp. 1862–1867.

[14] Thomas JR Hughes, Gonzalo R Feijóo, Luca Mazzei, and Jean-Baptiste Quincy. "The variational multiscale method—a paradigm for computational mechanics." In: *Computer methods in applied mechanics and engineering* 166.1-2 (1998), pp. 3–24.

[15] Volker John and Petr Knobloch. "On discontinuity—capturing methods for convection—diffusion equations." In: *Numerical mathematics and advanced applications*. Springer, 2006, pp. 336–344.

[16] CE Kees, Ido Akkerman, MW Farthing, and Y Bazilevs. "A conservative level set method suitable for variable-order approximations and unstructured meshes." In: *Journal of Computational Physics* 230.12 (2011), pp. 4536–4558.

[17] Sunitha Nagrath, Kenneth E Jansen, and Richard T Lahey Jr. "Computation of incompressible bubble dynamics with a stabilized finite element level set method." In: *Computer Methods in Applied Mechanics and Engineering* 194.42-44 (2005), pp. 4565–4587.

[18] Farzin Shakib, Thomas JR Hughes, and Zdeněk Johan. "A multi-element group preconditioned GMRES algorithm for nonsymmetric systems arising in finite element analysis." In: *Computer Methods in Applied Mechanics and Engineering* 75.1-3 (1989), pp. 415–456.

[19] Joseph Smagorinsky. "General circulation experiments with the primitive equations: I. The basic experiment." In: *Monthly weather review* 91.3 (1963), pp. 99–164.

[20] Mark Sussman, Peter Smereka, and Stanley Osher. "A level set approach for computing solutions to incompressible two-phase flow." In: *Journal of Computational physics* 114.1 (1994), pp. 146–159.

[21] Serge Toxopeus, Frans van Walree, and Rink Hallmann. "Manoeuvring and seakeeping tests for 5415M." In: *AVT-189 Specialists', UK* (2011).

[22] Serge Toxopeus, Hamid Sadat-Hosseini, Michel Visonneau, Emmanuel Guilmineau, Tin-Guen Yen, Woei-Min Lin, Gregory Grigoropoulos, and Frederick Stern. "CFD, potential flow and system-based simulations of fully appended free running 5415M in calm water and waves." In: *International Shipbuilding Progress* Preprint (2018), pp. 1–31.

[23] Christian H Whiting et al. "Stabilized finite element methods for fluid dynamics using a hierarchical basis." In: *Master's thesis (Rensselaer Polytechnic Institute Troy, New York, 1999)* (1999).

[24] Hongkai Zhao. "A fast sweeping method for eikonal equations." In: *Mathematics of computation* 74.250 (2005), pp. 603–627.