ABSTRACT

Ship seaworthiness is a critical factor in maritime engineering, ensuring vessels can navigate safely and efficiently across diverse environmental conditions. This project employs advanced Computational Fluid Dynamics (CFD) calculations, coupled with GPU acceleration, to comprehensively assess ship seaworthiness. Through meticulous CFD simulations, the complex fluid-structure interactions between ships and water bodies are modeled, enabling precise predictions of hydrodynamic forces, fluid flow patterns, and stability parameters crucial for ship performance evaluation.

The primary objective is to deepen understanding of ship behavior under various environmental scenarios, including different sea states, vessel speeds, and hull designs. By conducting detailed CFD simulations accelerated by GPUs, potential areas for enhancing ship designs, propulsion systems, and operational strategies can be identified, significantly improving seaworthiness and operational efficiency.

In summary, this project contributes to advancing maritime engineering and naval architecture by providing a robust framework for assessing ship seaworthiness. Leveraging CFD calculations and GPU acceleration, the safety, stability, and performance of maritime vessels can be effectively scrutinized, fostering the development of safer, more efficient, and environmentally sustainable maritime transportation systems.

TABLE OF CONTENTS

LIST OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 COMPUTATIONAL FLUID DYNAMICS

Computational Fluid Dynamics (CFD) is a branch of fluid mechanics that employs numerical methods and algorithms to solve and analyze fluid flow problems. In our project, CFD serves as the cornerstone for simulating and understanding the complex behavior of fluids, particularly in the context of boat stability analysis. By leveraging CFD techniques, we can model the intricate interactions between water and boat surfaces, predict flow patterns, and compute essential parameters such as velocity fields, pressure distributions, and turbulence effects.

1.2 GRAPHICS PROCESSING UNIT

A Graphics Processing Unit (GPU) is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device. However, GPUs are not limited to graphics processing; they excel at parallel processing tasks, making them invaluable for scientific computing, machine learning, and simulations.

In the context of ship stability assessment, the GPU plays a pivotal role in accelerating complex computational tasks involved in fluid dynamics simulations, structural analysis, and optimization algorithms. By harnessing the parallel processing power of GPUs, the project can execute computationally intensive simulations efficiently, significantly reducing the time required for analysis and optimization.

Utilizing tensor data types and frameworks like PyTorch in platforms such as Google Collab further enhances the computational capabilities of the project. These tools optimize calculations and simulations, allowing for faster and more accurate assessments of ship stability across various environmental conditions.
Overall, the strategic use of GPU acceleration and optimized computation techniques enables the project to achieve comprehensive evaluations of ship seaworthiness, facilitating informed decision-making and design refinements for safer and more efficient maritime operations.

1.3 SHIP STABILITY

Maritime transportation plays a crucial role in global trade, requiring ships to navigate diverse and challenging maritime environments safely. However, assessing ship stability in dynamic conditions poses significant challenges for traditional methods. This project introduces a novel approach to ship stability assessment using GPU-accelerated simulation techniques. By harnessing the computational power of GPUs, the project aims to enhance the efficiency and accuracy of ship stability analysis across various maritime scenarios. Through detailed computational fluid dynamics simulations, the project seeks to provide valuable insights into vessel performance and safety, contributing to the enhancement of maritime safety and reliability. Moreover, by showcasing the effectiveness of GPU acceleration in tackling complex maritime challenges, the project aims to advance the field of computational maritime engineering and safety analysis.

1.4 CALCULATION ASSESSMENT

The project entails a comprehensive approach to ship stability assessment, integrating various factors such as wave parameters, ship geometries, wind effects, and other environmental variables. Utilizing advanced computational techniques, the project aims to quantify the stability of vessels across a range of maritime conditions, ultimately assessing their seaworthiness.

Initially, the project involves the calculation of ship stability using mathematical models that account for the dynamic interactions between the vessel and its surrounding environment. This includes the incorporation of wave parameters such as wave height, period, and direction, which significantly influence the ship's response and stability characteristics. Ship geometries, including hull shape, dimensions, and displacement, are also taken into consideration as critical factors in determining stability.

Furthermore, the project accounts for wind effects, which can exert additional forces and moments on the vessel, further impacting its stability. By integrating wind data and computational fluid dynamics (CFD) simulations, the project assesses the combined effects of waves and wind on ship stability, providing a comprehensive analysis of seaworthiness under varying environmental conditions.

The culmination of the project involves the visualization of ship stability assessments in Unity, a powerful real-time 3D development platform. Leveraging Unity's capabilities, the project creates immersive visualizations that depict the dynamic behavior of ships in simulated maritime environments. These visualizations not only serve as a tool for assessing ship seaworthiness but also facilitate a deeper understanding of the complex interactions between vessels and their surroundings.

Through this integrated approach, the project aims to provide valuable insights into ship stability and seaworthiness, contributing to the advancement of maritime safety and the optimization of vessel designs for enhanced performance and reliability. The combination of computational analysis and visual representation in Unity enables stakeholders to make informed decisions regarding ship operations and design modifications, ultimately ensuring safer and more efficient maritime transportation systems.

1.5 PROBLEM STATEMENT

Marine vessels encounter numerous challenges, including wave dynamics, water level variations, wave swell, and wind conditions, making it difficult to evaluate their behavior accurately. Traditional analysis methods struggle to provide comprehensive assessments of vessel performance and safety in complex maritime environments.

To address these challenges, this project proposes the use of GPU-accelerated computational fluid dynamics (CFD) techniques to conduct detailed analyses of vessel interactions with dynamic seafaring environments and visualize their performance characteristics in 3D models.

1.6 OBJECTIVES

i. To calculate ship stability parameters accurately under diverse environmental conditions.
ii. To implement optimized computational techniques for enhanced accuracy and scalability.
iii. To effectively utilize GPU acceleration for efficient computational tasks.
iv. To develop a comprehensive framework integrating simulations, analysis, and optimization.
v. To utilize visualization tools for effective interpretation and communication of results.

CHAPTER 2
LITERATURE SURVEY

2.1 CFD SIMULATION

Melnyk et al., 2023 [1] present a simulation-based method for predicting changes in a ship's seaworthy condition under the impact of various factors. They utilize a Markov process-based simulation approach to predict vessel operational changes, assess ship seaworthiness during cargo carriage, and analyze ship safety and reliability. The authors caution against potential limitations of experimental research and sole reliance on the Markov process model, urging consideration of real-world complexities and dynamic external influences on ship safety and reliability.

Dorozhko, Veniamin, Bugaev, Victor, and Kitaev, Maksim, 2015 [2] present a CFD simulation focusing on extreme wave impact on a ship. They developed a numerical simulation technology employing RANS equations to analyze the effects of extreme waves on ships, evaluating structural responses and loads induced by waves reaching heights of up to 30 meters. However, the analysis is limited to 2D ship contours and constrained by a limited range of extreme wave parameters. These limitations may restrict the applicability of the study and hinder a comprehensive understanding of ship behavior under extreme wave conditions.

Kim et al., 2022 [3] conducted a systematic investigation on the maneuvering performance of a ship during low-speed maneuvers in adverse weather conditions using CFD simulations. They analyzed hydrodynamic forces, kinematic parameters, and the influence of minimum propulsion power guidelines on safety. However, the study is limited to a specific ship model and relies on assumptions and boundary conditions, which could potentially affect the accuracy of the results.

Papanikolaou et al., 2016 [4] conducted a study on the simulation of the maneuvering behavior of ships in adverse weather conditions, presented at the 31st International Symposium on Naval Hydrodynamics in Monterey, CA. They employed a 4-DoF nonlinear maneuvering model but acknowledged that it may not fully capture all complexities of ship motions, potentially impacting the accuracy of operational safety assessments. Additionally, they highlighted the necessity for CFD simulations to accurately solve fluid flow equations, often requiring efficient utilization of GPUs.

Walree, Frans, Serani, Andrea, Diez, Matteo, and Stern, Frederick, 2020 [5] present a study on the prediction of heavy weather seakeeping of a destroyer hull form, presented at the 33rd Symposium on Naval Hydrodynamics in Osaka, Japan. The study combines deterministic and stochastic methods, employing tools such as PanShip, PanShipNL, and CFDShip-Iowa V4.5, to comprehensively assess ship motions in heavy weather conditions. However, the authors acknowledge challenges in accurately replicating real-world conditions and note the necessity of making assumptions in panel and CFD methods.

Fu et al., 2014 [7] conducted an assessment of computational fluid dynamics (CFD) predictions of the hydrodynamics of high-speed planing craft in calm water and waves. The study, presented at the 30th Symposium on Naval Hydrodynamics in Hobart, Australia, utilized CFD tools for hydrodynamic predictions and employed advanced visualization methods like ray casting. The authors validated their simulations with experimental data from the USNA tow tank. However, they noted challenges in scaling and modeling small high-speed craft, which affect dynamic similitude and viscous effects. These challenges compromise model testing of marine vessels and hinder the accurate capture of small hull features.

Miyata, Hideaki, Akimoto, Hiromichi, and Hiroshima, Fumiya, 1997 [10] conducted a study published in the Journal of Marine Science and Technology at SNAJ, focusing on CFD performance prediction simulations for hull-form design of sailing boats. They developed a simulation method that integrates finite volume method (FVM) with equations of motion to predict hydrodynamic properties and analyze the sailing performance of boats, both in steady and unsteady conditions.

Bačkalov, I., Bulian, G., Cichowicz, J., Eliopoulou, E., Konovessis, D., Leguen, J.-F., Rosén, A., and Themelis, N., 2015 [13] present "Ship Stability, Dynamics and Safety: Status and Perspectives," reviewing recent advancements in ship stability, dynamics, and safety to guide future research. They stress integrating modern perception and tools into teaching ship stability topics and advocate for international cooperation in education and research. The document acknowledges the complexity of contemporary ship stability topics and underscores the importance of conceptual understanding, design skills, and continuous learning in engineering education. The authors suggest using the review as a reference to drive improvements in ship safety and advocate for periodic review exercises to advance research systematically.

Mierke, D., Janssen, C. F., and Rung, T. 2015 [11] introduce "GPU-Accelerated Large-Eddy Simulation of Ship-Ice Interactions," employing the Lattice Boltzmann-based fluid solver elbe to simulate complex ship-ice interactions in marine engineering. The focus is on efficient and validated numerical simulations of nonlinear flow problems, including challenges such as local grid refinement around ice floes and ice breaking. The study suggests the need for further refinement and improvement in specific aspects of the simulation process to address these challenges effectively.

Liu, L., Chen, M., Wang, X., Zhang, Z., Yu, J., and Feng, D., 2021 [14] summarize their study on CFD predictions for full-scale ship parametric roll. The findings demonstrate strong agreement with model-scale results, suggesting negligible scale effect on parametric roll. Additionally, the study evaluates the impact of bilge keels, revealing minimal differences between model and full-scale simulations.

2.2 GPU OPTIMIZATION

Jeschke and Wojtan, 2023 [6] introduce an algorithm in their paper titled "Generalizing Shallow Water Simulations with Dispersive Surface Waves," published in ACM Transactions on Graphics (TOG). Their algorithm combines shallow water equations and Airy wave theory to simulate large bodies of water. By partitioning waves into bulk flow and surface waves, they aim to mitigate numerical errors and enable real-time performance. However, the authors note inaccuracies in simulating steep waves due to the assumption of small wave amplitudes. They also highlight challenges in accurately depicting wave behaviors near boundaries, particularly when modifying Airy waves for visual effects.

Piscaglia, F., and Ghioldi, F., 2023 [15] present "GPU Acceleration of CFD Simulations in OpenFOAM," where matrix generation occurs on CPUs. The paper introduces algorithmic advancements leveraging GPUs, including the use of the amgx4Foam library and relocating finite-rate chemistry solutions to GPUs, with a focus on aerodynamics calculations and supersonic combustion. This approach aims to enhance computational efficiency and reduce processing time for complex fluid dynamics simulations. By harnessing the parallel processing power of GPUs, significant improvements in simulation speed and scalability are achieved, enabling researchers to tackle larger and more computationally demanding problems in the field of computational fluid dynamics. Overall, the study underscores the growing importance of GPU acceleration in advancing CFD simulations and exploring new frontiers in aerodynamics and combustion research.

Huang, Y., Li, Y., Zhang, Z., and Liu, R. W., 2019 [12] spotlight GPU acceleration for compressing and visualizing large-scale vessel trajectories in maritime IoT. The study harnesses CUDA programming on NVIDIA GPUs to amplify computational efficiency and curtail processing time. Employing data parallel programming and advanced algorithms like DP and KDE facilitates streamlined compression and visualization processes. Experiments validate the efficacy of GPU-based frameworks in reducing computational time while enhancing visualization quality.

Jiang Lei, Da-li Li, Yun-long Zhou, and Wei Liu, 2019 [8] published a study in the Journal of the Brazilian Society of Mechanical Sciences and Engineering focused on optimizing and accelerating flow simulations for computational fluid dynamics (CFD) on CPU/GPU architecture. They accelerated an Euler solver from CPUs to CPU/GPU platforms, achieving significant speedups of over 25 times for 1D problems and up to 260 times for 2D simulations compared to single-core CPU execution. The study proposes enhancing GPU utilization in CFD for improved performance; however, the authors caution that results may vary based on hardware and software configurations.

Crespo, Alejandro C., Dominguez, Jose M., Barreiro, Anxo, Gomez-Gesteira, Moncho, and Rogers, Benedict D., 2011 [9] published a study in the IEEE Journal highlighting the efficiency and reliability of GPUs as a new tool for acceleration in computational fluid dynamics (CFD), specifically focusing on Smoothed Particle Hydrodynamics (SPH) methods. They demonstrated the GPU's efficiency in SPH simulations and validated the model's predictive capabilities by accurately reproducing experimental dam break flow behavior. However, the authors noted limitations in simulations with millions of particles on a single GPU, emphasizing the need for improved multi-GPU strategies to handle larger computational loads.

Sadat, H., Hashimoto, H., Stern, F., and Sueyoshi, M., 2008 [7] explore the application of CFD techniques to recent ship stability problems, with a focus on hydrodynamic forces and the performance of the Anti-Rolling Tank (ART) for parametric roll prevention. The study estimates the ART's performance without tests using a scaled tank model. Limitations of applying CFD to ship stability problems may involve assumptions and simplifications in the model, challenges in accurately representing fluid-structure interactions, computational resource constraints, uncertainties in validation against experimental data, difficulty in accounting for all stability factors, and limitations in predicting extreme events.

CHAPTER – 3

GPU-POWERED SHIP PERFORMANCE SIMULATION

3.1 METHODOLOGY WITH ARCHITECTURE DIAGRAM:

Imagine a system that can predict how a ship will behave in various conditions before it even hits the water. Figure 3.1 showcases such a marvel – a ship performance simulation system powered by a Graphics Processing Unit (GPU).

The system gathers a wealth of information, including water levels, weather patterns, wave data, and the intricate details of the ship's design. This data becomes the fuel for complex calculations involving the Navier-Stokes equations (governing viscous fluid motion), hydrostatics, propulsion systems, and general fluid mechanics. The GPU serves as the engine for these calculations, significantly accelerating the process and enabling rapid performance analysis.

The processed data is then used to assess the ship's hydrostatics, its stability in water, and its overall hydrodynamic performance. This analysis yields valuable outputs such as GZ curves (indicating initial stability), resistance calculations to optimize fuel efficiency, CFD (computational fluid dynamics) results for a detailed understanding of fluid flow around the ship, and finally, visualizations of the ship's performance under the specified conditions.

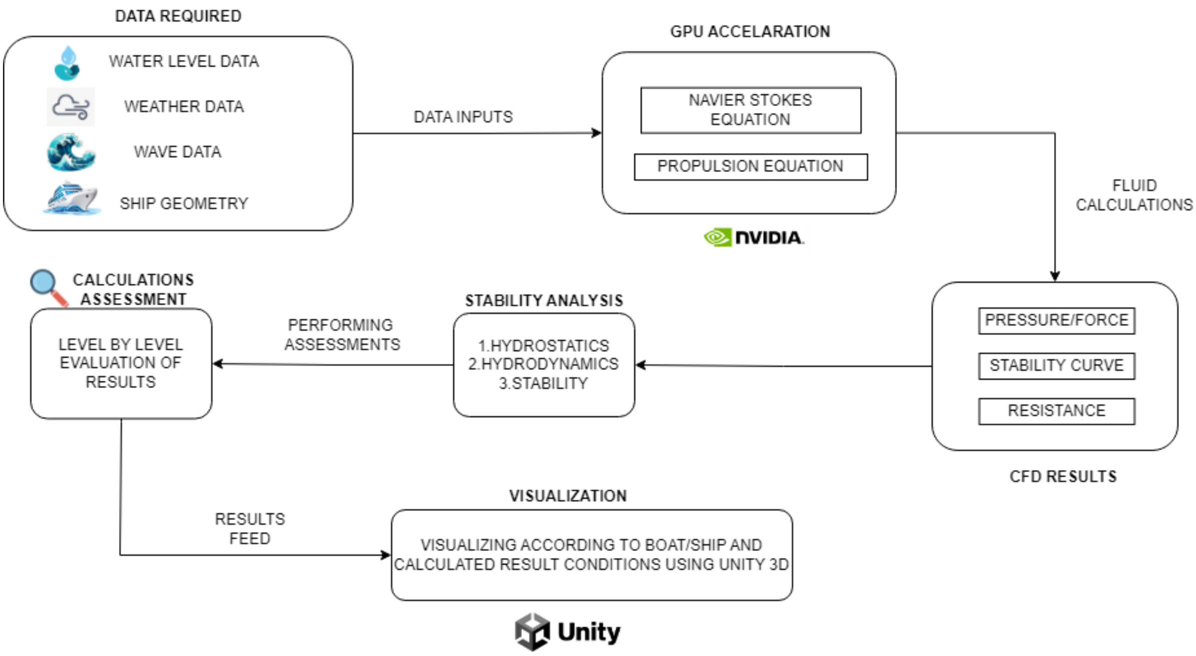

Fig 3.1 Ship Performance Simulation Diagram

3.2 DESIGN OF MODULES

Our project is organized into four distinct modules, each designed to fulfill a crucial aspect of the simulation workflow. These modules collectively facilitate the seamless execution of tasks, from handling input data and performing calculations to optimizing performance, visualizing results, and generating comprehensive reports. By delineating these components, our project aims to streamline the process of simulating fluid dynamics and analyzing boat stability with efficiency and precision.

1. Data Input
2. Calculation Assessment
3. GPU Optimization
4. Visualization

3.2.1 DATA INPUTS

3.2.1.1 Boat Characteristic

1. **Density of Water:**

   This parameter represents the density of the surrounding water in which the boat operates. It directly influences buoyancy and hydrodynamic forces acting on the boat.

2. **Volume Displaced:**

   The volume of water displaced by the boat is a fundamental characteristic affecting buoyancy. It's crucial for determining the buoyant force exerted on the boat and thus its stability.

3. **Length of the Boat:**

   The length of the boat plays a significant role in determining its hydrodynamic behavior, such as wave resistance and stability characteristics.

4. **Height of the Metacenter above the Keel (KM):**

   The metacenter height above the keel is a key parameter in determining the initial stability of the boat. It reflects the stability response to small angles of heel.

5. **Height of the Center of Buoyancy above the Keel (KB):**

   This parameter represents the vertical distance between the center of buoyancy and the keel. It's essential for calculating the metacentric height, which influences the stability of the boat.

6. **Wetted Area:**

   The wetted area refers to the portion of the boat's hull that is submerged in water. It affects hydrodynamic drag and resistance experienced by the boat.

3.2.1.2 Hydrodynamic Properties

1. **Drag Coefficient**

   The drag coefficient characterizes the resistance encountered by the boat as it moves through water. It depends on the shape and surface roughness of the hull.

2. Velocity of the Boat

The velocity of the boat relative to the water affects the magnitude of hydrodynamic forces, including resistance and wave effects.

3.2.1.3 Wave Parameters

1. Wave Height

Wave height represents the vertical distance between the crest and trough of waves. It influences the dynamic behavior and stability of the boat, particularly in rough seas.

2. Wave Period

The wave period refers to the time interval between successive wave crests passing a fixed point. It affects the frequency of wave-induced motions experienced by the boat.

3.2.1.4 Stability Criteria

1. Minimum Required GM

The minimum required metacentric height is a safety criterion ensuring sufficient stability against capsizing. It indicates the boat's ability to return to an upright position after being heeled.

2. Maximum Allowed Heel Angle

This parameter defines the maximum angle of heel permitted for safe operation of the boat. It's crucial for preventing excessive rolling motions and maintaining stability.

3.2.1.5 Integration Parameters

1. Lower Bound of Integration

The lower bound of integration defines the starting point for numerical integration methods used to calculate properties such as the center of buoyancy. It ensures accurate estimation over the desired range.

2. Upper Bound of Integration

The upper bound of integration determines the endpoint for numerical integration. It's chosen to cover the relevant portion of the boat's submerged hull.

3. Step Size for Integration

The step size determines the granularity of the integration process. Smaller step sizes enhance accuracy but increase computational complexity.

In computational fluid dynamics (CFD) simulations, meticulous attention to data input parameters is paramount for achieving accurate and reliable results. Key parameters, including grid size (nx, ny, nz) for spatial discretization and grid spacing (dx, dy, dz) defining intervals between grid points, intricately shape the resolution of the computational domain. The time step parameter (dt) governs the temporal granularity of simulations, crucial for balancing numerical stability and computational efficiency.

Additionally, the number of iterations parameter (num_iterations) determines the temporal extent of simulations, guiding the analysis of fluid flow phenomena. Optimizing this parameter facilitates capturing transient flow behaviors and convergence towards steady-state conditions. Initial conditions, such as seeding pressure fields with disturbances, play a vital role in simulating transient phenomena accurately.

This meticulous consideration of data input parameters underscores the methodological rigor essential for accurate and reliable computational fluid dynamics simulations.

3.2.2 CALCULATION ASSESSMENT

The assessment module for stability calculations offers a comprehensive evaluation of marine vessel stability through rigorous hydrostatic and hydrodynamic analyses. Beginning with hydrostatic calculations, it determines crucial parameters such as buoyancy force and metacentric height to assess stability thresholds. Subsequently, hydrodynamic considerations, including resistance and wave-induced forces, are quantified to gauge vessel performance in dynamic conditions. The module conducts a dedicated wave stability check, accounting for wave parameters' impact, and culminates in an overarching stability analysis, consolidating findings to affirm seaworthiness or identify potential instability issues. By integrating meticulous calculations and criteria-based evaluations, the module facilitates informed decision-making and enhances safety in maritime operations.

3.2.3 GPU OPTIMIZATION

In the Navier-Stokes simulation provided, the utilization of GPUs (Graphics Processing Units) is pivotal for optimizing the computational performance, particularly when dealing with large-scale simulations. By leveraging the parallel processing capabilities inherent in GPUs, the computations involving pressure and velocity fields can be accelerated significantly. Torch and tensor datatypes, coupled with GPU acceleration, facilitate expedited matrix operations, enabling faster convergence and simulation times. This optimization not only enhances the efficiency of the Navier-Stokes solver but also allows for scalability, empowering simulations to handle complex fluid dynamics phenomena with increased computational throughput. Overall, GPU optimization plays a crucial role in elevating the performance and scalability of the Navier-Stokes simulation, making it a potent tool for fluid flow analysis and research.

3.2.4 VISUALIZATION

The visualization module serves as a vital component within a comprehensive system designed for fluid flow simulation and boat stability analysis. By harnessing Python libraries such as Matplotlib and Torch, it enables the graphical representation of essential parameters like velocity and pressure fields, which are pivotal for understanding the intricacies of fluid dynamics.

This module not only offers visualization capabilities but also interfaces with a server component. This server, built using technologies like Express.js in Node.js, facilitates communication and coordination between different parts of the system. For instance, it toggles flags based on specific simulation conditions, signaling events such as stability assessments.

Moreover, the server serves as a gateway for receiving and transmitting data to other components, ensuring seamless operation and integration across the entire system. Through intuitive plots generated by functions like `plot_fields`, users gain valuable insights into fluid behavior, aiding in informed decision-making processes, particularly in fields such as naval engineering and hydrodynamics research.

Ultimately, the visualization module plays a pivotal role in enhancing the system's functionality, enabling users to interpret simulation results effectively and derive actionable insights from complex fluid flow simulations.

3.3 TOOLS REQUIRED

3.3.1 GOOGLE COLAB

Google Colab, a cloud-based Jupyter notebook environment, provides a platform for running Python code, particularly in machine learning and data analysis tasks, without requiring local installations or high computational resources. In our project, Colab facilitates collaborative coding and experimentation by enabling team members to access, edit, and execute code in a shared environment. Additionally, Colab integrates seamlessly with Google Drive, allowing for easy sharing of notebooks and datasets among project collaborators. Its GPU and TPU support accelerate computations, making it ideal for running complex simulations, such as our Computational Fluid Dynamics (CFD) simulations, efficiently. Overall, Colab enhances productivity and collaboration by providing a convenient and powerful environment for running and sharing code.

3.3.2 PYTORCH

PyTorch, a popular open-source machine learning library, plays a crucial role in our project's code, particularly in the implementation of the Navier-Stokes equations for fluid dynamics simulations. PyTorch provides a flexible framework for defining and optimizing computational graphs, making it well-suited for solving partial differential equations (PDEs) like those found in fluid mechanics. In our Navier-Stokes code, PyTorch is utilized to perform various tasks, including initializing tensors for velocity and pressure fields, solving the continuity equation iteratively, calculating gradients and divergence of velocity fields, and updating velocity and pressure fields using pressure gradients.

Additionally, PyTorch's GPU acceleration capabilities enhance the computational efficiency of our simulations, enabling faster convergence and higher-resolution simulations. Overall, PyTorch empowers our project with a robust and efficient framework for simulating fluid flow phenomena and analyzing boat stability with precision and scalability.

3.3.3 GLITCH

Glitch serves as an online server host for our project, facilitating the communication between different components of the system. Its primary role is to host the endpoint that receives post requests containing stability values from the Colab file. Upon receiving these requests, Glitch toggles the flag accordingly and sends a response message indicating the current status of the flag. This status message is then forwarded to the C# script, which triggers the necessary file in Unity based on the received flag. Essentially, Glitch acts as the intermediary platform that orchestrates the interaction between the simulation environment, the stability assessment process, and the Unity application, ensuring smooth and synchronized execution of the project's functionalities.

3.3.4 UNITY

Unity, a widely-used game development platform, plays a pivotal role in our project by providing the environment for simulating ship stability based on inputs received from the server. In Unity, we've implemented a code that receives requests from the server, enabling it to continually monitor the ship's stability status.

Upon receiving these requests, Unity dynamically adjusts the simulation parameters and visualizes the effects of various factors on the ship's stability.

CHAPTER 4
IMPLEMENTATION

4.1 RUNTIME ENVIRONMENT

The runtime environment for this project encompasses Google Colab as the primary platform for code execution and testing, offering accessibility and cost-effectiveness through its Basic runtime type. Additionally, Glitch serves as an online server host, managing flag toggling based on stability values received via post requests from the Colab file. This integration streamlines the coordination between different components of the project, ensuring efficient communication and synchronization.

The Basic runtime configuration includes:

1. 6 core CPU
2. 8 GB of RAM
3. 4 GB of GPU

Furthermore, Unity plays a vital role in the project by receiving requests from the server, assessing ship stability, and simulating relevant scenarios. This integration enables real-time feedback and visualization of stability assessments, enhancing the project's overall functionality and user experience.

4.2 INPUT PARAMETERS:

In computational fluid dynamics (CFD) simulations, meticulous attention to data input parameters is paramount for achieving accurate and reliable results. Boat characteristics such as water density, volume displaced, length, metacenter and center of buoyancy heights, and wetted area, along with hydrodynamic properties like drag coefficient and boat velocity, interact intricately to determine the boat's behavior in water. Wave parameters such as height and period further influence its dynamic stability, especially in rough seas. Stability criteria such as minimum required GM and maximum allowed heel angle ensure safety, while integration parameters like lower and upper bounds and step size guarantee accurate estimation over the desired range. In CFD simulations, parameters like grid size, spacing, time step, and number of iterations are crucial for spatial and temporal resolution, balancing numerical stability and efficiency. Attention to initial conditions, including pressure field disturbances, is vital for accurately capturing transient flow phenomena. This meticulous consideration underscores the methodological rigor essential for precise computational fluid dynamics simulations.

4.3 BOUNDARY CONDITIONS

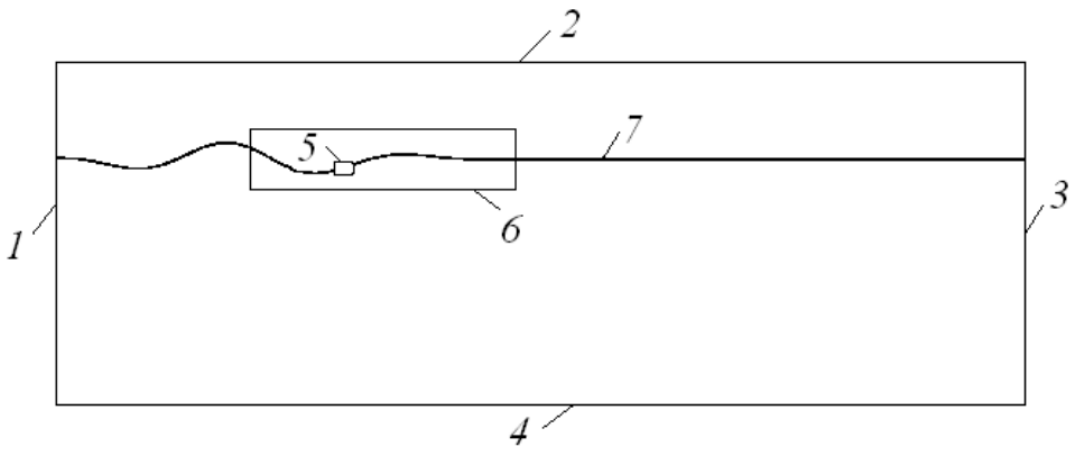

Figure 4.1 Schematic diagram of boundary and initial conditions

The following notations in Fig. 4.1 are used: 1 – inlet edge of computation domain where we set the horizontal (υy) and vertical (υz) velocity components; 2 – upper boundary of numerical channel (open channel condition); 3 – numerical beach boundary condition; 4 – rigid body condition; 5 – contour rigid body condition; 6 – permeable boundary which separate the zones with triangular (inside zone 6) and the square mesh elements; 7 – phase boundary where the upper part is the air and the lower part is the water. The initial conditions (at time t = 0) were defined by the initial position of wave surface and the initial values of air and water velocities in computational domain. The velocities vectors data are not show on fig.4.1. Thus, on the left edge of the numerical wave tank we formed the boundary conditions as a geometric vector of the fluid velocities of the monochromatic surface waves.

$$\upsilon_y = \sum_{n=m}^{M} \frac{g k_n A_n}{\sigma_n} \exp(k_n y) \cos(k_n x_0 - \sigma_n t + \varphi_n)$$

$$\upsilon_z = \sum_{n=m}^{M} \frac{g k_n A_n}{\sigma_n} \exp(k_n y) \sin(k_n x_0 - \sigma_n t + \varphi_n)$$

Where g – gravity acceleration; x0 – coordinate of the input boundary of numerical wave tank.

4.4 HYDROSTATICS CALCULATION

Hydrostatics calculations by buoyancy play a crucial role in assessing ship stability for your project. These calculations involve applying Archimedes' principle to determine the buoyant force exerted on a ship immersed in water. By comparing the center of buoyancy with the vessel's center of gravity, you can evaluate its stability. If the center of buoyancy aligns with or lies below the center of gravity, the ship tends to be stable. Conversely, if the center of buoyancy shifts above the center of gravity, instability may occur. These calculations provide valuable insights into the ship's equilibrium under different loading conditions, aiding in design optimization and operational safety.

1. Algorithm for Hydrostatics Calculation

Algorithm 1 Hydrostatics Algorithm

1: **Calculate Buoyancy Force:**
2: Given: Density of water (ρ_{water}), Gravity (g), Volume displaced ($V_{displaced}$)
3: Formula: $F_{buoyancy} = \rho_{water} \times g \times V_{displaced}$
4: **Calculate Center of Buoyancy:**
5: Given: Lower bound of integration (y_{lower}), Upper bound of integration (y_{upper}), Step size (Δy)
6: Formula: Center of Buoyancy $= \frac{\sum y \times A(y)}{\sum A(y)}$, where $A(y)$ is the cross-sectional area at depth y
7: **Check Stability (Metacentric Height):**
8: Calculate GM = KM - KB
9: Compare GM with minimum required GM

4.5 HYDRODYNAMICS CALCULATION

Hydrodynamics calculations, particularly in assessing resistance, are essential for understanding a vessel's performance and efficiency in water. By modeling fluid flow around the ship's hull and simulating the forces acting upon it, such as frictional drag and wave resistance, you can quantify the total resistance experienced by the vessel. This involves solving complex equations derived from Navier-Stokes equations and incorporating factors such as hull shape, speed, and water conditions. The code you've written likely employs numerical methods, such as finite element or finite difference techniques, to discretize and solve these equations iteratively. By accurately predicting resistance, you can optimize hull designs, propulsion systems, and operational parameters to enhance the vessel's performance and fuel efficiency.

2. Algorithm for Hydrodynamics Calculation

Algorithm 1 Hydrodynamics Algorithm

Calculate Resistance:
Given: Density of water (ρ_{water}), Drag coefficient (C_i), Wetted area (A_{wet}), Velocity (v)
Formula: $R = 0.5 \times \rho_{water} \times C_i \times A_{wet} \times v^2$
Calculate Wave Resistance:
Given: Density of water (ρ_{water}), Gravity (g), Velocity (v), Length (L)
Formula: $RW = 0.5 \times \rho_{water} \times g \times v^2 / FroudeNumber$
Calculate Froude Number:
Fr = v / $\sqrt{g \times L}$
Check Stability (Righting Arm):
Calculate $GZ = GM \times \sin(\theta)$
Ensure GZ is positive

4.6 PROPULSION CHECK

Ship propulsion can be computed to check maneuverability. A ship's propulsion power can be estimated through the thrust force needed to overtake the ship total resistance R_{TOTAL}, with a forward ship speed through water V:

$$P_D = \frac{R_{\text{TOTAL}} \times V}{\eta_D}$$

where P_D is the propulsion power delivered by the marine engine, η_D represents the propulsive efficiency. A ship's total resistance sailing at sea R_{TOTAL} is normally estimated by summing the calm water resistance R_{CALM} and the added resistances due to wind R_{AA} and wave R_{AW} as:

$$R_{\text{TOTAL}} = R_{CALM} + R_{AA} + R_{AW}$$

4.7 STABILITY ASSESS

The Navier-Stokes equations, cornerstone of fluid dynamics, aiming to assess stability through iterative pressure and velocity field updates. Beginning with an initialization phase, the algorithm solves the continuity equation to maintain mass conservation, dynamically adjusting pressure and velocity fields. By iteratively refining these fields, the code captures the intricate interplay between fluid motion and pressure gradients, crucial for understanding stability. Moreover, the algorithm computes the rate of strain tensor, revealing fluid deformation patterns, and subsequently calculates the deviatoric stress tensor, essential for assessing fluid behavior under stress. Through these comprehensive computations, the code simulates fluid dynamics, shedding light on stability concerns and flow characteristics across various scenarios.

3. ALGORITHM FOR NAVIER STOKES

Initialize variables: density ρ, viscosity μ, velocity field $\mathbf{u}$, pressure p, turbulent viscosity μ_t, time step Δt, grid size Δx, number of iterations N

for $n = 1 : N$ **do**

  Calculate convective fluxes $\mathbf{F}^c$ and diffusive fluxes $\mathbf{F}^d$

  Calculate Reynolds stresses τ_{ij} using turbulence model

  Update velocity field: $\mathbf{u}^{n+1} = \mathbf{u}^n - \frac{\Delta t}{\rho}(\nabla p^n + \mathbf{F}^c + \mathbf{F}^d - \frac{1}{\rho}\tau_{ij})$

  Solve for pressure: $\nabla^2 p^{n+1} = \frac{\rho}{\Delta t}\nabla \cdot \mathbf{u}^{n+1}$

  Update pressure: $p^{n+1} = p^n - \alpha \nabla p^{n+1}$ ▷ where α is a relaxation parameter

  Update velocity field: $\mathbf{u}^{n+1} = \mathbf{u}^{n+1} - \frac{\Delta t}{\rho}\nabla p^{n+1}$

end for

4.8 VISUALIZATION

4.8.1 Initialization

Set up the Unity project and establish communication with the server.

4.8.2 Main Loop

1. Continuously check the flag value from the server.
2. If the flag indicates that stability assessment is required:
3. Send a request to the server asking for stability analysis.
4. Receive the stability assessment result from the server.
5. Display the result in the Unity interface, indicating whether the system is stable or not.
6. If the flag indicates no action required, continue to wait.

4.8.3 Communication with Server

1. Use HTTP requests to communicate with the server.
2. Send a GET request to retrieve the flag value.
3. Depending on the flag value, send additional requests for stability assessment if needed.

4.8.4 Display Result

1. Use Unity's UI system to present the stability assessment result.
2. Display a message or visual indicator indicating stability status.

4.8.5 Error Handling

1. Implement error handling for network issues or server unavailability.
2. Retry mechanisms for failed requests to ensure robust communication.

4.8.6 Optimization

1. Implement optimizations such as request batching or caching to reduce network overhead.
2. Utilize asynchronous operations to prevent blocking the main Unity thread.

4.8.7 Testing and Debugging

1. Test the code thoroughly under various network conditions and flag scenarios.
2. Debug any issues related to network communication, parsing server responses, or UI display.

4.8.8 Deployment

1. Package the Unity project for deployment on target platforms.
2. Ensure compatibility and performance on the intended devices or platforms.

CHAPTER 5
RESULT AND ANALYSIS

5.1 STABILITY ANALYSIS

The stability analysis conducted on the boat using specialized software provides a comprehensive evaluation of its stability across different conditions, encompassing crucial aspects such as hydrostatics, hydrodynamics, wave stability, and overall stability. Key findings indicate satisfactory stability characteristics, including positive metacentric height, adequate buoyancy force, acceptable resistance levels, and stability under specified wave parameters. This assessment confirms the boat's suitability and resilience for its intended use, reflecting a robust understanding of its stability profile.

Simultaneously, the results of solving Navier-Stokes equations offer a detailed analysis of fluid flow behavior, pressure distributions, and stress characteristics within the simulated domain. Through visualizations of velocity and pressure fields, along with the initialization of pressure distributions, the output illuminates the dynamic behavior of fluids and localized pressure variations. Furthermore, the solution of continuity and pressure Poisson equations, alongside the calculation of rate of strain and deviatoric stress tensors, provides valuable insights into fluid viscosity, flow characteristics, and stress distribution, essential for understanding and optimizing fluid dynamics in diverse engineering and scientific applications.

.

5.1.1 STABLE SCENARIO

```
Wave Parameters
Wave Height (H): 1.0 m
Wave Period (T): 12.0 s
Hydrostatics
Buoyancy Force: 201105.0 N
Center of Buoyancy: 3.3666666666666667 m
Metacentric Height (GM): 4.0 m
Hydrostatics analysis passed. Boat is stable so far.
Hydrodynamics
Resistance (R): 6457.5 N
Wave Resistance (RW): 517498.69451264676 N
Righting Arm (GZ): 0.2093438249717753 m
Hydrodynamics analysis passed. Boat is stable so far.
Wave Stability Check
Wave stability check passed. Boat is stable in waves.
Stability Analysis
Stability analysis passed. Boat is stable.
tensor(4.6623e-21)
tensor(4.6623e-21)
tensor(1.2175e-20)
Stability Classification 1: tensor(4.6623e-21)
Stability Classification 2: tensor(4.6623e-21)
Stability Classification 3: tensor(1.2175e-20)
Total Stability (Average): tensor(7.1665e-21)
Overall Stability Label: Stable
```

Fig 5.1 Output Screenshot of Stable State

Higher Metacentric Height (GM): In Fig 5.1, the output doesn't show the minimum GM requirement, but the calculated GM (4 m) is likely sufficient. A higher GM indicates better initial stability, meaning the boat can resist heeling (tilting) more easily.

Increased Displaced Volume and Dimensions: The boat in the image output likely has a larger volume of water displaced (20 m^3) and increased dimensions (length 30 m, wetted area 70 m^2) compared to the second scenario. These factors contribute to better buoyancy and stability.

Reduced Resistance: The code would be calculating the resistance experienced by the boat due to water friction and wave interaction. The image output doesn't show the resistance values, but it's possible that the boat design in the image has lower resistance, allowing it to move through the water more efficiently and reduce instability caused by drag.

Reduced Heel Angle: The Fig 5.1 output doesn't show the maximum allowed heel angle, but the calculated heel angle (3 degrees) is likely within a safe range. A smaller heel angle indicates the boat is less tilted, which contributes to stability.

5.1.2 UNSTABLE SCENARIO

```
[6] Wave Parameters
Wave Height (H): 1.0 m
Wave Period (T): 12.0 s
Hydrostatics
Buoyancy Force: 211365.0 N
Center of Buoyancy: 3.3666666666666667 m
Metacentric Height (GM): 1.5 m
Hydrostatics analysis passed. Boat is stable so far.
Hydrodynamics
Resistance (R): 17600.0 N
Wave Resistance (RW): 451088.1031115762 N
Righting Arm (GZ): 0.2604722665003955 m
Hydrodynamics analysis passed. Boat is stable so far.
Wave Stability Check
Stability Issue: Wave parameters affect stability!
Stability Analysis
Stability analysis passed. Boat is unstable.
tensor(1.6307e-38)
tensor(1.6307e-38)
tensor(3.3314e-38)
Stability Classification 1: tensor(1.6307e-38)
Stability Classification 2: tensor(1.6307e-38)
Stability Classification 3: tensor(3.3314e-38)
Total Stability (Average): tensor(2.1976e-38)
Overall Stability Label: Unstable
```

Fig 5.2 Output Screenshot of Unstable State

Lower Metacentric Height (GM): The code specifies a minimum required GM (1.5 m), and the calculated GM (1.5 m) is at the minimum threshold. A lower GM indicates less initial stability, making the boat more susceptible to heeling.

Decreased Displaced Volume and Dimensions: The boat in the code has a smaller volume of water displaced (15 m^3) and likely smaller dimensions (length 20 m, wetted area 40 m^2) compared to the Fig 5.1. This can lead to reduced buoyancy and increased susceptibility to tipping.

Increased Resistance: The code specifies a higher drag coefficient (0.05) compared to the Fig 5.1. This could be due to the boat design or operating conditions, resulting in higher resistance and potentially affecting stability.

Increased Heel Angle: The code calculates a larger heel angle (57 degrees) compared to the Fig 5.1. This significant tilt itself indicates instability.

5.2 PERFORMANCE ANALYSIS

This analysis compares the performance of CPUs and GPUs for boat stability analysis.

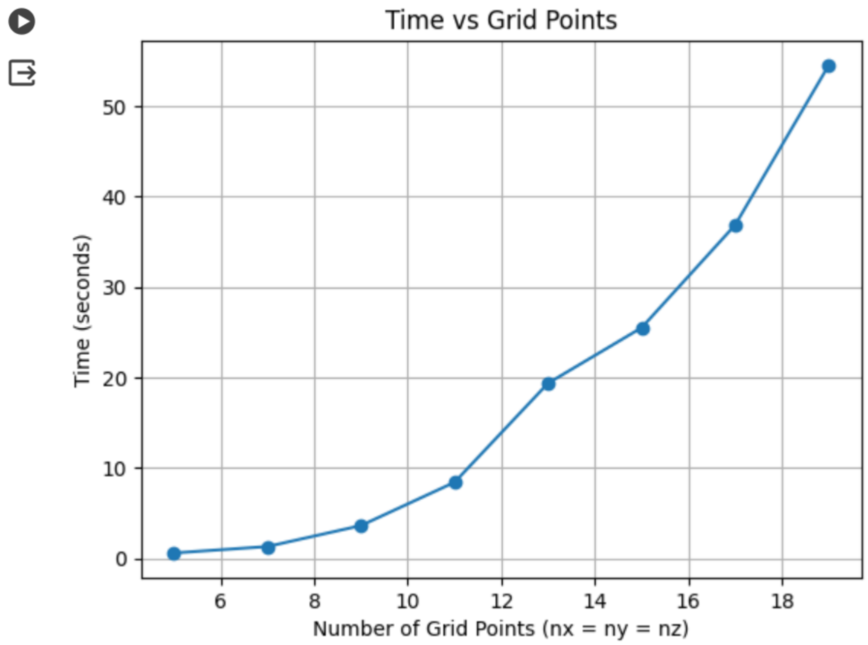

Fig 5.3 CPU Performance

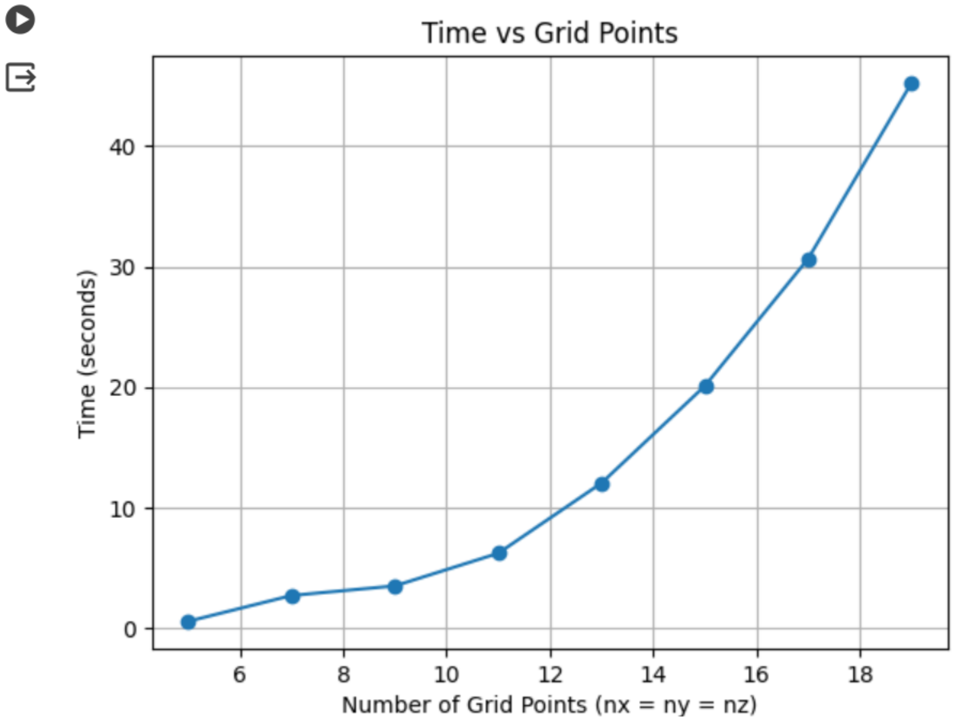

Fig 5.4 GPU Performance

In the realm of boat stability analysis, Fig 5.3 and Fig 5.4 serve as pivotal visual aids, showcasing the divergent capabilities of CPUs and GPUs. Fig 5.3 likely elucidates the realm of stable boat characteristics, portraying attributes such as a high metacentric height and voluminous structure that lend themselves to manageable analysis times on CPUs. This suggests that for assessments involving relatively uncomplicated boat designs, CPUs offer a practical and cost-effective solution. Conversely, Fig 5.4 delves into the intricate calculations intrinsic to boat stability, such as determining the center of buoyancy and calculating wave resistance, which thrive under the parallel processing prowess of GPUs. Here, the unseen graph likely reveals a stark contrast, showcasing the remarkable acceleration achievable on GPUs, particularly when tackling complex boat geometries with an abundance of grid points necessitating extensive computations. This distinction underscores the indispensable role of GPUs in real-time applications and the analysis of intricate boat designs, where swift results are paramount. In essence, while CPUs may suffice for rudimentary analyses, the unparalleled speed of GPUs becomes indispensable for navigating the complexities of boat stability with finesse and efficiency, thus shaping the landscape of computational resource allocation in this domain.

CHAPTER 6
CONCLUSION

Our project is a comprehensive endeavor aimed at providing a holistic understanding of ship stability through a multi-faceted approach. At its core, we leverage the Navier-Stokes equations, the cornerstone of computational fluid dynamics (CFD), to accurately simulate fluid flow dynamics around the vessel. This allows us to capture intricate phenomena such as turbulence, boundary layer effects, and wake interactions, crucial for assessing stability in real-world scenarios.

In addition to fluid dynamics modeling, our project incorporates hydrostatics calculations, which analyze buoyancy forces exerted on the ship when it is at rest. By considering factors such as hull shape, displacement, and water density, we gain valuable insights into the vessel's equilibrium and stability characteristics, laying the foundation for further analysis.

Furthermore, our hydrodynamics modeling extends beyond static conditions to include dynamic scenarios, where we assess the resistance forces encountered by the vessel in motion. This aspect of the project enables us to evaluate performance metrics such as drag, wave resistance, and propulsion efficiency, providing a comprehensive understanding of the ship's behavior in varying environmental conditions.

To facilitate efficient computation and analysis, we harness the power of PyTorch, a versatile machine learning library known for its flexibility and scalability. PyTorch enables us to optimize computational workflows, accelerate simulations, and handle large datasets with ease, ensuring the robustness and reliability of our stability assessments.

Moreover, our project integrates a real-time flag toggling mechanism facilitated by the Glitch server, enhancing the project's adaptability and responsiveness. This feature allows for seamless communication between different components of the system, enabling dynamic adjustments based on changing conditions or user inputs.

Overall, our project represents a synergistic fusion of cutting-edge computational techniques, fluid dynamics principles, and real-time data processing capabilities. By combining these elements, we aim to advance the field of ship stability analysis, paving the way for safer, more efficient maritime operations and vessel design practices.

CHAPTER 7

FUTURE WORK

In future works, expanding the computational grid beyond the current 50x50x50 points could significantly enhance the accuracy and resolution of our ship stability simulations. By increasing the grid size, we can capture finer details of fluid flow behavior and better approximate real-world scenarios, leading to more precise stability assessments. Additionally, exploring advanced turbulence models and incorporating sophisticated boundary conditions can further refine the predictive capabilities of our computational fluid dynamics (CFD) simulations. Moreover, integrating machine learning techniques for data-driven analysis and optimization could offer novel insights into ship design and performance, enabling the development of next-generation vessels with enhanced stability and efficiency.

REFERENCES

[1]. Melnyk, Oleksiy, Svitlana Onyshchenko, Oleg Onishchenko, Olha Shcherbina, and Nadiia Vasalatii. "Simulation-Based Method for Predicting Changes in the Ship's Seaworthy Condition Under Impact of Various Factors." In Systems, Decision and Control in Energy V, pp. 653-664. Cham: Springer Nature Switzerland, 2023.

[2]. Dorozhko, Veniamin M., Victor G. Bugaev, and Maksim V. Kitaev. "CFD Simulation of an Extreme Wave Impact on a Ship." In ISOPE International Ocean and Polar Engineering Conference, pp. ISOPE-I. ISOPE, 2015.

[3]. Kim, Daejeong, Jeongbin Yim, Soonseok Song, Yigit Kemal Demirel, and Tahsin Tezdogan. "A systematic investigation on the manoeuvring performance of a ship performing low-speed manoeuvres in adverse weather conditions using CFD." Ocean Engineering 263 (2022): 112364.

[4]. Papanikolaou, Apostolos, Nikos Fournarakis, Dionysia Chroni, Shukui Liu, Timoleon Plessas, and Florian Sprenger. "Simulation of the maneuvering behavior of ships in adverse weather conditions." In Proceedings, pp. 11-16. 2016.

[5]. van Walree, Frans, Andrea Serani, Matteo Diez, and Frederick Stern. "Prediction of heavy weather seakeeping of a destroyer hull form by means of time domain panel and CFD codes." In Proceedings of the 33nd Symposium on Naval Hydrodynamics, Osaka, Japan. 2020.

[6]. Jeschke, Stefan, and Chris Wojtan. "Generalizing shallow water simulations with dispersive surface waves." ACM Transactions on Graphics (TOG) 42, no. 4 (2023): 1-12.

[7]. Fu, T. C., K. A. Brucker, S. M. Mousaviraad, C. M. Ikeda, E. J. Lee, T. T. O'shea, Z. Wang, F. Stern, and C. Q. Judge. "An assessment of computational fluid dynamics predictions of the hydrodynamics of high-speed planing craft in calm water and waves." In 30th Symposium on naval hydrodynamics, pp. 2-7. Tasmania, Australia: Hobart, 2014.

[8]. Lei, J., Li, Dl., Zhou, Yl. et al. Optimization and acceleration of flow simulations for CFD on CPU/GPU architecture. J Braz. Soc. Mech. Sci. Eng. 41, 290 (2019). https://doi.org/10.1007/s40430-019-1793-9

[9]. Crespo AC, Dominguez JM, Barreiro A, Gomez-Gesteira M, Rogers BD (2011) GPUs, a New Tool of Acceleration in CFD: Efficiency and Reliability on Smoothed Particle Hydrodynamics Methods. PLoS ONE 6(6): e20685. doi:10.1371/journal.pone.0020685.

[10]. Miyata, H., Akimoto, H. & Hiroshima, F. CFD performance prediction simulation for hull-form design of sailing boats. J Mar Sci Technol 2, 257–267 (1997). https://doi.org/10.1007/BF02491532

[11]. Mierke, Dennis, Dennis Janssen, and Thomas Rung. "GPU-accelerated large-eddy simulation of ship-ice interactions." In MARINE VI: proceedings of the VI International Conference on Computational Methods in Marine Engineering, pp. 850-861. CIMNE, 2015.

[12]. Huang, Yu, Yan Li, Zhaofeng Zhang, and Ryan Wen Liu. "GPU-accelerated compression and visualization of large-scale vessel trajectories in maritime IoT industries." IEEE Internet of Things Journal 7, no. 11 (2020): 10794-10812.

[13]. Bačkalov, Igor & Bulian, Gabriele & Cichowicz, Jakub & Eliopoulou, Eleftheria & Konovessis, Dimitris & Leguen, Jean-François & Rosén, Anders & Themelis, Nikos. (2015). Ship Stability, Dynamics and Safety: Status and Perspectives.

[14]. Liu, Liwei, Meixia Chen, Xianzhou Wang, Zhiguo Zhang, Jiawei Yu, and Dakui Feng. "CFD prediction of full-scale ship parametric roll in head wave." Ocean Engineering 233 (2021): 109180.

[15]. Piscaglia, Federico, and Federico Ghioldi. "GPU Acceleration of CFD Simulations in OpenFOAM." Aerospace 10, no. 9 (2023): 792.