

Severi Kosonen

ON THE DIRECT STABILITY ASSESSMENT OF PARAMETRIC ROLLING WITH CFD

Preparation for the Second Generation Intact Stability Criteria

Master of Science Thesis
Faculty of Engineering and Natural Sciences
Examiner: Assoc. Prof. Lassi Paunonen
Dr. Petteri Laakkonen
MSc. Kari Kyyrö
March 2023

ABSTRACT

Severi Kosonen: On the direct stability assessment of parametric rolling with CFD
Master of Science Thesis
Tampere University
Master's Programme in Science and Engineering
March 2023

Parametric rolling is a phenomenon where a ship starts to uncontrollably oscillate with increasing amplitude. This is caused by the surrounding waves parametrically amplifying the naturally occurring small rolling motions. The phenomenon can be dangerous, but the irregular nature of the sea makes it rare.

The International Maritime Organization is drafting a new type of assessment for ship stability in wave-related phenomena. This assessment is known as the Second Generation Intact Stability Criterion (SGISC), and parametric rolling is one of its focuses. The ship design is studied at three levels, where the computational complexity gradually increases. The final level is referred to as Direct Stability Assessment (DSA), where the phenomenon is modeled as accurately as possible. One possible method for this phase is Computational Fluid Dynamics (CFD) simulations, where the fluid flow surrounding the ship is numerically resolved from the Navier-Stokes equations. This allows the ship's response to waves to be accurately resolved without making any assumptions about the design.

This thesis studies the possibility of using CFD in the DSA phase for the parametric roll phenomenon. First, a literature survey was conducted on the hydrodynamics of the phenomenon, CFD, and the SGISC. Then, multiple CFD simulations were conducted. These simulations included the actual DSA simulation as well as supplementary simulations for other phases of the assessment. It was found that CFD is a valuable tool for studying the vulnerability to parametric rolling. However, the computational power required to directly study the parametric rolling may still be too high for a typical shipyard.

Keywords: Second Generation Intact Stability Criteria, SGISC, Direct Stability Assessment, DSA, Computational Fluid Dynamics, CFD, Navier-Stokes equations, parametric roll, parametric resonance, Mathieu's equation, Floquet's theorem

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

TIIVISTELMÄ

Severi Kosonen: Parametrisen keinunnan suorasimulointia CFD:llä
Diplomityö
Tampereen yliopisto
Teknis-luonnontieteellinen DI-ohjelma
Maaliskuu 2023

Parametrinen keinunta on ilmiö, jossa laiva alkaa hallitsemattomasti keinua kasvavalla amplitudilla eli yhä suuremmilla poikkeamilla aluksen tasapainoasemasta. Ilmiö on seurausta laivaa ympäröivistä aalloista, jotka vahvistavat aluksen pieniä ja luonnollisia kallistusliikkeitä resonanssin omaisesti. Ilmiö voi olla vaarallinen, mutta meren aaltojen epäsäännöllinen luonne tekee siitä kuitenkin harvinaisen.

Kansainvälinen merenkulkujärjestö (IMO) laatii uutta arviointikriteeriä alusten vakaudelle aaltoihin liittyvissä ilmiöissä. Tämä kriteeri tunnetaan nimellä "Second Generation Intact Stability Criterion (SGISC)", ja parametrinen keinunta on yksi sen painopisteistä. Aluksen vakautta tutkitaan kolmessa tasossa, joissa laskennallinen kompleksisuus kasvaa vaiheittain. Viimeistä tasoa kutsutaan suorasimuloinniksi (eng. Direct Stability Assessment, DSA), ja siinä ilmiötä pyritään mallintamaan mahdollisimman tarkasti. Yksi mahdollinen menetelmä suorasimuloinnin toteuttamiseen on laskennallinen virtausdynamiikka (eng. Computational Fluid Dynamics, CFD), jossa alusta ympäröivä virtaus ratkaistaan numeerisesti Navier-Stokesin yhtälöistä. Menetelmä mahdollistaa aluksen tarkan reaktion tarkastelun erilaisten aaltojen tapauksissa.

Tässä diplomityössä tutkitaan, voidaanko CFD:tä käyttää parametrisen keinunnan tutkimiseen suorasimulaatiovaiheessa. Työn aluksi toteutettiin kirjallisuuskatsaus liittyen parametrisen keinunnan hydrodynamiikkaan, virtausdynamiikkaan ja SGISC:n arviointikriteeriin. Tämän jälkeen työssä toteutettiin useita erilaisia CFD-simulaatioita. Tuloksista voidaan huomata, että CFD on arvokas työkalu parametrisen keinunnan haavoittuvuuksien analysoinnissa. Se kuitenkin osoittautui myös olevan laskennallisesti kallis, minkä takia sen käyttö voi vielä olla epärealistista yleiselle telakalle.

Avainsanat: Second Generation Intact Stability Criteria, SGISC, Direct Stability Assessment, DSA, Numeerinen Virtausdynamiikka, CFD, Navier-Stokes yhtälöt, Parametrinen Keinunta, Parametrinen Resonanssi, Mathieun yhtälö, Floquetin lause

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

PREFACE

I would like to express my deepest gratitude to my supervisors Lassi and Kari for their guidance during this thesis. The fact that both of them spent their free time assessing my work continues to amaze me. I would also like to thank Petteri for agreeing to act as a separate examiner. Big thank you to the people at Rauma Marine Constructions, which enabled this work in the first place.

This thesis finalizes my educational career and now I can start to build my professional career. I feel fortunate that I have found a field that interests me and that an operator in this field is also interested in me. The academic part of me can always find something new to learn and the industrial part of me is excited to be part of building something as awesome as a floating hotel. Nobody can tell what the future will hold, but for now, I am excited to get to work. Thank you to everyone that had a part in this chapter in my life. (Now that I have officially moved to this small coastal city of Finland, I can start to address it as the natives do.)

Raum, 24th March 2023

Severi Kosonen

CONTENTS

1. Introduction	1
2. Hydrodynamics	4
2.1 Hydrostatics	4
2.2 Roll motions	10
2.2.1 Floquet's theorem	12
2.2.2 Stability of undamped Mathieu equation	18
2.2.3 Stability of damped Mathieu equation	22
2.3 Water wave theory	24
2.3.1 Airy wave	25
2.3.2 Stokes wave	28
2.3.3 Irregular waves	31
3. Computational fluid dynamics	35
3.1 Governing equations	35
3.1.1 Conservation of mass	36
3.1.2 Conservation of momentum	37
3.2 Reynolds Averaged Navier-Stokes equations	40
3.3 Turbulence modeling	42
3.3.1 Zero-equation models	43
3.3.2 One-equation models	44
3.3.3 Two-equation models	45
3.3.4 Higher order models	47
3.4 Finite volume method	49
3.4.1 Discretization process	49
3.4.2 Solving linear algebraic linear system	53
3.4.3 SIMPLE-algorithm	57
4. Second generation intact stability criteria	60
4.1 Vulnerability level 1	61
4.2 Vulnerability level 2	62
4.3 Direct stability assessment	65
5. Computational setup	68
5.1 OpenFOAM	68
5.1.1 Rigid body dynamics	70
5.1.2 Wave damping with forcing zones	73

5.2 Simulation Setups	76
5.2.1 Free roll decay	78
5.2.2 Parametric roll	80
6. Results	83
7. Conclusions and further work.	90
References.	93
Appendix A: Basic ship vocabulary and metrics	99
Appendix B: Tables	105

ABBREVIATION AND ACRONYMS

A	Coefficient matrix for linear system
a	Amplitude
AP	Aft perpendicular
B	Ship's beam or breadth
BiCG	Bi-Conjugate Gradient method
C	Monodromy matrix
C_1, C_2	Level 2 check coefficients
C_{2f}	Speed check coefficient in following waves
C_{2h}	Speed check coefficient in head waves
CFD	Computational Fluid Dynamics
CG	Conjugate Gradient method
C_m	Ship's midsection coefficient
CS	Control Surface
$C_{S,i}$	Individual case check coefficient
CV	Control Volume
D	Ship's depth
DNS	Direct Numerical Simulations
DSA	Direct Stability Assessment
ϵ	Turbulence eddy-dissipation
η	Surface height function
f	Frequency
FEM	Finite Element Method
F_n	Froude number
\mathbf{F}	Force vector
FP	Forward perpendicular
$\Phi(t)$	The fundamental solution matrix
FVM	Finite Volume Method

g	Gravitational constant
H	Wave height
I	Identity matrix
I_d	Inertia of waterplane at draft d
IMO	International Maritime Organisation
ITTC	International Towing Tank Conference
I_x	Moment of inertia around x -axis
k	Turbulence kinetic energy
λ	Wavelength
$\lambda_{A,i}$	The i -th eigenvalue of the matrix A
L	Length
LOA	Length overall
LPP	Length between perpendiculars
LWL	Length of waterline
μ	Dynamic viscosity
∇	Ship's volume of displacement
$S(f, \theta)$	Wave density spectrum
T_n	Natural roll period
ν	Kinematic viscosity
ν_t	Kinematic eddy-viscosity
ω	Angular frequency
ω_n	Natural angular roll frequency
p	Pressure
ϕ	Angle around the x -axis, i.e. the roll angle
ψ	Arbitrary flow property
RANS	Reynolds Averaged Navier-Stokes equations
Re	Reynolds number
ρ	Density
RMC	Rauma Marine Constructions
\mathbf{S}_f	Vector area of a surface
SGISC	Second Generation Intact Stability Criteria
Σ	Stress tensor

σ	Deviatoric stress tensor
τ	Dimensionless time
θ	Wave encounter angle
TUNI	Tampere Universities
u, v, w	Velocity components in their respective x , y , and z directions
\mathbf{u}	Velocity vector ($u_1 = u, u_2 = v, u_3 = w$)
W_i	Weighting factor
ξ	Velocity potential
ζ	Dimensionless damping coefficient
ζ_{ij}	Random phase number

1. INTRODUCTION

During the last decade, the shipping industry has taken a keen interest in building bigger and faster ships to support the growing international trading. To support bigger weight, these ships typically have a flared bow and flat stern, which has been observed to have poor stability, especially when interacting with waves [72] [47]. For example in 2014, a Danish container ship Svendborg Maersk was caught in a storm and suddenly experienced high-rolling action. The ship eventually inclined to a maximum of 41 degrees. No major injuries had occurred to the crew, but 517 containers were lost overboard and approximately 250 were damaged [6].



Figure 1.1. Svendborg Maersk arriving in Malaga after the incident [6]

While it is impossible to establish the precise weather conditions at the time of the incident, the investigation [6] still believes that the high roll angles were caused by a resonance effect from the waves. This phenomenon is known as *parametric roll* or *parametric resonance*, where the amplitude of the oscillating ship starts to gradually increase with each oscillation. The physics behind this phenomenon have been known for at least half a century and the first published theories date back to 1952 by Grim [24] and 1955 by Kerwin [34]. Loosely speaking, the parametric roll is analogous to the situation where a

person is pushed on a swing. The incoming waves "push" the ship back to its upright position and then "gives room" for the ship to incline to the other side. A more physical explanation of the phenomenon is that waves travelling through the hull alter the location of the centre of buoyancy. This causes variation to the *metacentric height* (see Section 2.1), which is an abstract concept to evaluate the ship's stability. Waves in favourable conditions can exploit this by creating high restoring moments when the ship is inclined and low restoring moments when the ship is at its upright position [52]. This causes the ship to experience a larger amplitude than what it experienced in the preceding period. However, given that the nature of waves in seas is random, the probability of encountering nearly regular waves is small, which makes the phenomenon rare. The parametric roll occurs mostly when the waves travel to the ship's direction i.e. head or following waves (see Appendix A), but can also occur in other scenarios i.e. the oblique waves [25].

The phenomenon is well-known in theory. However, no rigorous guidelines exist on determining the vulnerability of the ship to parametric rolling since different ship designs behave differently in waves. The current intact stability assessment is based on statistical data and considers the stability of the ship in calm waters. However, as of late, International Maritime Organisation (IMO) has been working on a new physics-based stability assessment named Second Generation Intact Stability Criteria (SGISC), where parametric roll is one of five failure events. This assessment will gradually study the ship's design from simple criteria checks up to a direct stability assessment i.e. a full probabilistic study involving multiple detailed simulations in realistically modelled irregular waves. [47]

This thesis is created in co-operation with Rauma Marine Constructions (RMC), which is a Finnish shipyard located at Rauma that specializes in ROPAX ferries, ice breakers and navy frigates. The primary goal of the thesis is to determine whether Computational Fluid Dynamics (CFD) would be a viable option in conducting the direct stability assessment step of SGISC, as CFD has shown of being a valuable tool in predicting parametric roll [22] [39]. As the phenomenon is rare at irregular seas, the question of whether the computational cost would be too high remains. Multiple CFD simulations are conducted for a concept hull to achieve the goal. A secondary goal of the thesis is to conduct a literature survey concerning SGISC, the hydrodynamics of the phenomenon, and CFD.

Chapter 2 focused on the underlying *hydrodynamics* of the phenomenon. This chapter begins by introducing *hydrostatics*, which is a branch of hydrodynamics that examines the stability of ships in calm waters. Next, the simplified roll equation is modified into a dimensionless *Mathieu's equation*, and it is analyzed with *Floquet's theorem*. Here the style of writing is more mathematical, but the results are analyzed in a more general fashion. Finally, a section on the water waves studies the derivation of Airy waves, extending this to Stoke's waves, and the generation of irregular waves.

Chapter 3 covers the theory of *Computational Fluid Dynamics* (CFD) by starting with

the derivation of the *Navier-Stokes equations*, which describe the motion of the fluid. Solving these directly is however computationally expensive, so Reynolds averaging is applied to derive the Reynolds Averaged Navier-Stokes (RANS) equations. Because of this averaging, turbulence needs to be modelled with a separate model. The partial differential equations are turned into algebraic equations and solved utilizing the Finite Volume Method (FVM).

Chapter 4 then breaks down the SGISC for the parametric roll section. The official guidelines divide the assessment into three sections with increasing complexity: *level 1*, *level 2*, and *Direct stability assessment* (DSA). In this thesis, levels 1 and 2 are presented and the DSA guidelines are analysed more in-depth.

Finally, several simulations are run as the practical part of the thesis. The computational setup is introduced in Chapter 5 and the results from the simulations are analyzed in Chapters 6 and 7. The simulations conducted are free roll decay and parametric roll in regular and irregular waves. The more standard way to determine whether the ship is vulnerable to parametric rolling is by directly solving the simplified equation of motion for roll angle. This requires knowing the ship's damping capabilities as well as the natural roll frequency. While empirical methods exist for finding these, a more accurate method to obtain these is by conducting a free roll decay simulation. The parametric roll in regular wave allows testing the CFD capabilities on parametric roll simulations. It also allows finding accurately the maximum amplitude for regular wave scenarios. The irregular version on the other hand reflects the actual simulations that would be part of actual DSA analysis. The waves are created realistically to model a specific sea state.

2. HYDRODYNAMICS

Hydrodynamics or fluid dynamics is a branch of physics that studies the motion of fluids. As ships float on the sea, this branch is crucial in naval architecture due to its applications in ship power efficiency, maneuverability, and most importantly, stability.

In this chapter, a theoretical review is taken of some of the hydrodynamic topics which relate to parametric roll. Hydrostatics, which studies the stability of ships in calm waters, is a good starting point. The concepts of metacenter \overline{GM} and the righting-lever \overline{GZ} are introduced as they give a mathematical definition for the ship's stability. Then, the equation of motion with respect to the roll angle is introduced and its stability criterion is derived using Floquet's theorem. This part of the thesis is also written in a more mathematical style including definitions, theorems, and proofs. Finally, a section about waves discusses the water wave theory and how to model irregular seas to a given state.

2.1 Hydrostatics

Ship hydrostatics is an important branch of naval architecture, as it studies the stability of floating objects. While the thesis has a heavy emphasis on the stability of the ship in waves, it is important to understand the basic stability concepts in a calm sea. This section is based on the hydrostatic books [5], [4], and [44] where readers are referred for further reading as well. For readers with no prior knowledge of vocabulary and metrics used in the naval field, Appendix A is recommended.

To study the stability of a floating body, it is helpful to sketch the free-body diagram and determine the effecting forces. As the length of the ship is much larger than the width, the ship is much more likely capsize due to the roll motions than the pitch motions. Therefore, we are interested in investigating the stability in a rolling motion and can neglect the x -axis. The system becomes two-dimensional, and the free-body diagram is drawn as follows.

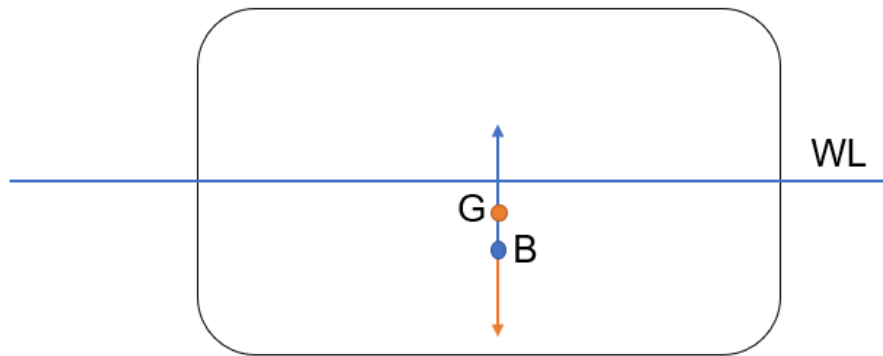


Figure 2.1. Free body diagram of the ship in calm water

The affecting downward force G is the gravitational force and depends on the mass as $G = mg$. The upward force B is the buoyant force and from the Archimedes principle, it must equal the weight of the displaced volume. Thus, the buoyancy force has a form $B = \rho g \nabla$, where ρ is the density of the fluid and ∇ is the volume that the hull displaces i.e. the volume of displacement. For the ship to be in equilibrium, these forces must be equal i.e.

$$mg = \rho g \nabla \quad (2.1)$$

The ship has multiple elements with some mass. Instead of drawing them all to the diagram, a better way is to use the concept of centroids, as in Figure 2.1. It allows us to draw an analogous figure, where all masses are combined so that the free-body diagram is similar. Mathematically speaking, given that the system is composed of individual massed m_1, m_2, \dots , the center of mass is defined as

$$\mathbf{x}_M = \frac{\sum \mathbf{x}_i m_i}{M}, \quad (2.2)$$

where M is the total mass of the system and \mathbf{x}_i are the relative locations of the mass elements. Similarly, the center of buoyancy is located at the centroid of the volume submerged.

If the ship is exposed to a small inclination angle, the submerged part of the hull changes slightly. This causes the center of buoyancy to shift as demonstrated in Figure 2.2.

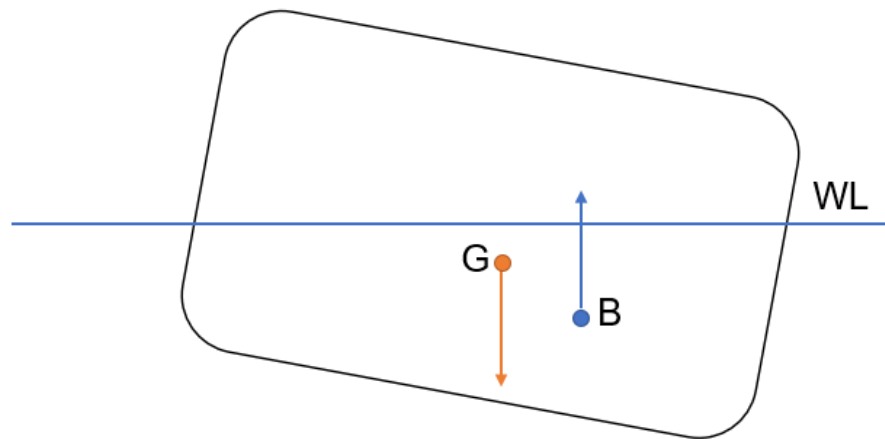


Figure 2.2. Free body diagram of an inclined ship in calm water

Since now the forces don't lie on the same vertical line, they produce a rotational moment. In this case, the moment is called restoring moment as it tries to rotate the ship back to its upright position. If the center of buoyancy would affect from the other side, the moment would rotate the ship even more making the system unstable. This behaviour has been known for centuries and the quest to find the criteria for stability goes back even to the early 1700s. In 1746 a French mathematician Pierre Bouguer published *Traité du Navire*, where he proposed a concept of metacenter. It is an abstract point that lies in the intersection of an action line from buoyant force and the inclination line through the center of gravity. In Figure 2.3, the point M_ϕ is the location of the metacenter for the ship that is inclined by an angle of ϕ . Note that here we keep the hull stationary and rotate the waterline, which allows viewing the base case and the rotated case in the same picture.

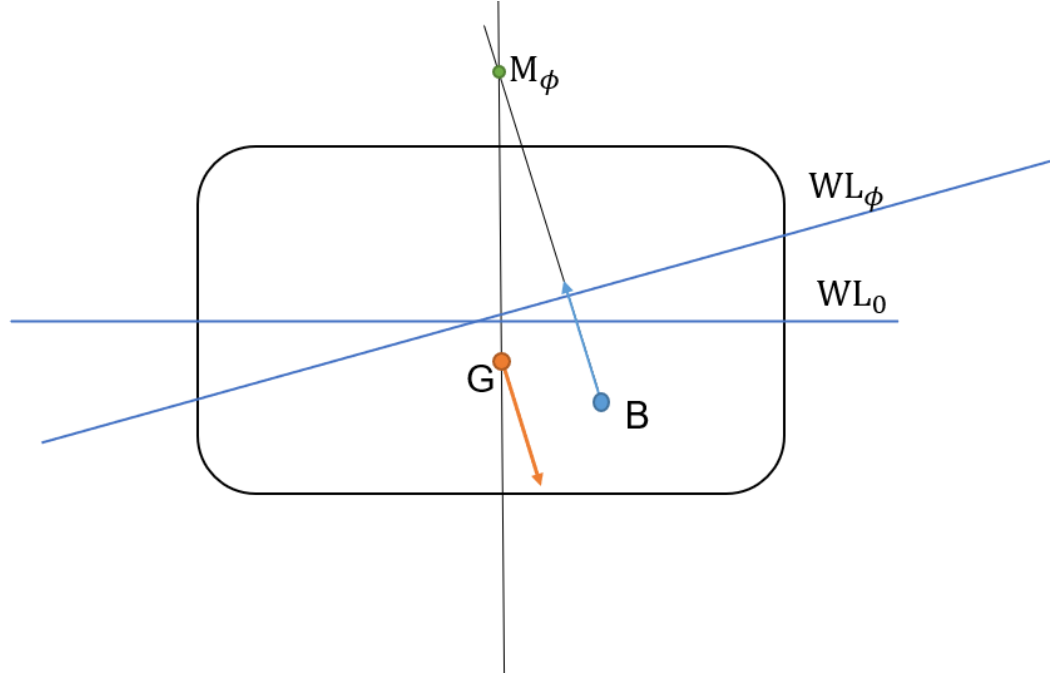


Figure 2.3. The concept of metacenter

As long as the metacenter lies higher than the center of gravity, the forces create a restoring moment for the hull and vice versa. Therefore, if we define the distance from the center of gravity to the metacenter as

$$\overline{GM} = \overline{KB} + \overline{BM} - \overline{KG}, \quad (2.3)$$

where K means the lowest point of the ship i.e. the keel, then we notice that the ship has a restoring moment whenever

$$\overline{GM} > 0. \quad (2.4)$$

From equation (2.3), the components for the stability become more clear. For example, by lowering the center of mass, the term \overline{KG} gets smaller, and thus \overline{GM} gets larger.

Since the metacenter is dependent on the location of the center of buoyancy and the center of buoyancy is dependent on the angle of inclination ϕ , we get that $\overline{GM} := \overline{GM}(\phi)$. In literature, the metacentric height at the ship's upright position is directly referred to as $\overline{GM}(0) = \overline{GM}$. We also have assumed that the center of mass doesn't move when the ship inclines. In reality, this value also alters slightly due to the free surface effect inside the ship's tanks. However, we neglect these for now for simplicity.

The \overline{GM} can be calculated from equation (2.3). The center of buoyancy is located at the centroid of the hull underwater and the center of gravity is defined by the loading condition, so the only unknown that remains is the metacentric radius \overline{BM} . This can be

calculated from

$$\overline{BM} = \frac{I_d}{\nabla}, \quad (2.5)$$

where I_d is the waterplane moment of inertia at draft d and ∇ is the volume of displacement. Given that the hull is symmetric, the waterplane moment of inertia is defined as

$$I_d = \frac{2}{3} \int_0^L y^3 dx, \quad (2.6)$$

where y is a curve that defined the ship's contour in the waterplane at draft d .

While the concept of \overline{GM} does explain whether the ship tends to incline back to the upright position, the parameter itself is rarely used directly in stability calculations, as the metacenter point is an abstract concept. A more applicable parameter would be the so-called \overline{GZ} , which is the leveraging arm and can be used to determine the actual restoring moment due to buoyancy. If we draw a line from G that is perpendicular to the action line from B to M , the intersection of these lines happens at point Z as illustrated in Figure 2.4.

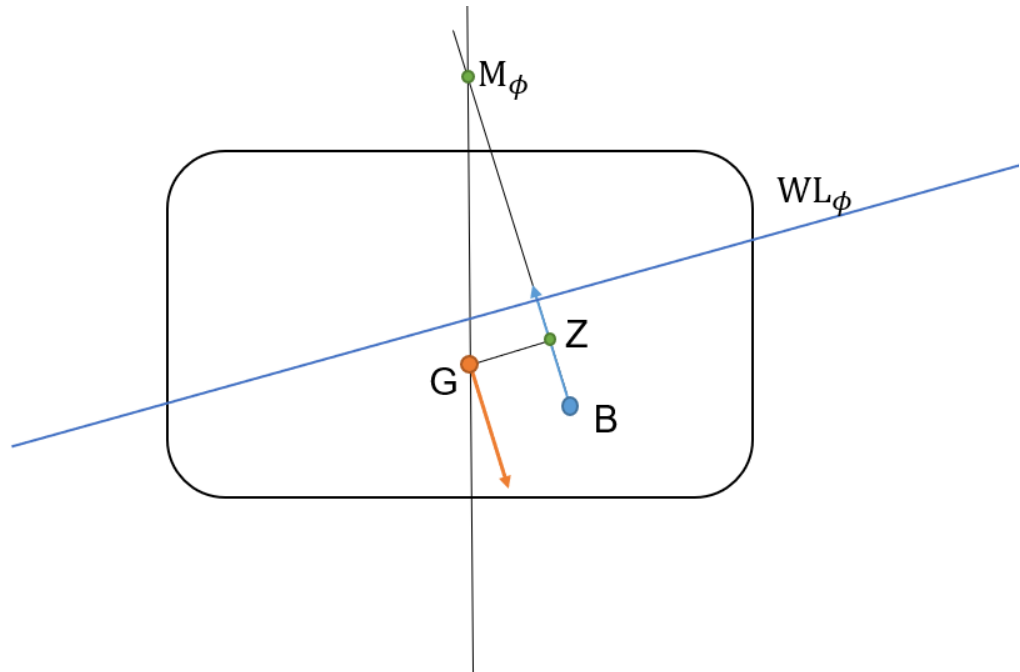


Figure 2.4. Moment arm \overline{GZ} illustrated

As with the metacentric height, the righting lever is also a function of the roll angle. Therefore we can calculate different \overline{GZ} values for different ϕ values and plot them all to a graph. This curve is often referred to as the *curve of statistical stability* or *GZ curve* and typically has the following form.

Loading condition LC00, GM=1.115 m

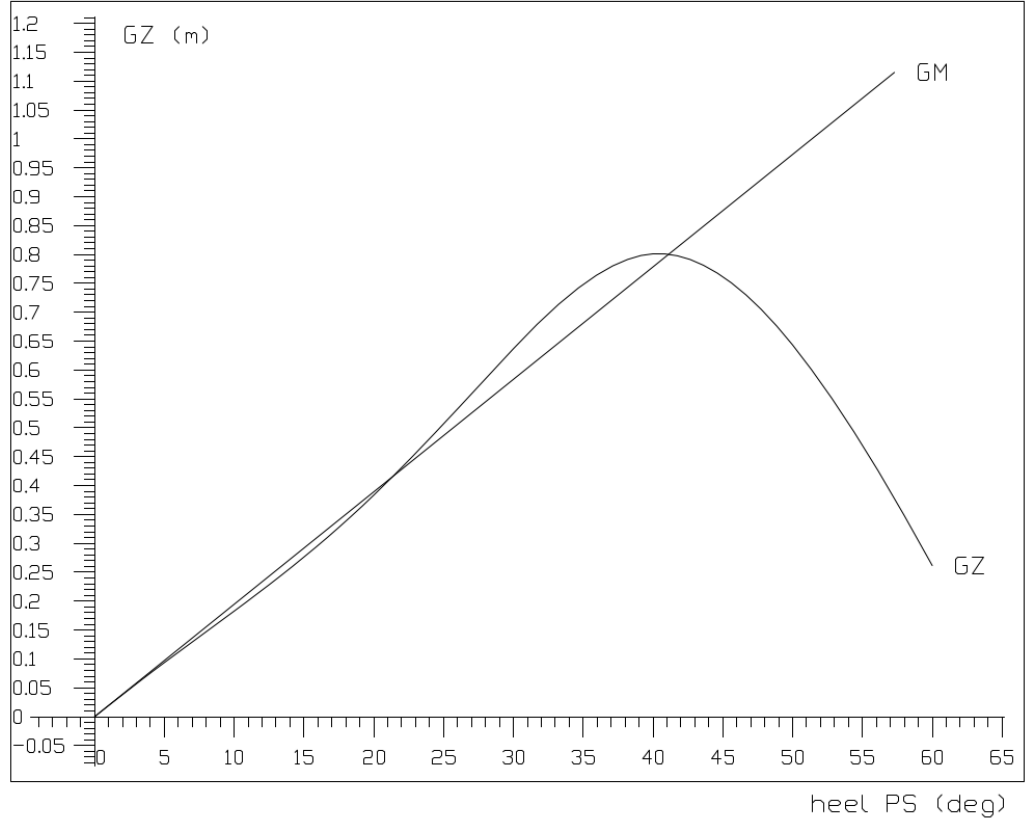


Figure 2.5. *GZ curve of the model used in the thesis, which is calculated by hydrostatics software NAPA*

One interesting property from the GZ curve is that the value initial metacenter \overline{GM} can be graphically extrapolated, by drawing a tangent line at $\phi = 0$ to the curve and interpolating the value at 1 radian (≈ 57.3 degrees). To see why this is the case, for small angles ϕ we can approximate

$$\overline{GZ} \approx \overline{GM} \sin(\phi). \quad (2.7)$$

Now looking at the tangent line when ϕ approaches 0, we see that

$$\lim_{\phi \rightarrow 0} \frac{\overline{GZ}}{\phi} = \lim_{\phi \rightarrow 0} \frac{\overline{GM} \sin(\phi)}{\phi} = \overline{GM} = \frac{\overline{GM}}{1}. \quad (2.8)$$

Verbally explained indeed the tangent at $\phi = 0$ equals the slope $\overline{GM}/1(\text{rad})$. Thus if the slope could be approximated geometrically, then the value of \overline{GM} is found from the slope at $\phi = 1$ radians. This is useful when verifying the \overline{GZ} curve as \overline{GM} is most often calculated by hydrostatic software.

For each ship model and loading condition, a unique \overline{GZ} curve can be sketched. The obvious benefit of this is that when simulating the ship motions, the restoring moment can

be explicitly evaluated from the precalculated curves. For example, in ITTC guidelines [16], the nonlinear roll equation for the complicated case is written as

$$\begin{aligned} & \ddot{\phi} + 2\zeta\omega_{\phi}\dot{\phi} + \delta\omega_{\phi}\dot{\phi}^3 + \\ & \omega_{\phi}^2[1 - h\cos(\omega_e t)]\left(1 - \frac{\phi^2}{\pi^2}\right)\phi + \\ & - c_3\omega_{\phi}^2\phi^3 - c_5\omega_{\phi}^2\phi^5 - \dots - c_{2n+1}\omega_{\phi}^2\phi^{2n+1} = 0. \end{aligned} \quad (2.9)$$

Here the coefficients c_{2n+1} are determined as the least square fit from the \overline{GZ} and describe the restoring moment.

2.2 Roll motions

Ship motion on the sea is analogous to a floating rigid body, so a second-order differential equation can be used to predict the movement. The full equation of motion is [56]

$$A\ddot{x} + B(\dot{x})\dot{x} + C(x)x = F(t), \quad (2.10)$$

where A is mass matrix, B is damping matrix, C is the stiffness matrix and F is external moments. The system is composed of a total of 6 equations, of which 3 are translational and 3 are rotational motions. The system is coupled by nature, but an uncoupled roll equation in the head or following waves can be approximated as [66]

$$I_x \frac{d^2\phi}{dt^2} + b \frac{d\phi}{dt} + \Delta(\overline{GM} + \delta\overline{GM})\phi = 0, \quad (2.11)$$

where ϕ is the roll angle, I_x is mass moment of inertia around the x -axis, b is a damping coefficient, Δ is displacement of hull, \overline{GM} is metacenter height in calm water and $\delta\overline{GM}$ is variation of metacentric height due to wave. For regular waves, this can be approximated as

$$\delta\overline{GM} = \overline{GM}c\cos(\omega t), \quad (2.12)$$

where c is fractional variation of \overline{GM} and ω is the wave encounter frequency. To study stability more using mathematical tools, equation (2.11) can be simplified by writing it in a dimensionless form. By noting that the natural frequency $\omega_n^2 = \Delta\overline{GM}/I_x$, the equation can be written as

$$\frac{d^2\phi}{dt^2} + \frac{b}{I_x} \frac{d\phi}{dt} + (\omega_n^2 + c\omega_n^2 \cos(\omega t))\phi = 0, \quad (2.13)$$

The damping term is approximated to be linear as $b/I_x = 2\alpha$, where α is constant. To write this in a dimensionless format, we write ζ as an arbitrary dimensionless constant and define the relation

$$\frac{b}{I_x} = 2\alpha = 2\zeta\omega_n. \quad (2.14)$$

Finally, after changing variable $\tau = \omega t$, we obtain a dimensionless format of the equation of motion as

$$\ddot{\phi}(\tau) + 2\zeta\sqrt{\delta}\dot{\phi}(\tau) + (\delta + \epsilon \cos \tau)\phi(\tau) = 0, \quad (2.15)$$

where $\delta = \omega_n^2/\omega^2$ and $\epsilon = c\delta$. If the damping term were discarded by setting the $\zeta = 0$, we would get

$$\ddot{\phi}(\tau) + (\delta + \epsilon \cos \tau)\phi(\tau) = 0, \quad (2.16)$$

which is also known as the *Mathieu equation*. Equation (2.16) can be derived in other applications such as a forced pendulum or a spring system. To understand the behaviour of the solution that satisfies equation (2.16), the system is solved numerically using Matlab's ode45-solver with an initial roll angle of 5 degrees. By setting $\delta = 0.3$ and $\epsilon = 0.05$, we get the solution presented in Figure 2.6.

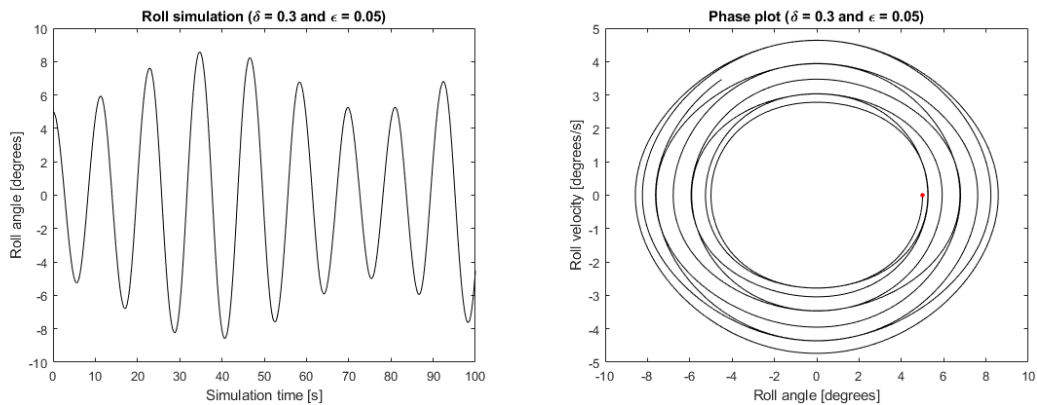


Figure 2.6. Roll simulation

The leftmost plot represents the ship's roll angle with respect to time and the rightmost plot is the so-called phase plot. This is a visualization technique where the solution in each timestep is plotted on a phase plane. In this case, the phase plane is the roll angle and the roll velocity, which itself is more of an abstract concept. However, by following

the trajectory of the plot, we can see how the solution evolves within the simulation and determine whether the solution would converge to a singular point, diverge to infinity or oscillate in a bounded region. Note that it is only a visualization technique and should not be used as a convergence criterion.

The solution in Figure 2.6 appears to be stable, as the roll angle oscillates in a bounded region which is supported by the phase plot. Thus, we can relatively confidently state that the result for these parameters is stable. By choosing $\delta = 0.2$ and $\epsilon = 0.15$, we get a solution presented in Figure 2.7.

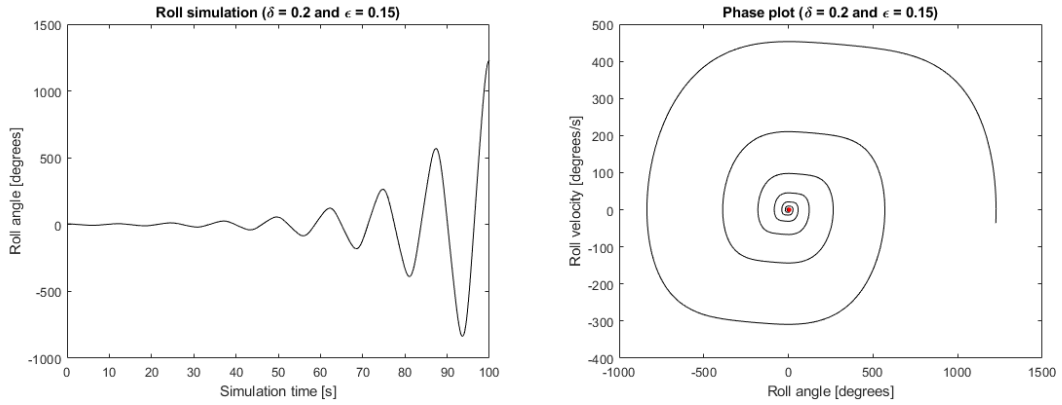


Figure 2.7. Roll simulation

Here the solution diverges in an oscillatory manner. Thus, we can conclude that for some parameters the system is stable while for others it is unstable.

2.2.1 Floquet's theorem

To study the stability of (2.16) for a pair (δ, ϵ) , a numerical simulation could be run to determine whether the solution would converge or diverge. However, it is rather difficult to determine the criteria of divergence, especially when the system is periodic. The time to determine that the system is diverging might take a long time, and the choice of initial conditions is no trivial task either. A more interesting approach would be to take aid from *Floquet's theorem*, which studies the behaviour of periodic systems. This subsection is based on books [11], [32] and [61]. Also, good supplementary sources recommended are theses [2] and [21] as well as lecture notes [7].

A Floquet system is defined as

$$\frac{d}{dt}x(t) = A(t)x(t), \quad (2.17)$$

where $A(t)$ is a T -periodic function $A(t) = A(t + T)$ for all $t \in \mathbb{R}$ and $x(t) \in \mathbb{R}^{n \times 1}$ is a solution to the system. First, we will define the fundamental solution matrix as follows.

Definition 2.1 (Fundamental solution matrix). ([61, p. 19]) Let $x_1(t), x_2(t), \dots, x_n(t)$ be linearly independent solutions of (2.17). The fundamental solution matrix $\Phi(t)$ is defined as

$$\Phi(t) = \begin{bmatrix} x_1(t) & x_2(t) & \dots & x_n(t) \end{bmatrix}. \quad (2.18)$$

The fundamental solution matrix is also referred to as a fundamental matrix or solution matrix in literature. One motivation for this is that the general solution can be presented using the fundamental solution matrix. The general solution for system (2.17) has a form

$$x(t) = c_1 x_1(t) + c_2 x_2(t) + \dots + c_n x_n(t), \quad (2.19)$$

where $x_i(t)$ are linearly independent solutions to (2.17), c_i are constant coefficients and $i = 1, 2, \dots, n$. By using the fundamental solution matrix, we present the general solution as

$$x(t) = \Phi(t)\mathbf{c}, \quad (2.20)$$

where $\mathbf{c} = \begin{bmatrix} c_1 & c_2 & \dots & c_n \end{bmatrix}^T$. Plugging (2.20) to the original system (2.17), we get

$$\begin{aligned} \frac{d}{dt}x(t) &= Ax(t) \\ \implies \frac{d}{dt}(\Phi(t)\mathbf{c}) &= A(t)\Phi(t)\mathbf{c} \\ \implies \left(\frac{d}{dt}\Phi(t)\right)\mathbf{c} + \underbrace{\Phi\left(\frac{d}{dt}\mathbf{c}\right)}_{=0} &= A(t)\Phi(t)\mathbf{c} \\ \implies \frac{d}{dt}\Phi(t) &= A(t)\Phi(t). \end{aligned} \quad (2.21)$$

Thus, the fundamental solution matrix satisfies the original system and therefore can be used to study multiple linearly independent solutions simultaneously. We will later also discover, that it has useful propriety in the stability of periodic systems. A few important lemmas arise from the definition.

Lemma 2.1. *Let $\Phi(t)$ be a fundamental solution matrix for system (2.17). Thus, $\Phi(t+T)$ is a fundamental solution matrix as well.*

Proof. As the $\Phi(t)$ is solution matrix, it satisfies the equation

$$\frac{d}{dt}\Phi(t) = A(t)\Phi(t). \quad (2.22)$$

Let $Z(t) := \Phi(t + T)$. We can directly calculate that

$$\frac{d}{dt}Z(t) = \frac{d}{dt}\Phi(t + T) = A(t + T)\Phi(t + T) = A(t)Z(t). \quad (2.23)$$

Finally, to see that $Z(t)$ is non-singular i.e. its columns are linearly independent, we have that $\det Z(t) = \det \Phi(t + T) \neq 0$ since $\Phi(t)$ is non-singular for all $t \in \mathbb{R}$ and $t + T \in \mathbb{R}$. Thus, we conclude that $Z(t)$ is a fundamental solution matrix. \square

The $\Phi(t + T)$ can be thought of as the set of solutions that have evolved into the next period. Thus, it would be useful to find a mapping of solutions from t to $t + T$. Luckily, such a matrix exists, which is shown next in a general form.

Lemma 2.2. *Let $\Phi(t)$ be a fundamental solution matrix for system (2.17). Now, $\tilde{\Phi}(t)$ is also a fundamental solution matrix if and only if it has a form $\tilde{\Phi}(t) = \Phi(t)C_0$, where C_0 is a non-singular constant matrix.*

Proof. " \Leftarrow ": Let $\Phi(t)$ be a fundamental solution matrix for (2.17) and let $\tilde{\Phi}(t) = \Phi(t)C_0$ where C_0 is a non-singular constant matrix. To show that $\tilde{\Phi}(t)$ is a fundamental solution matrix, consider

$$\frac{d}{dt}\tilde{\Phi}(t) = \frac{d}{dt}(\Phi(t)C_0) = \frac{d}{dt}(\Phi(t))C_0 = A(t)\Phi(t)C_0 = A(t)\tilde{\Phi}(t). \quad (2.24)$$

Furthermore, since $\Phi(t)$ and C_0 are both non-singular, $\tilde{\Phi}(t)$ must also be non-singular. Thus, $\tilde{\Phi}(t)$ is fundamental solution matrix.

" \Rightarrow ": Let $\Phi(t)$ be a fundamental solution matrix for system (2.17) and define $C(t) := \Phi^{-1}(t)\tilde{\Phi}(t)$, where $\tilde{\Phi}(t)$ is an arbitrary fundamental solution matrix. Since $\Phi(t)$ and $\tilde{\Phi}(t)$ are solution matrices, $C(t)$ is non-singular and we can write that

$$\tilde{\Phi}(t) = \Phi(t)C(t). \quad (2.25)$$

Now, define a constant matrix $C_0 := C(t_0)$ at a fixed time t_0 , and a matrix $Z(t) = \Phi(t)C_0$ for all $t \in \mathbb{R}$. Since

$$\frac{d}{dt}Z(t) = \frac{d}{dt}(\Phi(t)C_0) = \frac{d}{dt}(\Phi(t))C_0 = A(t)\Phi(t)C_0 = A(t)Z(t), \quad (2.26)$$

the $Z(t)$ is a solution matrix as well. Thus, we have two fundamental solutions

$$Z(t) = \Phi(t)C_0 \quad \text{and} \quad \tilde{\Phi}(t) = \Phi(t)C(t). \quad (2.27)$$

We see that when $t = t_0$, we get that

$$\tilde{\Phi}(t_0) = \Phi(t_0)C(t_0) = \Phi(t_0)C_0 = Z(t_0), \quad (2.28)$$

i.e. the systems equal at the value t_0 . Since the solution for the differential system with unique initial values must also be unique, the matrices in equation (2.27) must equal i.e. $C(t) = C_0$. Therefore, the matrix $C(t)$ is constant for all $t \in \mathbb{R}$, which concludes the proof. \square

Note that since the matrix C_0 is non-singular, we can define the initial solution $\Phi(0) = I$ by defining $\tilde{\Phi}(0) = C_0^{-1}$. The Lemma 2.2 also allows us to define the *monodromy matrix* as follows.

Definition 2.2 (Monodromy matrix). ([11, p. 191]) Let $\Phi(t)$ be a fundamental solution matrix for system (2.17). The monodromy matrix C is defined as

$$C = \Phi^{-1}(0)\Phi(T). \quad (2.29)$$

Finally, we define $C(t) = \Phi^{-1}(t)\Phi(t+T)$. Since C is constant due to Lemmas 2.1 and 2.2, we can calculate it at any time t . Thus, it equates to the monodromy matrix (2.29) and we get a relation

$$\Phi(t+T) = \Phi^{-1}(0)\Phi(T)\Phi(t). \quad (2.30)$$

Furthermore, by choosing the initial condition as $\Phi(0) = I$, this simplifies to

$$\Phi(t+T) = \Phi(T)\Phi(t). \quad (2.31)$$

This indicates that the solution might be determined by what is the solution at the first period $t = T$. We will later see that the eigenvalues of $\Phi(T)$ can be used to determine the stability of the system.

Theorem 2.1 (Floquet's theorem). ([11, p. 189]) Let $\Phi(t)$ be a fundamental matrix of (2.17) and let $\Phi(0) = I$. There exists an expression

$$\Phi(t) = P(t)e^{Bt}, \quad (2.32)$$

where $P(t)$ is T -periodic matrix and B is a non-singular constant matrix.

Proof. Let $\Phi(t)$ be a solution matrix for system (2.17) with $\Phi(0) = I$. Now the monodromy matrix has form

$$C = \underbrace{\Phi^{-1}(0)}_{=I} \Phi(T) = \Phi(T). \quad (2.33)$$

Also there exists a matrix B such that $e^{BT} = C$. Now define $P(t) := \Phi(t)e^{-Bt}$ for $t \in \mathbb{R}$. Since matrix exponential is invertible, we have that $\Phi(t) = P(t)e^{Bt}$. Thus it remains to show that $P(t)$ is periodic. Consider

$$\begin{aligned} P(t+T) &= \Phi(t+T)e^{-B(t+T)} \\ &= \Phi(t+T)e^{-Bt-BT} \end{aligned} \quad (2.34)$$

Now since the matrices $-Bt$ and $-BT$ commute, we can separate the matrix exponent as $e^{-Bt-BT} = e^{-Bt}e^{-BT}$ [33, p. 35]. Thus, the equation modifies as

$$\begin{aligned} \Phi(t+T)e^{-Bt-BT} &= \Phi(t)Ce^{-Bt}e^{-BT} \\ &= \Phi(t)e^{BT}e^{-Bt}e^{-BT} \\ &= \Phi(t)e^{-Bt} \\ &= P(t), \end{aligned} \quad (2.35)$$

which concludes the proof. □

Floquet's theorem is useful as it gives an expression for the solution matrix to study the stability of the solution just by choosing appropriate initial conditions. The periodic function $P(t)$ can be shown to be bounded, so the magnitude of the result depends solely on the matrix power e^{Bt} . Also, since B is a matrix, it makes more sense to study its eigenvalues to determine the criteria for stability. From Floquet's theorem, we can finally create a criterion to determine the stability criteria.

Theorem 2.2. *Let $\Phi(t)$ be the solution matrix of (2.17) with the initial solution $\Phi(0) = I$ and let $\lambda_{C,i}$ be the eigenvalues of the monodromy matrix C defined in Definition 2.2. The stability of the system is determined as follows:*

- *If $|\lambda_{C,i}| < 1$ for all $i \in 1, 2, \dots, n$, then the system is asymptotically stable.*

- If $|\lambda_{C,i}| \leq 1$ for all $i \in 1, 2, \dots, n$ and $|\lambda_{C,j}| = 1$ for some $j \in N$, then the system is stable.
- Otherwise the system is unstable.

Proof. Let $\Phi(t)$ be the solution matrix of system (2.17) and $\Phi(0) = I$. Floquet's theorem gives us an expression for the solution matrix as

$$\Phi(t) = P(t)e^{Bt}, \quad (2.36)$$

where $P(t)$ is a periodic matrix and B is constant matrix. Let (possibly complex) λ_B be the eigenvalue of matrix B and z its corresponding eigenvector. Now, since $Bz = \lambda_B z$, then we can write it also in the matrix exponential as $e^{Bt}z = e^{\lambda_B t}z$ [33, p. 35]. In the proof of Theorem 2.2.1 we defined $P(t) = \Phi(t)e^{-Bt}$, so $P(t)$ is continuous and as $P(t)$ is also periodic, it must be bounded. Therefore, we can evaluate

$$\|P(t)\| \leq \sup_{t \in \mathbb{R}} |P(t)| = \sup_{t \in [0, T]} |P(t)|, \quad (2.37)$$

and note $Q = \sup_{t \in [0, T]} |P(t)|$. Now, we can evaluate that

$$\begin{aligned} \|\Phi(t)z\| &= \|P(t)e^{Bt}z\| \\ &\leq \|P(t)\| \|e^{Bt}z\| \\ &\leq Q \|e^{\lambda_B t}z\| \\ &= Q |e^{\lambda_B t}| \|z\| \\ &= Q e^{\Re(\lambda_B)t} \|z\|. \end{aligned} \quad (2.38)$$

This implies that the boundedness of the solution $\Phi(t)$ depends on the real part of the eigenvalue λ_B . Finally, we want to find a relation between λ_B and the monodromy matrix. Let C be the monodromy matrix defined in Definition 2.2 and λ_C its eigenvalue. Like in the proof of Floquet's theorem, we have that $C = e^{BT}$. Thus, we can write that [32]

$$\lambda_C = e^{\lambda_B T}, \quad (2.39)$$

which further implies that

$$|\lambda_C| = e^{\Re(\lambda_B)T}. \quad (2.40)$$

As T must be positive, this expression gives the following conditions:

- $|\lambda_C| < 1$ if and only if $\Re(\lambda_B) < 0$
- $|\lambda_C| > 1$ if and only if $\Re(\lambda_B) > 0$
- $|\lambda_C| = 1$ if and only if $\Re(\lambda_B) = 0$

From this, we can draw the conclusions stated in the theorem. \square

One can employ the aforementioned theorem to ascertain the stability of a periodic system by computing the eigenvalues of its monodromy matrix $\Phi(T)$. The following subsections demonstrate this by utilizing numerical methods for the derived roll equations.

2.2.2 Stability of undamped Mathieu equation

The undamped Mathieu equation (2.16) is periodic with period $T = 2\pi$. Defining $x_1 = \phi$ and $x_2 = \dot{\phi}$, the corresponding Floquet's system has form

$$\dot{x}(\tau) = A(\tau)x(\tau), \quad (2.41)$$

where

$$x(\tau) = \begin{bmatrix} x_1(\tau) \\ x_2(\tau) \end{bmatrix} \quad \text{and} \quad A(\tau) = \begin{bmatrix} 0 & 1 \\ -\delta - \epsilon \cos \tau & 0 \end{bmatrix}.$$

To study the stability of the system, we need to calculate its monodromy matrices' eigenvalues. Defining initial solution of $\Phi(0) = I$ and using Definition 2.2, we get that

$$C = \Phi^{-1}(0)\Phi(T) = \Phi(T). \quad (2.42)$$

Now given the λ_C is the eigenvalue of the matrix, we can calculate them from the relation

$$\det(C - \lambda_C I) = 0 \implies \lambda_C^2 - \text{tr } C \lambda_C + \det C = 0 \quad (2.43)$$

To calculate the $\det C$, we can define $W(\tau) := \det \Phi(\tau)$. Note that when $\tau = T$, then $W(T) = \det C$. First, calculate

$$W(0) = \det \Phi(0) = \det I = 1. \quad (2.44)$$

Finally, to show that $W(\tau)$ is constant with any choice of τ , take the derivative of it with respect to time.

$$\begin{aligned}
\frac{d}{d\tau} W(\tau) &= \frac{d}{d\tau} \det \Phi(\tau) \\
&= \frac{d}{d\tau} (x_{11}x_{22} - x_{12}x_{21}) \\
&= \dot{x}_{11}x_{22} + x_{11}\dot{x}_{22} - \dot{x}_{21}x_{12} - x_{21}\dot{x}_{12}
\end{aligned} \tag{2.45}$$

For the individual terms, we can write the (2.41) as

$$\begin{aligned}
\frac{d}{d\tau} \Phi(\tau) &= A(\tau)\Phi(\tau) \\
\frac{d}{d\tau} \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\delta - \epsilon \cos \tau & 0 \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} \\
&= \begin{bmatrix} x_{21} & x_{22} \\ (-\delta - \epsilon \cos \tau)x_{11} & (-\delta - \epsilon \cos \tau)x_{12} \end{bmatrix},
\end{aligned} \tag{2.46}$$

which implies that

$$\begin{cases} \dot{x}_{11}(\tau) = x_{21}(\tau) \\ \dot{x}_{12}(\tau) = x_{22}(\tau) \\ \dot{x}_{21}(\tau) = (-\delta - \epsilon \cos \tau)x_{11}(\tau) \\ \dot{x}_{22}(\tau) = (-\delta - \epsilon \cos \tau)x_{12}(\tau) \end{cases}$$

After plugging these to (2.45) we get that $d/d\tau W(\tau) = 0$ i.e. the determinant of the monodromy matrix is constant. Therefore, the determinant can be calculated at any $\tau \in \mathbb{R}$, so we get

$$\det C = W(T) = W(0) = 1 \tag{2.47}$$

and the characteristic polynomial takes the form

$$\lambda_C^2 - \text{tr } C \lambda_C + 1 = 0. \tag{2.48}$$

We can't directly calculate the trace of the monodromy matrix as we don't know the fundamental solution matrix. However, by solving the characteristic polynomial, we get eigenvalues

$$\lambda_{C,1}, \lambda_{C,2} = \frac{1}{2} \left(\text{tr } C \pm \sqrt{(\text{tr } C)^2 - 4} \right). \quad (2.49)$$

This condition gives different cases that determine the magnitude of the eigenvalues.

Case 1 $|\text{tr } C| < 2$:

The term inside the square root becomes negative and the eigenvalues are complex conjugate pairs where

$$\Im(\lambda_{C,\pm}) = \sqrt{1 - \frac{(\text{tr } C)^2}{4}} \quad \text{and} \quad \Re(\lambda_{C,\pm}) = \frac{\text{tr } C}{2} \quad (2.50)$$

Now, we can directly calculate that

$$\begin{aligned} \lambda_{C,+} \lambda_{C,-} &= \left(\frac{\text{tr } C}{2} + i \sqrt{1 - \frac{(\text{tr } C)^2}{4}} \right) \left(\frac{\text{tr } C}{2} - i \sqrt{1 - \frac{(\text{tr } C)^2}{4}} \right) \\ &= \frac{(\text{tr } C)^2}{4} + \left(1 - \frac{(\text{tr } C)^2}{4} \right) \\ &= 1, \end{aligned} \quad (2.51)$$

and since the eigenvalues are complex conjugate pair, the eigenvalues must lie on a complex unit circle. Thus $|\lambda_{C,+}| = |\lambda_{C,-}| = 1$, and the system is **stable but not asymptotically stable**.

Case 2 $|\text{tr } C| > 2$:

Now the term inside the square root is positive and thus both $\lambda_{C,1}$ and $\lambda_{C,2}$ are real. Also now since either

$$|\lambda_{C,1}| > 1 \quad \text{or} \quad |\lambda_{C,2}| > 1, \quad (2.52)$$

the solution is **unstable**.

Case 3 $|\text{tr } C| = 2$:

In this case, the square root term becomes 0 and the eigenvalues are real-valued numbers. The solution is also periodic. If $\text{tr } C = 2$ then $\lambda_{C,+} = \lambda_{C,-} = 1$. In this case, the solution is **stable** with the period of T . If $\text{tr } C = -2$, then $\lambda_{C,+} = \lambda_{C,-} = -1$. In this case, the solution is also **stable** with the period of $2T$.

Now, using this criterion, the stability of the system can be determined either numerically or by other means. Numerically it can be done by

1. Choose δ and ϵ .
2. Solve numerically $x_1(T)$ from initial condition $[1, 0]^T$.
3. Solve numerically $x_2(T)$ from initial condition $[0, 1]^T$.
4. Construct monodromy matrix $C = \Phi^{-1}(0)\Phi(T) = \Phi(T) = [x_1, x_2]$.
5. If $|\text{tr } C| = |\text{tr } \Phi(T)| \leq 2$, the solution is stable. Otherwise, it is unstable.

A Matlab code was constructed to run the algorithm above for different combinations of δ and ϵ values. It took roughly 13 minutes to calculate the diagram with a relatively powerful computer. Figure 2.8 shows the stable and unstable regions.

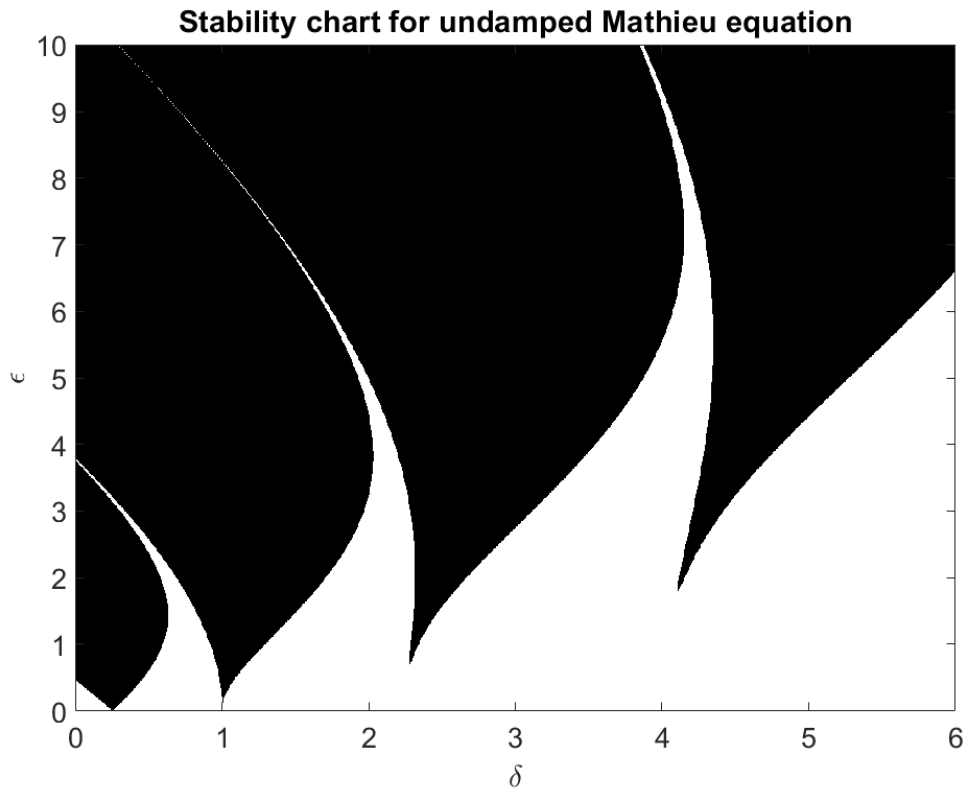


Figure 2.8. Stability chart for undamped Mathieu equation. The black region represents the unstable solution and the white region represents the stable solution

The diagram is also known as the Ince-Strutt diagram. There are also more analytical ways to find the regions for example by expanding the solution x_i to an infinite complex Fourier series, as done in [32]. However, for this thesis, the numerical solution suffices.

For roll equation, the parameters were defined as $\delta = \omega_n^2/\omega^2$ and $\epsilon = c\delta$, where ω_n is ships natural roll frequency, ω is the wave encounter frequency and c is fractional difference between \overline{GM} and \overline{GM}_{max} . These terms are defined this way just so the roll equation (2.11) could be turned into dimensionless. Thus, they might not have physical meaning and rather is a tool to keep the equations tidy. We see that δ is a fraction of natural roll frequency and incoming wave frequency. The variable ϵ on the other hand is

a function of δ and the fraction of \overline{GM} variation. Since \overline{GM} variation is greater in bigger waves, the variable ϵ is a function of wave height.

Now intuition tells that a bigger wave causes more instability, so it is interesting to see if the roll motions become unstable when the wave height is small. As per the last paragraph, this can be seen in when ϵ is the smallest. From Figure 2.8, the first instability region occurs near the δ value of $1/4$. This represents a case, where

$$\frac{\omega_n^2}{\omega^2} = \frac{1}{4} \implies \omega = 2\omega_n, \quad (2.53)$$

i.e. the encounter frequency is twice the natural roll frequency. Next instability closes to $\delta = 1$, where $\omega = \omega_n$. This region however is much more stable than previously. This trend goes on to all other "fangs" seen in Figure 2.8. Indeed we can say that the most unstable results occur when $\omega = 2\omega_n$, which is the observed worst case for parametric rolling.

A practical benefit of this is that if the ship could be assumed to have no damping, then just by knowing ω_n , ω , and c we could determine whether the ship would be vulnerable to parametric rolling without having to do any simulations. However, since every ship has some damping, the Ince-Strutt diagram cannot be utilized in actual shipbuilding and serves merely an academic purpose.

2.2.3 Stability of damped Mathieu equation

Subsection 2.2.2 discusses the stability of the undamped Mathieu equation, although in reality, there is no system that is completely devoid of damping. To include damping, we return to equation (2.15) and define $x_1(\tau) = \phi(\tau)$ and $x_2(\tau) = \dot{\phi}(\tau)$. The Floquet's system is now

$$\dot{x}(\tau) = A(\tau)x(\tau), \quad (2.54)$$

where

$$x(\tau) = \begin{bmatrix} x_1(\tau) \\ x_2(\tau) \end{bmatrix} \quad \text{and} \quad A(\tau) = \begin{bmatrix} 0 & 1 \\ -\delta - \epsilon \cos \tau & -2\sqrt{\zeta}\delta \end{bmatrix}.$$

Unlike previously, the determinant of does not equal to one, so the characteristic polynomial of the monodromy matrix is different. The product of the eigenvalues could be determined from Liouville's formula, but for this analysis, it suffices to calculate the eigenvalues numerically. Since the system has damping included, the solution is expected to

be mostly asymptotically stable. However, the stable but not asymptotically stable region (later bounded stable region) can be resolved from those eigenvalues, that have norm close to one. Here, the tolerance value is chosen as $5.0e - 02$ to exaggerate the region size, as choosing smaller tolerance would lead imperceptible region size. Plotting the stability chart for $\zeta = 0.15$ yields

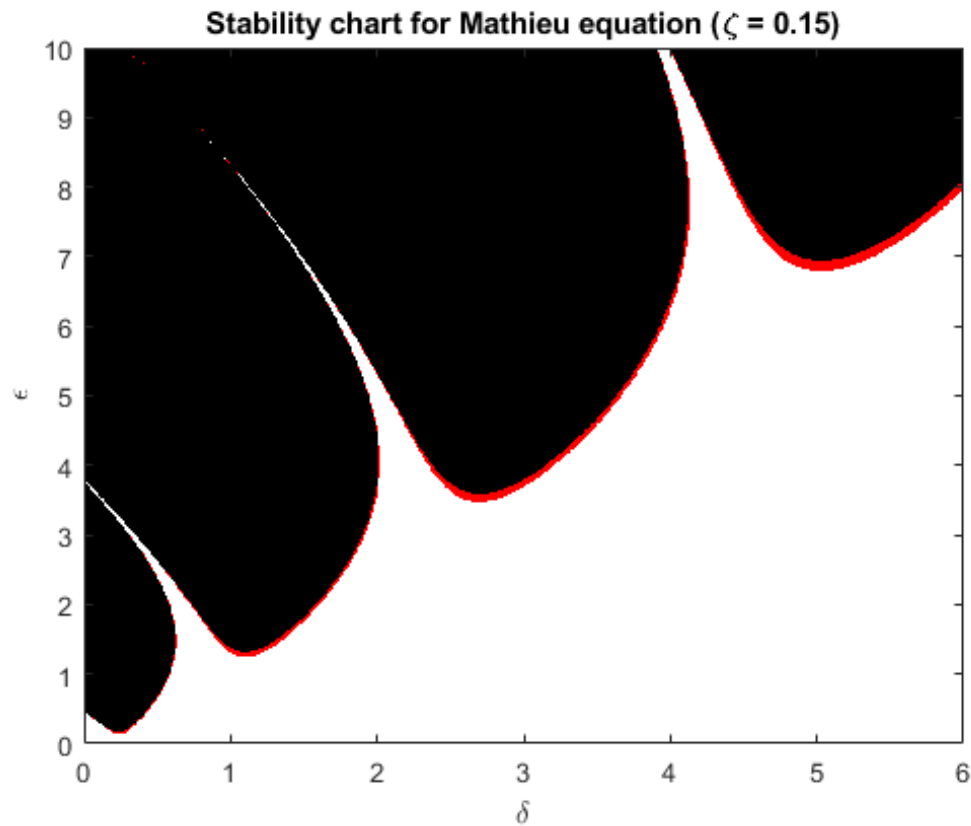


Figure 2.9. Stability chart for damped Mathieu equation. The black region represents unstable, the white region represents asymptotically stable, and the red region represents bounded stable region

Here we see that the shape of the fangs is similar to the undamped version. Also, the stable region is now much bigger, which is to be expected. If we set $\zeta = 0.05$, we get a figure that is somewhat between the previous two.

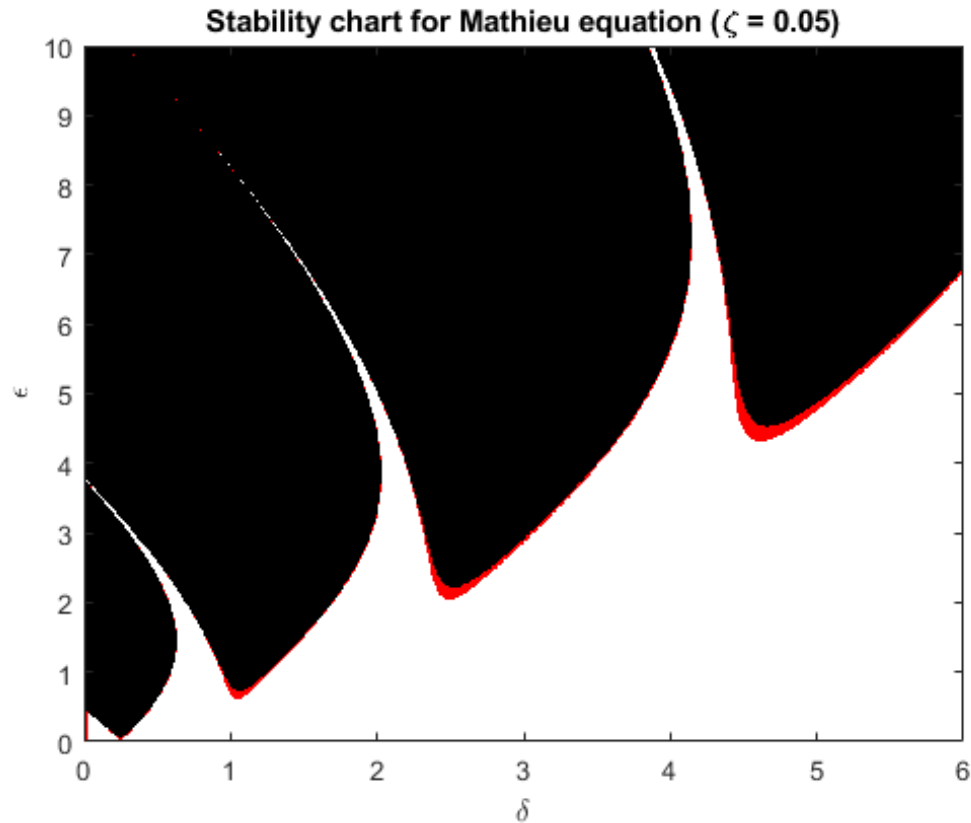


Figure 2.10. Stability chart for damped Mathieu equation. The black region represents unstable, the white region represents asymptotically stable, and the red region represents bounded stable region

The bounded stable region appears near the boundary of unstable and asymptotically stable regions, mostly near the resonance areas. This implies, that while the system would be stable, the rolling is less likely to completely dampen out when the conditions are near the resonance area. However, it also should be noted, that the red regions would be smaller if the tolerance value would be smaller. Thus, this system would practically be only asymptotically stable or unstable i.e. the roll amplitude would either diverge to infinity or converge to zero. In reality, the ship could also reach a maximum amplitude without capsizing. This introduces the importance of nonlinear terms for parametric rolling, which were neglected in this system. Therefore, the presented stability charts serve more as a mathematical explanation for when the parametric rolling would stay stable in the simplified case.

2.3 Water wave theory

Modeling a ship response in waves wouldn't be possible if waves themselves are modeled improperly. The concept of waves is familiar in many different fields, such as optics and signal processing. Although the surface waves that occur on the water bodies (later simply water surface waves) have similar features, they often are not in a perfect sinusoidal

form as they are composed of dense fluid (in this case water) and therefore the effect of gravity cannot be neglected. Thus, a theoretical approach is necessary to derive proper equations for linear and nonlinear water surface waves.

The typically used metrics are presented in Figure 2.11.

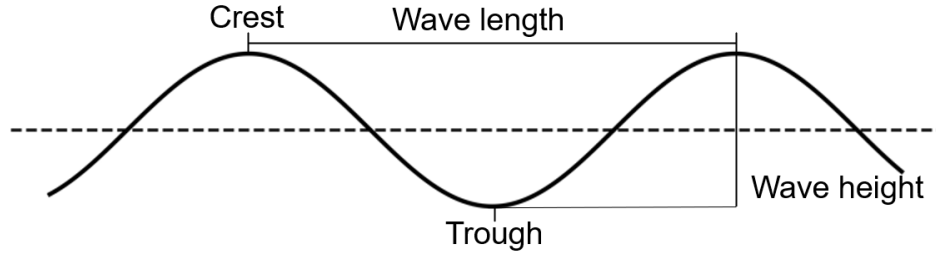


Figure 2.11. Wave characteristic definitions. The dotted line represents the calm sea level

Other important parameters are *wave frequency* f which is the number of wave crests passing a single point each second. Often in literature *angular wave frequency* ω or wave period T is used instead but they all are related as

$$f = \frac{\omega}{2\pi} = \frac{1}{T}. \quad (2.55)$$

The height of the wave H is described using the amplitude of the wave a , which is defined as $H = 2a$, and the length of the wave λ describes the wave number $k = 2\pi/\lambda$. The wave steepness s is defined as a fraction of height and length i.e. $s = H/\lambda$.

This section is based on books [28], [38] and [27], and Subsection 2.3.2 references [59]. Readers are also recommended to [8] for supplementary material.

2.3.1 Airy wave

Airy wave is a simple linear wave theory proposed in 1845 by George Biddell Airy in his work *Tides and Waves* [1] [18]. When deriving an expression for waves, the goal is to find some function η to express the surface height. First, the fluid should satisfy the conservation of mass given as

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0, \quad (2.56)$$

and the conservation of momentum

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + g_i. \quad (2.57)$$

We are going to derive these equations later in Section 3.1. Note that the stress term is neglected as the fluid is assumed to be ideal i.e. incompressible and with no viscosity. To solve the system, boundary conditions are required for the initial value problem. We can assume the y -axis to be symmetric, which then would correspond to an infinitely wide wave. The problem is reduced to two dimensions, where the x -axis tells the direction of the wave and gravity affects to the direction of the negative z -axis. Assuming that the wave is periodic, only z -directional boundary conditions are required.

For the bottom of the domain, the vertical velocity should be zero i.e. the domain is deep enough so that the effect of the surface doesn't have an effect. Given that h is the height of the domain, we have

$$w = 0, \quad \text{when } z = -h. \quad (2.58)$$

On the surface, the vertical velocity should equal to the rate of change of the surface level. This ensures that no fluid crosses the free surface boundary and the wave stays in its form. Mathematically speaking we have

$$w = \frac{\partial \eta}{\partial t} + u \frac{\partial \eta}{\partial x} + v \frac{\partial \eta}{\partial y}, \quad \text{when } z = 0. \quad (2.59)$$

where η represents the free surface height. Neglecting y -direction and assuming that the wave height is relatively small with respect to its length, we can approximate that $\partial \eta / \partial x \approx 0$. Now the boundary condition has a linear form

$$w = \frac{\partial \eta}{\partial t}, \quad \text{when } z = 0. \quad (2.60)$$

Finally, the pressure at the free surface should equal to the atmospheric pressure, so it will be only static. The dynamic pressure thus is

$$p = 0, \quad \text{when } z = \eta. \quad (2.61)$$

Finding the solution to this conventionally is hard. However, by defining a velocity potential ξ such that it satisfies

$$u = \frac{\partial \xi}{\partial x}, \quad v = \frac{\partial \xi}{\partial y} \quad \text{and} \quad w = \frac{\partial \xi}{\partial z}, \quad (2.62)$$

the problem is easier as ξ is scalar, unlike the velocity. In vector notation, this is written as $\mathbf{u} = \nabla \xi$. Since the curl of a gradient is zero for any function [35, p. 74], we obtain

$$\nabla \times \nabla \xi = 0 \implies \nabla \times \mathbf{u} = 0, \quad (2.63)$$

i.e. the velocity must satisfy the irrotationality constraint, where the fluid particles don't rotate around their own axes. For our problem, this approximation is valid since fluid rotationality often arises from turbulence and here the velocities are relatively small and no surfaces exist. Note that usually in literature, velocity potential is noted as ϕ , but here ξ was chosen to avoid confusion with roll angle. Modifying the mass equation with velocity potential yields

$$\frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 \xi}{\partial z^2} = 0, \quad (2.64)$$

which is Laplace's equation. The first two boundary conditions take the form

$$\frac{\partial \xi}{\partial z} = \begin{cases} 0, & \text{when } z = -h \\ \frac{\partial \eta}{\partial t}, & \text{when } z = 0 \end{cases} \quad (2.65)$$

The pressure boundary condition requires an equation for pressure, where we can take advantage of the conservation of momentum. Presenting velocities in potential form yields

$$\frac{\partial}{\partial t} \frac{\partial \xi}{\partial x_i} + \frac{\partial \xi}{\partial x_j} \frac{\partial}{\partial x_j} \frac{\partial \xi}{\partial x_i} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + g_i. \quad (2.66)$$

Changing the derivative order of the first term and writing the second term using the dot product rule [35, p. 74] and the irrotationality constraint (2.63) allows taking the spatial derivative as a common factor. Since gravity affects in the z-axis, we can write that $g_i = 0$ for $i = 1, 2$ and $g_3 = g$. Thus, we write the gravity term in alternative form so that the momentum equation is

$$\frac{\partial}{\partial x_i} \left(\frac{\partial \xi}{\partial t} + \frac{1}{2} \frac{\partial^2 \xi}{\partial x_j^2} + \frac{p}{\rho} + gz \right) = 0. \quad (2.67)$$

As the derivative is zero in all directions, the function inside the parenthesis must compute to some arbitrary constant C . By choosing $C = 0$, we get a modified boundary condition for pressure as

$$\frac{\partial \xi}{\partial t} + \frac{1}{2} \frac{\partial^2 \xi}{\partial x_j^2} + \frac{p}{\rho} + gz = 0. \quad (2.68)$$

This equation is known as *Bernoulli's equation* and is usually presented in a linear form where the nonlinear second term is neglected. Applying the boundary condition (2.61) to

this equation we get

$$\frac{\partial \xi}{\partial t} + g\eta = 0, \quad \text{when } z = 0. \quad (2.69)$$

Now, the problem closes to solving the Laplace equation (2.64) with boundary condition (2.65). One known analytical solution to this system is the surface function

$$\eta = a \sin(\omega t - kx) \quad (2.70)$$

with the velocity potential of

$$\xi = \frac{\omega a \cosh(k[h + z])}{k \sinh(kh)} \cos(\omega t - kx), \quad (2.71)$$

where a is the amplitude, ω is the angular frequency, h is the water depth and k is the wave number. Applying the kinematic boundary condition to this, we can plug the solution above to (2.69) that will yield

$$\omega^2 = gk \tanh(kh) \quad \text{or} \quad \lambda = \frac{gT^2}{2\pi} \tanh\left(\frac{2\pi h}{\lambda}\right) \quad (2.72)$$

where λ is the wavelength and T is the wave period. This is known as the dispersion relation of water surface waves. Given that the water depth is sufficiently large compared to the wavelength, we can approximate the $\tanh(kh) \approx 1$ to further simplify the relation.

2.3.2 Stokes wave

The Airy wave theory is a good approximation of wave theory and is good enough for most engineering problems. However, as the wave height grows, more non-linear terms become significant and the linear approximation will be less valid. The real-world waves do not have the form of a sine function, so to model real-world waves, a more accurate model is required.

One advantage of Airy wave theory is that it is an analytical solution to linear approximation. Utilizing this Sir George Stokes proposed a nonlinear model in 1847, which used the perturbation theory to obtain a higher-degree solution. The idea is to extend the solution into an infinitely long sequence using a small perturbation term s and then solve the next-order approximation with the previous-order solutions. Defining wave steepness as $s = ka$, the surface height and velocity potential are assumed to have the form

$$\begin{aligned}\xi &= s\xi_1 + s^2\xi_2 + s^3\xi_3 + \dots \\ \eta &= s\eta_1 + s^2\eta_2 + s^3\eta_3 + \dots\end{aligned}\tag{2.73}$$

and for example, the second-order form has solutions

$$\begin{aligned}\xi &= s\xi_1 + s^2\xi_2 + \mathcal{O}(s^3) = \xi_l + \xi_q + \mathcal{O}(s^3) \\ \eta &= s\eta_1 + s^2\eta_2 + \mathcal{O}(s^3) = \eta_l + \eta_q + \mathcal{O}(s^3)\end{aligned}\tag{2.74}$$

where subscript l represents a linear solution, q a quadratic solution and $\mathcal{O}(s^3)$ represent terms with the order of three or more and are neglected. These values are yet unknown, but to get the value in some boundary, we can utilize Taylor series expansion as

$$\xi = \xi\Big|_{z=0} + \eta\frac{\partial\xi}{\partial z}\Big|_{z=0} + \frac{1}{2}\eta^2\frac{\partial^2\xi}{\partial z^2}\Big|_{z=0} + \dots\tag{2.75}$$

and after applying (2.73) to this we get

$$\xi = s\xi_1 + s^2\left(\xi_2 + \eta_1\frac{\partial\xi_1}{\partial z}\right) + \mathcal{O}(s^3)\tag{2.76}$$

at boundary $z = 0$. Using the analytically derived Airy solution from Subsection 2.3.1, the problem reduces to finding expressions for ξ_q and η_q . Plugging the velocity potential into Laplace's equation (2.64) yields

$$\begin{aligned}&\frac{\partial^2}{\partial x^2}(\xi_l + \xi_q) + \frac{\partial^2}{\partial z^2}(\xi_l + \xi_q) = 0 \\ \Rightarrow &\underbrace{\frac{\partial^2\xi_l}{\partial x^2} + \frac{\partial^2\xi_l}{\partial z^2}}_{=0} + \frac{\partial^2\xi_q}{\partial x^2} + \frac{\partial^2\xi_q}{\partial z^2} = 0 \\ \Rightarrow &\frac{\partial^2\xi_q}{\partial x^2} + \frac{\partial^2\xi_q}{\partial z^2} = 0.\end{aligned}\tag{2.77}$$

Similarly, the boundary conditions are modified. The bottom boundary condition is simply

$$\frac{\partial\xi_q}{\partial z} = 0, \quad \text{when } z = -h\tag{2.78}$$

for similar reasoning as Laplace's equation. The kinematic and dynamic boundary conditions on the other hand are more difficult due to the nonlinear terms included. In literature,

the conditions are combined as they are both evaluated at the top of the domain at $z = 0$. Taking the total derivative of the (2.69) with nonlinear terms included yields

$$g \frac{d\eta}{dt} = -\frac{d}{dt} \left[\frac{\partial \xi}{\partial t} + \frac{1}{2} \left(\left(\frac{\partial \xi}{\partial x} \right)^2 + \left(\frac{\partial \xi}{\partial z} \right)^2 \right) \right] \quad (2.79)$$

Due to boundary condition (2.59), we can replace the term $d\eta/dt$ with $\partial \xi / \partial z$. Also, after expanding the total derivative on the right-hand side and including only the linear and the quadratic terms, we get the boundary condition on the surface as

$$\frac{\partial^2 \xi}{\partial t^2} + g \frac{\partial \xi}{\partial z} = -\frac{\partial}{\partial t} \left[\left(\frac{\partial \xi}{\partial x} \right)^2 + \left(\frac{\partial \xi}{\partial z} \right)^2 \right], \quad \text{when } z = 0. \quad (2.80)$$

Applying the perturbed solution to this yields the second-order boundary condition

$$\frac{\partial^2 \xi_q}{\partial t^2} + g \frac{\partial \xi_q}{\partial z} = -\eta_l \frac{\partial}{\partial z} \left[\frac{\partial^2 \xi_l}{\partial t^2} + g \frac{\partial \xi_l}{\partial z} \right] - \frac{\partial}{\partial t} \left[\left(\frac{\partial \xi_l}{\partial x} \right)^2 + \left(\frac{\partial \xi_l}{\partial z} \right)^2 \right], \quad (2.81)$$

when $z = 0$. Plugging the linear solution from the previous section yields finally the boundary condition

$$\frac{\partial^2 \xi_q}{\partial t^2} + g \frac{\partial \xi_q}{\partial z} = \frac{3a^2 g k \omega}{\sinh 2kh} \sin 2(kx - \omega t) \quad (2.82)$$

The problem reduces to finding the solution to equation (2.77) with boundary conditions (2.78) and (2.82). Such a solution can be found for example by using the Fourier infinite series decomposition, but this goes too in-depth for this thesis. Thus, the second-order solution has a form [59]

$$\xi_q = -\frac{3a^2 \omega \cosh(2k(z+h))}{8 \sinh^4(kh)} \sin 2(kx - \omega t) \quad (2.83)$$

and

$$\eta_q = -\frac{a^2 k \cosh(kh)}{4 \sinh^3(kh)} (2 + \cosh(2kh)) \cos 2(kx - \omega t) - \frac{a^2 k}{2 \sinh(2kh)} \quad (2.84)$$

and the full second-order solution is found by plugging these into the equation (2.74). Figure 2.12 shows how the second-order solution modifies the linear solution.

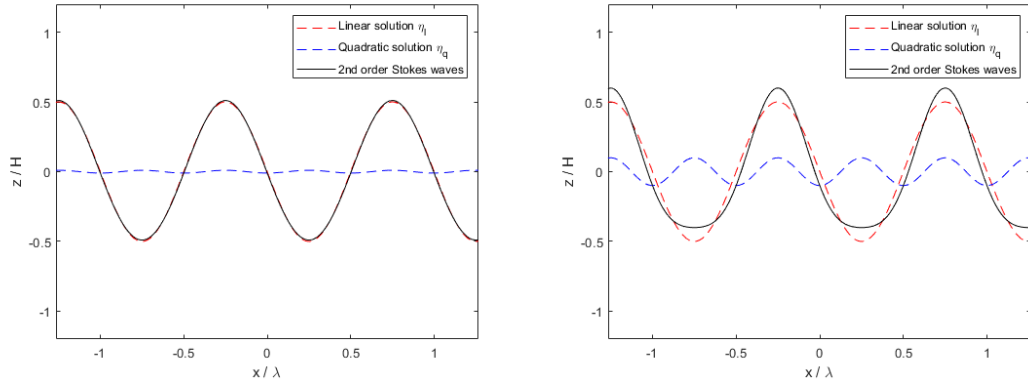


Figure 2.12. Stokes second order wave decomposition. The left figure consists wave with a steepness of 0.0127 and the right figure consists wave with a steepness of 0.1265.

We see that the wave trough becomes sharper, and the crest becomes flatter when the non-linear contribution is added. Figure 2.12 also shows the importance of non-linear parts in steeper waves.

Stokes wave theory can be expanded to any order of accuracy by deriving a new surface boundary condition and expression for ξ and η from the Taylor series and finding the solution to this new system. For example, a third-order solution is derived in [69] and even a fifth-order solution in [20]. In reality, the wave steepness has a limit before the waves start breaking or white-capping. For example, Miche [43] showed that the maximum wave height is

$$H_{\max} \approx 0.14L \tanh\left(\frac{2\pi h}{L}\right). \quad (2.85)$$

This implies that in deep waters, the maximum steepness is

$$s_{\max} = \frac{H_{\max}}{L} \approx 0.14. \quad (2.86)$$

Therefore, for most engineering applications, the linear waves suffice and the nonlinear waves act merely as a theoretical result.

2.3.3 Irregular waves

In the real world, the sea is never composed of regular waves, but rather looks like chaos where waves of varying sizes travel in different directions. Waves in the ocean are created by the surrounding environment, mostly by wind affecting long enough to the surface. Larger waves are created by storms that might have occurred even days before. Modeling a surface height of a sea is by no means a trivial task, as the randomness of the environment creates problems of its own. Thus, a more statistical approach is required.

The subsection is based on the book [28] and the lecture notes [73] are recommended for supplementary material.

Due to the multiple different wave types occurring in the irregular sea, it makes no sense to discuss a singular wave period or height. To characterize an irregular sea state, the concept of average period or significant wave height is used. The characteristic period for sea state can be described with, for example, the peak period T_p . It describes the wave period that occurs for the most probable wave in the sea state. Another characteristic period is the zero-crossing wave period, which describes the average period where the wave crosses the $z = 0$ level. The significant wave height H_s , on the other, hand describes the typical wave height of the sea state. These can be for example the root-mean-square wave height

$$H_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^N H_i^2} \quad (2.87)$$

or the mean of the highest one-third of waves as

$$H_{1/3} = \frac{1}{N/3} \sum_{i=1}^{N/3} H_i, \quad (2.88)$$

where the observed wave heights H_i are sorted in descending order.

A good thing about the linear solution above is that if two singular waves are the solution for the system, then their superposition is also a solution to the system. Thus, a typical irregular wave model superposes multiple Airy waves to produce the wave elevation height [77]. Mathematically speaking the water elevation is defined as

$$\eta = \sum_{i=1}^N a_i \cos(k_i x - \omega_i t + \zeta_i), \quad (2.89)$$

where a_i is the amplitude of wave, ζ_i is a random phase and $N \rightarrow \infty$ in an ideal case. As the waves are related by dispersion, the only unknown parameters for individual wave components are amplitude $a = H/2$ and wave period $T = 2\pi/\omega$ as the phase can be chosen randomly from $[0, 2\pi]$. Choosing all of the component parameters randomly would result in a random seaway, but it wouldn't be ideal as the actual sea always possesses some statistical characteristics. Thus, a more interesting way to define these parameters would be to use the sea energy density spectrum $S(f, \theta)$ which describes the density of a wave component for a frequency band and direction. Using this strategy, the irregular wave model slightly modifies to [23]

$$\eta = \sum_{j=1}^{N_\theta} \sum_{i=1}^{N_f} a(f, \theta) \cos(k_{ij}x - \omega_j t + \zeta_{ij}), \quad (2.90)$$

where the amplitude is defined as

$$a(f, \theta) = \sqrt{2S(f, \theta)\Delta f \Delta \theta}.$$

The energy density spectrum is then decomposed into energy and directional spread components as $S(f, \theta) = S(f)D(\theta)$, where the directional spread must satisfy $\int_{-\pi}^{\pi} D(\theta)d\theta = 1$. The energy spectra can be defined in numerous ways, and for example, the Pierson-Moskowitz spectrum is defined as [23]

$$S(f) = \frac{A}{f^5} \exp\left(-\frac{B}{f^4}\right), \quad (2.91)$$

where

$$A = \frac{5}{16} \frac{H_{1/3}^2}{T_p^4} \quad \text{and} \quad B = \frac{5}{4} \frac{1}{T_p^4}.$$

By sampling the spectrum, each wave component can be extracted to create a specific sea state. The IMO suggests using the Bretschneider wave spectrum similar to the previous one but with different coefficients. The spectrum is defined for zero crossing wave period T_z and significant wave height H_s as

$$S(\omega) = \frac{H_s^2}{4\pi} \left(\frac{2\pi}{T_z}\right)^4 \omega^{-5} \exp\left(-\frac{1}{\pi} \left(\frac{2\pi}{T_z}\right)^4 \omega^{-4}\right). \quad (2.92)$$

To compare these two spectra, the domain needs to be the same. Since the spectrum (2.92) is defined in the angular frequency domain, it is transformed into the frequency domain by relation

$$S(\omega)d\omega = S(f)df \Rightarrow S(f) = S(\omega) \frac{d\omega}{df} \underbrace{\Rightarrow}_{\omega=2\pi f} S(f) = S(\omega)2\pi. \quad (2.93)$$

Now, the spectra can be plotted in the frequency domain as follows.

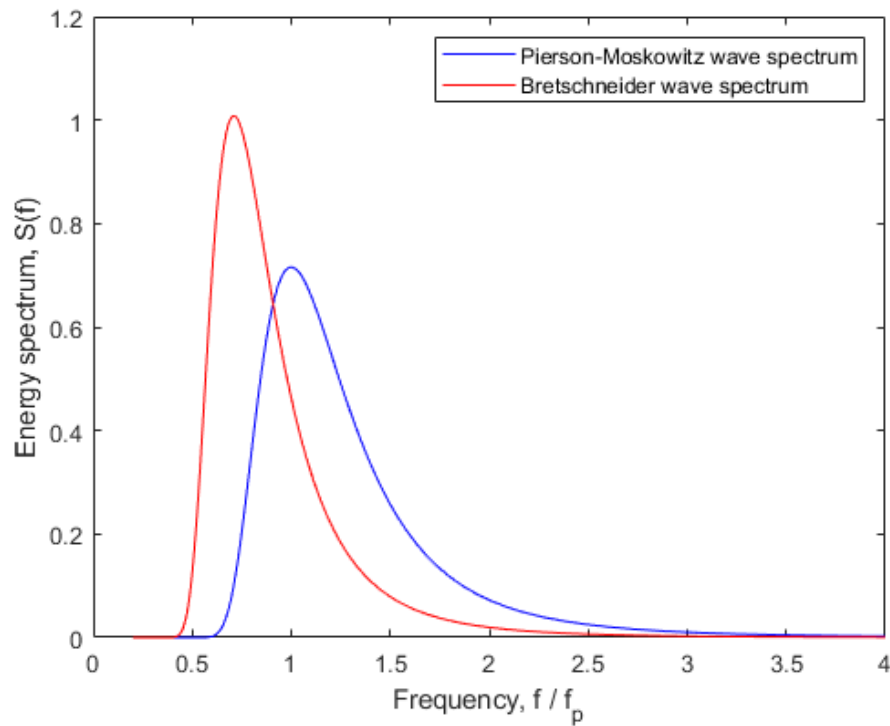


Figure 2.13. The comparison of Pierson-Moskowitz spectrum (2.91) with the Bretschneider spectrum (2.92) in frequency domain

Both spectra have a form of a typical Rayleigh distribution, which is the observed behavior of the sea within a short period. The peak of Pierson-Moskowitz occurs near the peak period, and Bretschneider's spectrum has its' peak shifted more towards the left. This is due to the Pierson-Moskowitz using peak period and Bretschneider using the zero crossing wave period, which is not the same. Thus, when using the spectrum and the data from the sea, care needs to be taken on what parameters each spectrum uses.

3. COMPUTATIONAL FLUID DYNAMICS

Computational Fluid Dynamics (CFD) is a numerical technique for obtaining the solution to the Navier-Stokes. These equations govern the conservation of mass, momentum, and energy in a form of a partial differential equation. The most popular method to conduct CFD simulations is called the Finite Volume Method (FVM), which discretizes the domain into a computational mesh, and the flow equations are turned into a linear system of equations. The flow properties are solved from this system by using numerical methods. However, to produce accurate results, the mesh needs to be fine making the computational complexity of the CFD one of its greatest issues. [14] [45]

The CFD has been originally inspired by the aeronautics and aerospace industry, but other industries such as automotive, chemical, or marine have also developed applications to utilize it [45]. The software to calculate CFD problem is nowadays so user-friendly that with only a few clicks, one can conduct a complete analysis. Interpreting the results without knowledge of the methods and theory can lead to errors. This chapter conducts a literature survey on the CFD to determine the methods and assumptions. The literature books recommended and used in this chapter include [45], [75], [65], [14], and [3].

3.1 Governing equations

As stated previously, the core equations of CFD calculations are the so-called Navier-Stokes equations i.e. the conservation of mass, momentum, and energy. This thesis does not cover the energy equation as it is not important to us. For more about the energy equations, readers are referred, for example, to [14]. The remaining conservation equations can be derived in numerous ways. One of them includes the use of the Reynolds Transport Theorem which is covered here. Before that, we need a couple of other useful definitions. Firstly the Leibniz integral rule is as follows.

Definition 3.1. ([26, p. 11]) ([45, p. 38]) Let $\psi = \psi(\mathbf{x}, t)$ a space and time dependent function and $\frac{\partial \psi}{\partial t}$ be continuous in integration volume $V(t)$. The *Leibniz integral rule* is defined as

$$\frac{d}{dt} \int_{V(t)} \psi dV = \int_{V(t)} \frac{\partial \psi}{\partial t} dV + \int_{S(t)} \psi (\mathbf{u}_s \cdot \mathbf{n}) dS, \quad (3.1)$$

where $S(t)$ is the boundaries of the volume $V(t)$ and \mathbf{u}_s is the velocity of the moving boundary.

For a special case where the boundaries are stationary, the second term of equation (3.1) vanishes and we obtain the relation

$$\frac{d}{dt} \int_V \psi dV = \int_V \frac{\partial \psi}{\partial t} dV. \quad (3.2)$$

Another important definition for CFD is the divergence theorem, which is used to turn a volumetric integral into a surface integral.

Definition 3.2. ([9, p. 457]) ([45, p. 35]) Let V be volume, S its boundary, and \mathbf{u} a velocity vector field defined inside V . Now the *divergence theorem* states that

$$\int_V \nabla \cdot \mathbf{u} dV = \int_S \mathbf{u} \cdot \mathbf{n} dS. \quad (3.3)$$

The divergence theorem is also known as Gauss' theorem and is used in deriving conservation equations and in the FVM. Finally, generalizing the Definition 3.1 into a material of mass inside the fluid field, we get the so-called Reynolds Transport Theorem.

Definition 3.3. ([37, p. 22]) ([65, p. 382]) Let E be extensive propriety of fluid and e intensive value of E . Now for control volume CV closed by control surface CS , the *Reynolds Transport Theorem* states a relation

$$\frac{dE}{dt} = \frac{d}{dt} \int_{CV} \rho e dV + \int_{CS} (\rho e)(\mathbf{u} \cdot \mathbf{n}) dS, \quad (3.4)$$

where ρ is density, \mathbf{u} is velocity and \mathbf{n} is the normal vector for the control surface.

In Definition 3.3, extensive property means that the property is proportional to the mass, and intensive property is independent of mass. Thus, we write that $e = E/m$. For example, fluids momentum $\mathbf{p} = m\mathbf{u}$ is an extensive property, and its corresponding intensive property is velocity \mathbf{u} .

The Reynolds Transport Theorem has a physical interpretation, where the variation of property E within a control volume is described as the sum of internal sources and sinks and the flux across the control surface boundaries. However, it rarely gets used in actual problems, but rather is used to derive conservation laws.

3.1.1 Conservation of mass

Let CV be a stationary control volume that co-insides the system of interest, and let CS be its boundary. To derive the conservation of mass equation, let $E = m$ be mass

in Definition 3.3. The corresponding intensive property computes as $e = E/m = 1$. Applying these to equation (3.4) we get that

$$\frac{dm}{dt} = \frac{d}{dt} \int_{CV} \rho dV + \int_{CS} \rho(\mathbf{u} \cdot \mathbf{n}) dS. \quad (3.5)$$

As the mass in the system is conserved, we write that $dm/dt = 0$. Applying Leibniz rule (3.1) to the first term and divergence theorem (3.3) to the second term, we get that

$$\int_{CV} \left(\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) \right) dV = 0. \quad (3.6)$$

As the control volume of zero is a trivial case, we require that the integrand of equation (3.6) is zero. Thus, we get the conservation of mass as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0. \quad (3.7)$$

This equation can be further simplified if incompressibility is assumed. Here, the density remains constant within streamline which is a good approximation for any liquid, but for example not to sound waves [45]. Therefore, the conservation of mass is defined as

$$\nabla \cdot \mathbf{u} = 0 \quad (3.8)$$

or, by using the Einstein notation, as

$$\frac{\partial u_j}{\partial x_j} = 0, \quad (3.9)$$

where u_1, u_2 , and u_3 are velocities and x_1, x_2 , and x_3 are coordinates in their respective x, y and z directions. The $(\nabla \cdot)$ operator is referenced as a divergence or "del" operator. Physically it has meaning that point is either a sink or a source. Thus, the conservation of mass (3.8) states that no point acts like a sink or a source i.e. no mass is created or destroyed.

3.1.2 Conservation of momentum

Like previously, we fix the CV and use the Definition 3.3 to write the extensive property $\mathbf{E} = \mathbf{p} = m\mathbf{u}$ as the momentum. The corresponding intensive property is now a velocity i.e. $\mathbf{e} = \mathbf{u}$. Note, that now we have a vector as a property of interest instead of a scalar, and could be written as three separate scalar equations as well. The Reynolds Transport Theorem (3.4) states that

$$\frac{d\mathbf{p}}{dt} = \frac{d}{dt} \int_{CV} \rho \mathbf{u} dV + \int_{CS} \rho \mathbf{u} (\mathbf{u} \cdot \mathbf{n}) dS, \quad (3.10)$$

and by applying the Leibniz rule (3.1) to the first term and the divergence theorem (3.3) to the second term, we obtain

$$\frac{d\mathbf{p}}{dt} = \int_{CV} \frac{\partial \rho \mathbf{u}}{\partial t} + (\nabla \cdot \mathbf{u}) \rho \mathbf{u} dV. \quad (3.11)$$

From Newton's second law, we know that the left-hand side of (3.11) equals the total force of the system. Now, since the control volume coincides with the system, the total force of the system equals the total force of the control volume, and the control volume is obtained by integrating the whole volume as

$$\frac{d\mathbf{p}}{dt} = \int_{CV} \mathbf{F} dV \quad (3.12)$$

Applying this to equation (3.11), we obtain the conservation of momentum equation as

$$\frac{\partial \rho \mathbf{u}}{\partial t} + (\nabla \cdot \mathbf{u}) \rho \mathbf{u} = \mathbf{F}. \quad (3.13)$$

The total force can be divided as [14]

$$\mathbf{F} = \mathbf{F}_B + \mathbf{F}_S, \quad (3.14)$$

where subscripts represent body and surface forces. The body forces act inside the control volume of the fluid parcel. The most dominant force is the gravitational force. Also when dealing with a rotating frame of reference, the Coriolis and the centrifugal forces arise. However, in the case where the frame of reference is fixed, these can be neglected. Thus we define body forces as

$$\mathbf{F}_B = \rho \mathbf{g} + \rho \mathbf{F}_{\text{source}}, \quad (3.15)$$

where $\mathbf{F}_{\text{source}}$ represents external forces, which usually are modeled as zero.

Surface forces are described by stresses acting on the control surface. These stresses are modeled as 3×3 tensor Σ , where diagonal elements represent the forces acting on the surface's normal direction, and the off-diagonal elements represent the shearing forces. The total surface force over the control volume is therefore

$$\int_{CV} \mathbf{F}_S dV = \int_{CS} \boldsymbol{\Sigma} \cdot \mathbf{n} dS, \quad (3.16)$$

and after applying the divergence theorem, we obtain that

$$\mathbf{F}_S = \nabla \cdot \boldsymbol{\Sigma}. \quad (3.17)$$

The off-diagonal elements of $\boldsymbol{\Sigma}$ occur due to friction. For fluid flows, this is referred to as viscosity. On the other hand, normal stresses occur due to friction and pressure acting on the surface. Pressure forces act in the opposite direction of the normal, so the total stress tensor can be modeled as

$$\boldsymbol{\Sigma} = -p\mathbf{I} + \boldsymbol{\sigma}, \quad (3.18)$$

where \mathbf{I} is identity matrix and $\boldsymbol{\sigma}$ is deviatoric stress tensor. Applying this to (3.17) yields

$$\mathbf{F}_S = -\nabla p + \nabla \cdot \boldsymbol{\sigma}. \quad (3.19)$$

Thus, the final equation for the conservation of momentum is

$$\frac{\partial \rho \mathbf{u}}{\partial t} + (\nabla \cdot \mathbf{u})\rho \mathbf{u} = -\nabla p + \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g} + \rho \mathbf{F}_{\text{source}}. \quad (3.20)$$

or written in the Einstein notation

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \sigma_{ij}}{\partial x_j} + \rho g_i + \rho F_{i,\text{source}} \quad (3.21)$$

where $i = 1, 2, 3$. Assuming incompressibility and by using the chain rule, the second term of the left-hand side of equation (3.21) is expanded as

$$\frac{\partial \rho u_i u_j}{\partial x_j} = \rho \left(u_i \frac{\partial u_j}{\partial x_j} + u_j \frac{\partial u_i}{\partial x_j} \right), \quad (3.22)$$

and applying the equation (3.9) to the first term, we get the incompressible conservation of momentum as

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{1}{\rho} \frac{\partial \sigma_{ij}}{\partial x_j} + g_i + F_{i,\text{source}}. \quad (3.23)$$

The equation (3.23) describes the motion of the incompressible fluid parcel with the pres-

sure gradients and the stresses. Further assuming that the fluid is Newtonian, the stress terms can be modeled as

$$\sigma_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \quad (3.24)$$

and the only unknowns that remain in the equation (3.23) are velocity and pressure.

3.2 Reynolds Averaged Navier-Stokes equations

The conservation of mass and momentum can be used to describe the motion of the fluid. This system exhibits four unknowns and has four equations. In theory, the system is complete and could be solved. However, a problem arises with the length scale of the flow. The most accepted theory of the smallest scale of turbulence is known as the Kolmogorov microscales, where the kinetic energy of turbulence dissipates to heat [45, p. 694]. The microscales for length η and time t_n are given as

$$\eta = \left(\frac{\nu^3}{\epsilon} \right)^{\frac{1}{4}} \quad \text{and} \quad t_n = \left(\frac{\nu}{\epsilon} \right)^{\frac{1}{2}}, \quad (3.25)$$

where ν is kinematic viscosity and ϵ is turbulence dissipation rate. As shown in the example in [12], the mesh using this length scale can grow unrealistically large even in simple simulations. Since most engineering problems aren't interested in the fine details of the flow, but rather the whole picture, a lot of redundant work would be conducted. Thus, the idea of averaging the solution variables to find the mean values becomes attractive. In Reynolds averaging, the space and time-dependent quantity $\psi(\mathbf{x}, t)$ is decomposed into a mean and a fluctuating part as

$$\psi(\mathbf{x}, t) = \overline{\psi(\mathbf{x})} + \psi'(\mathbf{x}, t), \quad (3.26)$$

where the mean quantity is defined by taking the time average as

$$\overline{\psi(\mathbf{x})} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{t_0}^{t_0+T} \psi(\mathbf{x}, t) dt. \quad (3.27)$$

On the other hand, the fluctuating part is the small and mostly random variation of the fluid property. The decomposition is illustrated in the figure below.

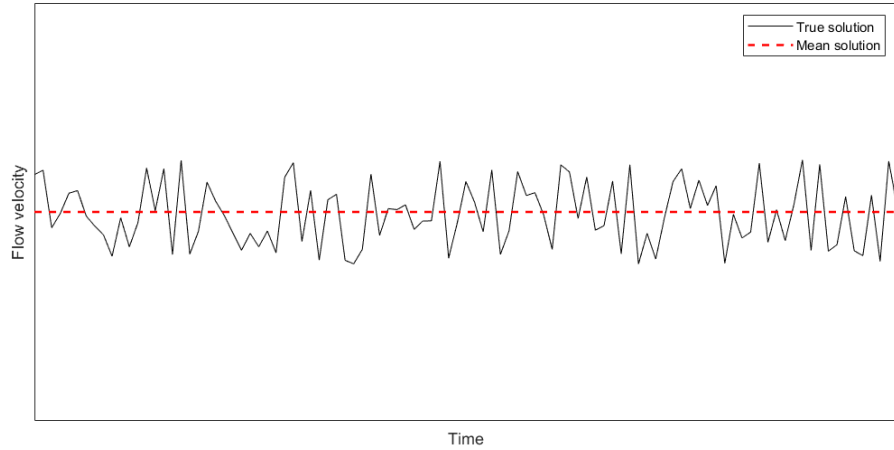


Figure 3.1. The red line corresponds to true velocity u and the dotted red line corresponds the mean value \bar{u}

Some systems might possess an oscillating or otherwise slowly changing steady-state solution. This requires the change of the time averaging definition (3.27) to

$$\overline{\psi(\mathbf{x}, t)} = \frac{1}{T} \int_t^{t+T} \psi(\mathbf{x}, t) dt, \quad (3.28)$$

where T is a sufficiently long time. However, for now, we can assume that the time-independent steady-state solution exists. [45]

Applying the Reynolds decomposition to the mass equation (3.9) and to the momentum equation (3.23), we get the Reynolds Averaged Navier-Stokes (RANS) equations as [14] [45]

$$\frac{\partial \bar{u}_j}{\partial x_j} = 0 \quad (3.29)$$

and

$$\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \frac{1}{\rho} \frac{\partial}{\partial x_j} (\bar{\sigma}_{ij} - \rho \overline{u'_j u'_i}) + \bar{g}_i + \bar{f}_{\text{source}}. \quad (3.30)$$

These equations are similar to the original Navier-Stokes equations with additional term $\overline{u'_j u'_i}$ on the right-hand side. This term is referred to as Reynolds stresses because its dimensions represent a typical stress variable. All other terms are now time-averaged, so the solution of this system is a steady-state solution i.e. the mean flow velocity and pressure from an infinitely long period.

When looking at our system, we notice that we have created a new problem. The term

$\overline{u'_j u'_i}$ is a symmetric tensor, which therefore creates six new unknowns, while the Reynolds averaging process didn't produce any new equations. Thus we have four equations and ten unknowns, so the system is not closed. Thus more equations are needed, which is further discussed in Section 3.3 about turbulence modeling.

3.3 Turbulence modeling

The Reynolds Averaged Navier-Stokes equations are not closed and thus require more information. We have four equations for velocity and pressure, but six problematic unknowns arise from Reynolds stress terms $\overline{u'_j u'_i}$, which describe the variance of fluctuating terms. The idea of modeling the turbulence is to evaluate these stress terms by a separate submodel. Although there is no one universal model as each has its ups and downs, a lot of research has been devoted to this subject. In this section, some of the most well-known models are introduced with varying complexity, and the chapter is based on [3], [45] and the works of Wilcox [78], who pioneered one of the most commonly used turbulence models.

The most simple model is to set this term to 0. Physically this means that the velocity doesn't fluctuate, i.e. it acts smoothly and thus has a steady state solution. This kind of flow is referred to as *laminar flow*, which occurs with a low Reynolds number. For example, in flow over a flat plate, the laminar region occurs when $Re < 5.0 \times 10^5$ [45, p. 72]. Laminar flow can be described as such flow, where streamlines don't cross each other. However, the flow is more often turbulent, and finding a laminar flow requires a specific setup.

The concept of *Eddy viscosity*, also known as the *Boussinesq hypothesis*, is the most commonly used approach in turbulence modeling. This idea was first proposed in 1877, and it assumes that turbulence has a dissipative effect on the mean flow, which can be modeled using a "virtual viscosity" term. Turbulence produces swirls known as eddies, and this term is used to refer to these swirling motions. Mathematically speaking, the Reynolds stresses are modeled as

$$-\rho \overline{u'_j u'_i} = \mu_t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{1}{3} \frac{\partial u_j}{\partial x_j} \delta_{i,j} \right) - \rho \frac{2}{3} k \delta_{i,j}, \quad (3.31)$$

where μ_t is the kinematic viscosity of turbulence, δ_{ij} is Kronecker delta and k is turbulence kinetic energy defined as $k = \frac{1}{2} \overline{u'_i u'_i}$. For incompressible flows, this equation simplifies to

$$-\rho \overline{u'_j u'_i} = \mu_t \left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i} \right) - \rho \frac{2}{3} k \delta_{i,j}. \quad (3.32)$$

One keynote of the model is that the Reynolds stress tensor is proportional to the mean

shear stress tensor. With available experiments and direct numerical simulations, this assumption has been tested and found that it fails rather often. While the hypothesis lacks truth, its derived models can predict the turbulence of simple flows relatively well. Therefore it is a more useful tool than using nothing for engineering purposes. [62]

The equation (3.32) now has 2 unknowns, turbulent kinematic viscosity μ_t and the turbulent kinetic energy k . Examining this further reveals that the non-diagonal terms are the first term of the right-hand side due to the Kroneker delta, and the diagonal terms are the second term of the right-hand side due to continuity constraint. As the normal stresses act likewise to pressure terms, we can modify pressure as

$$p^* = p - \frac{2}{3}\rho k \quad (3.33)$$

and solve for p^* instead of static pressure p . Therefore, only the turbulent kinematic viscosity μ_t is needed to close the problem.

3.3.1 Zero-equation models

The naming scheme "n-equation" models come from how many of these transport equations are constructed to solve the turbulence viscosity. In the zero-equation models, there are no separate partial differential equations but rather the problem is closed with simple algebraic models. One zero-equation model is Prandtl's mixture length model. We begin by noting that $\mu_t = \rho\nu_t$ and the

$$[\nu_t] = \left[\frac{L^2}{T} \right] \implies \nu_t \equiv l_{\text{mix}} u_{\text{mix}}, \quad (3.34)$$

where l_{mix} is mixing length and u_{mix} is mixing velocity. As the mixing happens at the molecular level, we can make a good approximation that

$$u_{\text{mix}} \approx l_{\text{mix}} \frac{\partial u}{\partial y}, \quad (3.35)$$

where the partial derivative is with respect to the normal to the wall. Thus, we derive the non-diagonal term for Reynolds stress as

$$\overline{\rho u'v'} = l_{\text{mix}}^2 \frac{\partial u}{\partial y}. \quad (3.36)$$

Here the mixing length can be evaluated from experimental results. When absence of experimental data, the evaluation of the mixing length can be difficult. Also, since in different parts of the geometry, different length scales are more present making the use

of this model unpractical. Therefore, it is not advised to be used, but rather it stands as a precursor for the world of turbulence modeling.

3.3.2 One-equation models

The one-equation models try to close the system by solving one extra partial differential equation, which usually is a transport equation for some turbulence quantity. In the previous chapter, a turbulent kinetic energy k was introduced, but it was bypassed by including it in the pressure term as (3.33). A full transport equation for this quantity can be derived as

$$\frac{\partial k}{\partial t} + u_j \frac{\partial k}{\partial x_j} = -\overline{u'_i u'_j} \frac{\partial u_i}{\partial x_j} - \epsilon - \frac{\partial}{\partial x_j} \left[\frac{1}{2} (\overline{u'_i u'_i u'_j}) + \frac{1}{\rho} (\overline{p' u'_j}) \right] + \frac{1}{\rho} \frac{\partial}{\partial x_j} \left[\mu \frac{\partial k}{\partial x_j} \right], \quad (3.37)$$

where ϵ is the dissipation of turbulence and the variance term $\overline{u'_j u'_i}$ is given by equation (3.32). Here we have created a bunch of new unknown terms, so the system is not closed once again. The third term of the right-hand side defines the transport of turbulence fluctuation, so instead of solving it directly, we can model it as a generic gradient-diffusion term. The fourth pressure fluctuation term is still problematic, but DNS calculations show that for incompressible flows it is relatively small, and can be grouped with the gradient-diffusion term. Thus we can make the assumption that

$$\frac{1}{2} (\overline{u'_i u'_i u'_j}) + \frac{1}{\rho} (\overline{p' u'_j}) = -\frac{\nu_t}{\sigma_k} \frac{\partial k}{\partial x_j}, \quad (3.38)$$

where σ_k is the closure coefficient known as the Prandtl-Schmidt number. The equation (3.37) simplifies to

$$\frac{\partial k}{\partial t} + u_j \frac{\partial k}{\partial x_j} = -\overline{u'_i u'_j} \frac{\partial u_i}{\partial x_j} - \epsilon + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right]. \quad (3.39)$$

The dissipation term ϵ is unknown, but pure dimensional analysis shows that $\epsilon \sim \frac{k^{3/2}}{l}$, so Prandtl postulated a closure

$$\epsilon = C_D \frac{k^{3/2}}{l}, \quad (3.40)$$

where C_D is a model coefficient and l is the typical length scale for the turbulence.

Another one-equation model comes from Spalart and Allmaras [71]. Here instead of solving the kinetic energy of turbulence, a modified version of turbulent viscosity is solved

in the transport equation

$$\frac{\partial \tilde{\nu}}{\partial t} + u_j \frac{\partial \tilde{\nu}}{\partial x_j} = c_{b1} \tilde{S} \tilde{\nu} - c_{w1} f_w \left(\frac{\tilde{\nu}}{d} \right)^2 + \frac{1}{\sigma} \frac{\partial}{\partial x_i} \left[(\nu + \tilde{\nu}) \frac{\partial \tilde{\nu}}{\partial x_i} \right] + \frac{c_{b2}}{\sigma} \frac{\partial \tilde{\nu}}{\partial x_i} \frac{\partial \tilde{\nu}}{\partial x_i}. \quad (3.41)$$

The kinematic viscosity is then solved as

$$\nu_t = \tilde{\nu} f_{v1}, \quad (3.42)$$

where f_{v1} is a damping function to simulate the law of the wall behaviour near the surface. The model coefficients are calibrated with respect to aerospace applications, so the model should be used only in this field.

3.3.3 Two-equation models

Two-equation models are by far the most widely used in the engineering industry. One weakness with one-equation models is that the turbulence scales must be provided by the user. While for some problems these scales can be evaluated thoroughly, applying these models to more complex geometries can result in faulty predictions. The idea of resolving the scale of turbulence with a separate transport equation seems a more logical approach since it would make the model complete.

The choice for the second parameter is still an active topic for research and no universal model has been accepted by the industry. The most popular choice is the turbulent dissipation rate ϵ as was discussed previously. Dimensional analysis shows that the length scale and eddy viscosity are related to the energy and dissipation rate as

$$l \sim \frac{k^{3/2}}{\epsilon} \quad \text{and} \quad \nu_t \sim \frac{k^2}{\epsilon}. \quad (3.43)$$

Thus the standard $k - \epsilon$ model by Launder and Spalding [36] solves the equation (3.39) for kinetic energy and dissipation rate, a transport equation [78]

$$\frac{\partial \epsilon}{\partial t} + u_j \frac{\partial \epsilon}{\partial x_j} = -C_{\epsilon 1} \frac{\epsilon}{k} u'_i u'_j \frac{\partial u_i}{\partial x_j} - C_{\epsilon 2} \frac{\epsilon^2}{k} + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_\epsilon} \right) \frac{\partial \epsilon}{\partial x_j} \right]. \quad (3.44)$$

is solved. Finally, the eddy viscosity is calculated as $\nu_t = C_\mu k^2 / \epsilon$, where C_μ is a model constant. By default, the value 0.09 is used for this constant.

Another popular parameter to describe the turbulence scale is the specific dissipation rate ω , which has a unit of $1/s$. Now the dimensional analysis states that

$$l \sim \frac{k}{\omega} \quad \text{and} \quad \nu_t \sim \frac{k^{1/2}}{\omega}, \quad (3.45)$$

so ω can be used as a turbulence length scale as well. The standard $k - \omega$ model developed by Wilcox [78] solves the kinetic energy as

$$\frac{\partial k}{\partial t} + \overline{u_j} \frac{\partial k}{\partial x_j} = -\overline{u'_i u'_j} \frac{\partial \overline{u_i}}{\partial x_j} - \beta^* k \omega + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right]. \quad (3.46)$$

and the specific dissipation term as

$$\frac{\partial \omega}{\partial t} + \overline{u_j} \frac{\partial \omega}{\partial x_j} = \alpha \frac{\omega}{k} \overline{u'_i u'_j} \frac{\partial \overline{u_i}}{\partial x_j} - \beta \omega^2 + \frac{\sigma_d}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} + \frac{\partial}{\partial x_j} \left[\left(\nu + \sigma \frac{k}{\omega} \right) \frac{\partial \omega}{\partial x_j} \right]. \quad (3.47)$$

Here the turbulent viscosity is resolved from relation $\nu_t = k/\omega$.

Both $k - \epsilon$ and $k - \omega$ serve as good borderline models for turbulence, but they have seen a lot of improvements and new variations of these models have appeared. One modified model is called *realizable* $k - \epsilon$, where the model is modified to be realizable. Consider equation (3.32) in form

$$\overline{u'_i u'_i} = \frac{2}{3} k - 2\nu_t \frac{\partial \overline{u_i}}{\partial x_i}. \quad (3.48)$$

With the standard $k - \epsilon$ model, we could get the result of $\overline{u'^2_i} < 0$, which would be non-physical. Another realizability constraint, that the equation might break, is so called Schwarz's inequality given as

$$(\overline{u'_i u'_j})^2 \leq \overline{u'^2_i} \overline{u'^2_j}. \quad (3.49)$$

To combat this, the realizable $k - \epsilon$ model solves the modified equation for the turbulent dissipation rate

$$\frac{\partial \epsilon}{\partial t} + \overline{u_j} \frac{\partial \epsilon}{\partial x_j} = C_1 S \epsilon + C_{\epsilon 1} \frac{\epsilon}{k} C_{3\epsilon} G_b - C_2 \frac{\epsilon^2}{k + \sqrt{\nu \epsilon}} + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_\epsilon} \right) \frac{\partial \epsilon}{\partial x_j} \right], \quad (3.50)$$

where

$$C_1 = \max \left[0.43, \frac{S \frac{k}{\epsilon}}{S \frac{k}{\epsilon} + 5} \right], \quad S = \sqrt{2 S_{ij} S_{ij}}, \quad S_{ij} = \frac{1}{2} \left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i} \right).$$

Also, the term C_μ used to calculate the eddy viscosity is no longer constant but rather a variable defined as

$$C_\mu = \frac{1}{A_0 + A_s \frac{kU^*}{\epsilon}}. \quad (3.51)$$

While the realizability constraints seem to make the model better, it doesn't automatically outperform the standard $k - \epsilon$ model, as shown in [63]. However, the realizable model is still thought of as the superior version for complex geometry generally.

One of the downfalls of the standard $k - \epsilon$ model is that it tends to overestimate the turbulence near boundary layers and in the existence of adverse pressure gradients. The standard $k - \omega$ was the first two equation model to combat these, but it has a problem of being sensitive to freestream conditions, whereas standard $k - \epsilon$ showed good results. Thus Menter [42] proposed a combined turbulence model, which utilized the best of both models using a blending function. Here the turbulent kinetic energy is calculated from

$$\frac{\partial k}{\partial t} + \bar{u}_j \frac{\partial k}{\partial x_j} = \tilde{P}_k - \beta^* k \omega + \frac{\partial}{\partial x_j} \left[(\nu + \sigma_k \nu_t) \frac{\partial k}{\partial x_j} \right]. \quad (3.52)$$

and the turbulent specific dissipation rate is calculated from

$$\frac{\partial \omega}{\partial t} + \bar{u}_j \frac{\partial \omega}{\partial x_j} = \alpha S^2 - \beta \omega^2 + \frac{\partial}{\partial x_j} \left[(\nu + \sigma_\omega \nu_t) \frac{\partial \omega}{\partial x_j} \right] + 2(1 - F_1) \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i}, \quad (3.53)$$

where F_1 is the blending function defined as

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

For other parameters, readers are referred to the [42]. This model is known as Menter's $k - \omega$ SST and it is thought by many as the most complete two-equation model so far. However, for some specific cases, other models might give better results as the models often get tailored and calibrated to those cases. But for an unknown case with complex geometry, $k - \omega$ SST has shown the most consistent results.

3.3.4 Higher order models

The two-equation models are considered as complete models, as they calculate the magnitude (usually from the kinetic turbulent energy k) and the scale (usually from dissipation rate ϵ or specific dissipation rate ω). The need to solve any higher dimensional model is

often redundant in engineering problems, as they either impose more numerical instability or computational cost. Therefore these models are often used mainly in the academic field.

The Reynolds stress transport model tries to solve the Reynolds stresses by imposing a transport equation directly as [60]

$$\frac{\partial \overline{\rho u'_j u'_i}}{\partial t} + \frac{\partial \rho u_k \overline{u'_j u'_i}}{\partial x_k} = D_{T,ij} + D_{L,ij} + P_{ij} + \phi_{ij} - \epsilon_{ij} + F_{ij}, \quad (3.54)$$

where D_T is the turbulent diffusion term, D_L is the molecular diffusion term, P is the production term for stresses, ϕ is pressure strain, ϵ is dissipation and F is the production due rotations. Each of these terms then has a separate transport equation.

The idea of Reynolds averaging is to separate the flow into a mean and fluctuating part. This allows the system to solve a steady-state solution and the computational grid doesn't need to be resolved down to the smallest turbulence length scale. However, it only models the turbulence, and in the cases where turbulence is dominant, RANS equations can give erroneous results. To properly resolve complex temporal cases, a Large Eddy Simulation (LES) applies a filtering operator to the computational grid. This allows the turbulence scales to be directly solved, where the grid is fine enough, and the remaining parts can be modeled. The generalized filter operator is defined as

$$\overline{u}_i(x, t) = \int \int \int G(x - \xi; \Delta) u_i(\xi, t) d^3 \xi, \quad (3.55)$$

where $\overline{u}_i(x, t)$ is noted as resolvable velocity and $G(x - \eta, \Delta)$ is a normalized filter function. The volume-average box filter is defined as

$$G(x - \xi; \Delta) = \begin{cases} \frac{1}{\Delta^3}, & |x_i - \xi_i| < \frac{\Delta x_i}{2}, \\ 0, & \text{otherwise} \end{cases}, \quad (3.56)$$

where Δ is the volume of the cell. Finally, the subgrid-scale velocity is defined as

$$u'_i = u_i - \overline{u}_i. \quad (3.57)$$

When applying the filter operator to Navier-Stokes equations, the subgrid-scale velocities produce similar turbulence terms as in RANS calculations. These later then need to be resolved in a separate submodel.

3.4 Finite volume method

For some simple cases, the Navier-Stokes equations can be solved analytically, for example, a laminar flow inside a pipe. However, for more complex geometry, the only applicable way to obtain information on the flow is to solve the system numerically, where the FVM comes in. The goal is to discretize the geometry into small control volumes i.e. *cells* and to turn the partial differential equation into algebraic equations. The smaller this control volume is, the more accurate the result will be, but this will increase the total number of cells which increases the computational cost.

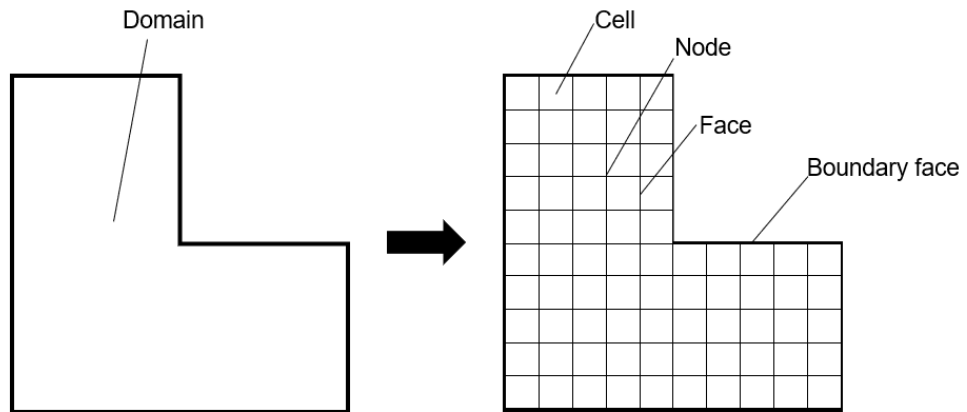


Figure 3.2. The discretization process visualized

Although the FVM has similarities with the finite element method (FEM), it is often kept as superior in fluid calculations due to its conservative nature. This comes from the fact that flux entering each cell must be equal to flux leaving the fluid. Also, the FVM is considered to be easier to implement and more stable than FEM. [45]

This section is based on the books [45], [15], and [75]. For supplementary reading, the Ph.D. thesis from Jasak [31] is recommended.

3.4.1 Discretization process

To understand FVM, let's consider a typical scalar transport equation

$$\underbrace{\frac{\partial \rho \psi}{\partial t}}_{\text{temporal term}} + \underbrace{\nabla \cdot \rho \mathbf{u} \psi}_{\text{convective term}} - \underbrace{\nabla \cdot (\rho D_{\psi} \nabla \psi)}_{\text{diffusive term}} = \underbrace{S_{\psi}}_{\text{source term}}, \quad (3.58)$$

where ψ is a flow variable and D_{ψ} is diffusion coefficient. Now, the technique involves dividing the domain into multiple small control volumes and solving this relation within

each volume. These control volumes are referred to as cells and they must not overlap each other. Taking the integral with respect to the CV , we get

$$\int_{CV} \frac{\partial \rho \psi}{\partial t} dV + \int_{CV} \nabla \cdot \rho \mathbf{u} \psi dV - \int_{CV} \nabla \cdot (\rho D_\psi \nabla \psi) dV = \int_{CV} S_\psi dV. \quad (3.59)$$

Applying the divergence theorem 3.2 to the convective term to modify the volumetric integral into a surface integral. Thus, the convective term modifies to

$$\int_{CV} \nabla \cdot \rho \mathbf{u} \psi dV = \int_{CS} (\rho \mathbf{u} \psi) \cdot \mathbf{n} dS \quad (3.60)$$

where CS is the control surface over the CV . Now, defining the control volumes to be some arbitrary polygons, i.e. the boundaries are composed of flat faces, the integration within the whole control surface can be expressed as the sum of integrals over individual faces i.e.

$$\int_{CS} (\rho \mathbf{u} \psi) \cdot \mathbf{n} dS = \sum_f \left(\int_f (\rho \mathbf{u} \psi) \cdot \mathbf{n} dS \right) \quad (3.61)$$

and by applying Gaussian quadrature for one point, we can approximate the integral as [45] [15]

$$\int_f (\rho \mathbf{u} \psi) \cdot \mathbf{n} dS \approx ((\rho \mathbf{u} \psi)_f \cdot \mathbf{n}) S_f = \mathbf{S}_f \cdot (\rho \mathbf{u} \psi)_f, \quad (3.62)$$

where \mathbf{S}_f is a vector pointing to the surface normal direction with a magnitude of surface area and the subscript f represents that the values are evaluated at the face center. Similarly, for the diffusion term, we have that

$$\int_{CV} \nabla \cdot (\rho D_\psi \nabla \psi) dV \approx \sum_f \mathbf{S}_f \cdot (\rho D_\psi \nabla \psi)_f \quad (3.63)$$

Assuming the source term to be uniform within the control volume, the integral can be evaluated directly. The source term is usually separated into a constant part S_c and a variable dependent part $S_p \psi$. Thus, we write

$$\int_{CV} S_\psi dV = S_c V_p + S_p V_p \psi. \quad (3.64)$$

Lastly, the temporal term is also evaluated directly, and temporal discretization is applied

in a form of an integral from t to $t + \Delta t$. A typical scheme for evaluating the temporal derivative is either the first-order forward Euler or the second-order Crank-Nicolson [45]. For now, we can assume to find a steady state solution so the temporal term will be neglected. The semi-discretized equation has a form

$$\sum_f \left(\rho \mathbf{u} \psi - \rho D_\psi \nabla \psi \right)_f \cdot \mathbf{S}_f = S_c V_p + S_p V_p \psi. \quad (3.65)$$

The equation above still contains unknown terms ψ_f and $(\nabla \psi)_f$, which are the flow variable and its gradient evaluated in the face of the boundary cells. As the values are known in the cell center from the initial guess or previous time-step, the respective face values can be interpolated or gradient evaluated numerically.

The face value ψ_f can be obtained by interpolation. As this interpolation is done to determine the divergence term in (3.58), the choice of the interpolation method is known as the divergence scheme in CFD software. Many different-order schemes exist, for example, the popular first-order upwind scheme is defined as

$$\psi_f = \begin{cases} \psi_C, & \mathbf{u}_f \cdot \mathbf{S}_f > 0 \\ \psi_N, & \text{otherwise} \end{cases} \quad (3.66)$$

where subscript C points to the cell center and subscript N points to the neighboring cell center. The scheme thus chooses the value at the face as the value in the cell by the flux direction. The scheme is simple and found to be stable, but it is also found to include high numerical diffusion to the solution. For second-order upwind, the face value is evaluated as

$$\psi_f = \psi_C + (2\nabla \psi_C - \nabla \psi_f) \cdot \mathbf{d}_{CF}, \quad (3.67)$$

where \mathbf{d}_{CF} is the distance vector from the cell center to the face. While this is a higher accuracy scheme, it might produce more stability problems.

The typical way to solve the gradient of ψ in FVM is to use Green-Gauss' or least-square method. For example, the Green-Gauss gradient is derived as

$$V_p \nabla \psi_C = \int_{CV} \nabla \psi dV = \int_{CS} \psi \mathbf{n} dS = \sum_f \psi_f \mathbf{S}_f. \quad (3.68)$$

The gradient term $(\nabla \psi)_f \cdot \mathbf{S}_f$ at the equation (3.65) is evaluated at the face of the cell. It brings difficulties as the vector between neighboring cell centroids might not be parallel with respect to the wall normal vector. The angle between these two vectors is called

non-orthogonality and it is often used as a parameter for mesh quality. To reduce the error caused by non-orthogonality, the surface normal vector is separated into orthogonal and correction terms as [31]

$$\mathbf{S}_f = \underbrace{\Delta}_{\text{orthogonal contribution}} + \underbrace{\mathbf{k}}_{\text{non-orthogonal contribution}}, \quad (3.69)$$

where Δ is vector parallel to the line from cell centers and \mathbf{k} is the "residual" part. The orthogonal part can be evaluated as a typical numerical gradient, and the non-orthogonal part by other means. Therefore the surface normal gradient is evaluated as

$$\begin{aligned} (\nabla\psi)_f \cdot \mathbf{S}_f &= (\nabla\psi)_f \cdot \Delta + (\nabla\psi)_f \cdot \mathbf{k} \\ &= |\Delta| \frac{\psi_f - \psi_C}{d_{CF}} + (\nabla\psi)_f \cdot \mathbf{k} \end{aligned} \quad (3.70)$$

Now we still need to decide the length of Δ , which affects the overall stability. One choice is to make $|\Delta|$ such long that \mathbf{k} is perpendicular to Δ . This ensures that \mathbf{k} is the smallest possible, and the method is therefore referred to as the minimum correction. Another choice is to make \mathbf{k} perpendicular to \mathbf{S}_f , which then forces the importance of the term involving ψ_C and ψ_N when non-orthogonality increases. This method is referred to as the over-relaxed approach. For example, OpenFOAM uses the over-relaxed approach in its "corrected" surface normal scheme [17].

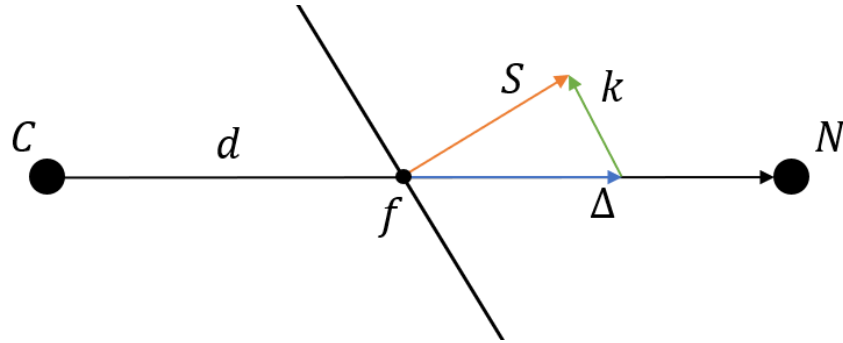


Figure 3.3. Non-orthogonality correction. Figure is extension from [31]

Thus applying these numerical techniques to (3.65) we finally get the algebraic relation

$$a_C \psi_C + \sum_f a_F \psi_F = b_C, \quad (3.71)$$

where a_C , a_F , and b_C are some constants depending on the choice of discretization used. For some cells that have a wall, but no neighbor behind it, a boundary condition must be specified. The two typical boundary conditions are the Dirichlet condition, where the value

of the boundary is specified as constant, and the Neumann condition, where the gradient or flux through the boundary is specified.

3.4.2 Solving linear algebraic linear system

Now when all of the control volumes have discretized, we get N equations in the form (3.71). It can be written as a linear system of equations $A\psi = b$, which allows all of the equations to be solved simultaneously. For example, one could find the inverse of A directly using Gaussian elimination. However, this process has the complexity of $\mathcal{O}(n^3)$, and as a typical mesh could have millions of cells, this would be unnecessarily time-consuming. A more suitable way would be to use a numerical algorithm and solve the system iteratively. This subsection additionally uses the book [58] as a reference.

One of the most popular algorithms is the so-called Gauss-Seidel method, which is an improved version of Jacobi's method. In Jacobi's method, each variable is algebraically solved one by one using the result from the previous step. Gauss-Seidel improves this by incorporating the most up-to-date result when solving the system. Mathematically speaking the i -th variable is solved by

$$\psi_i^{(n)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} \psi_j^{(n)} - \sum_{j=1+i}^N a_{ij} \psi_j^{(n-1)} \right), \quad (3.72)$$

where $\psi_i^{(n)}$ is the variable in current time step and $\psi_i^{(n-1)}$ is from previous time-step. It is also less memory intensive since the values can be updated to the original solution vector instead of creating a new one.

Another method for solving the linear system of equations is called the conjugate gradient (CG) which again is an improved version of the steepest descent method. This type of method is based on an optimization problem where given that A is a symmetric positive definite, the idea is solving ψ such that the energy of the system is defined as

$$Q(\psi) = \frac{1}{2} \psi^T A \psi - b^T \psi \quad (3.73)$$

is the minimum. This is done by choosing the initial guess $\psi(0)$ and traveling some arbitrary length to the negative gradient direction, as it descends the most at that point. Mathematically we have

$$\psi^{(n+1)} = \psi^{(n)} + \alpha_n \mathbf{d}^{(n)}, \quad (3.74)$$

where α_n is the length travelled and \mathbf{d} is the search direction. Determining α_n is no trivial

task, as a bad choice might lead to a situation, where the algorithm would never find the minimum. The term is often tied to the residual $\mathbf{r} = b - A\psi$, such that as the solution gets closer to the minimum, the residual decreases, and thus the length traveled decreases. To find this criterion, we can take

$$\frac{d}{d\alpha^{(n)}} Q(\psi^{(n+1)}) = 0, \quad (3.75)$$

which finds the α_n that minimizes Q . Now expanding from this, we get

$$\alpha_n = \frac{(\mathbf{r}^{(n)})^T \mathbf{r}^{(n)}}{(\mathbf{r}^{(n)})^T A \mathbf{r}^{(n)}}. \quad (3.76)$$

As the matrix A is symmetric, the gradient of $Q(\psi)$ equates to the system $A\psi - b$, and so the negative of this gradient equates to the residual \mathbf{r} . Thus, the search direction vector can be evaluated as the residual vector, which then finalizes the method of steepest descent. The conjugate gradient method then improves this by ensuring, that each search direction is A -orthogonal to other search directions. This criterion is defined by

$$(\mathbf{d}^{(n)})^T A \mathbf{d}^{(m)} = 0, \quad (3.77)$$

where $n \neq m$. As the search directions are always unique, the solution should converge better as it won't get stuck choosing the same directions near the minimum point. Thus by this constraint, the conjugate gradient method operates with the following algorithm:

1. Chose initial direction as the initial residual $\mathbf{d}^{(0)} = \mathbf{r}^{(0)}$.
2. Solve the step length as
$$\alpha_n = \frac{\mathbf{d}^{(n)T} \mathbf{r}^{(n)}}{\mathbf{d}^{(n)T} A \mathbf{d}^{(n)}}$$
3. Solve the new solution as $\psi^{(n+1)} = \psi^{(n)} + \alpha_n \mathbf{d}^{(n)}$
4. Calculate residual $\mathbf{r}^{(n+1)} = \mathbf{r}^{(n)} - \alpha_n A \mathbf{d}^{(n)}$
5. Find the new search direction as $\mathbf{d}^{(n+1)} = \mathbf{r}^{(n+1)} + \beta_n \mathbf{d}^{(n)}$, where

$$\beta_n = \frac{\mathbf{r}^{(n+1)T} \mathbf{r}^{(n+1)}}{\mathbf{r}^{(n)T} \mathbf{r}^{(n)}}$$

6. Repeat this from step 2 until the residuals are small enough.

One downside of the conjugate gradient method is that it requires matrix A to be symmetric. For example, when discretizing the divergence term using upwind scheme (3.66), the coefficient matrix becomes asymmetric as the scheme uses the value from only one of

the neighboring cells. Thus, the resulting discretization might not always result in a symmetric coefficient matrix, which then renders the conjugate gradient method unusable. However, a modified version called the bi-conjugate gradient method (BiCG) solves this problem by solving a new system

$$\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} \hat{\psi} \\ \psi \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}, \quad (3.78)$$

where $\hat{\psi}$ is a dummy variable. The new system has a guaranteed symmetric coefficient matrix, and the normal conjugate gradient method can be used. It should be noted, that as the size is twice as large, the computational cost is increased as well. Therefore, it is preferred to use the original CG when A is known to be symmetric. A stabilized version of BiCG has been developed by Van Der Vorst [76], which is considered to be a successor for the BiCG. Here it is first noted, that the residual \mathbf{r} and the dummy residual $\hat{\mathbf{r}}$ must satisfy

$$\langle \mathbf{r}^{(n)}, \hat{\mathbf{r}}^{(m)} \rangle = 0, \quad (3.79)$$

where $\langle \cdot, \cdot \rangle$ is an inner product, and that the residuals can be written as $\mathbf{r}^{(n)} = P_n(A)\mathbf{r}^{(0)}$ where $P_n(A)$ is a n -degree polynomial. Thus we can write

$$\begin{aligned} \langle \mathbf{r}^{(n)}, \hat{\mathbf{r}}^{(m)} \rangle &= \langle P_n(A)\mathbf{r}^{(0)}, P_m(A^T)\hat{\mathbf{r}}^{(0)} \rangle \\ &= \langle P_m(A)P_n(A)\mathbf{r}^{(0)}, \hat{\mathbf{r}}^{(0)} \rangle \\ &= 0 \end{aligned} \quad (3.80)$$

which implies that $\hat{\mathbf{r}}^{(m)} = P_m^2(A)\mathbf{r}^{(0)}$. Thus, the dummy residual can be evaluated without having to do any matrix multiplication with A^T , which makes the algorithm more efficient. Van Der Vorst also noticed that if we define $Q(A)$ such that vectors $\mathbf{r}^{(m)}$ and $Q_n(A^t)\hat{\mathbf{r}}^{(0)}$ are perpendicular, then the inner product

$$\langle Q_m(A)P_n(A)\mathbf{r}^{(0)}, \hat{\mathbf{r}}^{(0)} \rangle = 0 \quad (3.81)$$

will also satisfy, i.e. the polynomial with respect to the dummy variable doesn't need to be the same as the original one. Thus, by defining

$$Q_i(A) = (I - \omega_1 A)(I - \omega_2 A) \dots \quad (3.82)$$

and choosing suitable coefficients ω_i , the algorithm can be made to converge more smoothly and rapidly.

A third technique to solve linear systems is the so-called *multigrid method*, where the idea is to agglomerate the fine mesh into a coarse one and then to "smooth" the solution, for example, by a few iterations of the Gauss-Seidell algorithm. This is then repeated until the grid is very coarse and the solution could be obtained by inverting the coefficient matrix with sufficiently low computational cost. With this solution, the fine grid is returned in reverse steps with smoothing at each level. This method is one *v-cycle* of the multigrid method and could be operated multiple times if necessary. The process is illustrated in the figure below.

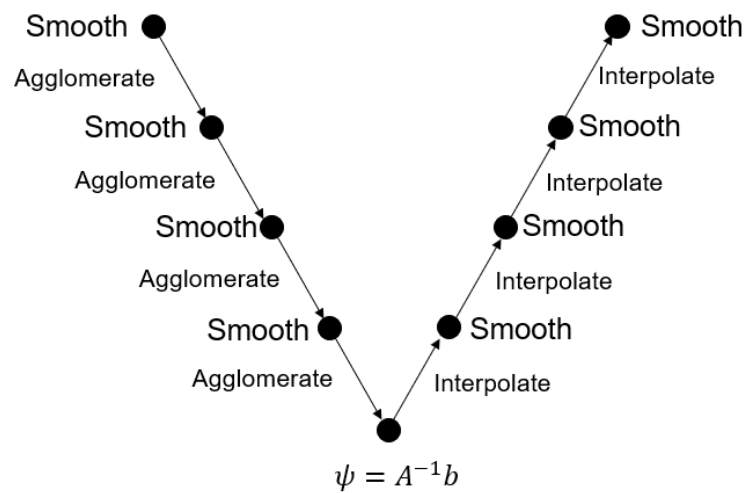


Figure 3.4. One V-cycle of multigrid method illustrated.

The reason why this method is effective comes clear when observing the evolution of error produced in for example Gauss-Seidell method. The error can be decomposed into different sine waves with different frequencies and amplitudes using the Fourier method. When a Gauss-Seidell iteration is performed, the high-frequency error modes are effectively diminished but the low-frequency modes stay more or less unaffected. Thus, more iterations are required to diminish the lower frequency modes as well, which is not efficient. This also explains why the convergence of the Gauss-Seidell method starts rapidly but the rate of convergence slows down after some iterations. However, by changing the mesh size, these low-frequency modes become relatively higher and thus Gauss-Seidell smooths them more effectively. Therefore multiple different error modes can be eliminated by going systematically from the fine mesh to the coarse mesh and smoothing the solution within each level.

Due to the nature of the process, the method is often implemented by using recursion. The agglomeration process can be done either by geometrically agglomerating nearby cells into one big cell but can be done purely algebraically where the adjacent rows of the

coefficient matrix are combined into one. The latter is preferred when the mesh is complex but the geometrically agglomerating makes more sense as the adjacent cell values are often close to each other.

Lastly, a general way to improve the convergence rate of all the mentioned methods is by altering the systems' spectral properties. This is called preconditioning, and mathematically speaking it is defined as

$$P^{-1}A\psi = P^{-1}b, \quad (3.83)$$

where P is referred to as the preconditioning matrix. The solution to this system is the same as the original. The idea is to choose such P that the solution converges faster. For example, this can be achieved by choosing P that is easily invertible and $P^{-1} \approx A^{-1}$. The Gauss-Seidel preconditioning matrix is defined as $P = D + L$, where D is the diagonal part and L is the lower triangle of A .

3.4.3 SIMPLE-algorithm

While the RANS equations with turbulence model could now be solved, no explicit equation for pressure exists. The conservation of mass rather acts as a restriction than a standalone equation of motion. Also, the same pressure term appears in all three momentum equations, making the system coupled and the solution needs to be found simultaneously. To resolve this, Patankar and Spalding [51] came up with a procedure, which later caught the name SIMPLE coming from Semi-Implicit Method for Pressure-Linked Equations. The idea of this procedure is to guess the initial solution and correct it afterward. First, we predict the momentum with the pressure from the previous iteration, then we calculate the new pressure derived from the momentum equation, and finally, we correct the velocity. Practically this is a little bit more involved as the equations need to be discretized first. The discretized momentum equation is written in the form

$$M\mathbf{u} = b - \nabla \cdot p, \quad (3.84)$$

where M consists of coefficients, \mathbf{u} is velocity, b consist contribution from source terms and p is the pressure. The pressure term is explicitly not discretized yet since we need it to derive the pressure equation. Also, we can assume that the system has no source terms, so $b = 0$. At this point, the preliminary velocity is solved by using the pressure from the previous iteration. It is noteworthy that now this velocity doesn't necessarily satisfy the continuity equation. Next, we will separate the coefficient matrix M such that the equation (3.84) turns to

$$A\mathbf{u} = H(\mathbf{u}) - \nabla p, \quad (3.85)$$

where A is the diagonal part of M and $H(\mathbf{u})$ is "what remains" i.e. the contribution from neighboring cells and the temporal discretization. The velocity solved from (3.85) yields

$$\mathbf{u} = \frac{H(\mathbf{u})}{A} - \frac{1}{A}\nabla p. \quad (3.86)$$

Now to obtain the equation for pressure, we can utilize the continuity equation (3.7) and obtain

$$\nabla \cdot \mathbf{u} = 0 \Rightarrow \nabla \cdot \left(\frac{H(\mathbf{u})}{A} \right) = \nabla \cdot \left(\frac{1}{A} \nabla p \right). \quad (3.87)$$

The equation above is in a form of a typical Laplace's equation and can be solved for pressure. Now, with the updated pressure, the velocity can be corrected with (3.86), which in theory will satisfy the momentum and mass equations. The SIMPLE algorithm is presented below:

1. Define the boundary conditions
2. Solve the momentum predictor from (3.84).
3. Relax the velocity
4. From the predicted velocity, find A , $H(\mathbf{u})$ and $(H(\mathbf{u})/A)_f$. The last term is the predicted flux across the faces.
5. Solve the pressure from (3.87) with relaxing. This step is repeated N times, where N is the number of non-orthogonal corrections.
6. Update the fluxes with new pressure
7. Relax the pressure
8. Update the velocity with new pressure
9. Update the boundary conditions
10. If solution not converged, proceed till next iteration to step 2

For curious readers, the source code for the SIMPLE algorithm implemented in OpenFOAM can be found in [68]. When deriving the pressure equation from momentum prediction, we quietly discarded the correction term for the fluxes, i.e. $H(\mathbf{u}')$, where \mathbf{u}' is the velocity correction. For the final solution, this doesn't have an effect as the term approaches zero when the solution converges. However, during the iteration phases, this term might have an impact, which then causes the solution process to slow down or even

diverge. To resolve this, pressure and velocity can be relaxed, which mathematically is defined as

$$\psi_{\text{new}} = \psi_{\text{old}} + \alpha(\psi_{\text{predicted}} - \psi_{\text{old}}). \quad (3.88)$$

The α is usually between 0 and 1 which results in the case of under-relaxation. Finding the optimum values of α is not trivial, but for example, [45] derived that $\alpha_p \approx 1 - \alpha_u$, where subscripts p and u depicts the relaxation values for pressure and velocity respectively. Also, Peric studied the behaviour of the relaxation factors in his thesis and suggests that $\alpha_p = 0.2$ and $\alpha_u = 0.8$ is valid for most cases [53].

4. SECOND GENERATION INTACT STABILITY CRITERIA

International Maritime Organisation (IMO) is a specialized agency of the United Nations and has been tasked to improve maritime safety, emission control, and other maritime-related topics. One of its ongoing developments is the so-called Second Generation Intact Stability Criteria (SGISC), which is a set of performance-based stability criteria to assess stability conditions for ship designs. The reason for this development is that the previous version of the intact stability code only assessed the stability in absence of waves. As some ship designs stability fluctuates greatly in waves due to their hull's design, this might not be sufficient enough to determine whether the design is vulnerable to wave-related phenomena such as parametric rolling. While IMO has introduced methods for assessing the maximum amplitude for parametric rolling [16], they are derived from regular wave scenarios and use calm sea hydrostatics.

The SGISC focuses on five failure event modes, which occur due to the interaction of the waves and the ship. These modes are *parametric roll*, *dead-ship condition*, *pure loss of stability*, *excessive acceleration* and *broaching/surf-riding*. This thesis primarily focuses on the parametric roll, but the guidelines and failure criteria are similar in all cases. [40]

The SGISC proposes a method, where the analysis is divided into three primary sections: *level 1*, *level 2* and *Direct Stability Assessment (DSA)*. Finally, if all of these fail, the ship design might be approved by applying operational guidance or limitations. This includes, for example, limiting the operation to a certain set of sea states or training the crew to be prepared when the ship begins to sail in dangerous territory. The chart below visualizes this design flow for each failure event. [40]

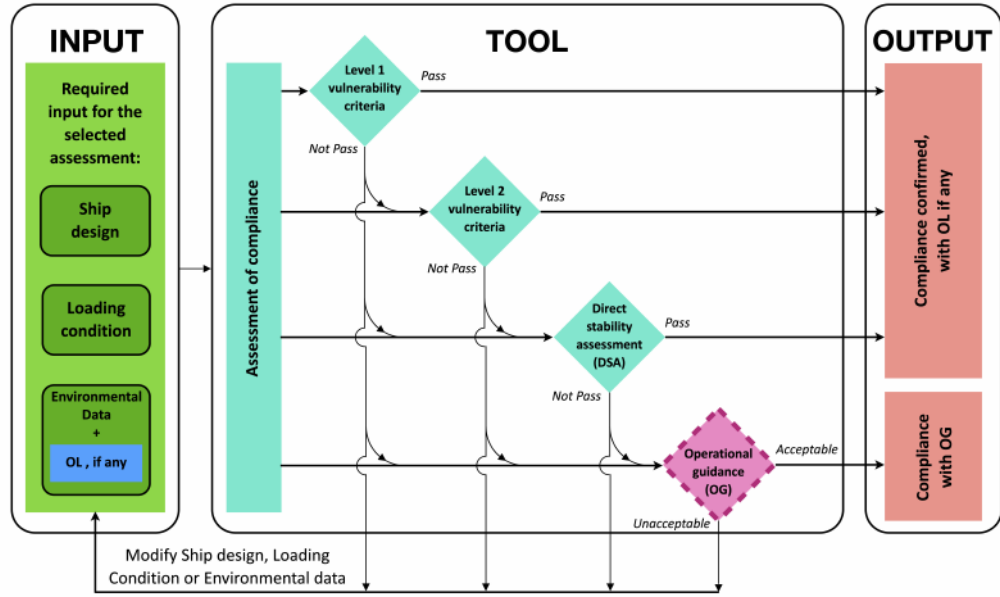


Figure 4.1. Flowchart of stability analysis in SGISC [55]

The use of the SGISC is not going to be mandatory, but it allows Administrations to assess and approve ship designs. Thus, it will be necessary for shipyards to be able to conduct these assessments. This chapter is based on the official interim guidelines [50] as well as on works [47] and [40]. Also, a brief overview of typical vocabulary and metric definitions in maritime applications has been collected to Appendix A.

4.1 Vulnerability level 1

Level 1 assessment is the most conservative and straightforward. It tests whether the design is even close to being vulnerable to the failure event with simple criteria using common hydrostatic parameters. For parametric roll, the design is not vulnerable if the fluctuation of the metacentric height (see Section 2.1) due to the interaction with waves is small enough i.e.

$$\frac{\delta \overline{GM}}{\overline{GM}} \leq R_{PR}, \quad (4.1)$$

where

$$R_{PR} = \begin{cases} 1.87, & \text{if the ship has sharp bilge} \\ 0.17 + 0.425 \left(\frac{100A_k}{LB} \right), & C_m > 0.96 \\ 0.17 + (10.625C_m - 9.775) \left(\frac{100A_k}{LB} \right), & 0.94 < C_m < 0.96 \\ 0.17 + 0.2125 \left(\frac{100A_k}{LB} \right), & C_m < 0.94, \end{cases}$$

A_k is total projected area of bilge keels, L is length, B is beam and C_m is largest section

coefficient (see Figures A.2 and A.6). The parameter R_{PR} can range from 0.17 for designs with no bilge keels up to 1.87 where the bilge itself is sharp. In equation (4.1) the \overline{GM} describes the metacentric height in calm water (absence of waves) and $\delta\overline{GM}$ is an amplitude of metacentric height variation defined as

$$\delta\overline{GM} = \frac{\overline{GM}_{\max} - \overline{GM}_{\min}}{2}. \quad (4.2)$$

where the maximum and minimum \overline{GM} are calculated from metacentre in wave with steepness coefficient s_w of 0.0167 (see beginning of Section 2.3). Simplified version of $\delta\overline{GM}$ can be obtained by

$$\delta\overline{GM} = \frac{I_H - I_L}{2\nabla} \quad \text{only if} \quad \frac{\nabla_D - \nabla}{A_w(D - d)} \geq 1, \quad (4.3)$$

where I_H is the moment of inertia of the waterplane at the wave crest, I_L is the moment of inertia of the waterplane at the wave trough (see equation (2.6)), ∇ is a volume of displacement, D is a moulded depth at the side to the main deck and ∇_D is a volume of displacement at waterline D (see Appendix A). Thus, the level 1 assessment requires the ability to calculate the moment of inertia of waterplane at some draft but otherwise is relatively straightforward.

4.2 Vulnerability level 2

If level 1 fails, then the analysis moves to level 2. This level goes more in-depth into the mathematics behind the phenomenon and includes multiple different wave cases or *sea states*. These states represent the typical types of waves that occur in a given time period. For example, the waves typically in storms have large wave height and length whereas wind that affects for short time to the water surface generates waves with small height and length. Typically sea states are presented with significant wave height H_s and peak period T_p and the wavelength can be solved from the dispersion relation (2.72) (see Subsection 2.3.3).

Due to the geometric difference, all seas have a unique set of sea states that appear more often. These states can be determined by observing the sea with buoys and constructing a density plot, which describes the probability for a given wave type to occur. For example, Finnish Meteorological Institute offers data from buoys in a form of peak period and significant wave height from 30 minute observation period [29]. For the year 2020, the density plot of the Baltic sea is presented in the figure below.

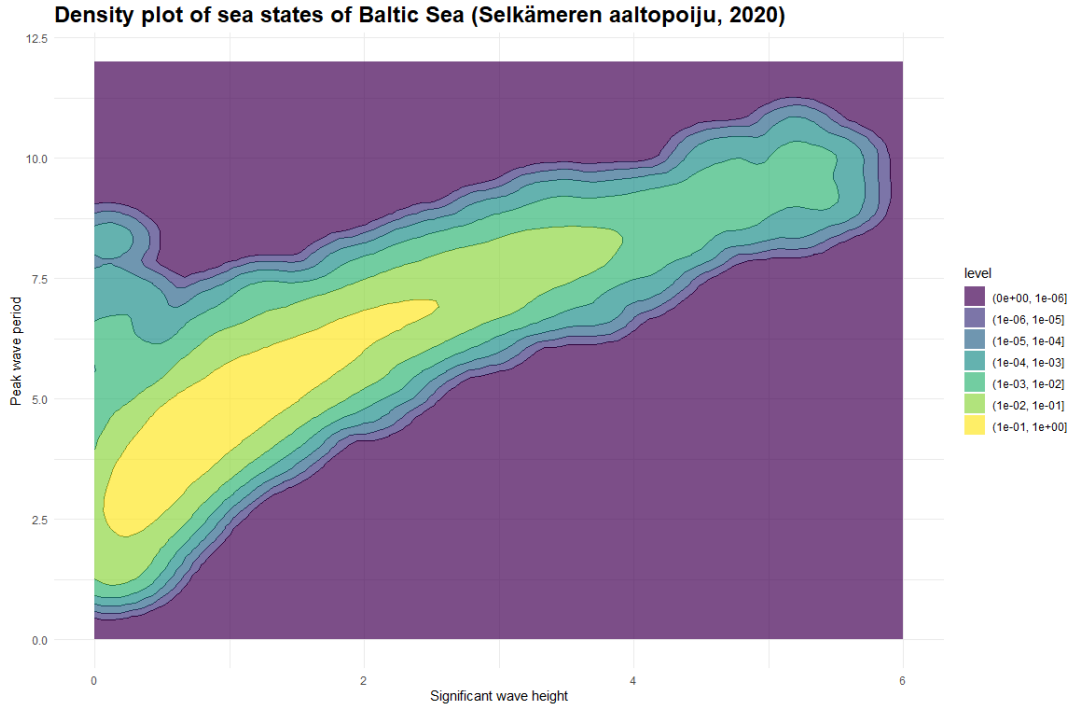


Figure 4.2. Density plot of sea states observations from Baltic sea in 2020 [29]

While a density plot gives a visual representation of the sea characteristic, a more useful format for computations is so called *wave scatter diagram*. It is essentially a discretized version of the density plot that shows the number of times a specific sea state was measured. This table can be used to discard rare sea states when conducting the analysis.

For the level 2 assessment of parametric roll, the ship design is not vulnerable if

$$\max(C_1, C_2) \leq 0.06, \quad (4.4)$$

where C_1 and C_2 are coefficients from two different type of tests. The first test coefficient C_1 tests different wave cases and calculates two criteria similar to level 1. This coefficient is calculated as

$$C_1 = \sum_{i=1}^N W_i C_{S,i}, \quad (4.5)$$

where N is total number of sea states, W_i is a weighting factor and coefficient $C_{S,i}$ is defined as

$$C_{S,i} = \begin{cases} 0, & \text{if either (4.6) or (4.7) is satisfied} \\ 1, & \text{otherwise.} \end{cases}$$

The first requirement calculates the amplitude of metacentre variation like in the vulnerability level 1 assessment. Now instead of having constant wave steepness, several different waves are imposed. The first requirement is

$$\overline{GM}(H_i, \lambda_i) > 0 \quad \text{and} \quad \frac{\delta \overline{GM}(H_i, \lambda_i)}{\overline{GM}(H_i, \lambda_i)} < R_{PR}, \quad (4.6)$$

where $\overline{GM}(H_i, \lambda_i)$ is mean metacentric height in waves with wave height H_i and wavelength λ_i , $\delta \overline{GM}(H_i, \lambda_i)$ is metacentric height variation calculated from (4.2) and R_{PR} is defined similarly as in level 1. The second requirement tests whether the ship's velocity is far enough from the range where the parametric rolling occurs most likely. It is defined as

$$u_s < \left| \frac{2\lambda_i}{T_n} \sqrt{\frac{\overline{GM}(H_i, \lambda_i)}{\overline{GM}}} - \sqrt{g \frac{\lambda_i}{2\pi}} \right|, \quad (4.7)$$

where u_s is the ship's service velocity, T_n is the natural roll frequency of the ship and \overline{GM} is metacentric height in calm sea. In absence of wave data, the table B.1 describes the default suggested cases. If other wave scatter diagram is chosen, then the weighting factor can be obtained by dividing the number of observations by the total number of observations.

The second test coefficient C_2 from (4.4) is defined as

$$C_2 = \frac{1}{25} \left(\sum_{i=1}^{12} C_{2h}(K_i Fn) + \frac{1}{2} (C_{2h}(0) + C_{2f}(0)) + \sum_{i=1}^{12} C_{2f}(K_i Fn) \right), \quad (4.8)$$

where $Fn = u_s / \sqrt{Lg}$ is a Froude number, K_i is a speed factor (see table B.2), the subscript h means simulations where ship is travelling in head seas and f to following seas respectively (see Appendix A). These are further calculated as

$$C_{2h}(Fn) = \sum_{i=1}^N W_i C_{S,i} \quad \text{and} \quad C_{2f}(Fn) = \sum_{i=1}^N W_i C_{S,i},$$

where again W_i is a weighting factor, N is the total number of cases from the wave scatter diagram and $C_{S,i}$ is either 0 or 1 depending on the simulation results. The $C_{S,i}$ can be obtained by solving a roll equation for the given sea state to determine the maximum amplitude of the oscillations. However, this would lead to many simulations given that (4.4) already has a total of 26 assessments for different velocities. Thus, for a given Froude number, a simplified method involves solving the maximum roll angle for a set of sea states defined as

$$\lambda = L \quad \text{and} \quad h_j = 0.01jL, \quad (4.9)$$

where $j = 0, 1, \dots, 10$, L is the length of the ship and balanced sinkage and trim are assumed. Now for each sea state, the representative wave height H_{r_i} is obtained by filtering waves within the ship length. Finally, the maximum roll angle for the sea state is then obtained by linear interpolation from the maximum angles calculated with wave heights h_j . If the maximum roll angle exceeds 25 degrees, then C_i is set to 1 and otherwise to 0.

4.3 Direct stability assessment

The last step of the assessment is DSA. Unlike the previous two levels, this level doesn't have a straightforward procedure, but rather the idea is to conduct a probabilistic analysis for all sea states. This practically means finding the average failure rate for any given sea state and loading condition. Failure rate means the number of failure events expected to occur within a given period. The SGISC guidelines criteria are chosen to ensure that the average failure rate would not exceed $2.6e - 03$ per ship per year. The guidelines define a failure event as a roll angle exceeding 40 degrees or lateral acceleration exceeding 9.81 m/s^2 , but administrations may make these more strict on demand. Also, the time to failure is defined as a simulation time until the failure event occurs.

To obtain the average failure rate for a given sea state, a simulation is run until a failure event occurs. The seaway modelled should be irregular and for parametric roll, the simulation should include roll, pitch, and heave motions. These simulations can be conducted by any method as long as the requirements are met. The software requirements have a section for general requirements and then specific requirements for specific failure events. The general requirements consist for example that the forces are calculated appropriately using Froude-Krylov forces defined as

$$\mathbf{F} = \int_{S_w} p \mathbf{n} dS, \quad (4.10)$$

where S_w is the submerged surface, p is the pressure acting on the surface and \mathbf{n} is the normal vector from the surface. The specific requirement for parametric roll might be that the maximum roll amplitude should not be underpredicted from the maximum angle of \overline{GZ} by more than 10%.

Computational Fluid Dynamics (CFD) is a method to solve Navier-Stokes equations computationally. These equations describe the fluid flow which is composed of velocity and pressure. Therefore CFD allows the free surface to be directly resolved and by applying a general rigid body simulation to the ship, the ship's response to the waves can be accurately obtained. In CFD, the software requirements are naturally fulfilled and only

the validation remains. However, one downfall of this method is that it is computationally expensive.

The environmental conditions depict three different ways to conduct the stability analysis: *full probabilistic*, *probabilistic criteria in design situations*, and *deterministic criteria according to design situations*. Full probabilistic assessment tests all mentioned sea states in all defined loading conditions. This inherently is very time-consuming since the number of different scenarios alone produces a massive number of simulations for one loading condition. Thus, computing a full probabilistic analysis might be computationally too expensive even for efficient methods. It also is redundant to calculate many of the mentioned sea states, as they appear rarely. IMO has proposed a method of probabilistic analysis according to different design situations. Here the sailing conditions are predetermined and for example, to parametric roll, the wave direction needs to be either head or following and the ship velocity can be kept as zero. Also, only the sea states that have probability density over 10^{-5} are included. This discards many redundant cases from the total simulation set which significantly reduces the total computational cost of the analysis.

Modelling the irregular waves consists of creating a set of random phases for multiple regular wave components and summing them together (see Subsection 2.3.3). For each set, the resulting waves might seem different even if the actual sea state remained constant. Due to this randomness, the calculated time to failure for a given sea state is only an estimate and the true time to failure lies somewhere within the confidence interval. The SGISC guidelines state that the upper boundary value should be used as the average time to failure to have a nonconservative estimate. Shigunov showed in [67], that when the number of simulations with different initial phases is increased, the mean estimated time to failure converges and the 95% confidence interval gets smaller. It is also shown in [67] that typically for any ship type, at least 200 simulations are required to get the upper boundary in 5% accuracy from the estimated value. This makes the probabilistic criteria for design states still too inapplicable for example to model tests. Therefore IMO has proposed a deterministic criterion, where for each sea state, at least a 15-hour period should be tested. This method however introduces inaccuracy and needs to be taken into account.

The process can further be accelerated with statistical extrapolation. As the cases are generated from the wave scatter diagram, most of them share a common peak period for different significant wave heights. Given that the peak period is constant, the average time to a failure T can be evaluated as

$$\ln T = A + \frac{B}{H_s^2}, \quad (4.11)$$

where A and B are model coefficients, and H_s is a significant wave height. IMO suggests

that when constructing (4.11), at least three cases are resolved where the significant wave height deviates by $2m$ from each other. Thus, when studying cases with the same peak period, only three simulations are required to be resolved and the rest of them can be extrapolated. For more extensive research from the extrapolation, readers are recommended to [67]. Other extrapolation methods could also be used as long as they have been validated and used cautiously.

5. COMPUTATIONAL SETUP

The goal of this thesis is to determine the possibility of using CFD in the DSA phase. CFD can be used majorly in two ways in parametric rolling assessment. First, the obvious way is by directly simulating the ship's motion in a given sea state to complete the DSA phase. Another way to use CFD while studying the parametric roll is to conduct a simple free decay simulation to determine important hydrostatic parameters i.e. the natural roll period and the damping coefficients. These both are conducted in this thesis. The simulations were performed using OpenFOAM and postprocessing was done with ParaView and Python (programming language).

5.1 OpenFOAM

As seen previously, the numerics of CFD calculations can be rather difficult, and creating software that can perform such simulations is far from being an easy task. Some simplified cases might be doable with computational programming languages, like Matlab or Python. However, the task of solving cases involving complex geometries quickly becomes very complicated. The software needs to be able to create meshes from complex geometries, utilize the FVM technique to solve partial differential equations, and post-process the solution. This also needs to be computationally efficient in memory and time usage while being memory safe and sufficiently simple for the user.

Multiple commercial software exists, but they can be expensive with licenses that can "range from 10000\$ to even 50000\$ per year depending on the added extras, the number of users, whether multiple licenses are required for parallel computation, and the commercial or academic nature of the license" [10]. However, with the rise in popularity of the open-source model, the CFD tools developed by Henry Weller and Hrvoje Jasak were published in 2004 with the title Open-source Field Operation And Manipulation i.e. OpenFOAM. This software has seen a lot of development since then and continues to be developed and used in the academic and industrial fields.

OpenFOAM is a C++ module containing numerical mathematics, file handling, algorithms for solving CFD-related problems, and more. Although OpenFOAM is primarily used to handle CFD-related problems, the library is general enough that solvers for other fields have also been implemented. For example, financialFoam solves the Black-Scholes equa-

tion to price commodities [49]. As OpenFOAM is an open-source project, anyone can download the source code and create models to fit their needs. These models can be kept private, but many of them also get pushed back to OpenFOAM and they get released in the official version. OpenFOAM offers good documentation for programmers and users and a lot of users help in different forums.

As of the time of writing this thesis, OpenFOAM doesn't contain any official graphical user interface, but rather pre-built Linux binaries are operated from the command line. Thus, a Linux-based computer is required, but Windows users can use Windows Subsystem for Linux [30] for instance. The typical simulation case is structured inside a folder with consistent naming described in the figure below.

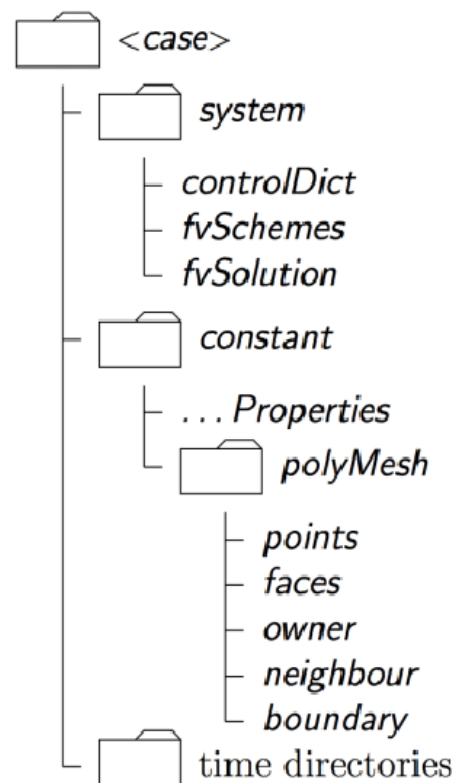


Figure 5.1. The general structure of an OpenFOAM directory [64]

The main operation is separated into three folders: system, constant and time. The system folder contains parameters defining the system parameters. Each case always contains fvSolution for numerical solution parameters, fvSchemes for discretization schemes, and controlDict for general simulation parameters. The constant folder contains the mesh and other case-dependent constant parameters, such as turbulence properties, thermo-physical properties, gravity, and so on. Finally, the different time folders contain the information from each time step. The value of the time step is the folder's name, and thus the "0" folder contains initial values as well as the boundary conditions.

The coding style of OpenFOAM has been optimised so that writing mathematical expressions within the code represents closely to the real expression. For example, the generic

scalar transport equation

$$\frac{\partial \psi}{\partial t} + \nabla \cdot \mathbf{u}\psi - \nabla \cdot (D_\psi \nabla \psi) = S_\psi, \quad (5.1)$$

for arbitrary parameter, ψ is constructed, relaxed and solved in OpenFOAM as follows.

```
fvScalarMatrix scalarEqn
(
    fvm::ddt(psi)
  + fvm::div(phi, psi)
  - fvm::laplacian(D, psi)
  ==
    fvOptions(psi)
);

scalarEqn.relax(relaxCoeff);

scalarEqn.solve();
```

Figure 5.2. OpenFOAM coding style demonstrated

The software automatically discretizes, solves the system, and stores the result in the "psi" named variable. The different schemes are defined in the system folder but can be explicitly set within the code if desired. This kind of style makes it easy to implement expressions and is easily understandable by someone not so profound in programming.

OpenFOAM itself offers a wide variety of postprocessing modules, but they only save the data into a file and thus need further manipulation to gain visual presentation. However, to visualize the results in a cell-by-cell manner, OpenFOAM has developed a module in ParaView such that it can read OpenFOAM cases. ParaView itself is a postprocessing software that gives freedom for users to define methods while offering itself multiple different ways to data visualization.

The following subsections cover some important submodules that are utilized in the practical part of the thesis. The idea of these is to cover the physics of the modules as well as familiarize the programming style of OpenFOAM. The Subsection 5.1.2 also gives an overview of the process of implementing a custom submodel.

5.1.1 Rigid body dynamics

One of the most appealing reasons to use CFD for DSA analysis is that as the fluid motion is directly resolved, the ship's response to the surrounding fluid can be accurately resolved. To update the ship's position, a generic submodel on rigid bodies is can be applied. By default, OpenFOAM offers two kinds of rigid body motion solvers: *rigidBodyDynamics* and *sixDoFRigidBodyMotions*. The rigidBodyDynamics model is based on the idea that the rigid body is composed of multiple bodies, which are linked together with joints. On the other hand, sixDoFRigidBodyMotion solves the movement of one rigid

body from the forces and moments caused by the fluid interacting with the body. The latter of these is used in this thesis as now the ship can be thought of as one rigid body. For more information about the rigidBodyDynamics method, the reader is advised to the source code of OpenFOAM [70] and the works of Featherstone [19].

The basic idea of rigid body simulations is to utilize Newton's second law to find the displacement and rotation matrix for the new timestep. Newton's second law is given as

$$\sum \mathbf{F} = m\mathbf{a}, \quad (5.2)$$

where \mathbf{F} is the force component, m is the mass and \mathbf{a} is the acceleration. The forces are divided into two components: fluid flow-induced and external. The fluid flow-induced forces are caused by the flow pushing and shearing the body and are often referred to as buoyant forces. The surface normal forces in a single face are defined as

$$\mathbf{F}_n = \rho \mathbf{S}_f (p_f - p_{\text{ref}}), \quad (5.3)$$

where the p_{ref} is the ambient pressure. The surface tangential force, on the other hand, is defined as

$$\mathbf{F}_p = \mathbf{S}_f (\mu \mathbf{R}_{\text{dev}}), \quad (5.4)$$

where μ is dynamic viscosity and \mathbf{R}_{dev} is deviatoric stress tensor. Similarly, the rotational forces i.e. the moments are calculated as a typical torque

$$\boldsymbol{\tau} = \mathbf{r} \times \mathbf{F}, \quad (5.5)$$

where \mathbf{r} is the distance from the face to the centre of rotation referred to as a moment arm and $\mathbf{F} = \mathbf{F}_n + \mathbf{F}_p$. To get the total force and moment affecting the faces, the effect from the individual faces is summed up.

With the calculated forces, equation (5.2) is used to find the acceleration of the rigid bodies. By choosing a small timestep, we can assume that the acceleration is constant from t_{old} to t_{new} . The velocity and displacement can now be evaluated directly as

$$\mathbf{u} = \int_{t_{\text{old}}}^{t_{\text{new}}} \mathbf{a} dt \implies \mathbf{u}_{\text{new}} \approx \mathbf{u}_{\text{old}} + \mathbf{a} \Delta t \quad (5.6)$$

and

$$\mathbf{x} = \int_{t_{\text{old}}}^{t_{\text{new}}} \mathbf{u} dt \implies \mathbf{x}_{\text{new}} \approx \mathbf{x}_{\text{old}} + \mathbf{u}_{\text{old}}\Delta t + \frac{1}{2}\mathbf{a}\Delta t^2. \quad (5.7)$$

The sixDoFRigidBodyMotion model allows the user to use slightly modified versions, for example, Newmark's method [46], where parameters γ and β are defined to relax the equations. The velocity is evaluated as

$$\mathbf{u}_{\text{new}} = \mathbf{u}_{\text{old}} + (1 - \gamma)\mathbf{a}_{\text{old}}\Delta t + \gamma\mathbf{a}_{\text{new}}\Delta t \quad (5.8)$$

and displacement is evaluated as

$$\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{old}} + \mathbf{u}_{\text{old}}\Delta t + \left(\frac{1}{2} - \beta\right)\mathbf{a}_{\text{old}}\Delta t^2 + \beta\mathbf{a}_{\text{new}}\Delta t^2. \quad (5.9)$$

These apply to rotational movement also. In this case, instead of mass or forces, we talk about inertia or moments respectively. The Newmark's method implemented in OpenFOAM is shown as follows.

```
// Update the linear acceleration and torque
updateAcceleration(fGlobal, tauGlobal);

// Correct linear velocity
v() =
    tConstraints()
    & (v0() + aDamp()*deltaT*(gamma_*a() + (1 - gamma_)*a0()));

// Correct angular momentum
pi() =
    rConstraints()
    & (pi0() + aDamp()*deltaT*(gamma_*tau() + (1 - gamma_)*tau0()));

// Correct position
centreOfRotation() =
    centreOfRotation0()
    + (
        tConstraints()
        & (
            deltaT*v0()
            + aDamp()*sqr(deltaT)*(beta_*a() + (0.5 - beta_)*a0())
        )
    );

// Correct orientation
vector piDeltaT =
    rConstraints()
    & (
        deltaT*pi0()
        + aDamp()*sqr(deltaT)*(beta_*tau() + (0.5 - beta_)*tau0())
    );
Tuple2<tensor, vector> Qpi = rotate(Q0(), piDeltaT, 1);
Q() = Qpi.first();
```

Figure 5.3. Newmark's method implementation in OpenFOAM

Finally, now the body can be translated and rotated. However, just directly moving the body would create problems as the cells neighbouring the body would get detached or overlap with the body. Thus, the sixDoFRigidBodyMotion resolves this by morphing the mesh near the body as well. Two parameters, innerDistance d_i and outerDistance d_o are

introduced with the following purpose. Given that d is the shortest distance from some point to the rigid body, the point is moved as

- If $d < d_i$, then the point is moved in a similar fashion as the rigid body.
- If $d_i < d < d_o$, then the point is moved using spherical linear interpolation with the interpolation factor defined as the relative distance.
- If $d > d_o$, then the mesh is not moved.

This is developed by calculating a scale factor that is 1 at d_i and linearly decreases to 0 at d_o . Then using this scale factor, the point transformations can be done with the following code.

```
// Calculate the transformation seption from the initial state
septionion s
(
    centreOfRotation() - initialCentreOfRotation(),
    quaternion(Q().T() & initialQ())
);

tmp<pointField> tpoints(new pointField(initialPoints));
pointField& points = tpoints.ref();

forAll(points, pointi)
{
    // Move non-stationary points
    if (scale[pointi] > SMALL)
    {
        // Use solid-body motion where scale = 1
        if (scale[pointi] > 1 - SMALL)
        {
            points[pointi] = transform(initialPoints[pointi]);
        }
        // Slerp septionion interpolation
        else
        {
            septionion ss(slerp(septionion::I, s, scale[pointi]));

            points[pointi] =
                initialCentreOfRotation()
                + ss.invTransformPoint
                (
                    initialPoints[pointi]
                    - initialCentreOfRotation()
                );
        }
    }
}
```

Figure 5.4. Mesh transformation function in OpenFOAM

The model also allows the definition of restraints and constraints. Typical restraint would be for example a spring that creates reaction force whenever the body is not at its equilibrium. Constraints can be used to disable some degrees of motion which can be useful when simplifying the problem.

5.1.2 Wave damping with forcing zones

In many CFD marine applications, water waves are an important part of the simulation. Waves can either be explicitly created at the boundaries or can appear in the form of a wake when a ship travels through a fluid. As these waves eventually reach the domain

boundary, some fraction of them might get reflected due to boundary conditions. This has the effect of creating an unwanted error in the actual calculations. Increasing domain size helps, but as it increases computational complexity, other methods are more tempting. This subsection is a theoretical overview of the implemented forcing zones, which are further studied in the Ph.D. thesis written by Peric [54].

Generally, wave damping can be divided into *active* and *passive* methods. In passive damping methods, the waves are damped passively, for example, by including a gradually increasing slope at the bottom of the domain. This has the effect of waves breaking like on a real beach. Another passive method is the so-called "numerical beach", where the grid is stretched near the boundary in hopes of increasing numerical diffusion error. While this might dampen the waves, it is often not wise to purposely introduce numerical error into the simulation, as this can cause unwanted effects. Thus, passive types are seldom used in reality and the focus is shifted to active wave-damping techniques.

Active damping methods include finding a way that actively changes the part of the simulation to dampen the wave. This can be, for example, a specific absorbing boundary condition that has the idea of containing values derived from the desired analytical solution. This can have the effect of no reflection, but practically implementing such a condition can be tricky due to numerous factors. Another active damping method is the so-called forcing zones where the solution is gradually forced into a reference solution. For example, a reference solution might be calm water conditions where the vertical velocity of the fluid is zero and the free surface height is specified. However, if done improperly, the zone itself can act like a wall and reflect the wave even sooner. In this thesis, a module to include a forcing zone is developed in OpenFOAM.

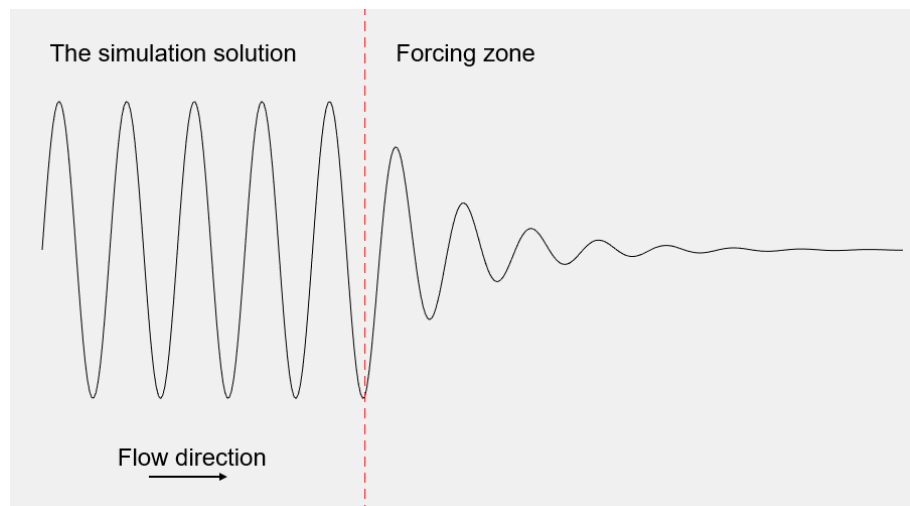


Figure 5.5. The idea of forcing zones, where the solution is gradually forced into a reference solution.

Including a forcing zone to the CFD workflow can be achieved with a source term defined as

$$f_{\text{source}} = \gamma b(x)(\psi_{\text{ref}} - \psi), \quad (5.10)$$

where γ is a constant representing forcing strength, $b(x)$ is a blending function, ψ is the flow variable which is calculated and ψ_{ref} is a desired solution for the flow variable. Peric introduced multiple choices of blending functions and concluded that no universal optimum exists while constant and linear should be avoided [54]. Thus, the exponential blending function is chosen as

$$b(\mathbf{x}) = \frac{e^{(\lambda-d(\mathbf{x}))/\lambda} - 1}{e^1 - 1}, \quad (5.11)$$

where λ is the width of the forcing zone and d is the distance to the boundary from point \mathbf{x} .

Damping waves can be achieved by having a reference solution with no vertical velocity i.e. $u_z = 0$. To implement this in OpenFOAM, the modified version of *interFoam* is created with a modified momentum equation to include the source term (5.10). First before any actual iterations, a variable $w(\mathbf{x}) = \gamma b(\mathbf{x})$ is calculated for each cell.

```
forAll(cells, id)
{
    const scalar d1 = Foam::sqrt(Foam::pow(xMax - cells[id][0], 2));
    const scalar d2 = Foam::sqrt(Foam::pow(yMax - cells[id][1], 2));
    const scalar d3 = Foam::sqrt(Foam::pow(xMin - cells[id][0], 2));
    const scalar d4 = Foam::sqrt(Foam::pow(yMin - cells[id][1], 2));

    const scalar d = min(d1, min(d2, min(d3, d4)));

    if (d <= lambda) {
        w[id] = gamma * min(
            max(
                (Foam::exp(Foam::pow((lambda - d) / lambda, 1)) - 1) / (Foam::exp(1.0) - 1),
                0
            ),
            1
        );
    }
}
```

Figure 5.6. Calculation of forcing zone parameter

Now, when constructing the momentum equation, the source term is applied simply as

```
forAll(mesh.C(), i) {
    UTarget[i] = vector(U[i].x(), U[i].y(), 0.0);
}

fvMatrix<vector> dampSource
(
    UTarget*w*rho - fvm::Sp(rho*w, U)
);

fvVectorMatrix UEqn
(
    fvm::ddt(rho, U) + fvm::div(rhoPhi, U)
    + MRF.DDt(rho, U)
    + turbulence->divDevRhoReff(rho, U)
    ==
    dampSource
    + fvOptions(rho, U)
);
```

Figure 5.7. The source term inclusion

Otherwise, the module is analogous to the default *interFoam*. The module is controlled with the entry "relaxFieldFactors" inside the constant directory. Here the boundary coordinates and the forcing zone parameters are set. Choosing the optimum γ and λ is further discussed in [54], but finding the values by trial and error can also produce acceptable conditions.

5.2 Simulation Setups

The ship model for the calculations was provided by RMC in an STL format. The ship is a concept hull with an "axe bow" and bilge keels removed to make the ship less effective in damping the oscillations. This ship type is typically not considered vulnerable to parametric rolling, but the simulations offer a further understanding of the practicality of the assessment. The ship design is presented in the figure below.

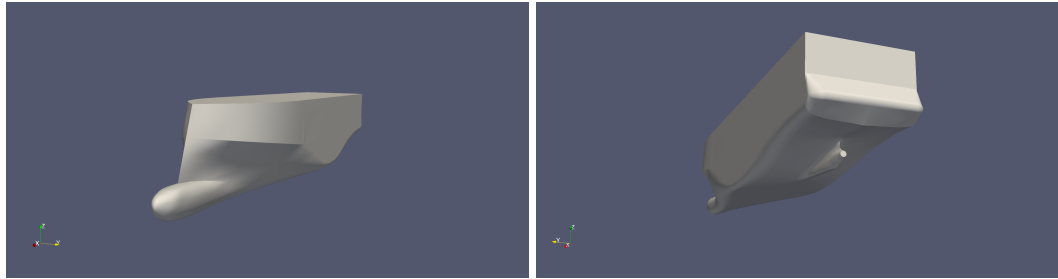


Figure 5.8. The model used in calculations

The loading condition is chosen so that the draft $T = 5m$ and the ship is at an even keel. The \overline{GM} is chosen as smaller than a typical value would be as it increases the vulnerability of parametric rolling. The table below describes some of the hydrostatic parameters for the ship.

	Model scale	Full scale
Scaling factor λ	$\frac{1}{15}$	-
Length overall L_{OA}	7.1 m	105.5 m
Beam B	1.2 m	18.4 m
Displacement Δ	1530 kg	5164 t
Roll inertia I_x	310 kg m ²	2.25e08 kg m ²
\overline{KM}	0.574 m	8.61 m
\overline{GM}	0.074 m	1.11 m

Table 5.1. Hydrostatic parameters for the ship's model

The solver used in the simulation is an in-house version of *interFoam* named *interFoam-Damping*, where waves can be damped with forcing zones explained in Subsection 5.1.2.

The interFoam is a multiphase transient solver that utilizes the volume of fluid method to resolve the free surface. Here a variable α is defined as a fraction function that allows tracking the volume of certain fluid. This variable is evolved in time by equation

$$\frac{\partial \alpha}{\partial t} + \mathbf{u} \cdot \nabla \alpha = 0, \quad (5.12)$$

which represents the typical transport equation with only the convection term. This makes α follow the velocity field streamlines. As α is a fraction function, it is required to be bounded within interval $[0, 1]$ which can be achieved by applying limiting factors to the variable flux. OpenFOAM uses a *Multidimensional Universal Limiter for Explicit Solution* (MULES) method to ensure boundedness for which more information can be found in Santiago's Ph.D. thesis [41]. With the α variable, two-phase flows resolve the density and viscosity in the cell simply by

$$\begin{aligned} \rho &= \alpha \rho_{\text{water}} + (1 - \alpha) \rho_{\text{air}} \\ \mu &= \alpha \mu_{\text{water}} + (1 - \alpha) \mu_{\text{air}}, \end{aligned} \quad (5.13)$$

where now $\alpha = 1$ means that cell contains water, $\alpha = 0$ means that cell contains air and when $0 < \alpha < 1$, then the cell is mixture of both phases.

Instead of solving the pressure p directly, interFoam solves alternative pressure defined as

$$p_{\text{rgh}} = p - \rho(\mathbf{g} \cdot \mathbf{h}), \quad (5.14)$$

which is a pressure without hydrostatic contribution. In OpenFOAM, this variable is labeled as "p_rgh". The field \mathbf{h} represents the height from a reference level, and for air-water simulations, it is usually set to the free surface. Disabling the hydrostatic contribution from the calculations makes the postprocessing and setting of the boundary conditions easier.

In free roll decay, the ship is inclined to some initial angle ϕ_0 and then is released. The ideal result is that the ship starts to roll towards its upright position and continues to the other side to some angle ϕ_1 , and then repeats the process. If the roll angle was plotted with time, the resulting figure would represent a damped oscillation. This simulation can be used to determine two parameters: natural roll period T_n and roll damping coefficients. The natural roll period is a time period where the ship naturally rolls from one maximum to the opposing side and back. While it can be approximated by empirical methods, the CFD method is the most accurate numerical one as it doesn't have any assumptions on

the hull form. For example, ITTC approximates the natural roll frequency as

$$T_n = \frac{2\pi}{\omega_n} = \frac{2\pi}{\sqrt{mg\overline{GM}/(I_x + J_x)}}, \quad (5.15)$$

where m is mass, \overline{GM} is initial metacentric height, I_x is inertia around roll axis and J_x is hydrodynamic roll moment of inertia. The roll damping coefficients on the other hand are used when simulating the roll by directly solving the roll equations. The roll-damping coefficients are often separated as

$$B(\dot{\phi}) = B_1\dot{\phi} + B_2|\dot{\phi}|\dot{\phi} + B_3\dot{\phi}^3 + \dots \quad (5.16)$$

where the B_i are constant coefficients. These can be obtained by comparing the logarithmic decrement of two adjacent amplitudes and fitting a curve into these. Since this thesis does not cover this, the readers are advised to the Ph.D. thesis [56] for further reading.

In the parametric roll simulations, the ship is also inclined by a small angle ϕ_0 and released into oscillating motion, while the simulation boundaries generate the specified sea state. The idea is to track the roll angle for enough time to determine the maximum angle. In the irregular wave scenario, the free was tracked at a point two meters before the ship. The point is far enough such that the radiation and diffraction-induced waves don't have an effect.

5.2.1 Free roll decay

In this thesis, free roll decay was conducted in model and full scale. For both cases, the system settings were the same with minor exceptions. The initial roll angle was chosen as $\phi_0 = 2$ degrees and the solution was obtained for roll and heave motions.

The turbulence was resolved using $k - \omega$ SST developed by Menter [42]. The oscillating ship can produce some small amplitude waves that travel to the boundaries and might get reflected, disturbing the free oscillation. However, these waves were effectively damped using the forcing zone technique explained in Subsection 5.1.2. The timestep Δt was chosen as constant 0.0005 instead of dynamically modifying it with CFL criteria. This criterion ensures that if a fluid particle were observed, it wouldn't skip any cells between iterations [45]. Thus, the CFL criterion would lead to a relatively large timestep value since the fluid motion is mostly stationary. This can be problematic for the rigid body model, which requires a small timestep to accurately calculate the accelerations.

The discretization scheme was mostly second-order while limiting the cell values to ensure the boundedness of the extrapolation. The pressure term was calculated using

GAMG, and velocity and turbulence parameters by Gauss-Seidel. For the mesh motion, the sixDoFRigidBodyState module was utilized.

Due to the nature of the case, the mesh could be kept relatively coarse and the domain small. The model scale mesh had a total of $1.2M$ cells, where roughly 94% of the cells were hexahedral, and the full-scale simulation had roughly $2M$ cells, with 93% of the cells being hexahedral. The hull and the waterline were refined to increase the accuracy in capturing the free surface. Both of the meshes had less than 70 of maximum non-orthogonality and 4 of skewness. Thus, the mesh quality is sufficient.

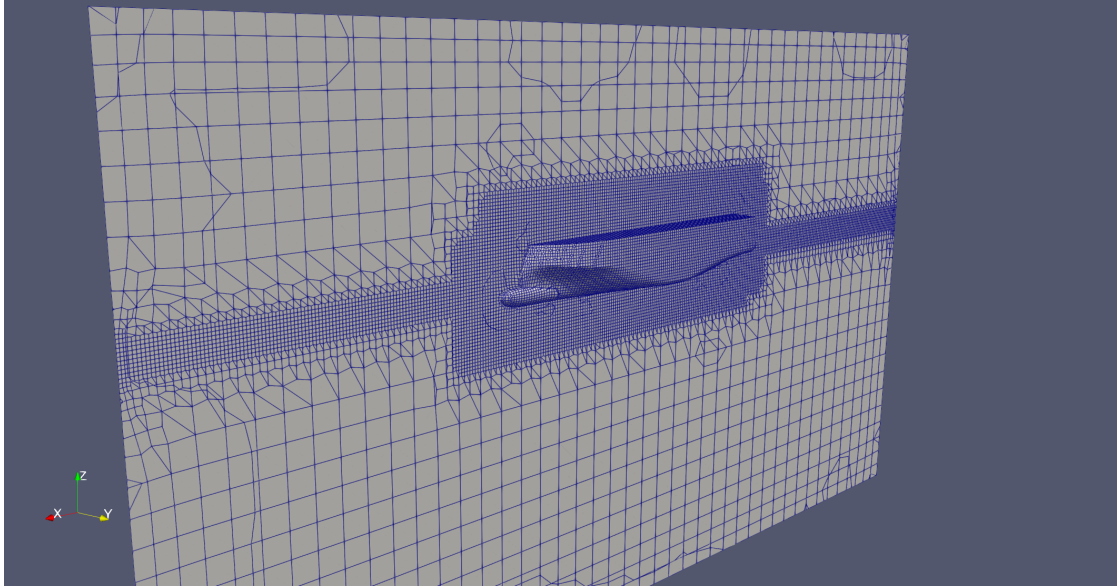


Figure 5.9. The free decay simulation mesh in model scale

The domain is defined as the top boundary for a pressure outlet and the rest of the surfaces are defined as typical wall boundaries. The specific boundary conditions are listed in the table below. For an in-depth explanation of the specific condition, the reader is referred to the OpenFOAM documentation [48].

	Walls	Top
U	noSlip	pressureInletOutletVelocity
p_{rgh}	fixedFluxPressure	totalPressure
k	kqRWallFunction	inletOutlet
ω	omegaWallFunction	inletOutlet
ν_t	nutkRoughWallFunction	zeroGradient
α	zeroGradient	inletOutlet

Table 5.2. Boundary conditions for free decay simulations

5.2.2 Parametric roll

Three different types of parametric roll simulations were conducted: regular waves in model scale, regular waves in full scale, and irregular waves in model scale. The goal of the simulations is to simulate for long enough so that the phenomenon is visible when the roll angle is plotted with respect to time. The simulations were solved for roll, heave, and pitch motions. Yaw motion was explicitly constrained but some yaw motions are going to appear from the combination of roll and pitch.

For regular waves, a worst-case scenario was created to determine the ship's ability to resist the parametric roll phenomenon. This also acts as proof of concept to determine whether this method captures the phenomenon. A worst-case scenario is defined when the wavelength is roughly the ship's length i.e. $\lambda_w \approx L_{OA}$, and the wave encounter period is half of the ship's natural roll period i.e. $T_e = 0.5T_n$. The encounter period is defined by the angular frequency as

$$\omega_e = \omega_w - \frac{\omega_w^2 u_s \cos \mu}{g}, \quad (5.17)$$

where μ is the encounter angle, such that $\mu = 0$ in following waves. Since the waves are related by dispersion relation (2.72), defining a regular wave can be achieved with wave period and wave height. To obtain the parameters for the worst-case scenario, the following algorithm can be used.

1. Calculate the wave period T_w that will satisfy the dispersion relation (2.72) such that $\lambda_w \approx L_{OA}$.
2. Calculate the wave height from the steepness s as $H_w = Ls$, where L is the length of the ship.
3. Note that $\omega_w = 2\pi/T_w$ and $\omega_e = 2\pi/(0.5T_n)$.
4. Calculate the ship velocity as

$$u_s = \frac{g}{\omega_w^2}(\omega_e - \omega_w)$$

The velocity is computed as 0.424 m/s for model-scale simulation and 1.196 m/s for full-scale simulation. For the irregular wave scenario, the ship's velocity was set to zero as stated in the SGISC guidelines. In the simulations, the natural roll period is obtained from the free decay simulations and steepness is chosen small enough so that first-order Stokes waves can be used.

For the irregular waves, two different simulations were conducted in model scale to get some reference on different cases. Both of the simulations had a peak period of $T_w \approx$

2.2 s so that the majority of the waves simulated will be approximately the length of the ship, and the significant wave heights were 0.2 m and 0.5 m . The components of each wave are created by the method of spectral analysis explained in Subsection 2.3.3. The ship was kept stationary and the mean wave direction is head waves.

The turbulence was resolved with $k - \omega$ SST and the initial angle $\phi_0 = 2$ degrees. The waves were damped with forcing zones only at the end of the domain as the damping since sides were modelled as a slip condition which mostly guided the flow towards the back of the domain. Time-step was dynamically modified to ensure that the maximum Courant number wouldn't exceed 1.5. The discretization and numerical settings were chosen the same as in the free decay simulations.

The mesh had to be further refined in parametric roll simulations than in free decay since the fluid had intentional motion. The after-side of the domain was extended to include a bigger area for the wake. An anisotropic refinement was also applied to the free surface level to increase the precision without increasing the cell count. Thus the total cell count was roughly 2.8 M cells in the regular wave case and 3.6 M in the irregular wave case. The maximum non-orthogonality was below 70 degrees and skewness below 4.

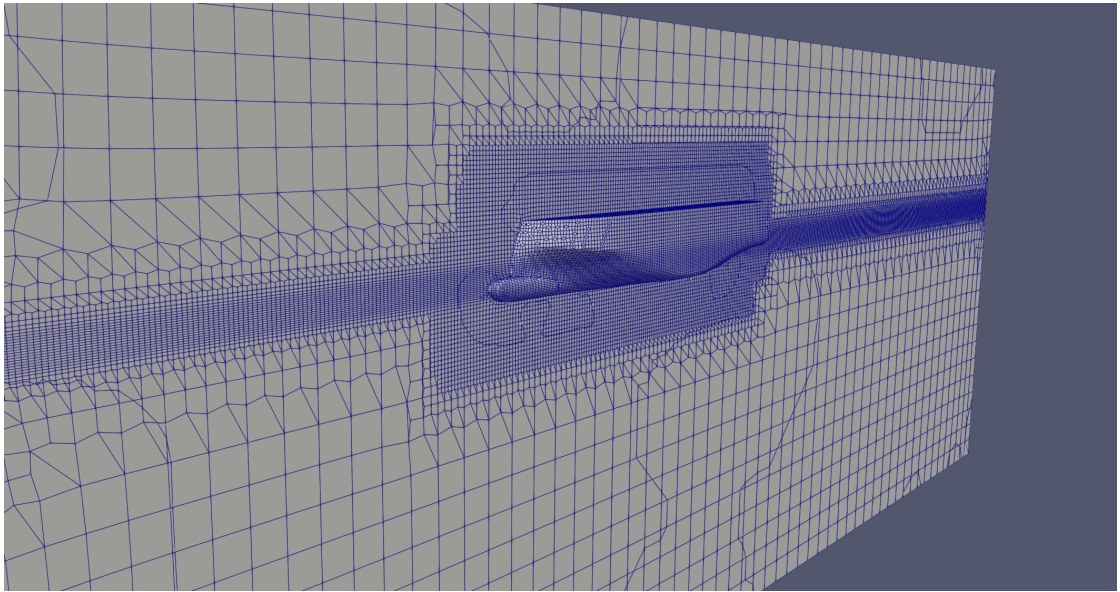


Figure 5.10. *The parametric roll simulation mesh*

The idea of the boundary conditions was to have a velocity inlet, pressure outlet, and symmetry for the side boundaries. The specific boundary conditions are listed in the table below.

	hull	inlet	outlet	sides	atmosphere	bottom
U	mWV	wV	zG	sP	pIL	nS
p_{rgh}	fFP	fFP	tP	sP	tP	fFP
k	kqRWF	fV	iO	sP	iO	kqRWF
ω	oWF	fV	iO	sP	iO	oWF
ν_t	nkRWF	fV	zG	sP	zG	nkRWF
α	zG	wA	zG	sP	iO	zG

Table 5.3. *Boundary conditions for parametric roll*

The acronyms for table 5.3 are as follows: mWV is movingWallFunction, wV is waveVelocity, zG is zeroGradient, sP is symmetryPlane, pIL is pressureInletOutletVelocity, nS is noSlip, fFP is fixedFluxPressure, tP is totalPressure, kqRWF is kqRWallFunction, fV is fixedValue, iO is inletOutlet, oWF is omegaWallFunction, nkRWF is nutkRoughWallFunction, and wA is waveAlpha. The specific action of each condition can be read from OpenFOAM documentation [48].

6. RESULTS

The nature of CFD calculations being computationally expensive became evident in this thesis. This was due to the system used having only 12 computational cores, which is relatively low for these types of simulations. The results are shown in a figure with four plots: roll, heave, pitch, and yaw motions. Each plot represents the respective motion as a function of simulation time.

In the free decay simulation, the ship was inclined two degrees and then released. The result of the model scale simulation is presented in the figure below.

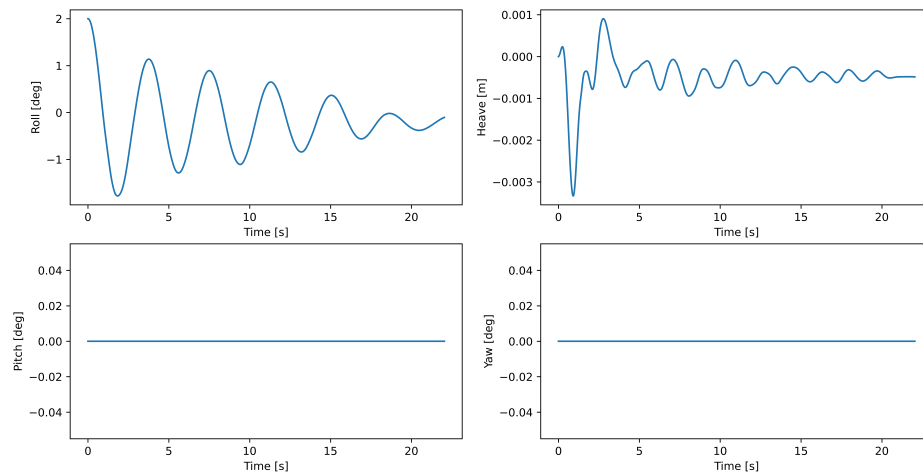


Figure 6.1. Free roll decay simulation in model scale

The simulation took roughly one day to complete for the system described above. As the simulation had two degrees of freedom, the movement only occurs in heave and roll motions. Altogether, the ship acted as a damped harmonic oscillator which is shown by the roll motion. The time between two local maxima was measured to calculate the natural roll period. Since these were not the same for each period, the average from all periods was taken. The average roll period is calculated as 3.720 s . For reference, the equation (5.15) estimates the roll period for this ship as 3.710 s .

Next, a free decay simulation was conducted in full scale for which the result is presented in the figure below.

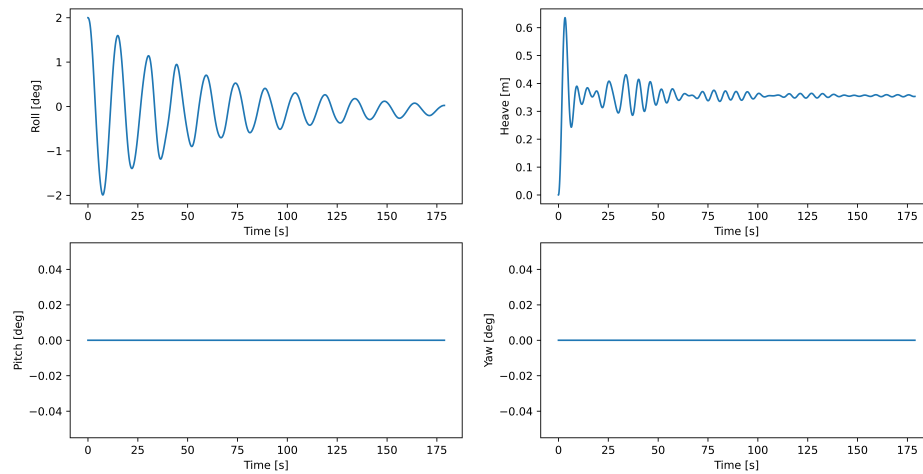


Figure 6.2. Free roll decay simulation in full scale

In the full-scale simulation, the roll angle also acted as a damped harmonic oscillator. The simulation was run for a much longer time since the roll period naturally increases as the ship's size increases. The average natural roll period was measured as 14.90 s . For comparison, if the model scale natural roll period is scaled to full scale using Froude scaling laws, the estimated roll period equals 14.41 s , which is relatively close to the measured value.

The parametric roll simulations were more time-consuming than the free decay simulations. The model scale motion plot is shown in the figure below.

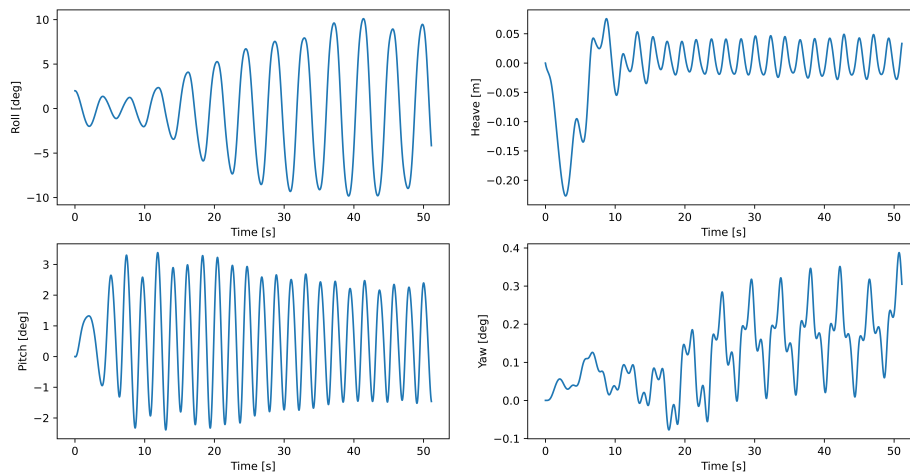


Figure 6.3. The state of motion in model scale parametric roll simulation for waves with $s = 0.0285475$

The roll angle starts at the initial angle $\phi_0 = 2$ degrees and starts to heave normally around its upright position. After around 7 seconds, the maximum amplitude of each period starts to gradually increase. After around 40 seconds, the maximum amplitude

settles to a maximum amplitude of 10 degrees, which any following period will also follow.

The ship responds to the waves by pitching upwards when the wave crest is at the bow, and downwards when the trough is at the bow. Thus, since the wave period is regular, the ship experiences a regular pitching oscillation with the same period as the waves. The pitching in Figure 6.3 shows that the hull indeed encounters a wave roughly twice every period, which creates the resonance effect.

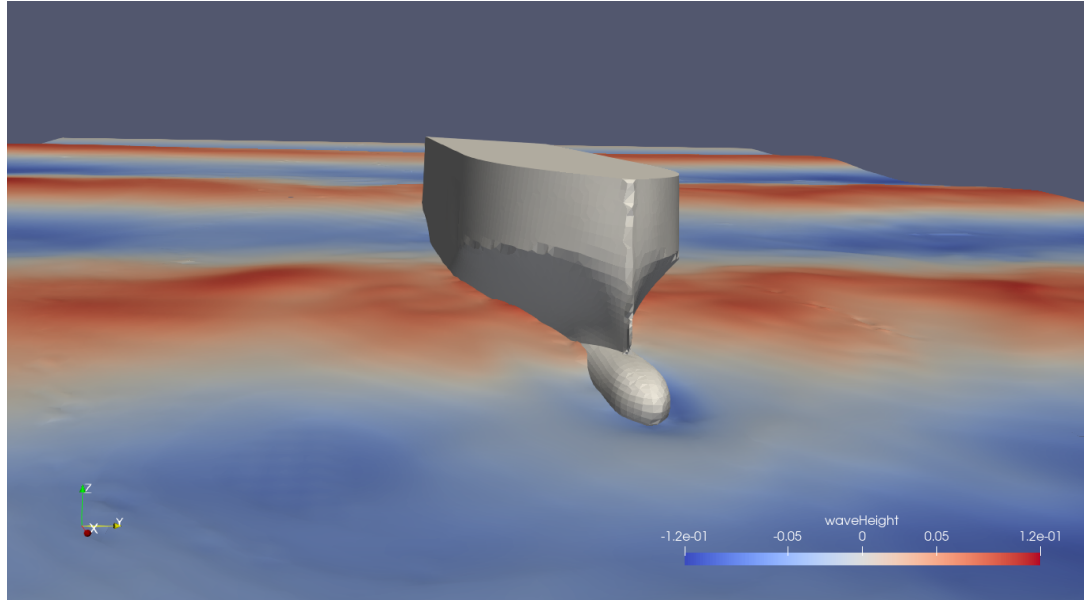


Figure 6.4. The simulation state at $t = 51$ s

The result of the full-scale simulation is shown in the figure below.

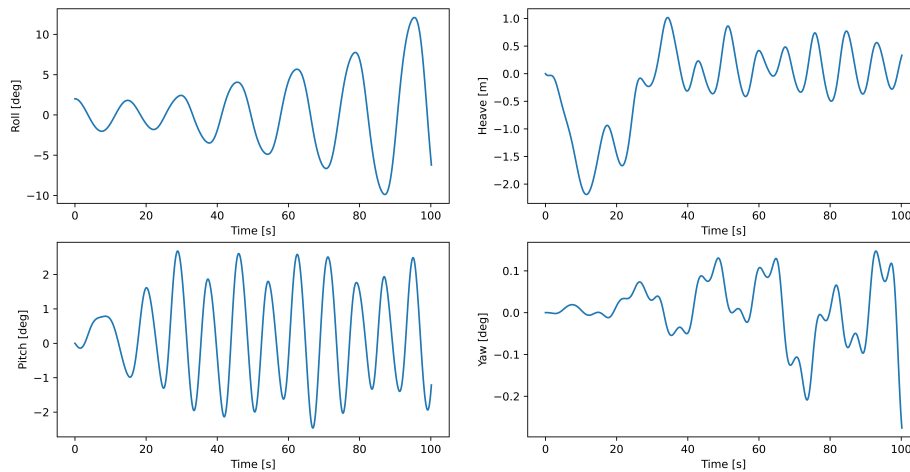


Figure 6.5. The state of motion in full-scale parametric roll simulation

Like in model scale, the initial roll angle is exceeded after a couple of oscillation periods and the pitching period is half of the rolling period. As expected, the full-scale ship moves more slowly compared to the model scale. Thus, the total simulation time needs to be

greater to find the maximum amplitude.

The simulation began promisingly, but crashed with no apparent explanation at the 100-second mark. The residuals showed normal behaviour and nothing else indicates the reason for this crash either. The most probable cause is the altered mesh having a negative effect on the simulations. Since time was limited and the irregular section more important, the focus was shifted to the next scenario.

The last simulation scenario was a model-scale parametric roll in the irregular waves. The wave components were created using the Pierson Moskowitz spectrum (2.91) for a total of 57 periods and the mean wave direction as head waves. For the first simulation, the resulting motion is shown in the figure below.

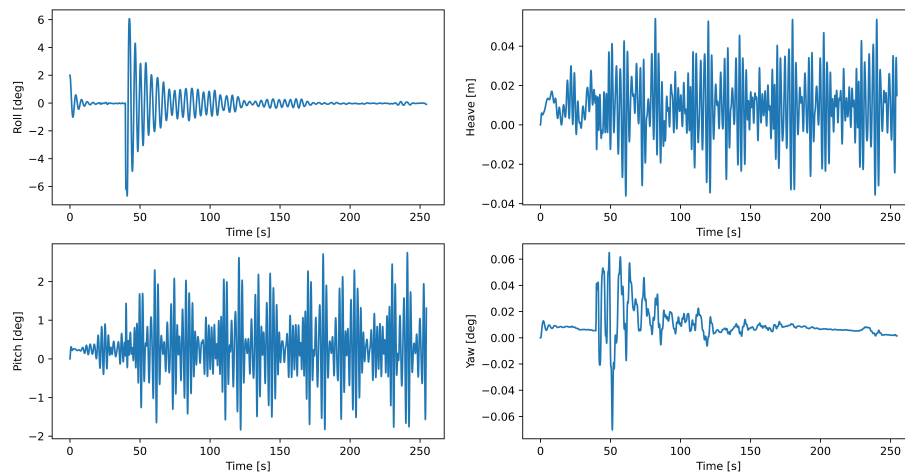


Figure 6.6. Parametric roll in irregular wave with $T_p = 2.12\text{ s}$ and $H_{1/3} = 0.2\text{ m}$.

The simulation took well over three weeks to produce a total of 251 seconds of simulated time. This was sufficiently long for the ship to encounter a sufficient amount of waves. The wave model used an 18-second ramp time which increases the simulation stability. After this, the heave and pitch showed that the irregular waves were fully developed. The free surface was tracked at a point near the ship which is shown in the figure below.

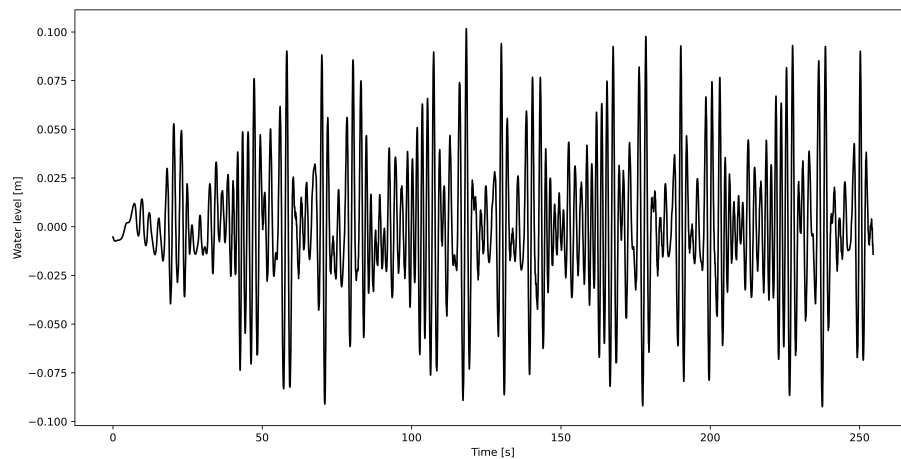


Figure 6.7. *Free surface tracking in the first irregular simulation*

One notable anomaly occurs around the 40 second mark where the roll angle suddenly drops to -6 degrees. This behaviour is not typical for the parametric roll phenomenon, and the most probable cause for this is a user or a software error. The anomaly only occurred once and the rest of the simulation ran smoothly.

While the phenomenon didn't occur like in regular waves, the rolling didn't dampen completely either. This can be seen at around the 75-second mark, where the amplitude stayed at the same level for multiple periods. Since now the ship velocity is zero, the encounter period deviates slightly from the worst-case scenario. Thus, the results show how rare the phenomenon can be, and that a much longer simulation would have been required to catch the phenomenon.

The final simulation was similar to the previous but the significant wave height $H_{1/3}$ was increased to 0.5 m . At the same time, a new set of wave phases was created, which resulted in a unique-looking sea. The result of that simulation is presented in the figure below.

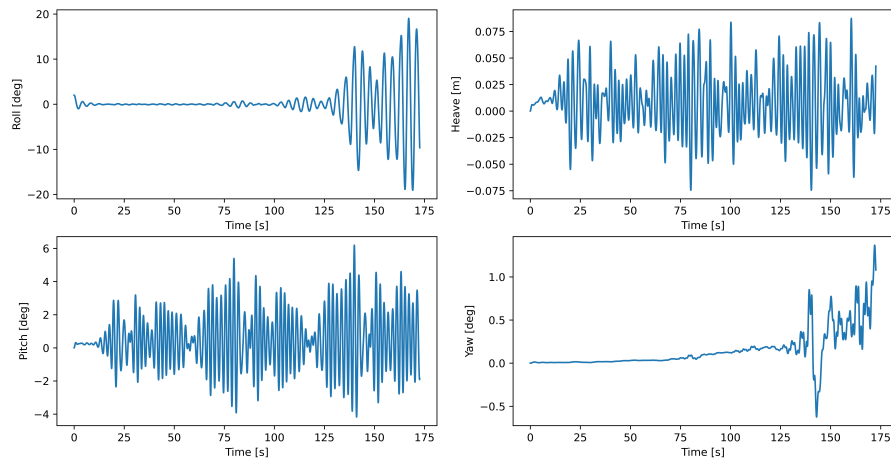


Figure 6.8. Parametric roll in irregular wave with $T_p = 2.12\text{ s}$ and $H_{1/3} = 0.5\text{ m}$.

The simulation took roughly three weeks for the given system. Unlike in the first simulation, these results show how the magnitude of the oscillating gradually begins to increase. The waves developed gradually for the first 15 seconds which lead the motions to be calm. When the waves are fully developed, the pitch angle and heave start to oscillate in response to these waves. The roll angle initially was damped but remained oscillating with a small amplitude. After about the 100-second mark, the rolling action starts to build up and the amplitude of the oscillations starts to increase. The maximum amplitude reaches around 20 degrees for both sides. The free surface at a point near the ship is shown in the figure below.

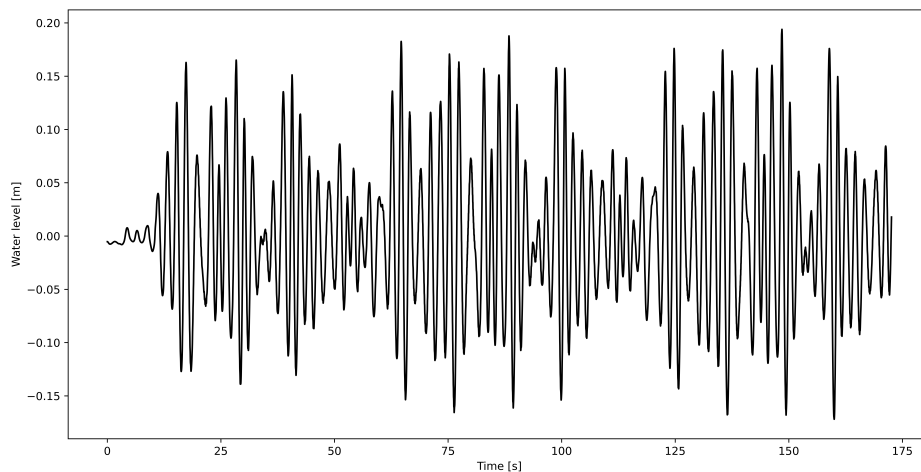


Figure 6.9. Wave height tracking in the second irregular simulation

The wave pattern represents a typical irregular wave, where every wave differs from the others. However, the state is not completely random due to the similar wave heights. Thus, the wave model has successfully created a sea state where the characteristics of

the waves are distinguishable. One observation from both of the irregular wave simulations is that wave height seems to be smaller than the respective significant wave height $H_{1/3}$. When creating the wave components, a small simulation was conducted to verify that the components indeed produce a visually correct sea state. For example, the latter irregular simulation had a following verification plot.

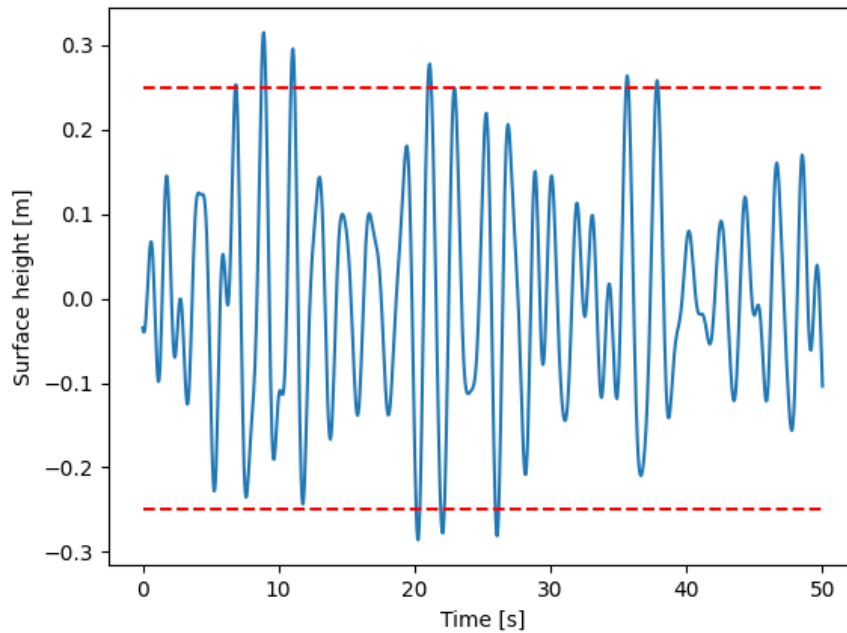


Figure 6.10. Verification simulation for sea state with $T_p = 2.12\text{ s}$ and $H_{1/3} = 0.5\text{ m}$.

The red dotted lines represent the boundaries of the wave height 0.5 m . Indeed, the wave height is much greater in the verification process than in the actual OpenFOAM simulation. Thus, the wave implementations in OpenFOAM might possess some nuances that should be further investigated. However, the encountered wave had enough height to produce meaningful results.

7. CONCLUSIONS AND FURTHER WORK

The new upcoming stability assessment SGISC is being developed to encourage that the new generation ship design would include the ship's stability in waves. While this is beneficial for everyone, carrying out the said assessment will create a new burden for the shipyards. Thus, this thesis was set to study the criteria assessment as well as to gain a deeper understanding of the software requirements in practice. To keep this thesis within reasonable size, only parametric roll was included from the five failure events.

The simplified roll equation was studied in Section 2.2. By utilizing Floquet's theorem, an algorithm was constructed to find a stability criterion for the system. Applying this to the dimensionless roll equation, stable and unstable regions could be plotted in a graph. The stable region in the system with damping is indeed larger than in the system with no damping. Also, all systems became more unstable when the encountered wave frequency approached twice the natural roll frequency. This is explained by the waves parametrically amplifying the small rolling motions, which has also been observed to be the worst-case scenario for parametric rolling.

The level 1 and 2 phases of SGISC are relatively simple and could be performed using computational and hydrodynamic software. The DSA phase on the other hand conducts a probabilistic study of the ship's interaction with the irregular waves. The guidelines for DSA allow the analysis to be conducted by any means as long as the requirements are met. With the computational power getting cheaper, a CFD becomes more and more available. As it computes the fluid flow itself, the method requires no assumptions for the hull interaction with waves as it is resolved in the process. Thus, conducting the DSA using CFD seems an attractive choice as long as the time for the analysis can be kept reasonable.

Creating CFD software from scratch is by no means a trivial task and some commercial licenses can cost up to 50000\$ per year. Therefore, a free open-source project called OpenFOAM can be an attractive choice. The free license is a big benefit, but other benefits include the ability to modify source code allowing users to implement their required models. The difficulty of open-source projects comes with the caveat that if something goes wrong, then the user is expected to find the solution. Therefore, operating it requires an understanding of the physics behind it and the ability to read and write software

code.

If the ship roll is simulated by using the equation of motion, the damping coefficients and the natural roll period are often required. The recommended method to obtain these is by model testing, but most often this is not possible due to practical reasons. The next best option is by conducting a free roll decay simulation in CFD. In this thesis, a respective model and full-scale simulations are run to determine the practicality of these simulations. The results of these are displayed in Figures 6.1 and 6.2. In both cases, the roll experienced a decaying motion where the ship gradually converged to its upright position. From these graphs, the natural rolling periods were calculated as the time between two troughs. When comparing the measured roll period to the equation (5.15) and scaling with Froude's number, the magnitudes matched pretty well indicating the validity of the methods.

The actual parametric roll simulations were conducted first in regular waves for both model and full scale. The hull design tested is generally not vulnerable to parametric roll. However, by decreasing the metacentric height and removing bilge keels, the stability in waves was decreased and the parametric rolling was more observable even in steeper waves. The worst-case scenarios were created to fully capture the phenomenon. The result for model and full-scale regular waves are displayed in Figures 6.3 and 6.5 respectively. These figures reveal that the parametric roll occurred: the roll angle gradually increases in each period and finally settles around the maximum value. The pitch angle period shows that the waves indeed occur twice the natural rolling period, which also confirms that the hull interaction with waves was successful.

For the full-scale parametric roll, one anomaly occurred at the end of the simulation. The software closed without giving any error even with the residuals being normal. Further inspections showed that the crash happened somewhere in the GAMG loop while calculating pressure. Changing the scheme to conjugate gradient advanced the simulation a little further, but after a while, the residuals diverged. Given more time, this could have been investigated further to find out whether the problem was in software or mesh. However, the simulation managed to run 100 seconds in which the parametric roll phenomenon did manage to occur.

The last two simulations were irregular sea in model scale by the means of SGISC. The sea condition is generated by superposing singular wave components, and these components were created by spectral methods to represent a realistic sea condition. Then, the simulations were run for enough long time to determine what the maximum roll angle would be. The result for these simulations is shown in Figures 6.6 and 6.8 respectively.

The first irregular simulation lasted for about 250 seconds, which allowed the ship to encounter multiple waves. Although the simulation ran mostly smoothly, one anomaly appeared at around the 40-second mark, where the roll angle suddenly dropped to about

—6 degrees. This is most likely a software or user-related error rather than the actual effect of resonance and thus could use further investigation. This however allowed the ship to regain some momentum and the event occurred only once.

In the second irregular simulation, a more typical result can be seen. The simulation differed from the first irregular simulation by having a larger significant wave height. This allowed the ship to gain momentum due to the resonance, which eventually led the ship to incline close to 20 degrees. However, the simulation started to slow down the convergence rate in the end, which is an undesirable outcome.

To summarize, using OpenFOAM to conduct parametric roll resonance was promising as the results from the simulations behaved as to be expected. However, with the current setup of a 12 core computer, the simulation time took unreasonably long, and thus, conducting the full SGISC analysis is practically impossible. However, with the computational power getting cheaper, CFD becomes more accessible and could be a viable method in the future. A good compromise would be to conduct the analysis with some simpler method and switch to CFD for individual simulations when necessary.

This thesis introduces the theory behind CFD calculations, hydrodynamics, and parametric rolling, but only touches the surface of the actual computations. Some further work ideas are presented below.

- A couple of anomalies appeared in the simulations which could use further research. The following topics arose during the thesis:
 - The wave height measured in irregular waves didn't match the selected significant wave height. This could be due to the mesh quality near the surface, possible active damping of the wave, or an error by the user.
 - The simulations had some anomalies where the simulation started to slow or diverge while the residuals remained normal. The most probable cause for these is that the rigid body simulations alter the mesh and thus the mesh quality changes. Therefore, an overset method could prove to be a better method for moving the mesh.
- Comparing the results from OpenFOAM to some other commercial CFD software would expose if the free open-source software could compete with actual commercial software.

REFERENCES

- [1] G.B. Airy. *Tides and Waves*. B. Fellowes, 1845. URL: <https://books.google.fi/books?id=41dSAAAAcAAJ>.
- [2] Fatimah Hassan Albasrawi. "Floquet Theory on Banach Space". eng. Masters's Thesis. 2013.
- [3] Bengt Andersson et al. *Computational Fluid Dynamics for Engineers*. eng. Cambridge: Cambridge University Press, 2011, pp. 1–. ISBN: 1107018951.
- [4] Bryan Barrass and D. R Derrett. *Ship Stability for Masters and Mates (7th Edition)*. eng. London: Elsevier, 2012. ISBN: 0080970931.
- [5] Adrian. Biran. *Ship hydrostatics and stability*. eng. Second edition. Boston: Butterworth-Heinemann, 2014. ISBN: 0-08-098290-5.
- [6] The Danish Maritime Accident Investigation Board. "SVENDBORG MÆRSK - Heavy weather damage on 14 February 2014". In: (Sept. 2014), p. 28.
- [7] The University of British Columbia. "Chapter 3 - Basic Floquet Theory". In: (), p. 40.
- [8] Michael Brorsen. *Non-linear Waves*. English. DCE Lecture notes 9. Denmark: Department of Civil Engineering, Aalborg University, 2007.
- [9] James J. Callahan. *Advanced Calculus A Geometric View*. eng. 1st ed. 2010. Undergraduate Texts in Mathematics. New York, NY: Springer New York, 2010. ISBN: 1-280-39141-3.
- [10] Goong Chen et al. "OpenFOAM for computational fluid dynamics". eng. In: *Notices of the American Mathematical Society* 61.4 (2014), pp. 354–363. ISSN: 0002-9920.
- [11] Carmen. Chicone. *Ordinary Differential Equations with Applications*. eng. 2nd ed. 2006. Texts in Applied Mathematics, 34. New York, NY: Springer New York, 2006. ISBN: 0-387-35794-7.
- [12] Henry Weller Chris Greenshields. *The cost of simulating turbulence*. 2022. URL: <https://doc.cfd.direct/notes/cfd-general-principles/the-cost-of-simulating-turbulence> (visited on 10/13/2022).
- [13] Wikimedia Commons. *File:Schlingerkiel.jpg* — *Wikimedia Commons, the free media repository*. [Online; accessed 21-February-2023]. 2022. URL: <https://commons.wikimedia.org/w/index.php?title=File:Schlingerkiel.jpg&oldid=655857289>.
- [14] *Computational Fluid Dynamics An Introduction*. eng. 3rd ed. 2009. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. ISBN: 3-540-85056-2.
- [15] *Computational fluid dynamics for incompressible flow*. eng. First edition. Boca Raton, FL: CRC Press, 2020. ISBN: 9780367809171.

- [16] International Towing Tank Conference. "Predicting the Occurrence and Magnitude of Parametric Rolling". In: (June 2021), p. 23.
- [17] *Corrected surface-normal gradient scheme*. 2023. URL: <https://www.openfoam.com/documentation/guides/latest/doc/guide-schemes-sng-rad-corrected.html>.
- [18] Alex D.D. Craik. "The origins of water wave theory". eng. In: *Annual review of fluid mechanics* 36 (2004), pp. 1–28. ISSN: 0066-4189.
- [19] Roy. Featherstone. *Rigid Body Dynamics Algorithms*. eng. 1st ed. 2008. New York, NY: Springer US, 2008. ISBN: 1-4899-7560-8.
- [20] John D Fenton. "A Fifth-Order Stokes Theory for Steady Waves". eng. In: *Journal of waterway, port, coastal, and ocean engineering* 111.2 (1985), pp. 216–234. ISSN: 0733-950X.
- [21] E. Folkers. "Floque's Theorem". eng. Bachelor's Thesis. Apr. 2018, p. 31.
- [22] Alistair Galbraith and Evangelos Boulougouris. "Parametric Rolling of the Tumble-home Hull using CFD". In: June 2015.
- [23] J.-C. Gilloteaux, A. Babarit, and A.H. Clément. "Influence of wave spectrum spreading on the production of the SEAREV wave energy converter". eng. In: *Proceedings of the International Offshore and Polar Engineering Conference*. 2007, pp. 415–420. ISBN: 1880653680.
- [24] O. Grim. *Rollschwingungen, Stabilität und Sicherheit im Seegang*. Bericht. Schiffbau-Versuchsanstalt, 1952. URL: <https://repository.tudelft.nl/islandora/object/uuid:3e0c3df6-e117-418a-a28d-b8dc54cece0e?collection=research>.
- [25] Min Gu, Shu-xia Bu, and Jiang Lu. "Study of parametric roll in oblique waves using a three-dimensional hybrid panel method". eng. In: *Journal of hydrodynamics. Series B* 32.1 (2020), pp. 126–138. ISSN: 1001-6058.
- [26] *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. eng. Place of publication not identified: U S Department of Commerce, 1972. ISBN: 1-59124-217-7.
- [27] A.J Hermans. *Water Waves and Ship Hydrodynamics: An Introduction*. ger ; eng. 1. Aufl. Dordrecht: Springer Science + Business Media, 2011. ISBN: 9400700954.
- [28] Leo H. Holthuijsen. *Waves in oceanic and coastal waters*. eng. Cambridge: Cambridge University Press, 2007. ISBN: 1-107-08636-1.
- [29] *Ilmatieteen laitos - Havaintojen lataus*. 2023. URL: <https://www.ilmatieteenlaitos.fi/havaintojen-lataus> (visited on 04/01/2023).
- [30] *Install Linux on Windows with WSL*. 2023. URL: <https://learn.microsoft.com/en-us/windows/wsl/install>.
- [31] Hrvoje Jasak. "Error Analysis and Estimation for the Finite Volume Method With Applications to Fluid Flows". In: *Direct M* (Jan. 1996).
- [32] Dominic Jordan and Peter Smith. *Nonlinear Ordinary Differential Equations: An Introduction for Scientists and Engineers*. eng. Oxford: Oxford University Press, Incorporated, 2007. ISBN: 0199208247.

- [33] Jer-Nan Juang. *Identification and control of mechanical systems*. eng. Cambridge: Cambridge University Press, 2001. ISBN: 1-107-12108-6.
- [34] J.E. Kerwin. "Notes on rolling in longitudinal waves". eng. In: *International shipbuilding progress* 2.16 (1955), pp. 597–614. ISSN: 0020-868X.
- [35] Edgar A. Kraut. *Fundamentals of Mathematical Physics*. eng. Dover Books on Physics. Newburyport: Dover Publications, 2013. ISBN: 0-486-13160-2.
- [36] B.E. Launder and D.B. Spalding. "The numerical computation of turbulent flows". eng. In: *Computer methods in applied mechanics and engineering* 3.2 (1974), pp. 269–289. ISSN: 0045-7825.
- [37] L. Gary Leal. *Advanced transport phenomena : fluid mechanics and convective transport processes*. eng. Cambridge series in chemical engineering. Cambridge: Cambridge University Press, 2007. ISBN: 1-107-16480-X.
- [38] Pengzhi Lin. *Numerical Modeling of Water Waves: An Introduction to Engineers and Scientists*. eng. Independence: Taylor & Francis Group, 2008. ISBN: 0415415780.
- [39] Liwei Liu et al. "CFD prediction of full-scale ship parametric roll in head wave". eng. In: *Ocean engineering* 233 (2021), pp. 109180–. ISSN: 0029-8018.
- [40] Kyle E. Marlantes, Sungeun (Peter) Kim, and Lucas A. Hurt. "Implementation of the IMO Second Generation Intact Stability Guidelines". In: *Journal of Marine Science and Engineering* 10.1 (2022). ISSN: 2077-1312. DOI: 10.3390/jmse10010041. URL: <https://www.mdpi.com/2077-1312/10/1/41>.
- [41] Santiago Márquez Damián. "An Extended Mixture Model for the Simultaneous Treatment of Short and Long Scale Interfaces". PhD thesis. Mar. 2013. DOI: 10.13140/RG.2.1.3182.8320.
- [42] F. R. Menter, M. Kuntz, and R. Langtry. "Ten Years of Industrial Experience with the SST Turbulence Model". In: ed. by K. Hanjalic, Y.Nagano, and M.Tummers. Begell House, Inc., 2003. Chap. xx, pp. xx–xx.
- [43] M. Miche. "Mouvements ondulatoires de la mer en profondeur constante ou décroissante". In: 1944.
- [44] Colin S Moore. *Intact stability*. eng. Principles of naval architecture Intact stability. Place of publication not identified: Society of Naval Architects, 2010. ISBN: 0-939773-74-0.
- [45] F. Moukalled. *The Finite Volume Method in Computational Fluid Dynamics An Advanced Introduction with OpenFOAM® and Matlab*. eng. 1st ed. 2016. Fluid Mechanics and Its Applications, 113. Cham: Springer International Publishing, 2016. ISBN: 3-319-16874-6.
- [46] Nathan M. Newmark. "A Method of Computation for Structural Dynamics". eng. In: *Transactions of the American Society of Civil Engineers* 127.1 (1962), pp. 1406–1433. ISSN: 0066-0604.

- [47] Petacco Nicola. "Second Generation Intact Stability criteria: Analysis, Implementation and Applications to significant ship typologies". PhD thesis. May 2019, p. 199. DOI: 10.15167/petacco-nicola_phd2019-05-24.
- [48] *OpenFOAM boundary conditions*. 2023. URL: <https://www.openfoam.com/documentation/guides/latest/doc/openfoam-guide-boundary-conditions.html> (visited on 07/02/2023).
- [49] *OpenFOAM financial solvers*. 2023. URL: https://www.openfoam.com/documentation/guides/latest/api/group__grpFinancialSolvers.html.
- [50] International Maritime Organization. "Interim Guidelines on the Second Generation Intact Stability Criteria". In: (Dec. 2020), p. 58.
- [51] S.V. Patankar and Spalding D.B. "Paper 5 - A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows". eng. In: *Numerical Prediction of Flow, Heat Transfer, Turbulence and Combustion*. United Kingdom: Elsevier Ltd, 1983, pp. 54–73. ISBN: 9780080309378.
- [52] J. Randolph Paulling. "Parametric Rolling of Ships – Then and Now". eng. In: *Contemporary Ideas on Ship Stability and Capsizing in Waves*. Vol. 97. Fluid Mechanics and Its Applications. Dordrecht: Springer Netherlands, 2011, pp. 347–360. ISBN: 9400714815.
- [53] Milovan Peric. "A finite volume method for the three-dimensional fluid flow in complex ducts". In: (Jan. 1985), p. 299.
- [54] Robinson Peric. "Minimizing undesired wave reflection at the domain boundaries in flow simulations with forcing zones". doctoralThesis. Technische Universität Hamburg, 2019. ISBN: 978-3-89220-713-9. DOI: 10.15480/882.2394. URL: <http://hdl.handle.net/11420/3286>.
- [55] Nicola Petacco and Paola Gualeni. "IMO Second Generation Intact Stability Criteria: General Overview and Focus on Operational Measures". In: *Journal of Marine Science and Engineering* 8 (July 2020), p. 494. DOI: 10.3390/jmse8070494.
- [56] Henry Peter Piehl. "Ship Roll Damping Analysis". eng. In: (Apr. 2016), p. 153.
- [57] Daniel Prasanna. *The ship motions at sea*. 2014. URL: <https://discover.hubpages.com/travel/theshipmotionsatsea> (visited on 10/01/2023).
- [58] Alfio. Quarteroni. *Numerical Models for Differential Problems*. eng. 1st ed. 2009. MS&A, Modeling, Simulation and Applications, 2. Milano: Springer Milan, 2009. ISBN: 1-280-38443-3.
- [59] M. Rahman. "Fundamentals concerning stokes waves". eng. In: *Advances in Fluid Mechanics*. Vol. 9. 1996, pp. 289–298.
- [60] *Reynolds stress model*. 2023. URL: [https://www.cfd-online.com/Wiki/Reynolds_stress_model_\(RSM\)](https://www.cfd-online.com/Wiki/Reynolds_stress_model_(RSM)) (visited on 09/01/2023).
- [61] D.A. Sánchez. *Ordinary Differential Equations and Stability Theory: An Introduction*. Dover Books on Mathematics. Dover, 1979. ISBN: 9780486638287. URL: https://books.google.fi/books?id=Qr38tJ7P%5C_okC.

- [62] François G. Schmitt. "About Boussinesq's turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity". eng ; fre. In: *Comptes rendus. Mécanique* 335.9 (2007), pp. 617–627. ISSN: 1631-0721.
- [63] Rawaa Shaheed, Abdolmajid Mohammadian, and Hossein Kheirkhah Gildeh. "A comparison of standard k–epsilon and realizable k–epsilon turbulence models in curved and confluent channels". eng. In: *Environmental fluid mechanics (Dordrecht, Netherlands : 2001)* 19.2 (2019), pp. 543–568. ISSN: 1567-7419.
- [64] Ahmed Shama. "Study of Microfluidic Mixing and Droplet Generation for 3D Printing of Nuclear Fuels". PhD thesis. Sept. 2017. DOI: 10.13140/RG.2.2.24723.78887.
- [65] Edward J. Shaughnessy. *Introduction to fluid mechanics*. eng. Place of publication not identified: Oxford University Press, 2005. ISBN: 1-68015-283-1.
- [66] Irfan Ahmad Sheikh. *Parametric Roll Instability of Ships*. eng. 2008.
- [67] Vladimir Shigunov. "Direct stability assessment: pragmatic solutions". In: *Ship Technology Research* 64.3 (2017), pp. 144–162. DOI: 10.1080/09377255.2017.1392074. URL: <https://doi.org/10.1080/09377255.2017.1392074>.
- [68] *simpleFoam implementation in OpenFOAM*. 2023. URL: <https://develop.openfoam.com/Development/openfoam/-/tree/master/applications/solvers/incompressible/simpleFoam> (visited on 01/31/2023).
- [69] Zhi-yao Song et al. "On the universal third order Stokes wave solution". In: *Science China Earth Sciences* 56 (Jan. 2012). DOI: 10.1007/s11430-012-4523-z.
- [70] *Source code of rigidBodyDynamics*. 2023. URL: <https://develop.openfoam.com/Development/openfoam/-/tree/master/src/rigidBodyDynamics> (visited on 01/02/2023).
- [71] P.R. Spalart and S.R. Allmaras. "One-equation turbulence model for aerodynamic flows". eng. In: *RECHERCHE AEROSPATIALE* 1 (1994), pp. 5–21. ISSN: 0034-1223.
- [72] Metin Taylan. "On the parametric resonance of container ships". In: *Ocean Engineering* 34 (May 2007), pp. 1021–1027. DOI: 10.1016/j.oceaneng.2006.04.007.
- [73] Delft University of Technology. "Chapter 5 - Waves". In: (), p. 48. URL: https://ocw.tudelft.nl/wp-content/uploads/7.OffshoreHydromech1_Waves.pdf.
- [74] *Using the Right-Hand Rule*. 2023. URL: <https://www.khanacademy.org/test-prep/mcat/physical-processes/magnetism-mcat/a/using-the-right-hand-rule> (visited on 10/01/2023).
- [75] H. K. Versteeg. *An introduction to computational fluid dynamics : the finite volume method*. eng. 2nd ed. Harlow: Pearson/Prentice Hall, 2007. ISBN: 978-0-13-127498-3.
- [76] H. A. van der Vorst. "Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems". In: *SIAM Journal on Scientific and Statistical Computing* 13.2 (1992), pp. 631–644. DOI: 10.1137/0913035. eprint: <https://doi.org/10.1137/0913035>. URL: <https://doi.org/10.1137/0913035>.

- [77] Yan-ying Wang. "The Specialist Committee on Waves Final Report and Recommendations to the 23 rd ITTC". In: 2016.
- [78] David Wilcox. *Turbulence Modeling for CFD*. 3rd ed. 2006. DCW Industries, 2006. ISBN: 1-928729-08-2.

APPENDIX A: BASIC SHIP VOCABULARY AND METRICS

This appendix explains the basic vocabulary and metrics that are common in the marine field and it is recommended for readers with little to no prior knowledge of the matter and it should at least cover every term used within this thesis. This section is based on [5] where readers are referred for further reading.

The dynamical system should be invariant i.e. independent of the choice of coordinate system. However, it is useful to define a universally common coordinate system such that precise definitions can be constructed. The international standards ISO 7460 and 7463 as well as the Germany Standardisation Institute recommend the following coordinate system for naval applications.

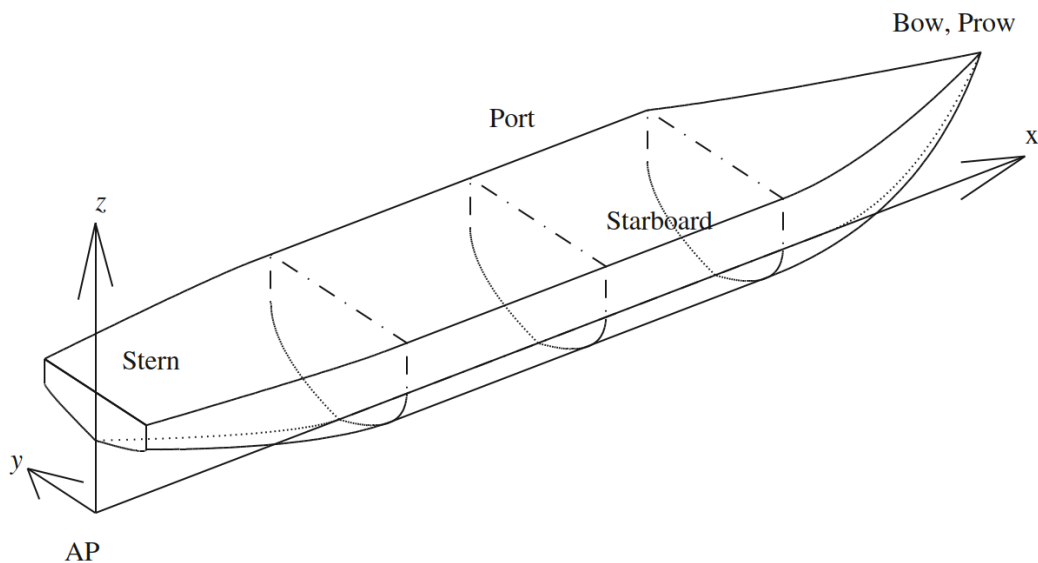


Figure A.1. Ship coordinate system proposed by DIN 81209-1 [5]

The positive x-axis points to the forward direction of the ship, the positive y points to the port side, the negative y to the starboard side, and the negative z-axis points to the direction of gravity. The forward (fwd) part of the ship is called the bow and the behind of the ship is called the after (aft) or stern. Using this coordinate system we can define that *head seas* is a state, where waves travel exclusively to negative x-direction, and in *following seas*, to positive x-direction. The origin lies at the aft part of the ship.

Often in hydrostatics, the ship geometry is sliced with respect to a plane that has a normal to one of the main coordinates. The resulting 2D image represents the ship contour on that plane. The xy -planes are called *waterlines*, xz -planes are called *buttocks* or *verticals*, and yz -planes are called *stations* or *frame sections*. From stations, we can see the definition of some commonly used metrics. *Draft* (T) is a distance from the bottom of the ship, i.e. the keel, to the waterline, *beam* or *breadth* (B) is the width of the ship, *depth* (D) is the distance from the keel to the main deck and *freeboard* (f) is the shortest distance from the waterline to the main deck up to which the watertight bulkheads (transversal watertight partitions) extend. These metrics are calculated from the outer part of the outer shell of the hull. If however, these metrics were to be calculated to the inner shell, then the term would be *moulded depth* or *moulded beam* etc. Normally these metrics are relatively the same, but for example, warships can have sonars below the keel. This changes the depth of the ship as the lowest point of the ship is lower, but moulded depth would remain the same. These metrics are illustrated below.

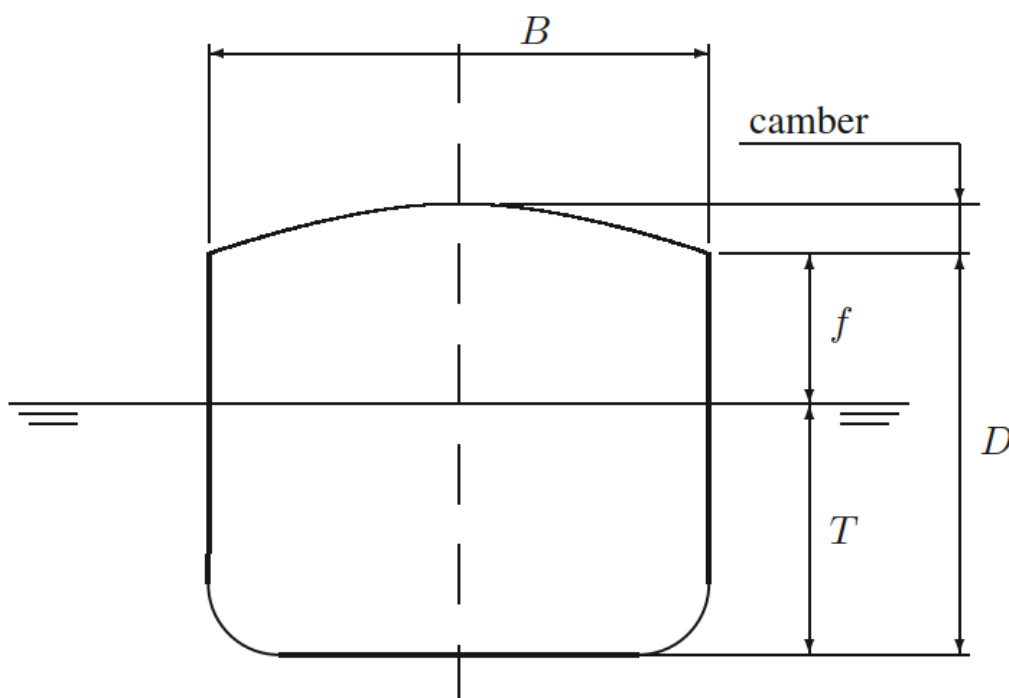


Figure A.2. Definition of draft, beam, depth and freeboard [5]

The *load line* or the *Plimsoll line* is a waterline up to where the ship can be loaded. It is often marked into the hull by "load line markings" to aid in loading the ship. The primarily used load line is so called *design summer load line*, which indicates the load line during summer at typical seawater density. For example, Lloyd's register then defined other load lines including tropical fresh water, fresh water, tropical seawater, winter temperature sea water, and winter North Atlantic load lines.

The length of the ship can be defined in numerous ways. One definition is *length between perpendiculars* which is the distance from the forward perpendicular (FP) to the aft

perpendicular (AP). Forward perpendicular is a vertical line perpendicular to the load line and where the bow and the free surface cross. Aft perpendicular, on the other hand, is a vertical line perpendicular to the load line that goes through the axis of the rudder stock. Other length definitions include *length at waterline* (LWL), which is the length measured at the waterline, and *length over all* (LOA) which is the maximum length of the hull. These are represented in the figure below.

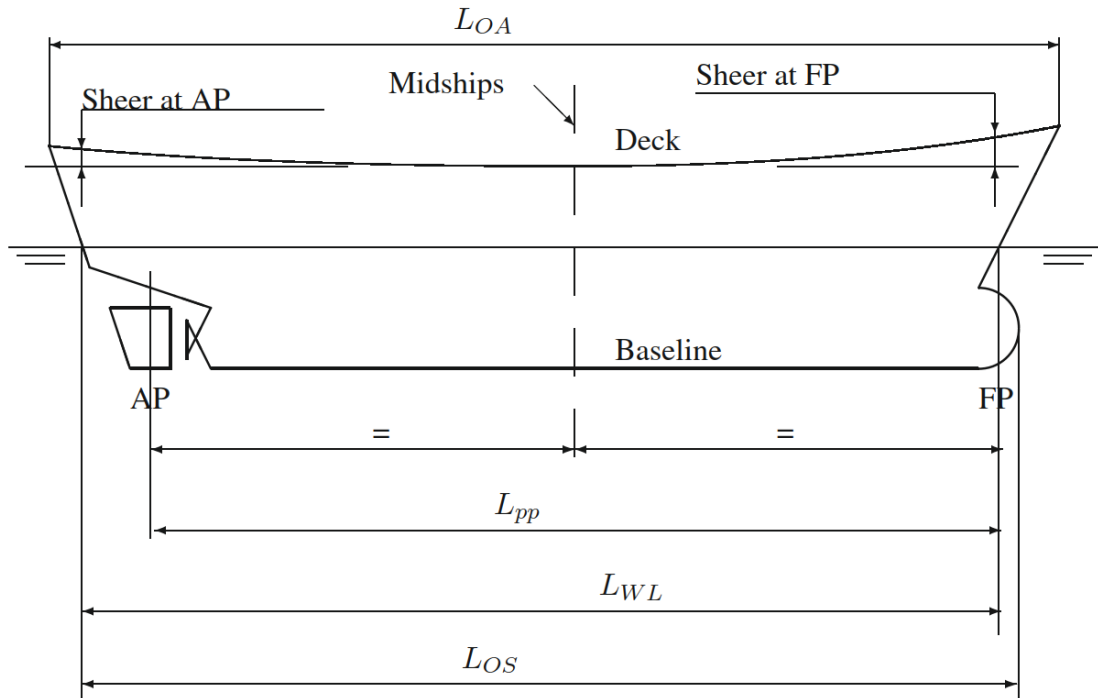


Figure A.3. Definition of ship lengths. [5]

The definition of ship motions is presented in the figure below. The translational motions are referred to as surge, heave, and sway, and rotational movements are referred to as roll, pitch, and yaw. To obtain the sign of the rotations, one can use the right-hand rule [74] with respect to the coordinate system defined before. For example, when the ship pitches so that the bow of the ship sinks and the stern rises, the pitch angle is positive.

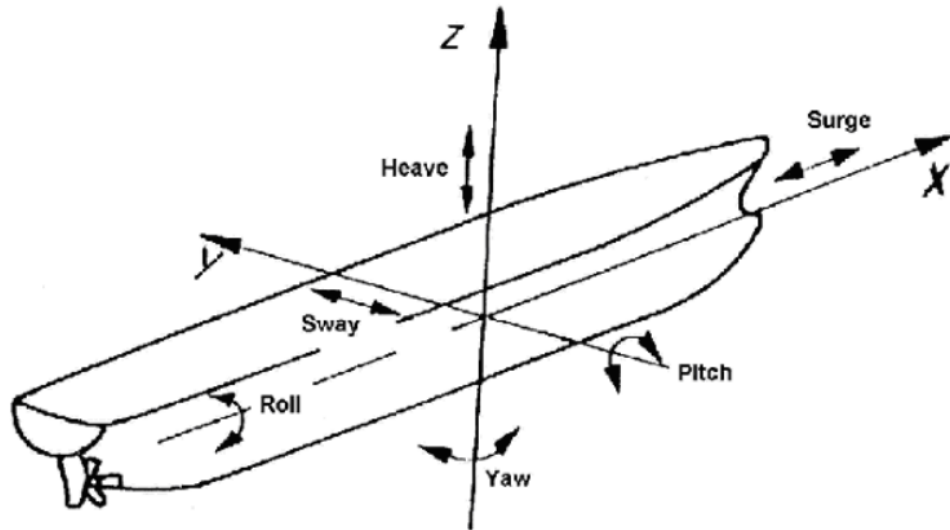


Figure A.4. Motions of the ship [57]

One of the passive ways to reduce rolling motions is with so-called bilge keels. Bilge itself is the curved part of the ship where the keel and side are combined. This is often a round shape, but a sharp bilge or chine would have a sharp edge. Bilge keel on the other hand is an additional plating that follows the flow streamlines and is located at the bilge. It can dampen the rolling action greatly while its streamlined shape doesn't increase the resistance by too much. The figure below clarifies these concepts.



Figure A.5. Bilge and bilge keel definition. The rightmost figure shows actual bilge keel [13]

Often when talking about the hull's design, a useful metric type is the so-called coefficient of form. These often explain the relationship between some parts of the hull to the extremes of that part. The *block coefficient* C_B explains the fraction between the displacement volume ∇ to the other extremes in A.2 i.e.

$$C_B = \frac{\nabla}{LBT}. \quad (\text{A.1})$$

The midship coefficient C_M on the other hand is a ratio of the midship section area to the beam and draft i.e.

$$C_M = \frac{A_M}{BT}, \quad (\text{A.2})$$

where A_M is the area of the section from the middle of the ship. Other types of coefficients can also be defined and multiple of these can be found in [5]. The figure below presents the previous two coefficients.

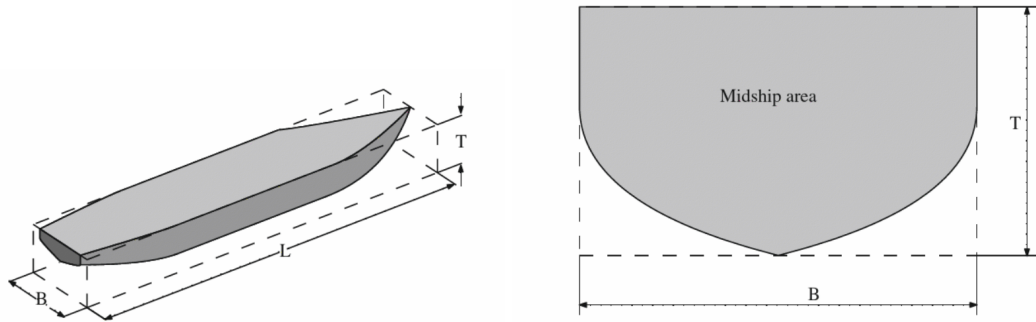


Figure A.6. The definition of block coefficient (the left figure) and the midship coefficient (the right figure) [5]

Although the term might be self-explanatory, a word about the meaning of loading conditions might be in order. As the ship is built, it has some weight due to the parts of the ship. This weight is referred to as *lightweight* and it's the minimum weight the ship has. To obtain the weight of the ship, the total mass of passengers, crew, liquids, cargo, etc. is added to the lightweight. A *deadweight* is the maximum allowed weight that the ship can support excluding the lightweight itself. When cargo is loaded into the ship, it is placed at some fixed point within the ship. As this cargo has some mass, the ship's centre of gravity is shifted by the equation (2.2). Therefore the location of the cargo affects the stability and hence the different loading conditions have different results for the stability assessments.

Two important dimensionless coefficients in hydrodynamics are the *Froude number* (F_n) and the *Reynolds number* (Re) and they both represent a ratio of some flow properties. The Froude number is defined as

$$F_n = \frac{u}{\sqrt{gL}}, \quad (\text{A.3})$$

where u is the fluid velocity, g is the gravitational constant and L is a characteristic length of the flow. It represents the ratio of inertia to the weight of the fluid element. In ship hydrodynamics, the Froude number is used to determine the wave-making resistance of the ship. On the other hand, the Reynolds number is defined as

$$Re = \frac{uL}{\nu}, \quad (\text{A.4})$$

where u is the flow velocity, L is the characteristic length of the flow and ν is kinematic viscosity of the fluid. This dimensionless number represents the ratio between inertia forces and viscous forces. It often is used to determine whether the fluid flow is laminar or turbulent.

APPENDIX B: TABLES

Wave case id	Weight factor W_i	Wavelength λ_i	Wave height H_i
1	0.000013	22.574	0.350
2	0.001654	37.316	0.495
3	0.020912	55.743	0.857
4	0.092799	77.857	1.295
5	0.199218	103.655	1.732
6	0.248788	133.139	2.205
7	0.208699	166.309	2.697
8	0.128984	203.164	3.176
9	0.062446	243.705	3.625
10	0.024790	287.931	4.040
11	0.008367	335.843	4.421
12	0.002473	387.440	4.769
13	0.000658	442.723	5.097
14	0.000158	501.691	5.370
15	0.000034	564.345	5.621
16	0.000007	630.684	5.950

Table B.1. Wave cases for parametric rolling evaluation [50]

i	K_i
1	1.0
2	0.991
3	0,966
4	0.924
5	0.866
6	0.793
7	0.707
8	0.609
9	0.500
10	0.383
11	0.259
12	0.131

Table B.2. Speed factor K_i [50]