Technische Universität München
TUM School of Engineering and Design

Real-time simulation of the ship-rotorcraft dynamic interface using local recalculation with the Lattice-Boltzmann method

Jakob Bludau

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Ilkay Yavrucuk

Prüfende der Dissertation:

1. Prof. Dr.-Ing. Manfred Hajek
2. Prof. Dr. Jürgen Rauleder

Die Dissertation wurde am 22.5.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 17.07.2024 angenommen.

Abstract

Rotorcraft ship deck operations require highly skilled pilots as the rotorcraft's handling constantly changes due to the aerodynamic interaction with the ship. Pilots learn to master the situation by extensive training in costly training flights. An alternative is flight simulation, which reduces training costs while improving safety. The flight simulator model is required to adequately predict these complex aerodynamic interactions in real-time. All models that aim to predict these interactions are called ship-rotorcraft dynamic interface models.

This thesis describes, implements, verifies, and validates a real-time capable ship-rotorcraft dynamic interface model. The model, named the Local Recalculation Model, uses a Lattice-Boltzmann-based fluid solver to combine the wake of the rotorcraft with the ship wake inside a small domain around the rotor. This thesis investigates the model's capability to predict the stationary and dynamic behavior of the rotorcraft in free flight, ground effect, wall effects, and recirculations behind the hangar doors of a NATO Generic Destroyer. The verification compares the model's predictions to flight tests, wind tunnel data, and algebraic models. Furthermore, a comparison with an overset simulation validates the model's capability to predict ship-rotorcraft interactions during an approach of an MBB Bo 105 onto a NATO Generic Destroyer.

The results indicate that the Local Recalculation Model adequately predicts the physical phenomena of ship-rotorcraft interaction. However, it fails to predict the ground effect in hover and forward flight. The fluctuations predicted by the Local Recalculation Model during the approach are comparable to a reference simulation using the overset technique. Nevertheless, assessing the model's validity for flight simulation of rotorcraft ship deck operation requires a piloted evaluation of the model.

Zusammenfassung

Start und Landung von Hubschraubern auf fahrenden Schiffen stellt hohe Anforderungen an die Piloten, da die aerodynamische Interaktion der Rotoren mit dem Nachlauf des Schiffes das Steuerverhalten des Hubschraubers fortlaufend ändert. Das damit einhergehende Risiko lässt sich durch intensives Training der Piloten in entsprechenden Szenarien zwar reduzieren, allerdings ist der finanzielle Aufwand signifikant. Pilotentraining in Flugsimulatoren bietet eine Alternative, die sowohl die Kosten als auch das Risiko für Mensch und Maschine reduziert. Jedoch erfordert dies ein echtzeitfähiges Flugsimulationsmodell, welches die komplexen aerodynamischen Interaktionen zwischen Schiff und Hubschrauber abbildet. Modelle, welche diese Interaktion vorhersagen, werden im Englischen als “ship-rotorcraft dynamic interface model‘‘ bezeichnet.

Diese Doktorarbeit beschreibt, implementiert, verifiziert und validiert ein echtzeitfähiges Schiff-Hubschrauber-Interaktionsmodell. Das “Local Recalculaton Model‘‘ genannte Modell benutzt einen Lattice-Boltzmann Fluidlöser, um die Interaktion der Nachläufe des Rotors und des Schiffs abzubilden. Die vorliegende Arbeit untersucht die Fähigkeit des beschriebenen Modells, stationären und instationären Hubschrauberfreiflug, Bodeneffekt, Wandeffekte und Rezirkulation hinter dem Hangar eines NATO Generic Destroyers zu berechnen. Dafür werden die Vorhersagen des Modells mit Flugversuchsdaten, Messdaten aus Windtunnelversuchen und algebraischen Modellen verglichen. Außerdem werden die vom Modell vorhergesagten Rotorkräfte und -momente während eines Anfluges einer MBB Bo 105 auf den NATO Generic Destroyer mit einer Simulation mit “overset‘‘-Technik verglichen.

Die Ergebnisse zeigen, dass das “Local Recalculation Model‘‘ die physikalischen Phänomene in Schiff-Hubschrauber-Interaktionen adäquat vorhersagt. Allerdings erzielt es nur ungenügende Vorhersagen in Bezug auf den Bodeneffekt im Schwebe- und Vorwärtsflug. Die vom Modell berechneten Fluktuationen der Rotorkräfte und -momente während des Anfluges sind vergleichbar zur Referenzrechnung mittels “overset‘‘-Technik. Jedoch sind Simulatorstudien mit Marinepiloten nötig, um die Validität des Modells für schiffgestützte Hubschraubereinsätze zu bewerten.

Dedicated to helicopters with their absurd mechanical design and history.
For the countless times they reminded me of how little I know, how much there is to discover, and how hard it is to not fool yourself.
Trying to understand them was a truly humbling experience.

Acknowledgments

My thanks go to my doctoral advisor, Professor Dr.-Ing. Manfred Hajek, thank you for your mentorship during this thesis. Furthermore, I would like to thank the German Aerospace Center DLR for providing me with the MBB Bo 105 flight test data and Wolfgang von Grünhagen for providing the necessary information to interpret the data files. Additionally, I want to thank the Office of Naval Research of the United States of America for supporting the investigations in this thesis and the insightful discussions at the annual meetings.

Many thanks go to Dr. Jürgen Rauleder, now a Professor at the Georgia Institute of Technology, for the ongoing discussions that inspired countless iterations of this thesis's thoughts.

My special thanks go to the office colleagues Andreas Reiser and Bastian Horvat for all the grand discussions about technical details, software design, life goals, reason, morals, and politics. There was always time for jokes or serious talks, whatever was appropriate. Furthermore, the countless pair-programming hours and fierce software design discussions with Bastian were crucial to the success of this thesis.

Last but not least, thanks go to Martina Thieme for her incredible support during all the time. Together with your colleagues at the institute, you made the time enjoyable.

And to all the unnamed but essential persons in my life - thank you for your support!

Table of Contents

List of Figures

List of Tables

Nomenclature

Superscripts

$*$	marks an intermediate value
$+$	computed from populations moving out of the domain
0	computed from populations moving tangential to the boundary and $\hat{f}_0$
$\boldsymbol{\chi}$	of Lagrangian-Eulerian formulation
$\boldsymbol{X}$	of Eulerian formulation
$\boldsymbol{x}$	of Lagrangian formulation
$\hat{\star}$	Discrete value of $\star$
∞	Reference value in far distance
$\bar{}$	Mean value
$\prime$	Used to mark a variable as post collision
ψ	of blade revolution angle ψ
$\star$	Used to mark a variable as different to without the mark
$\star^T$	Transposed tensor $\star$
aer	of aerodynamic forces
blades	of blades
cell	of cell
cg	of the center of gravity
cosine	of cosine coefficients
eq	in thermodynamic equilibrium
h	in hover
Mx	of rotor moment around x-axis
My	of rotor moment around y-axis

$NatoGD$	of NatoGD
neq	non equilibrium part
oge	Out of ground effect
pp	of Pitt-Peters inflow model
ref	Reference value
SGS	of sub-grid stress model
$sine$	of sine coefficients
$static$	of non-moving entity
T	of rotor thrust
$Wall$	of wall
$wave$	of wave

Symbols

β	Flap angle of blade
$\boldsymbol{\chi}$	Coordinate system in Lagrangian-Eulerian space
$\boldsymbol{\Phi}$	Projection from Lagrangian-Eulerian space into Lagrangian space
$\boldsymbol{\Psi}$	Projection from Lagrangian-Eulerian space into Eulerian space
$\boldsymbol{\varphi}$	Projection from Eulerian space into Lagrangian space
$\boldsymbol{\xi}$	Molecular velocity
$\boldsymbol{c}$	Mean speed of sound vector
$\boldsymbol{I}$	Identity tensor
$\boldsymbol{n}$	Outward pointing normal vector
$\boldsymbol{Q}$	Q tensor
$\boldsymbol{S}$	Strain rate tensor
$\boldsymbol{u}$	Velocity vector
$\boldsymbol{v}$	Induced velocity vector
$\boldsymbol{X}$	Coordinate system in Eulerian space
$\boldsymbol{x}$	Coordinate system in Lagrangian space

$\boldsymbol{x}$	Position vector
$\Delta\boldsymbol{x}$	Discrete spatial distance between cells
Δt	Discrete time step of LBM scheme
ϵ^{star}	Area fraction $\frac{A^{\star}}{A^{cell}}$
$\frac{D\star}{Dt}$	Total derivative of $\star$ with respect to time
ν	Dynamic viscosity
Ω	Rotational velocity
ω	Relaxation frequency
Φ	Roll angle of fuselage
ϕ	Angle of local blade element with respect to the unloaded rotor disk
Ψ	Yaw angle of fuselage
ψ	Revolution angle of blade
ρ	Density
σ	Rotorshaft angle with respect to fuselage
Θ	Nick angle of fuselage
ζ	Lead-lag angle of blade
a	Reach factor of LRM
C	non-dimensional coefficient
c_S	Speed of sound in LBM
$d\star$	Differential of $\star$
dA_C	Differential particle cross section
f	Particle distribution function $f(\boldsymbol{x}, \boldsymbol{\xi}, t)$
i	Index
J	Momentum
Kn	Knudsen-number
L	Length
m	Total particle mass

Ma	Mach number
N	Resolution
n	Number of cells
n	Number of particles
O	Order in big O notation
o	Offset
P	Power
R	Boltzmann constant
R	Rotor radius
R	Vector space
r	Radial coordinate of blade
s	Smagorinksy constant
T	Temperature
t	Time
TKE	Turbulent Kinetic Energy
w	Weight factor in LBM
z	Acoustic impedance

Subscripts

be	in coordinate system of corresponding blade element
cg	in body-fixed coordinate system with origin in center of gravity
g	in geodetic coordinate system with origin in center of gravity
i	ith component of tensor
j	jth component of vector
k	kth element of vector
$lattice$	in non-dimensional unit system of LBM lattice
mr	in body-fixed main rotor coordinate system
ned	in north-east-down coordinate system with origin in center of gravity
$physical$	in physical unit system
tr	in body-fixed tail rotor coordinate system

Acronyms

ALE	Arbitrary Lagrangian-Eulerian.
BEM	Blade Element Method.
BGK	Bhatnagar-Gross-Krook Model.
DES	Detached Eddy Simulation.
DLR	Deutsches Zentrum für Luft- und Raumfahrt.
DNS	Direct Numerical Simulation.
FAA	Federal Aviation Administration.
FVWM	Free Vortex Wake Method.
GPGPU	General Purpose Graphics Processing Unit.
KIT	Karlsruhe Institute of Technology.
LBM	Lattice-Boltzmann Method.
LES	Large Eddy Simulation.
LGA	Lattice Gas Automata.
LRM	Local Recalculation Model.
MBB Bo 105	Messerschmitt-Bölkow-Blohm Bo 105.
MRT	Multiple-Relaxation Times.
MTOW	Maximum Takeoff Weight.
NatoGD	NATO Generic Destroyer.
RAM	Random Access Memory.
SFS	Simple Frigate Shape.
SFS2	Simple Frigate Shape 2.
SSS	Shift and Swap Streaming Model.
URANS	Unsteady Reynolds Averaged Navier-Stokes.
VPM	Viscous Particle Method.

VTOL	Vertical Take-off and Landing.

Clarification on Self Citation

This work amends the three key papers [1, 2, 3] by the author of this thesis. All of them originated from the authors work at the institute. Nevertheless, the used code changed constantly as errors were fixed, parts redesigned, and models refined. Therefore, all plots in this document are created with one state of the used code and therefore are not cited from the papers. The used code is available in the institute gitlab:

```
https://gitlab.lrz.de/HTMWTUM/olb-ripoff.git
branch: jb_dissertation
```

1 Introduction

Ship deck take-off and landing are high-risk phases in naval rotorcraft operations. To counter the increased risk, pilots extensively train the respective maneuvers. However, direct training of rotorcraft take-off and landing on a moving ship's deck is expensive. Flight simulators are a promising tool for reducing the cost of training and increasing safety in ship deck take-off and landing. Therefore, a flight simulator model that captures the physical interactions of ship and rotorcraft is needed. This model would allow pilot training for various wind and sea state conditions in the safe environment of rotorcraft flight simulators. Besides the expectation of significant cost savings compared to training flights, the pilots' learning experience improves. The possibility to reiterate an approach in the same conditions, or do multiple consecutive runs in varying conditions, allows for compacter training cycles that can adapt to the student's needs. Furthermore, student errors during the simulation do not pose immediate threats to the life of the crew or result in high costs due to loss of or damage to the rotorcraft or ship. The reduced risk and fewer consequences of errors result in reduced stress during training, thus allowing for a high-quality learning environment.

The advantages of a model suitable for rotorcraft ship deck landing have inspired the scientific community for a long time. Continuous research created various models of the ship-rotorcraft interaction with different levels of detail. At their core, all these so-called ship-rotorcraft dynamic interface models aim to predict the interactions between the ship's and rotorcraft's wake. This wake interaction disturbs the inflow of the rotor, which changes the forces and moments produced by the rotor. The pilot feels this interaction, as these rotor forces and moments accelerate the rotorcraft. Depending on the desired flight path, the pilot adjusts the control inputs in response to the acceleration. Figure 1.1 depicts the major components of this network of interactions in abbreviated form.

Inside the figure, (—) arrows represent the interactions that ship-rotorcraft dynamic interface models aim to predict. Additionally, (—) arrows mark the general control loop of rotorcrafts, including the pilot. During ship-rotorcraft operations, all components of this network interact simultaneously. Additionally, most of the interactions are non-linear. A flight simulator model for pilot training must adequately capture all interactions in Fig. 1.1. Furthermore, using contemporary hardware, the model must be fast enough to advance its simulation time faster than the wall clock time.

The simplest ship-rotorcraft dynamic interface models superimpose velocity fluctuations extracted from a precomputed ship wake onto the rotor inflow resulting from algebraic dynamic inflow models. Although these models compute the results fast enough for flight simulation, the missing interaction of the rotor thrust perturbing the

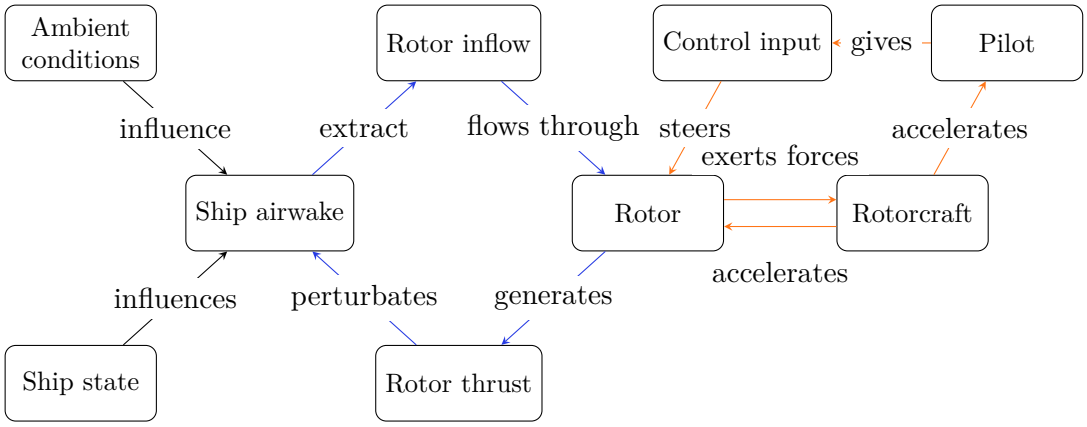


Figure 1.1: Sketch of the physical interactions present in rotorcraft ship deck operations. The (—) arrows mark the rotorcraft-ship dynamic interface, (—) arrows mark the general control loop of rotorcrafts.

ship wake removes an essential interaction in flight close to the ship's deck. More elaborate and recently proposed models build large databases of high-order CFD results. The databases include assorted rotor thrust distributions at different positions in the ship wake. Nevertheless, as the database does not contain all possible states of the interaction of the ship and rotorcraft, interpolation of the states during the flight simulation is necessary. Apart from the errors introduced by the interpolation, these models do not incorporate the full non-linearity of the interactions.

This thesis describes a new model capable of capturing the non-linearity of the interactions in the ship-rotorcraft dynamic interface while being fast enough to be used in piloted flight simulation. This model uses a comparatively small fluid domain that surrounds the rotorcraft. Inside this domain, the Lattice-Boltzmann method calculates the interaction of the ship's and rotorcraft's wakes. The ship air wake, which requires a large domain due to its turbulent nature, is calculated and tabulated for a selected set of ambient conditions and ship geometries in a preprocessing step. During the flight simulation, boundary conditions introduce the tabulated data into the domain surrounding the rotorcraft. As this model recalculates the ship air wake including the influence of the rotor inside the domain that surrounds the rotorcraft, it is called Local Recalculation Model (LRM).

This thesis focuses on the following central investigations to evaluate the LRM in the context of ship-rotorcraft interaction :

- Can the LRM model the rotorcraft-ship dynamic interface realistically enough to improve piloted simulator training?
- Does the computational efficiency enable real-time capable simulation of rotorcraft ship deck operations?

A positive answer to both questions implies that the LRM can enhance pilot training for naval operations in flight simulators. However, with a negative answer to the second

question, the model can still improve controller design for ship-rotorcraft operations or be used to investigate ship-helicopter operation limits.

Structure of this thesis

The development of the LRM in this thesis follows the V-model. The V-model enforces a systematic and structured approach to development tasks. It decomposes the system development lifecycle into dependent steps which include verification and validation to ensure compliance of the result with the requirements set at development start. The model's left side describes the refinement of the definition, the bottom is the implementation, and the model's right side describes the test and integration steps. In general, the level of abstraction in the V-model decreases from top to bottom. Figure 1.2 depicts the V-model.

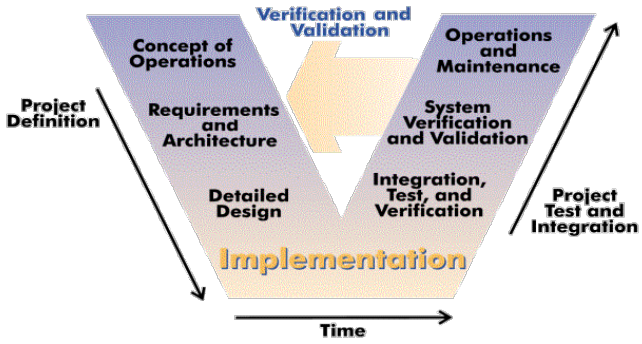


Figure 1.2: V-model depicting a system development lifecycle. Taken from [4].

Most chapters of the thesis can be associated with steps in the V-model. Nevertheless, this thesis excludes the "Concept of Operation" and "Operation and Maintenance" steps, which form the top level in the V-model. Chapter 2 formulates physical and computational requirements a ship-rotorcraft dynamic interface model has to fulfill to be used in piloted flight simulations. Therefore, it forms the first half of the V-model's "Requirements and Architecture" step. With the requirements set, chapter 3 presents an overview of existing models for the ship-rotorcraft dynamic interface and evaluates their compliance with the requirements. Following this, chapter 4 presents the central difference between the LRM and the other models and highlights the key idea. It corresponds to the "Architecture" step. Chapter 5 decomposes the physical interactions shown in Fig. 1.1 into domains and interfaces. Therefore, it represents the the V-model's "Detailled Design" step. After the decomposition, chapter 6 describes the modeling of the domains and the implementation of the interfaces. This chapter represents the V-model's "Implementation" step. The chapters 7 and 8 verify that the calculated ship wake and the LRM fulfill the requirements specified in chapter 2. Therefore, they correspond to the ascending right side and horizontal arrow between the descending and ascending part of the V-model. Chapter 9 validates the LRM's

capability to predict the ship-rotorcraft interaction during an approach. Finally, chapters 10 and 11 summarize the findings, conclude the thesis, and present an outlook and ideas for further work.

2 Requirement Specification

Prior to decomposing the model into domains, a set of requirements is established against which a ship-rotorcraft dynamic interface model can be verified. The requirements originate from the central questions of this thesis listed in chapter 1. This derivation of requirements from the overarching goals corresponds to the first part of the "Requirements and Architecture" stage in the V-model. If a ship-rotorcraft dynamic interface model fulfills the requirements, it adequately represents the physics of ship deck landing and is fast enough to be used in piloted flight simulation.

2.1 Relevant physical phenomena

The dynamic rotor inflow is coupling the rotorcraft with surrounding objects [5, 1, 3]. Figure 1.1 in the introduction depicts this for the case of an interaction with a ship. The air that streams through the rotor, referred to as inflow, transports the information from surrounding objects into the rotor. Thus everything that changes the inflow into the rotors influences the rotorcraft. Therefore, all ship-rotorcraft dynamic interface models are inflow models.

First, as all approaches feature straight flight paths, the ship-rotorcraft dynamic interface model needs to correctly predict the rotorcraft dynamics, power, and controls in hover and stationary forward flight [6]. Furthermore, it is required to predict the inertia effect of the inflow correctly [6, 1, 7]. A positive step input in the collective can illustrate this. After the step input, the pitching angle of the blade is higher, while the inflow is still the same as before the step input due to the inertia of the airflow. Thus the blade's angle of attack increases, and therefore the forces of the rotor increase. After the step input, the inflow reacts to the forces of the rotor, which accelerates the fluid flowing through [7]. Due to the increasing inflow, the blade angle of attack reduces, which reduces the rotor thrust. Because of the inflow's inertia, the rotor produces a surplus of thrust until the inflow is adapted [1].

Apart from correctly predicting the inflow in undisturbed flight, the ship-rotorcraft dynamic interface model needs to capture all effects that can change the inflow during the ship deck approach and landing. For simplicity, this thesis restricts the wind to the direction the ship moves (bow into the wind). Nevertheless, other angles between ship's path and wind might be relevant for pilots [8, 9].

During the approach, the number of physical phenomena that are concurrently present increases. In the early approach stage, the helicopter flies in the ship wake but not over the deck. The resulting distance to the ship structure limits the interaction to the ship wake disturbing the rotors. Furthermore, in this stage, the rotorcraft

flies fast relative to the air, which increases the stability of the rotorcraft in the case of a conventional helicopter [10].

A partial ground effect can occur once the rotorcraft flies above the landing deck. Aerodynamic investigations of ship-rotorcraft interaction use models like the Simple Frigate Shape 2 (SFS2) [11, 8] or the NATO Generic Destroyer (NatoGD) [12, 9], which feature a flat landing deck in the stern. Figure 5.2 shows a schematic of the NatoGD. As the rotorcraft is flying over the landing deck, parts of the rotor experience a ground effect of a possibly inclined and moving ship deck. Other parts of the rotor not positioned over the landing deck interact with the sea's surface, which can take various complex shapes. Furthermore, in this stage, the rotorcraft flies slower, which decreases stability [10]. Additionally, the interaction between the rotor and ship wake persists. In this stage, the rotor forces actively change the ship wake and flow separations at the ship structure.

If the ship features a hangar, the doors are often close to perpendicular to the flight deck; see Fig. 5.2 for the NatoGD. As the doors act aerodynamically like a backward-facing step, they produce an area with flow separation downstream. During the final stages of the approach, the rotorcraft flies close to the moving deck and hangar while interacting with, and influencing, the ship wake. Therefore, the separations behind the ship superstructure can create a considerable variation in the inflow of the rotors of an approaching rotorcraft. Furthermore, the hangar doors and the flight deck can create a recirculation area when deflecting the rotor wake. As the wake of the rotor develops, the deck and doors deflect it in front and above the rotorcraft. This deflection leads to the rotor drawing in its own wake.

In summary, a ship-rotorcraft dynamic interface model needs to capture the following physical phenomena:

- Control and power in stationary forward flight
- Dynamic behavior of the inflow and rotorcraft
- (Partial) ground effect of the moving ship deck
- Wall effects of moving hangar
- Aerodynamic wake of a ship
- Recirculation of the rotor wake

All these phenomena are of varying importance in different stages of the approach of a rotorcraft onto a ship deck. Furthermore, the requirement concerning the aerodynamic wake of a ship needs to be further specified to be testable. Section 2.2 discusses this requirement in detail.

2.2 Relevant turbulent spectra of ship wake

Experimental [11, 12] and numerical [9, 13] investigations show that, in general, the wake of naval vessels that feature a rotorcraft landing spot is turbulent. Neverthe-

less, not all turbulent scales in the wake are equally important for a rotorcraft flying through. Generally, the inertia of manned rotorcraft acts as a low-pass filter, filtering high frequent loads at the rotor blades from influencing the flight path [9]. Furthermore, as conventional rotorcraft feature flexible blades, disturbances in the flow through the rotor cause the blades to bend, acting as an additional filter. While the resulting high-frequency loads cause vibration inside the rotorcraft, they do not influence the flight path of the rotorcraft.

Nevertheless, disturbances in the $0.2Hz$ to $2Hz$ frequency range require a high pilot workload to compensate [9]. These frequencies are in the closed-loop pilot response frequency range [14, 15], so they can cause pilot-induced oscillations [9]. Therefore, all turbulent spectra that can induce disturbances of the rotor forces in this frequency range must be present in the ship wake used for a ship-rotorcraft dynamic interface model.

Disturbances with frequencies below $0.2Hz$ lead to a reaction of the rotorcraft that the pilot might need to compensate for [9]. Especially during take-off and landing on the moving ship deck that feature flight close to solid walls and ground, the pilot has to prevent undesired rotorcraft motion. Apart from vortices in the turbulent wake, the separation areas behind the ship superstructure can induce forces of the rotors in this frequency band. Therefore, this thesis requires a correct prediction of these flow separation areas.

In summary, a ship wake used for a ship-rotorcraft dynamic interface model needs to fulfill the following requirements:

- Correctly predict flow separation areas behind the ship superstructure
- Capture the turbulent nature of the ship wake
- Predict the turbulent spectra that induce disturbances of $2Hz$ and lower in the rotorcraft

2.3 Solving the equations in real-time

As flight simulators use the ship-rotorcraft interface model to train pilots, the model must fulfill a set of computational requirements. Besides the ability to run on contemporary hardware, the most restrictive requirement states that the model has to answer within a time limit. Furthermore, it has to logically and correctly compute the answer. Therefore, the model must not repeat a previous answer when reaching the time limit while calculations are unfinished. Shin and Ramanathan [16] define this set of requirements with the term “real-time”.

Additionally to the real-time requirements, the severity of a failure to comply must be specified. Kopetz provides three categories for this [17]:

- Hard = missing a deadline is a total system failure.
- Firm = infrequent deadline misses are tolerable but may degrade the system’s quality of service. The usefulness of a result is zero after its deadline.

- Soft = the usefulness of a result degrades after its deadline, thereby degrading the system's quality of service.

The term "deadline" refers to the latest time the result is required [17].

This thesis defines a ship-rotorcraft dynamic interface model as usable for pilot training if it is at least firm real-time capable. Furthermore, the flight simulator has a specified maximal reaction time to pilot inputs to be usable. The Federal Aviation Administration (FAA) requires a Level-D simulator to have a maximal reaction time to the pilot's inputs of $100ms$ [18]. This thesis adopts this exact requirement.

In summary, a ship-rotorcraft dynamic interface model needs to fulfill the following computational requirements:

- Run on contemporary hardware
- Firm real-time
- Reaction time $< 100ms$

A failure to comply with the computational requirements prevents the usage of the ship-rotorcraft dynamic interface model in piloted flight simulators. Thus, the computational requirements are imperative, and evaluation is binary, true or false. In contrast, a failure to correctly capture one or more of the required physical phenomena reduces the model's generality, but the use in piloted simulations is still technically possible. Furthermore, the simulation's fidelity regarding physical phenomena requires evaluation on a scale. Therefore, this thesis uses the ability to capture the physical phenomena as a rating scheme to assess usefulness. Nevertheless, this rating only applies to ship-rotorcraft dynamic interface models that fulfill the computational requirements for piloted simulations.

3 Related Work

This chapter summarizes relevant work on ship-rotorcraft dynamic interface modeling in order of complexity. This order correlates with the historic order, as the shortcomings of earlier models motivated the subsequent increase in complexity of later models. Furthermore, the models are evaluated with respect to the requirements given in chapter 2. The chapter starts with a brief description of the rotorcraft flight dynamics model and ship models. These are the starting points for the development of ship-rotorcraft dynamic interface models. This chapter is outside the V-model on which this thesis bases its structure. Nevertheless, it connects to the V-model's requirement specification.

Rotorcraft flight and ship air wake simulation

All scientific investigations into the interaction of ship and rotorcraft start with suitable models for the corresponding ship and rotorcraft. A successful way of modeling the mechanics of a rotorcraft is the Blade Element Method (BEM) [19, 20]. It combines the rigid body dynamics of the fuselage in flight with aerodynamic forces from polars for fuselage and stabilizers. Rigid or flexible beams represent the rotor blades connected to the hub [19, 20]. The individual blades are segmented span-wise into blade elements to calculate the aerodynamic forces and moments. With a known angle of attack, the airfoil polars of the blade element give the resulting forces and moments. The sum of the forces and moments of the elements acts on the blade and defines its movement [19, 20]. With the blade forces and moments known, the movement of the fuselage results from Newton's second law.

Nevertheless, calculating the angle of attack of the blade elements requires knowledge of the flow velocity through the rotor, which depend on the rotor forces. Models that calculate these flow velocities through the rotor from the forces are referred to as inflow models [19, 20]. With low requirements on computational power, algebraic inflow models like the Pitt-Peters model became successful early [21]. Based on Krinner's flow field of a disk with arbitrary pressure distribution in a frictionless fluid[22], Pitt and Peters [7] formulated a simplified expression for the inflow resulting from pressure distributions typical to helicopter rotors [21]. Many extensions based on this model emerged since, e.g. [23, 24, 25, 21]. These algebraic inflow models proved adequate for piloted simulation [6], primarily due to the excellent prediction of the rotorcraft behavior in slow forward flight [26, 6]. Furthermore, the combination of BEM and algebraic inflow models are frequently used for rotorcraft analysis and simulation [21]. Figure 3.1 depicts the interactions captured by a BEM in combination with an inflow model.

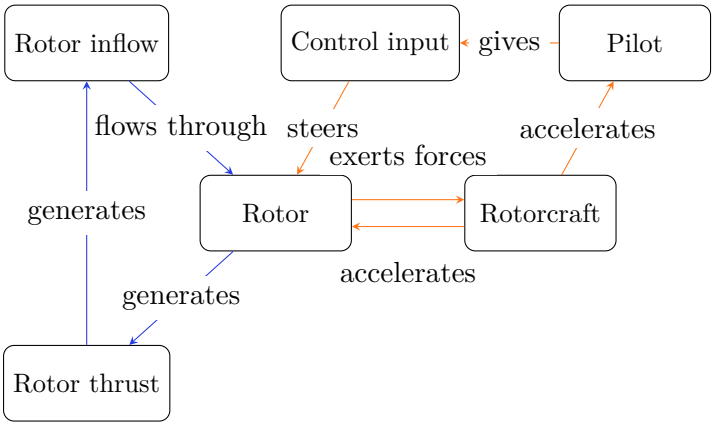


Figure 3.1: Part of the interactions in Fig. 1.1 that is captured by BEM based rotorcraft simulation models.

Naturally, with piloted flight simulation available and helicopters in naval operations, ship-rotorcraft interaction moved into the focus. In order to investigate the interaction, suitable ship geometries need to be defined. The Simple Frigate Shape (SFS) and SFS2 models reduce the ship to rectangular blocks and prisms. This reduction in geometric complexity allows easy comparison between different approaches to compute the air wake of the ship. Furthermore, it allows easy manufacturing of models for wind tunnel investigations. Both versions of the SFS were investigated numerically e.g. [9, 27, 28, 29, 30] and experimentally e.g. [31, 11, 32, 33]. Despite the SFS's success, newer ships use different designs with wider hulls [9]. Therefore, an updated ship geometry for scientific investigations, the NatoGD, was created [9, 12]. Besides defining the ship's geometry and motion profiles in different sea states, wind tunnel measurements are included [12]. First numerical investigations of the ship wake of the NatoGD have already been published [9, 34]. Figure 3.2 depicts the information in Fig. 1.1 that depends on the ship definition.

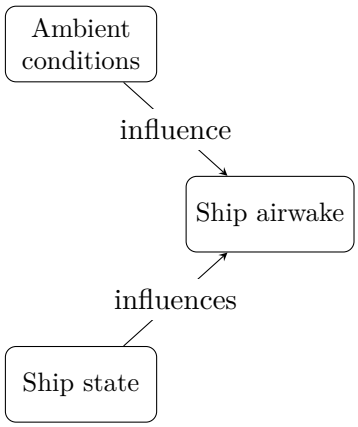


Figure 3.2: Part of the interactions in Fig. 1.1 that is captured by ship simulation models.

One- versus two-way ship-rotorcraft interaction

In general, ship-rotorcraft dynamic interface models represent one- or two-way coupled models. They differ in the directions of the interactions they represent. This transfers to how the rotorcraft simulation in Fig. 3.1 is connected to the ship simulation in Fig. 3.2. One-way coupled models simulate the rotorcraft as a standalone rotorcraft simulation with an inflow model but add disturbances to the inflow, which originate from a ship wake [8, 35, 36, 37, 38, 13, 39]. Figure 3.3 depicts the interaction in these models. Notably, there is no interaction between the rotor thrust and the ship wake.

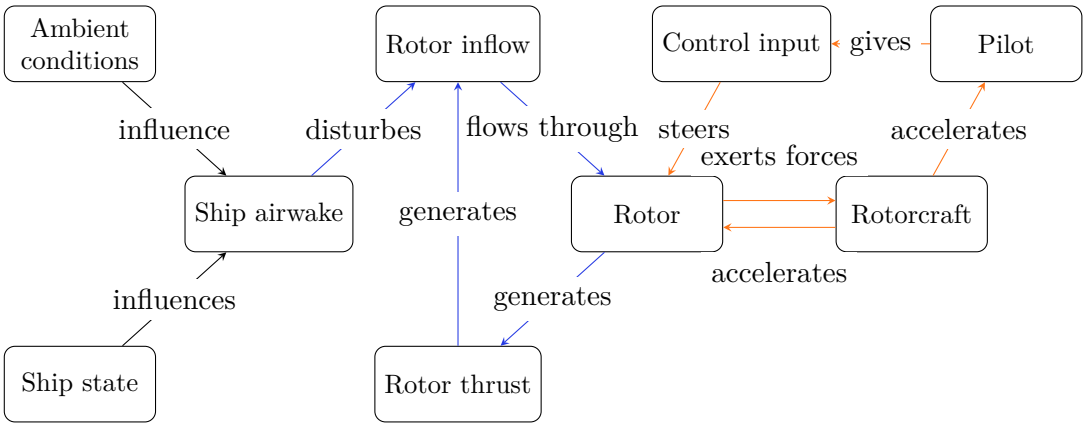


Figure 3.3: Part of the interactions in Fig. 1.1 that is captured by one-way coupled ship wake with BEM based rotorcraft simulation models.

One-way coupling models provide an essential advantage because the real-time capability defined in section 2.3 is imperative for piloted flight simulations. As the ship wake does not depend on the rotor thrust, but merely on the ambient conditions and the ship geometry, it can be pre-computed [13, 38]. During the ship wake simulation, velocities and density are tabulated. Subsequent simulations use a rotorcraft flight dynamics model that extracts disturbances to the rotor inflow from the previously tabulated ship wake data [9, 38, 13, 40, 39]. Additionally, this allows the model to adapt the solver settings to the desired fidelity for the ship wake. Unsteady Reynolds Averaged Navier-Stokes (URANS) was successfully used to calculate the ship wake for interaction with rotorcraft simulations using SFS type geometries [38, 41, 13]. Furthermore, Detached Eddy Simulation (DES) proved to be a successful alternative to investigate SFS type ship rotorcraft interaction [40, 42, 39].

With efficient data retrieval from the tabulated ship wake, one-way coupled models fulfill the real-time requirements defined in section 2.3. Furthermore, the aerodynamic wake of the ship, dynamic behavior of the inflow, and control in stationary flight are captured adequately [13]. Nevertheless, they omit vital interaction and non-linearity of the fluid flow and thus fail to capture some physical phenomena listed as requirements for a ship-rotorcraft dynamic interface model in section 2.1 [38, 34]. Specifically, a moving ship's ground and wall effects are neglected [34]. Furthermore, as these two

combined effects can cause recirculation of the rotor wake, this phenomenon is not captured. However, these phenomena gain importance with decreasing distance to the landing spot. Therefore, the last phase of landing a rotorcraft on a ship deck is not captured adequately by one-way coupled models.

In contrast, two-way coupled models can capture all physical phenomena listed in the requirements in section 2.1. They achieve this by capturing the entire interaction of ship wake and rotor as depicted in Fig. 1.1. Rotor and ship reside in connected or overlapping domains. Unfortunately, the required real-time capability is difficult to achieve with this approach and URANS or DES. These methods require much computation to advance the flow field for a given timestep. Nevertheless, they allow imposing difficult boundary conditions and complex geometries which enable the simulation of all phenomena specified in section 2.1. Furthermore, URANS and DES are extensively validated for many flow phenomena. Unfortunately, when using coarse resolutions to reduce the computational load, significant errors in the accuracy are introduced [43]. Furthermore, most solvers that use these methods do not return a result within a given time. For example, when using an iterative solver, the computation time might vary between individual time steps in the same simulation.

Methods like Free Vortex Wake Method (FVWM) are tailored to rotor wakes, thus allowing accurate results with coarse resolutions [20], even in real-time [44]. Techniques like mirroring create flat, solid walls or ground. Nevertheless, imposing boundary conditions to represent complex geometries like a ship superstructure is difficult. The Viscous Particle Method (VPM) is another method for rotor wake simulation[45, 46]. Like the FVWM, it discretizes the vorticity shed by the rotor blades and can achieve real-time capability [47]. Similarly to FVWM, representing complex geometries as boundary conditions proves difficult. Therefore, these models struggle to predict the required interaction with the ship geometry.

The Lattice-Boltzmann Method (LBM) discretizes the domain into volumes like URANS and DES, thus providing comparable generality concerning complex geometries. Additionally, it is an intrinsically parallel method to solve the Navier-Stokes equations [48, 49, 50]. Therefore, LBM is uniquely suited for calculations on one or multiple General Purpose Graphics Processing Unit (GPGPU) and shows near-ideal scaling starting at small domain sizes. Furthermore, the computation time per timestep is constant due to the use of an explicit time stepping scheme [48, 49]. This makes the LBM suitable for fluid simulations that focus on fast predictions of flow fields. Nevertheless, the efficiency reduces the fidelity of the solution.

This work uses the LBM to calculate the ship wake of the NatoGD and uses the results to test the ship-rotorcraft dynamic interface model presented in this thesis. Ashok et al. [34] also simulated the NatoGD using LBM, but with a different set of boundary conditions and local grid refinement. The SFS type ships were simulated using LBM and compared to experimental data in previous studies by the author [3] and others [30, 28]. These studies indicate that the LBM is capable of simulating ship wakes.

When applying LBM to rotor wake calculations in combination with BEM based rotorcraft simulation, it proved capable of predicting the rotor inflow [1, 2]. Further-

more, depending on the available GPGPU, it achieved real-time capability [51, 52, 1]. Nevertheless, in previous publications of the author, the real-time computable resolution did not adequately predict control inputs in stationary forward flight [2]. Based on these results, Horvat investigated the reaction of a Messerschmitt-Bölkow-Blohm Bo 105 (MBB Bo 105) to the wakes of wind turbines in a wind park in non-real-time [53]. This investigation helped inspiring the strategy for the ship-rotorcraft dynamic interface model described in this thesis.

The recent simulations done by [54] are a promising alternative ship-rotorcraft dynamic interface model to the one presented in this thesis. Their approach uses immersed boundaries to simulate the rotorcraft and ship in a single domain in real-time. The domain is located around the ship with the rotorcraft above the landing deck. Nevertheless, with the comparatively small domain located around the ship, an approaching rotorcraft might need to cross from the outside into the domain which remains to be modeled. But the novel approach can fulfill all the requirements given in chapter 2. The used LBM solver is based on the solver presented in this thesis. Due to the study being published quite recently no detailed analysis is made in this thesis.

4 Local Recalculation Model for Combining Ship and Rotorcraft Wake

This thesis investigates a novel ship-rotorcraft dynamic interaction model named Local Recalculation Model (LRM). It aims to fulfill all the requirements listed in chapter 2. Therefore, it uses the computational efficiency of LBM on GPGPUs to compute the two-way interaction of ship and rotorcraft. Furthermore, it reduces the computational requirement by only calculating the interaction near the rotor. The following chapter describes the essential concept and the reasoning behind the LRM on an abstract level. Thus, this chapter corresponds to the "Architecture" part of the "Requirements and Architecture" step of the V-model.

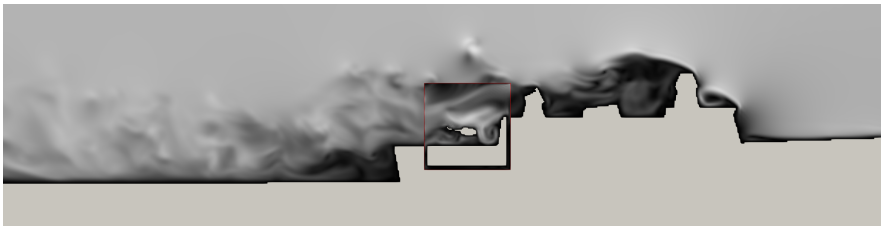
- *Problem*: Chapter 3 highlights that the requirements listed in chapter 2 are demanding. Accounting for the relevant physical phenomena requires two-way coupled models. Nevertheless, the added complexity results in high computational costs that interfere with the required real-time capability.

- *Idea*: To resolve this, the LRM only computes the interactions of ship and rotorcraft near the rotor inside a small domain. In fluid dynamics, the interaction of objects originates from the interaction of their wakes. This influence of the wake on the object reduces with distance [55]. Therefore, a sufficiently large domain around the objects captures the interaction. However, interaction with parts or entire objects outside the domain is neglected. Nevertheless, the wake of objects outside the domain can be simulated separately and transported by convection into the domain. This pre-computation and convection into the domain captures the interaction of the object's wake with the fluid inside the domain but is limited to the domain's size. As the LRM aims to predict the inflow into the rotors, focusing on the interaction close to the rotor is reasonable. Furthermore, the rotor forces create the rotorcraft's reaction to the pilot's control inputs. To illustrate the concept of the LRM, Fig. 4.1 depicts the ship wake, and the combination of ship wake and LRM domain. The figure represents a look ahead to chapter 9. The figure shows the center plane of a ship wake simulation without rotorcraft in 4.1a. This pre-computed wake is transported into the red domain representing the LRM domain in 4.1b. Inside the LRM domain, a fluid solver combines the ship's wakes and rotor, which is called local recalculation in this thesis. Furthermore, wall boundary conditions represent the ship geometry.

  Aerodynamically the geometries of SFS type ships and the NatoGD represent bluff bodies. This results in well-defined points for flow separations and shear

layers robust to disturbances downstream. Thus, computing the aerodynamic interaction of the rotor and ship inside a small domain around the rotor should capture most of the ship-rotorcraft interaction.

(a) Ship wake simulation



(b) Ship wake and LRM simulation

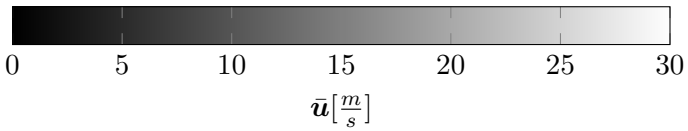


Figure 4.1: Mean velocity $\bar{\boldsymbol{u}}$ in center plane of a ship wake simulation and a LRM simulation using the data of the ship wake to generate boundary conditions.Grey solids represent the voxelized hulls and ground. Boundary of the LRM domainin red.

- *Ansatz*: To allow the domain of the LRM that contains the rotor to follow the rotorcraft during the piloted simulation, the discretization must allow free movement in space. The corresponding boundaries must transport information about the upstream ship wake into the domain. Inside the domain, wall boundary conditions extracted from the ship geometry allow to capture the influence of the ship geometry on the fluid flow. Figure 4.2 depicts the interactions of the LRM for piloted simulations. It highlights that the LRM represents all interactions present during rotorcraft ship deck operations.

- *Result*: The LRM decouples the ship rotorcraft interactions into three parts. Part one is the information far upstream of the rotor. At this distance, the model assumes no influence on the rotor; thus, the fluid state can be pre-computed. The second part is the resolved non-linear interaction inside the LRM domain that contains the rotor. The size of this domain needs to be investigated and selected to fulfill all computational requirements of 2.3. With the size selected,

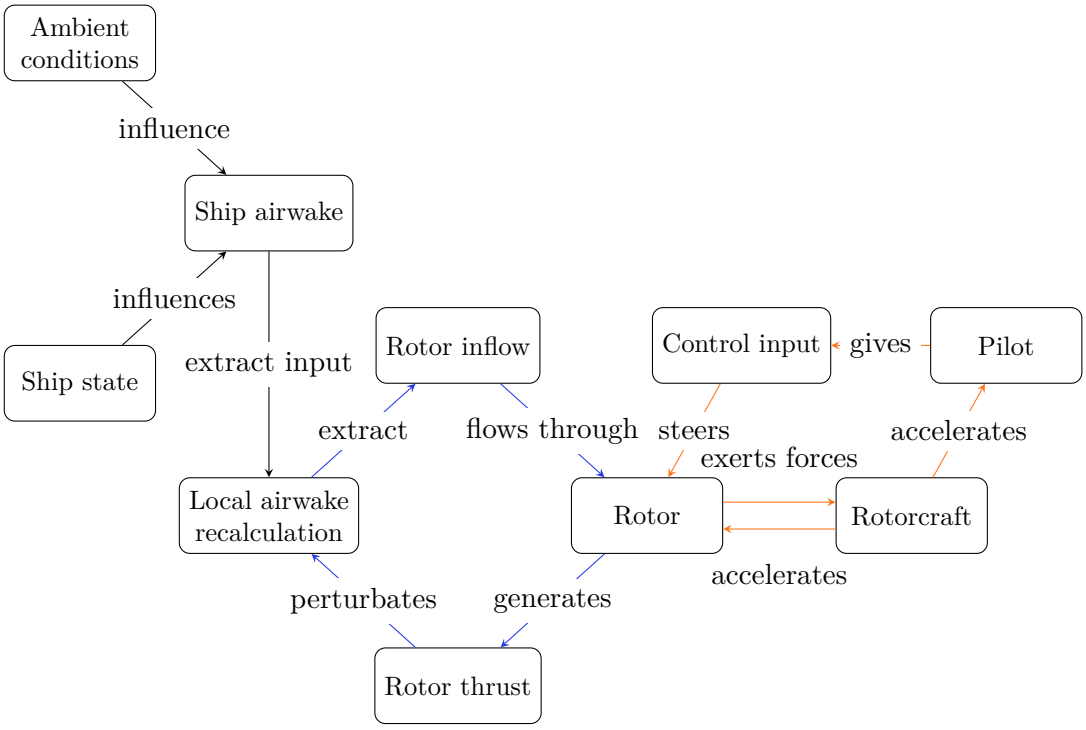


Figure 4.2: Sketch of the physical interactions in rotorcraft ship deck operations using local recalculation of the ship airwake inside the LRM domain. The (——) arrows mark the rotorcraft-ship dynamic interface, (——) arrows mark the general control loop of rotorcrafts.

the simulation must be verified to capture all critical physical phenomena given in 2.1. The third part consists of the information downstream of the domain. As the model assumes the combined ship-rotorcraft wake does not influence the rotor at this distance, the fluid state is discarded when leaving the LRM domain.

The strategy of the LRM allows to construct a ship-rotorcraft dynamic interface model that captured the two-way interaction of ship and rotorcraft within a small domain. Furthermore, prescribing the fluid state extracted from a pre-computed ship wake at the boundaries of the LRM domain allows to capture the influence of the part of the ship geometry that is outside of the LRM domain. Combining this strategy with the scaling of the LBM on GPGPUs results in a computationally efficient model that is real-time capable. Therefore, the LRM is a novel approach to the ship-rotorcraft dynamic interface problem. An intermediate state of the LRM was already published during development [3].

5 Decomposition into Physical Domains and Interface Definition

To implement the LRM, the interaction shown in Fig. 4.2 in the previous chapter is split into domains and interfaces. This thesis defines the domains corresponding to the physics that are active within, e.g., fluid mechanics or solid body mechanics. This chapter describes the tasks of the domains and interfaces used in the LRM. Therefore, it corresponds to the V-model's "Detailed Design" stage. While individual sections in the following describe the domains, the interface specifications get appended to the section of one of the domains they interface. Figure 5.1 depicts these domains and interfaces as colors.

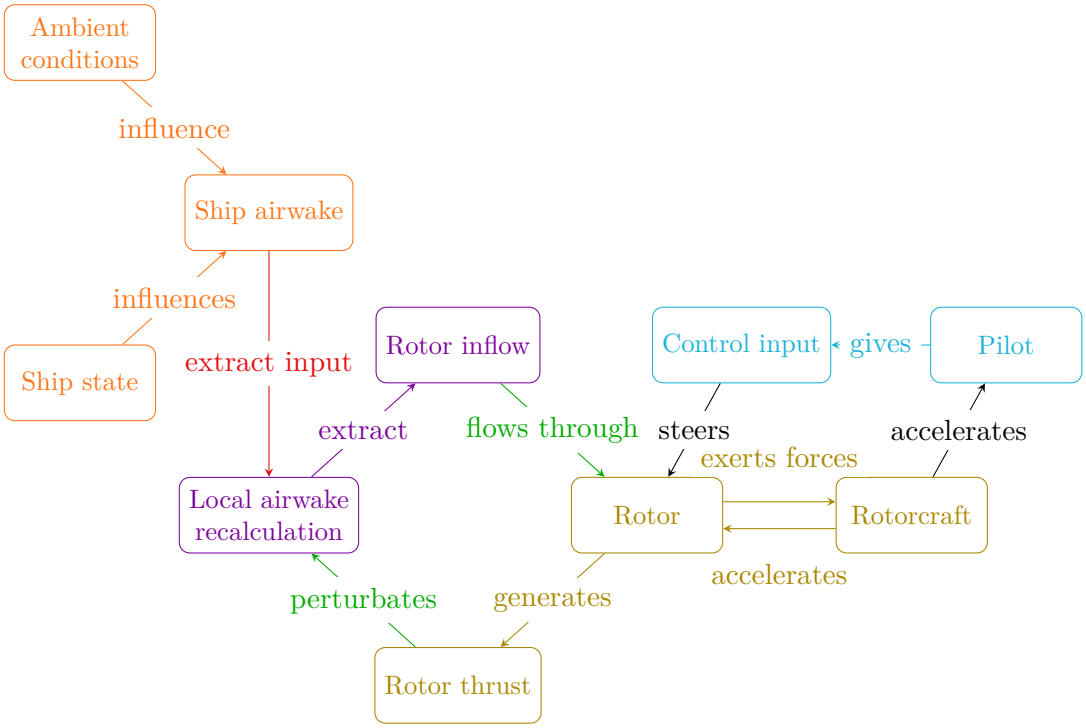


Figure 5.1: Sketch of the physical interactions in rotorcraft shipdeck operations using the LRM. Ship wake calculation shown in (—) which is connected to the LRM domain in (—) via the domain boundary interface (—). The rotor interface (—) connects the LRM with the rotorcraft flight dynamics (—) which is connected (—) to the pilot (—).

5.1 Ship movement and air wake simulation

Chapter 4 motivates that the ship wake can be precomputed without the interaction that the rotorcraft produces, for the LRM. This results from the LRM using the ship wake information to generate boundaries for the LRM domain (—) but recomputing the combined ship-rotorcraft interaction within the domain. Nevertheless, the LRM relies on ship wake simulation fulfilling the requirements in section 2.2 being available.

This section describes the simulation of the ship wake, (—) in Fig. 5.1, on an abstract level. Fortunately, the absence of interaction with the rotorcraft in this simulation allows tailoring domain and simulation parameters to the physics of the ship wake. Nevertheless, the domain size of the simulation needs to be large enough to prevent the boundary conditions from significantly influencing the ship wake. Therefore, the used domain has to span multiple lengths of the ship in the main wind direction and sufficient distance of the boundaries to the ship in the remaining directions. Additionally, the resolution of the domain needs to be fine enough for the used fluid solver to predict the physical phenomena specified in section 2.2. Furthermore, as ship motion plays an important role for the ship-rotorcraft dynamic interface, the used discretization has to support moving the ship geometry relative to the stationary ambient wind. As the ambient wind creates and transports the ship wake, the direction in which the bulk of the wake is developing is known. Therefore, the size of the subsection of the ship domain that is stored for the LRM to use can be defined to contain only a subset of the domain used to simulate the ship wake. This allows to save disk space and allows faster extraction of boundary conditions for the LRM domain. This thesis uses the NatoGD ship geometry as it is a better representation of contemporary naval vessels than the SFS types [9]. Furthermore, motion profiles and wind tunnel data are available [12]. Fig. 5.2 presents a back, side, and top view of the NatoGD.

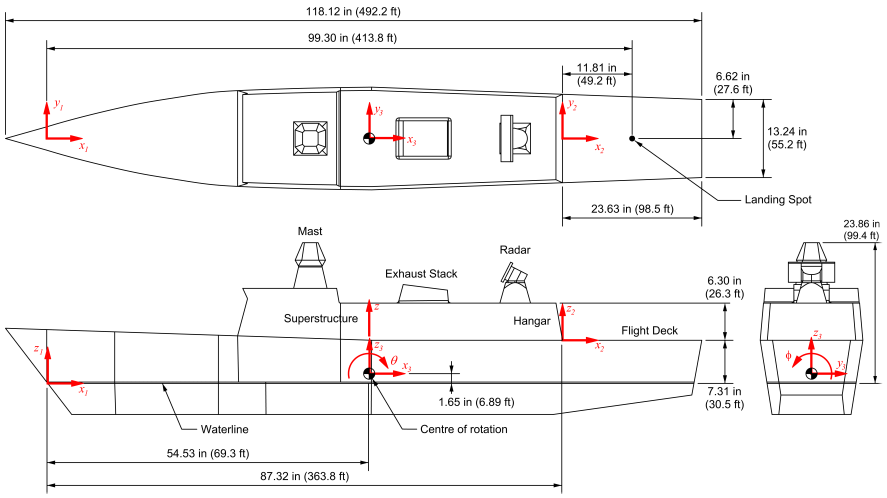


Figure 5.2: Schematic of the NATO Generic Destroyer. Dimensions in inches for a 1:50-scale model and feet for the full-scale ship (brackets). Taken from [12].

5.2 Local Recalculation Model domain as dynamic inflow model

The LRM domain, (—) in Fig. 5.1, forms the central part of the ship-rotorcraft dynamic interface model. It places the rotor in the center of the domain, as the flow in the vicinity of the rotor matters most for the ship rotorcraft interaction [3]. The fluid solver used for simulating the domain must support the free motion of the domain as the domain moves through space with the rotor. A cubical domain allows the definition of straight boundary conditions along the faces. Furthermore, it features almost equal distances between the rotor and boundaries. Nevertheless, there exists no restriction on the shape as long as the used solver can discretize it. Figure 5.3 shows an exemplary cubic domain used in this thesis, including a centered rotor and a MBB Bo 105 fuselage.

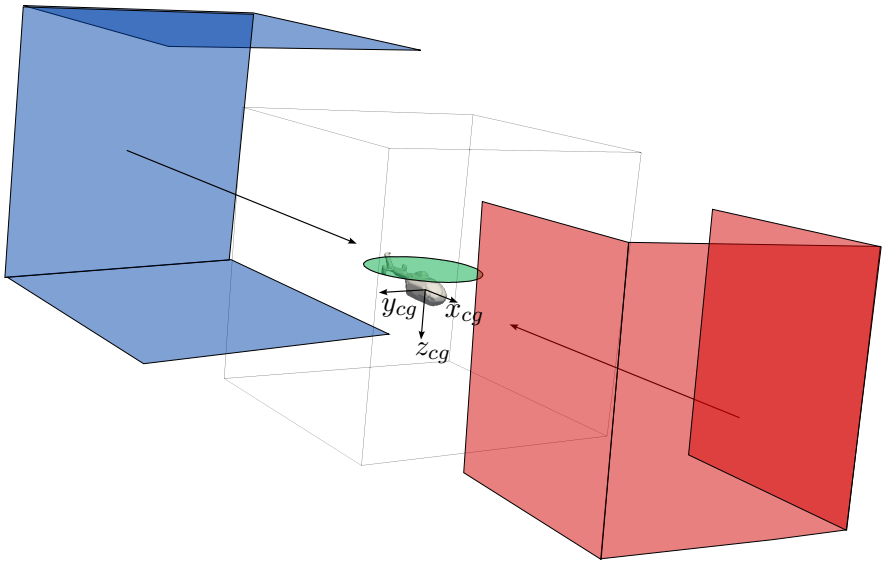


Figure 5.3: Schematic view of the LRM domain. Blue faces (—) represent impedance boundary conditions, red faces velocity boundary conditions (—) that represent the interface to the feed in information from the precomputed ship wake. The green rotor disk (—) represents the interface between the LRM and the rotorcraft flight dynamics simulation. Solids represent the fuselage of a MBB Bo 105. The coordinate system in the center of gravity is displayed. Adapted from [3].

The (—) interface that extracts the input from the precomputed ship wake in Fig. 5.1 uses the boundaries at the red faces of the LRM domain. At these faces, Dirichlet boundary conditions prescribe the fluid state of the ship calculation at the respective point in time and space [53, 3]. From the pilot's point of view, these faces are at the front and sides of the LRM domain. These directions represent the possible directions information passes into the domain, given that the rotorcraft approaches a ship headwind $\pm 15 deg$. The selection also allows the simulation of approaches with slight relative wind direction, as shown by Owen [9].

The blue faces of the LRM domain represent faces through which momentum must be allowed to enter or exit the domain freely [53, 3]. As the rotor adds momentum to the fluid inside the domain, free entry and exit of fluid into the domain is required. Otherwise, the wake of the rotor and ship would be reflected in the boundaries and interfere with the simulation at later steps. Because the rotorcraft has one dominant flight direction and rotors add momentum to the fluid predominantly along the shaft axis, the cubic domain's back, top, and bottom face should allow fluid entry and exit.

Wall boundary conditions represent the ship geometry inside the LRM domain [53, 3]. The wall boundaries allow the capture of the ground, wall effects, and recirculation that occur when flying the rotorcraft close to the ship superstructure like the hangar doors [34, 3]. Nevertheless, stability requires to not impose wall boundary conditions in the cells that form the two outermost cell layers of the cube [3]. Thus even if these cells lie within the ship geometry, no wall boundary is imposed.

With the interface to the precomputed ship wake, free outflow of fluid, and the ship geometry, the LRM domain can predict the ship rotorcraft interaction happening inside it. The LRM domain interfaces with the rotorcraft flight dynamics simulation, (—) in Fig. 5.1, by exchanging the rotor thrust and inflow. The (—) interactions in Fig. 5.1 represent this exchange. This abstract interaction happens at the rotor disk, represented by the green disk in Fig. 5.3. This thesis limits the interaction to the main rotor for simplicity. Nevertheless, the LRM does not restrict the number of rotors [3]. Theoretically, it captures the interaction between all rotors inside the domain.

For the rotorcraft flight dynamics model, this exchange of thrust and inflow represents a function that returns the inflow for a given thrust. As the LRM is precisely this type of function, it is an inflow model. Furthermore, if the fluid solver that calculates the flow inside the LRM domain is unsteady, the LRM is a dynamic inflow model.

5.3 Rotorcraft flight dynamics model

Generally, a rotorcraft is a multibody system of connected, flexible bodies. Various levels of detail exist when simulating rotorcraft movement as a reaction to the forces acting on the bodies [19]. Nevertheless, the LRM representing an inflow model requires the rotor discretization to use the inflow information to calculate the rotor forces. This requirement originates from the definition of the (—) interface in Fig. 5.1 with the LRM domain in the last section. For example, surrogate rotorcraft models do not feature a physical representation of the rotor internally, which prevents coupling with the LRM. Nevertheless, surrogate models of the complete interaction of ship and rotorcraft circumvent this restriction but represent stand alone rotor-ship dynamic interface models [56].

The rotorcraft flight dynamics models, (—) in Fig. 5.1, can calculate the rotorcraft movement as a reaction to the inflow, rotor state, and control inputs [20, 10, 19]. Nevertheless, the rotorcraft flight dynamics model must be validated for each rotorcraft it models. Furthermore, the model should require few computational resources and allow real-time simulation. Otherwise the computational requirements of section 2.3

can not be fulfilled. The rotorcraft flight dynamics model used in this thesis fulfills these requirements [57].

In order to represent the position of the rotorcraft's center of gravity $\boldsymbol{x}^{cg}$ and attitudes Φ, Θ, Ψ, this thesis uses the coordinate systems shown in Fig. 5.4. The revolution angle of the blade ψ, lead-lag ζ, and flap β angles describe the position of individual segments of blades, called blade elements. The local pitching angle of the blade element ϕ is defined relative to a virtual plane with the rotor shaft axis as normal vector. Furthermore, the position of the blade element depends on the shaft tilt angle.

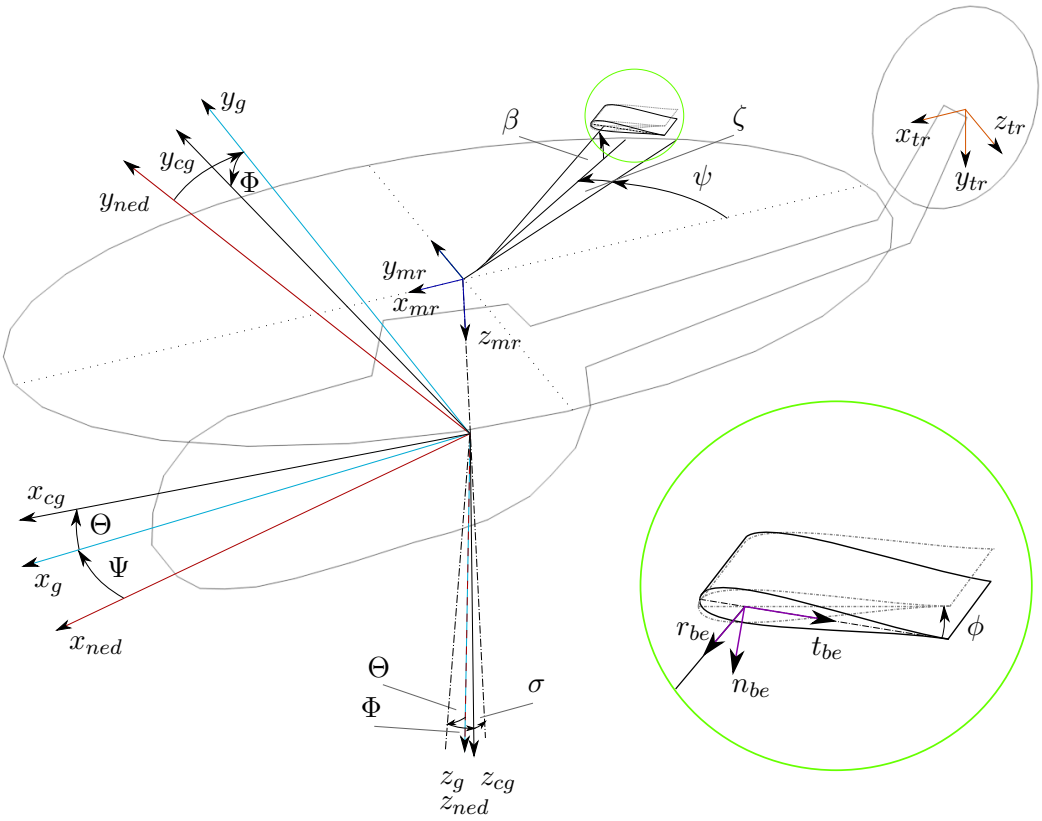


Figure 5.4: Schematic view of the rotorcraft with used coordinate systems. NED system in (—), geodetic system (x_g, y_g, z_g) in (—), body-fixed system (x_{cg}, y_{cg}, z_{cg}) with origin in the center of gravity in (—), body-fixed main rotor system (x_{mr}, y_{mr}, z_{mr}) in (—), body-fixed tail rotor system (x_{tr}, y_{tr}, z_{tr}) in (—), and local blade element system (n_{be}, t_{be}, r_{be}) in (—). Attitudes Φ, Θ, and Ψ, lead-lag ζ and flap β angles, rotorshaft-angle σ, local blade element pitching angle ϕ, and revolution angle of blade ψ. Adapted from [58].

5.4 Flight simulator to interface with human pilots

The pilot and the rotorcraft flight dynamics model must interact to allow a human pilot, (——) in Fig. 5.1, to steer the rotorcraft. For a human pilot, a flight simulator represents the interface shown in (——) in Fig. 5.1. The flight simulator generates a visual representation of the surroundings of the rotorcraft for the pilot to see, thus creating a sense of acceleration. Furthermore, the flight simulator digitalizes the pilot's control inputs and forwards them to the rotorcraft flight dynamics model. Nevertheless, this thesis does not include studies with human pilots in a flight simulator. For comparability, the approaches flown are predefined trajectories or tabulated control inputs.

5.5 Base algorithm for simulating one timestep

All the interactions in Fig. 5.1 happen simultaneously and continuously during an approach of rotorcraft on a ship deck. Implicit time-stepping schemes excel at solving equations of multiple domains and interfaces simultaneously. Nevertheless, the increased computational load of implicit schemes compared to explicit time stepping favors the latter in achieving the requirement of firm real-time for the LRM. This thesis uses explicit times stepping schemes to achieve real-time capability on the available hardware. Therefore, the solvers that solve the fluid flow in the ship domain, the LRM domain, and the rotorcraft flight dynamics model use an explicit scheme to integrate in time. Figure 5.5 sketches the main algorithm. It depicts the order of the interface exchange and the advancing of domains for a time step.

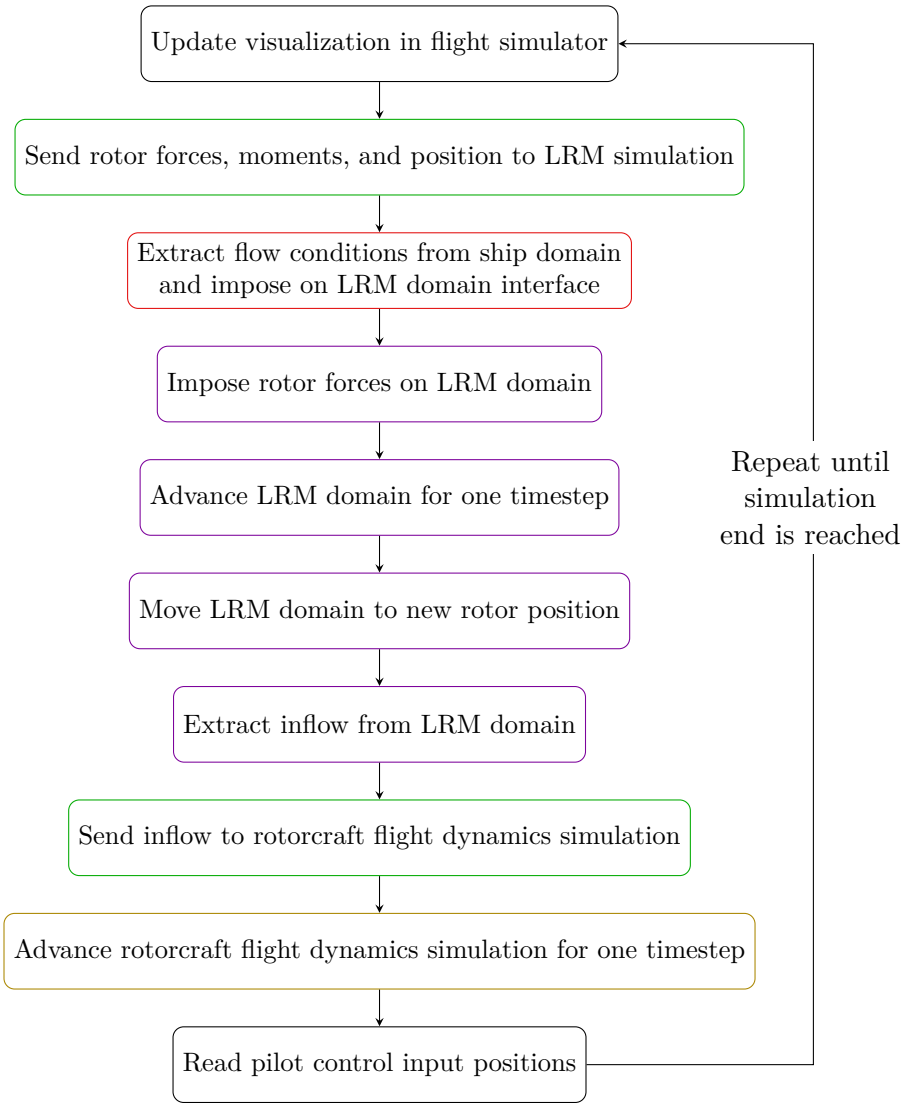


Figure 5.5: Schematic of the algorithmic steps of a LRM simulation. Color of the steps indicates the domain or interface of the LRM the step is a part of.

6 Modeling of Physical Domains and Interface Implementation

This chapter describes the modeling of the domains and implementation of interfaces that chapter 5 defines. Therefore, it corresponds to the V-model's "Implementation" stage. The chapter describes the physical modeling but refrains from showing the source code. Although multiple combinations of solvers, domain discretizations, or interface implementations are available, this thesis uses the LBM for the fluid simulation of the ship and LRM domain. This method to solve the Navier-Stokes equations excels at using GPGPUs to calculate fluid flows with coarse resolutions. Section 6.1 gives details of the used LBM solver. This solver is used for the the ship wake, described in section 6.2, and the simulation of the LRM domain in section 6.4. Section 6.3 details the interpolation of tabulated fluid states onto the interface of the LRM domain. The interface described in section 6.5 couples the LRM domain with the rotorcraft flight physics simulation described in 6.6.

6.1 Fluid simulation using the Lattice-Boltzmann Method

The Lattice-Boltzmann Method is an explicit time-stepping method for solving hyperbolic conservation laws [59, 60]. Therefore, it also solves the Navier-Stokes equations for incompressible flow [61, 49, 48]. The solver used in this thesis is openLB [62] with modifications to allow moving grids and computations on GPGPUs. Although the scheme models the fluid as weakly compressible [63], it can be extended for solving compressible flows [64, 65, 66]. In general, the LBM uses particles that at time t are located at spatial position $\boldsymbol{x}$ and move with the molecular velocity $\boldsymbol{\xi}$, which is the superposition of the macroscopic velocity $\boldsymbol{u}$ with the speed of sound $\boldsymbol{c}$. Due to their movement, the particles can collide and exchange information. Although the Boltzmann equation does not restrict the type of particles or the information carried, the particles can be imagined as fluid particles in this thesis to ease understanding. The LBM solver used in this thesis is an extension of OpenLB [67] for GPGPUs.

6.1.1 The Boltzmann equation

A fluid particle receives its properties by mixing the properties of the molecules that form the fluid. In any fluid with a temperature $T > 0K$, the particles travel a certain distance at their velocity until they collide with other particles. However, as approximately 10^{19} particles form $1cm^3$ of gas in normal conditions [48], tracking each particle exceeds computational resources quickly. Furthermore, technical simulations,

especially this thesis, target fluid flow simulation around structures with lengths of $L^{NatoGD} \approx 150m$. The Knudsen-number Kn describes this relation between the mean distance traveled between collisions and the typical physical length of the simulated phenomenon.

At low $Kn < 0.01$ the fluid formed by the particles acts as a continuum [48, 49]. Therefore, the behavior of the fluid does not require tracking individual fluid particles. Thus, a function describing the distribution of the particles over the spatial space $\boldsymbol{x}$ and molecular velocity space $\boldsymbol{\xi}$ suffices to describe the fluid state [49, 48]. With the transition from individual particles to a particle distribution function, the description type transitions from discrete to statistical. With this distribution function depending on space and velocity while evolving in time, it depends on seven parameters:

$$f(\boldsymbol{x}, \boldsymbol{\xi}, t)$$

The Boltzmann equation formulates the evolution of f in terms of a transport and collision term [48]:

$$\frac{Df}{Dt}|Transport = \frac{Df}{Dt}|Collision \tag{6.1}$$

Thus, the Boltzmann equation states that all transport-related terms of the total derivative of f with respect to time form an equilibrium with all collision-related terms of the total derivative of f with respect to time. Without going into detail, the transportation term, including forces acting on the fluid, is derived to [48]:

$$\frac{Df}{Dt}|Transport = \frac{\partial f}{\partial t} + \xi_i \frac{\partial f}{\partial x_i} + \frac{F_i}{m}\frac{\partial f}{\partial \xi_i}$$

This formula incorporates a volume force $\boldsymbol{F}$ acting on the fluid, which consists of a total particle mass m. Furthermore, the formula uses tensor notation and Einstein's summation convention. Broken down, the transport terms of equation 6.1 consist of $\frac{\partial f}{\partial t}$, the convection term $\boldsymbol{\xi}\frac{\partial f}{\partial x_i}$, and the forcing term $\frac{F_i}{m}\frac{\partial f}{\partial \xi_i}$. It describes the change in the distribution function f due to the transport of particles.

The right side of equation 6.1 describes the change in f due to collisions of the particles. In the infinitesimal control volume ΔV at spatial position $\boldsymbol{x}$, particles with the velocity $\boldsymbol{\xi}$ collide with particles with the velocity $\boldsymbol{\xi}^\star$. Nevertheless, not all particles that are inside ΔV collide. For particles to collide, the path of the particles has to cross in the proper condition. The differential particle cross-section dA_C encapsulates this condition [48].

As the particle distribution f describes the state of all particles in the fluid, it interacts with itself during the collision. With the notion $f^\star = f(\boldsymbol{x}, \boldsymbol{\xi}^\star, t)$, the total collision terms of equation 6.1 can be formed by integrating over $\boldsymbol{\xi}^\star$ and A_C as [48]:

$$\frac{Df}{Dt}|Collision = \int_{\boldsymbol{\xi}^\star}\int_{A_C} (f'f'^\star - ff^\star)(\boldsymbol{\xi}^\star - \boldsymbol{\xi})dA_C d\boldsymbol{\xi}^\star$$

Which uses $'$ to refer to the variable post-collision. Using this equation 6.1 is written in its mostly used form [48]:

$$\frac{\partial f}{\partial t} + \xi_i \frac{\partial f}{\partial x_i} + \frac{F_i}{m} \frac{\partial f}{\partial \xi_i} = \int_{\boldsymbol{\xi}^\star} \int_{A_C} (f' f'^\star - f f^\star)(\boldsymbol{\xi}^\star - \boldsymbol{\xi}) dA_C d\boldsymbol{\xi}^\star \tag{6.2}$$

The equilibrium distribution at maximum entropy

The Boltzmann equation is an integro-differential equation as shown by formulation in 6.2. Solving this complex integro-differential equation for domains with respective boundaries is resource-intensive. Nevertheless, for the limits of Kn, the equation is simple enough to be solved analytically. For $Kn \to \infty$, the particles have an infinite mean free path they can travel without colliding; thus, the collision term of equation 6.2 vanishes. In this form, the particles travel along their path only influenced by the volume force applied [48, 49]. In contrast, for $Kn \to 0$, the particles have zero mean free path they can travel before colliding. Thus, the transportation terms of equation 6.2 equal 0 [48, 49].

The limit of $Kn \to 0$ allows identifying the particle distribution for thermal equilibrium. In the thermal equilibrium particles collide but the distribution function f is constant, thus [48]:

$$0 = \int_{\boldsymbol{\xi}^\star} \int_{A_C} (f' f'^\star - f f^\star)(\boldsymbol{\xi}^\star - \boldsymbol{\xi}) dA_C d\boldsymbol{\xi}^\star$$

The trivial solution resulting in the integration to be zero is if the collision does not change $ff^\star$ [48]:

$$f' f'^\star = f f^\star$$

In order to connect this with the equations for mass, momentum, and kinetic energy of all particles that form f, the natural logarithm is taken [48]:

$$lnf' + lnf'^\star = lnf + lnf^\star$$

The following comparison with the conservation equations for particles in elastic collisions traveling at a mean speed of sound $\boldsymbol{c}$ is made [48]:

$$\begin{aligned} m' + m'^\star &= m + m^\star \\ \boldsymbol{c}m' + \boldsymbol{c}m'^\star &= \boldsymbol{c}m + \boldsymbol{c}m^\star \\ 0.5\boldsymbol{c}^2 m' + 0.5\boldsymbol{c}^2 m'^\star &= 0.5\boldsymbol{c}^2 m + 0.5\boldsymbol{c}^2 m^\star \end{aligned}$$

Using the ansatz $lnf = Am + Bm\boldsymbol{c} + 0.5Cm\boldsymbol{c}^2$, the Maxwell equilibrium distribution is derived [48]:

$$f^{eq}(\boldsymbol{x}, \boldsymbol{u}) = \frac{n}{(2\pi RT(\boldsymbol{x}))^{3/2}} e^{-\frac{(\boldsymbol{\xi}-\boldsymbol{u})^2}{2RT(\boldsymbol{x})}} \tag{6.3}$$

It states that the particle distribution for n particles in thermodynamic equilibrium is normal. This normal distribution has the mean value of the velocity $\boldsymbol{u}$ and variance of $\sqrt{RT}$ depending on the Boltzmann constant R and the temperature $T(\boldsymbol{x})$. As the Maxwell equilibrium distribution represents the state of maximal entropy at a given $\boldsymbol{u}$, any particle distribution f evolves into the Maxwell distribution over time. As the collisions of particles cause this evolution, the f of fluid flows relaxes faster at low Kn [48].

6.1.2 Discrete Lattice-Boltzmann equation

Using a computer to solve the identified governing equations for fluid particles requires two additional steps. First, the distribution function f needs to be discretized. Second, a computationally efficient model for the effect of the collisions needs to be defined, as a direct evaluation the integrals in the collision term of equation 6.2 requires substantial computational resources.

Discretizing the particle distribution and collision term

As all f in fluid flows evolve towards f^{eq} and the low Kn in technical flows indicates that this evolution towards the equilibrium is fast compared to the time scales of the fluid flow [48], several simplifications of the collision term of equation 6.2 were proposed. Two widely used ones are the Bhatnagar-Gross-Krook Model (BGK) [68] and Multiple-Relaxation Times (MRT) [69] collision model. This thesis focuses on the BGK collision model because of its computational efficiency. The central idea is to model the collision in equation 6.2 by reducing the difference of f and f^{eq} [68]:

$$\int_{\boldsymbol{\xi}^\star}\int_{A_C}(f'f'^\star - ff^\star)(\boldsymbol{\xi}^\star - \boldsymbol{\xi})dA_C d\boldsymbol{\xi}^\star = \omega(f^{(eq)} - f) \tag{6.4}$$

$$\omega = \frac{c_s^2}{\nu} \tag{6.5}$$

In this formula, the relaxation frequency describes the frequency of particle collisions. Despite its simplicity, the BGK has proven to give good results in many technical areas [48, 49]. Furthermore, the Boltzmann equation with the BGK collision term solves the Navier-Stokes equations for flows with low Kn. The statistical analysis used to derive this connection is called Chapman-Enskog expansion [70, 48, 49]. Nevertheless, the details of this analysis are outside of this thesis's scope.

With the simplified collision operator, the right side of the Boltzmann equation 6.2 is simple to compute. Nevertheless, the particle distribution f still spans the complete fluid domain. In order to efficiently compute the temporal evolution of f it has to be discretized. A common way to do this is to split the domain into spatial cubic elements with discrete molecular velocities $\hat{\boldsymbol{\xi}}$ [48, 49].

In the resulting discrete Boltzmann equation, discrete values $\hat{f}_i$ describe the information of the particle distribution f that travels with the velocity $\hat{\boldsymbol{\xi}}_i$ [48]. The discrete

$\hat{f}_i$ in all cubes form the discretization of f. The literature usually refers to the $\hat{f}_i$ as populations, the cubic elements as cells, and the cartesian grids formed by the cells as lattices. Using the BGK collision model and neglecting the volume force it reads [48]:

$$\frac{\partial \hat{f}_i}{\partial t} + \hat{\xi}_{ij}\frac{\partial \hat{f}_i}{\partial x_j} = \hat{\omega}\Delta t(\hat{f}_i^{(eq)} - \hat{f}_i) \tag{6.6}$$

$$\hat{\omega} = \frac{c_s^2}{\nu + \Delta t c_s^2/2} \tag{6.7}$$

Nevertheless, the discrete velocities $\hat{\boldsymbol{\xi}}_i$ still need to be defined. For ease of use, the $\hat{f}_i$ collide at the center point of a cell by definition [48]. Furthermore, it is advantageous if the $\hat{\boldsymbol{\xi}}_i$ point to the center point of the next cell in a distance of $\Delta\boldsymbol{x}$, while satisfying $\Delta\boldsymbol{x} = \hat{\boldsymbol{\xi}}_i\Delta t$. This results in the population $\hat{f}_i$ traveling from the center of the current cell to the center of the next cell in $\hat{\boldsymbol{\xi}}_i$ direction during the transportation step of equation 6.7. Nevertheless, as the macroscopic velocity $\boldsymbol{u}$ of the fluid is not known beforehand and can vary over time, the $\hat{\boldsymbol{\xi}}_i$ are non-constant. However, this would require interpolation, warping of the cells, or other complicated calculations, which must be avoided.

One way to circumvent this problem is restricting the flow to low Mach numbers $Ma = \boldsymbol{u}/\boldsymbol{c} < 0.1$. At $Ma = 0$, the discrete velocities equal the discrete speeds of sound in that direction $\hat{\boldsymbol{\xi}}_i = \boldsymbol{u} + \hat{\boldsymbol{c}}_i = \hat{\boldsymbol{c}}_i$. Thus, using the $\hat{\boldsymbol{c}}_i$ as speeds for the populations $\hat{f}_i$ to travel at guarantees that the $\hat{f}_i$ move exactly to the next cell position in one time step. Unfortunately, for $Ma \neq 0$ this introduces a Ma-dependent error [48]. Nevertheless, it allows the assembly of a regular grid with populations traveling from one cell to the corresponding neighbor cell in the time step Δt [48, 49].

With cubic cells, the distances between the center points of neighboring cells are longer in the diagonal directions than in the directions aligned with the coordinate system. Nevertheless, all populations travel to the next cell in the same Δt. To achieve this, the value of populations $\hat{f}_i$ associated with the diagonal velocities $\hat{\boldsymbol{c}}_i$ is reduced to account for the increased velocity [48].

To represent the pressure, $\hat{\boldsymbol{c}}_0 = \boldsymbol{0}$ is defined, which represents the particles not traveling to another cell during the transportation step. Nevertheless, multiple sets of $\hat{\boldsymbol{c}}_i$ exist to use for three-dimensional space [48]. These sets, called lattice descriptors, are invariant to rotations and adhere to the naming scheme of DaQb with a representing the rank of the space they discretize and b the number of $\hat{\boldsymbol{c}}_i$ used [48, 49]. The descriptors D3Q15, D3Q19, and D3Q27 represent popular choices for three-dimensional simulations [49]. This thesis uses the D3Q19 as it compromises between accuracy and computational efficiency due to using 19 $\hat{\boldsymbol{c}}_i$ only [49]. Figure 6.1 depicts a cell of the D3Q19 lattice descriptor with the corresponding $\hat{\boldsymbol{c}}_i$.

Nevertheless, equation 6.7 is still continuous in time. However, with the populations moving from cell to cell with the velocities defined in the descriptor, using this Δt as a time step for the scheme is advantageous [48, 49]. The resulting scheme features two steps. In the transportation step, the populations move between the cells. This motion

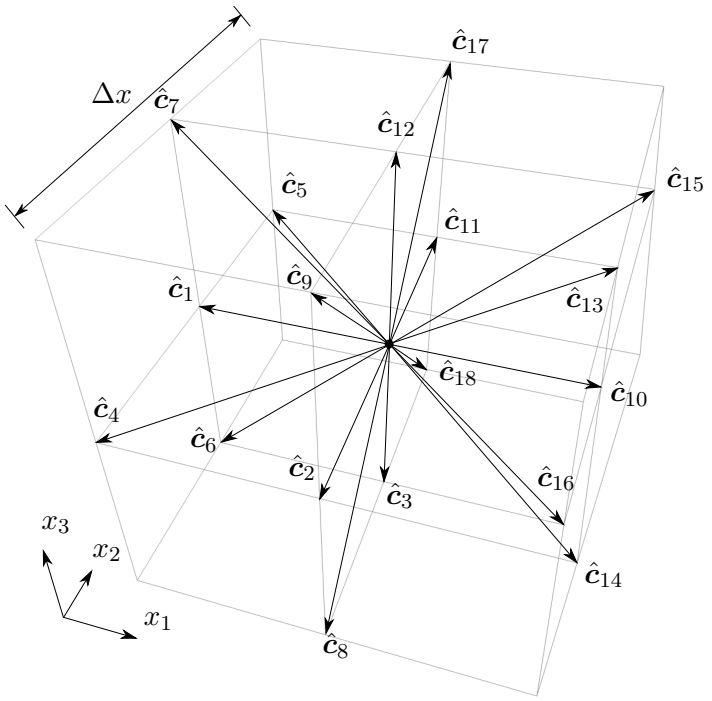


Figure 6.1: Schematic view of the D3Q19 lattice descriptor for a cell with length Δx and the numbered $\hat{\boldsymbol{c}}_i$ as arrows. $\hat{\boldsymbol{c}}_0$ not shown.

is referred to as the streaming of populations [48, 49]. In the following collision step, the particles collide at the center points of the cells [48, 49].

Therefore, the transportation step's left side of equation 6.7 can be simplified. Due to the populations not colliding during the transportation step, they travel unchanged along their characteristic for $\hat{\boldsymbol{c}}_i \Delta t$, equal to the distance to the next cell point. Thus, the solution to the transportation step on the left side of equation 6.7, referred to as a streaming step, equals [48]:

$$\hat{f}_i(\boldsymbol{x} + \hat{\boldsymbol{\xi}}_i \Delta t, t + \Delta t) = \hat{f}_i(\boldsymbol{x}, t)$$

In order to formulate the collision step of equation 6.7, the discrete equilibrium distribution is defined [48]:

$$\hat{f}_i^{eq}(\boldsymbol{x}, \hat{\boldsymbol{\xi}}_i, t) = \rho w_i (1 + \frac{\boldsymbol{u}\boldsymbol{c}_i}{c_S^2} + \frac{(\boldsymbol{u}\boldsymbol{c}_i)^2}{2c_S^4} - \frac{\boldsymbol{u}^2}{2c_S^2}) \tag{6.8}$$

Which uses the numerical non-dimensionalized speed of sound $c_S = 1/\sqrt{3}$ of the LBM and the descriptor-dependent weight factors w. Table 6.1 lists the w_i.

i	w
0	1/3
1,2,3,10,11,12	1/18
4,5,6,7,8,9,13,14,15,16,17,18	1/36

Table 6.1: Weight factors w of the D3Q19 lattice descriptor for the index i. Taken from [67].

With the solution of the transportation step and the $\hat{f}_i^{eq}$ to be used with the BGK model, the discrete scheme becomes:

$$\text{Collision step:} \quad \hat{f}_i'(\boldsymbol{x}, t) = \hat{f}_i(\boldsymbol{x}, t) + \hat{\omega}\Delta t(\hat{f}_i^{eq}(\boldsymbol{x}, t) - \hat{f}_i(\boldsymbol{x}, t)) \tag{6.9}$$

$$\text{Streaming step:} \quad \hat{f}_i(\boldsymbol{x} + \hat{\boldsymbol{\xi}}_i\Delta t, t + \Delta t) = \hat{f}_i'(\boldsymbol{x}, t) \tag{6.10}$$

After both scheme steps are complete, the solver completes one time step. The scheme is considered second-order accurate in space and time dependent on Ma [48, 49]. Furthermore, the computation is perfectly parallel because the collision step only uses the populations local to the cell. Intricate data storage models exist for populations that eliminate the streaming step. Section 6.1.2 details the streaming scheme used in this thesis. With this, the LBM is a perfectly parallel scheme with robust ideal scaling, perfect for GPGPUs [71, 72].

Nevertheless, the time step Δt restricts the performance of the LBM scheme. Because the time step results from the cell size Δx and the speed of sound c by $\Delta t = \Delta x\ c$. It results from the resolution required for accurate predictions of the given problem and the fluid properties. In order to circumvent this restriction, the speed of sound c can be artificially decreased for larger Δt. Dividing by this reference values gives non-dimensionalized fluid properties and allows the time step adaptation. The LBM solver used in this thesis provided automatic non-dimensionalization. Nevertheless, reducing c influences errors in the scheme, most notably the Ma dependent error the descriptor introduces [48, 49].

With the LBM defined, the macroscopic fluid values density ρ, velocity $\boldsymbol{u}$, and stress σ can be calculated from the populations $\hat{f}_i$ [48, 49]:

$$\rho = \sum_i \hat{f}_i \tag{6.11}$$

$$u_j = \sum_i \hat{c}_{ij}\hat{f}_i \tag{6.12}$$

$$\sigma_{ij} = \sum_\alpha \hat{c}_{\alpha,i}\hat{c}_{\alpha,j}(\hat{f}_\alpha - \hat{f}_\alpha^{eq}) \tag{6.13}$$

Smagorinsky subgrid stress model with volume force

For fluid simulations, the Reynolds number Re indicates whether inertial or viscous forces dominate the flow. Most technical flows, including rotorcraft and ship wakes, have high Re, which leads to highly turbulent wakes [55]. Unfortunately, fluid solvers

struggle to compute flows at high Re because instabilities from rounding and modeling errors are not sufficiently dampened [43, 73].

Introducing artificial viscosity can stabilize the simulation depending on its type. In this regard, the LBM used in this thesis behaves like a Large Eddy Simulation (LES). Therefore, it simulates the Navier-Stokes equations directly but has a specific minimal size of eddies it can simulate [43]. This minimal size corresponds to $\approx 2\Delta x$. In contrast, if $2\Delta x$ is small enough to represent the smallest eddies in the Kolmogorov cascade [74], the simulation is referred to as Direct Numerical Simulation (DNS). In order for LES simulations to model the turbulence a DNS captures, the effect of the eddies that the grid fails to represent has to be modeled by sub-grid stress models [43].

In contrast to simulations that model turbulence on an abstract level like URANS, the turbulence model in LES simulations may only model the dissipative effects of all scales that are smaller than what the discretization can describe [43]. A widely-used model is the Smagorinksy turbulence model [75, 76], which adds viscosity ν^{SGS} to account for sub-grid stresses in the grid [43].

The Smagorinsky model modifies the discrete relaxation frequency $\hat{\omega}$ by adding viscosity. To estimate the sub-grid stresses, it uses the local stresses $\hat{\sigma}$ from which the intermediate term $\hat{S}$ is calculated [77, 52]:

$$\hat{S} = \frac{\sqrt{\nu^2 + 18s^2\sqrt{\sigma_{ij}\sigma_{ij}}} - \nu}{6s^2}$$

This formula uses the Smagorinksy constant s. This term modifies the discrete relaxation frequency depending on the local stress and the Smagorinksy constant:

$$\hat{\omega} = \frac{1}{3(\nu + s^2\hat{S}) + 0.5}$$

This thesis uses the Smagorinsky turbulence model to stabilize the otherwise diverging boundary conditions. As the stability depends on the geometry, domain size, and flow velocities, the sections describing the respective simulation give the used value of s.

So far, the preceding explanations have excluded forces acting on the volume for simplicity. As forces acting on a cell's volume accelerate the particles, the forces change the populations. Nevertheless, multiple ways exist to add the force's contribution to the population. This thesis uses the forcing scheme described in [78]. Therefore, the first half of the force is applied to compute an intermediate velocity $\boldsymbol{u}^*$ [78, 62]:

$$\boldsymbol{u}^* = \boldsymbol{u} + 0.5\Delta t\boldsymbol{F}$$

The subsequent calculation of the $\hat{f}^{eq}$ uses this intermediate velocity to compute the Maxwell distribution in the collision step of equation 6.10. The other half of $\boldsymbol{F}$ is added to the populations post-collision $\hat{f}'$ [78, 62]:

$$\hat{f}_i' = \hat{f}_i' + \rho(1 - \frac{\hat{\omega}}{2})w_i \sum_j (\frac{\hat{c}_{ij} - u_j^*}{c_s^2} + \frac{\boldsymbol{c}_i\boldsymbol{u}^*}{c_s^4}\hat{c}_{ij}F_j)$$

Streaming step

With collision and discretization of volume forces defined, the streaming step is the last part of equation 6.10 that still needs to be defined. The trivial option, called AB pattern, consists of two identical data fields that contain all cells that form the domain. During the streaming step, the populations $\hat{f}_i$ are read from data field A at position $\boldsymbol{x}$ and stored in data field B at $\boldsymbol{x} + \hat{\boldsymbol{\xi}}_i \Delta t$ [67]. Nevertheless, this extra step of reading and writing data to memory without any calculation can be avoided [79], which also offers a substantial reduction in memory footprint [80]. Prominent versions are the A-A pattern [81] and Esoteric-twist [82] schemes.

This thesis uses a combination of the A-A pattern, and Esoteric-twist called Shift and Swap Streaming Model (SSS) developed by a collaboration of the author with the Karlsruhe Institute of Technology (KIT) [79]. It flattens the three-dimensional cell indices in the domain with the index hash [79]:

$$i^{cell} = i_1 * n_2 * n_3 + i_2 * n_3 + i_3 \tag{6.14}$$

Which depends on i_1, i_2, i_3 as the three-dimensional cell indices, and n_2, n_3 representing the number of cells in the respective directions [79]. It then allocates a linear data array for every $\hat{f}_i$ with a length of the domain plus the absolute value of the offset $o_i = \hat{c}_{i1} * n_2 * n_3 + \hat{c}_{i2} * n_3 + \hat{c}_{i3}$ [79]. The block of arrays for all $\hat{f}_i$ are called fluid data in the following.

In order to transport the populations in the domain of cells without extra memory operations, the scheme requires the distinction between an even and odd collision step. In this context, even and odd refer to the count of executed collision steps since the simulation starts modulo two. The zeroth step of the simulation is defined to be even. After the allocation, it creates an even pointer that maps the $i^{cell} = 0$ to the fluid data position o_i for the population $\hat{f}_i$ if $o_i > 0$ and to zero otherwise [79]. Furthermore, it creates an odd pointer with the same mapping but assigned to the population $\hat{f}_i$ pointing in the opposite direction in the lattice descriptor.

In even collision steps, the populations $\hat{f}_i$ are read at the respective i^{cell} but stored in the array of the $\hat{f}_i$ with the opposite $\hat{\boldsymbol{c}}_i$, except for $\hat{\boldsymbol{c}}_0$ [79]. Thus, reading from the odd pointer with its index mapping in the next step automatically moves the populations of one cell in the respective $\hat{\boldsymbol{c}}_i$ direction [79]. This scheme eliminates the streaming step of equation 6.10.

6.1.3 Boundary conditions

The discretized algorithm allows the simulation of the fluid behavior in the presence of forces. Nevertheless, suitable boundary conditions are required to simulate technical fluid flows, such as the ship wake or rotor inflow calculation, in this thesis. On the one hand describing the ship and fuselage geometry requires boundary conditions that form solid slip and non-slip walls. On the other hand in and outflow and boundary conditions that let information pass freely out of the domain are necessary for the LRM.

Bounce back for modeling walls

Simulating the rotorcraft fuselage or the ship using wall boundaries requires stick and slip wall boundary conditions. In the LBM, these boundary conditions mirror the populations that travel from the fluid into the wall [48]. Depending on only mirroring normal components or normal and tangential components, the resulting boundary acts as a slip or non-slip wall [48].

To impose a non-slip boundary between a cell at $\boldsymbol{x}^F$ inside the fluid and the wall, the outward pointing normal vector of the wall $\boldsymbol{n}^{Wall}$ is calculated. Then, populations $\hat{f}_i$ that travel towards the wall with $\hat{\boldsymbol{c}}_i$ are reflected (bounced back) along $-\hat{\boldsymbol{c}}_i$ during the streaming step [48]. By reflecting both the normal and tangential component of $\hat{\boldsymbol{c}}_i$ the velocity at the middle of $\boldsymbol{x}^F$ and the wall computes to $\boldsymbol{u}^{Wall} = 0$.

Slip wall boundaries apply the same basic approach but only reflect the vector component of $\hat{\boldsymbol{c}}_i$ that is normal to the wall. In this case, the boundary reflects the populations like photons in a mirror. This results in a normal component of the velocity of $u_n = 0$ but without changing the u_t components of the velocity at the boundary [48].

Despite being able to model many different wall-bound problems, these boundaries introduce a discretization error of $O(\Delta x)$ [48]. Thus, numerous formulations with lower-order error for curved boundaries, e.g., [83, 48, 84], were introduced and compared for their stability in various flow conditions [85, 86].

Although the higher-order formulations allow more freedom in the boundary geometry, this thesis uses the simple bounce-back scheme for slip and non-slip boundaries. With its real-time context, the LRM favors the minimal computational load introduced by the simple bounce-back schemes over accuracy.

Velocity and pressure boundary condition for in- and outlet

Apart from forces and wall boundaries, parts of the domain boundary can act as inflow and outflow. In contrast to wall boundaries, information can travel in and out of the domain at these boundaries. The populations that move from inside the domain to the outside transport information to the boundary. Therefore, the boundary condition generates the populations that enter the domain by combining incoming information with the user-set outside state. These boundary conditions represent Dirichlet or Neumann-type boundary conditions. Although both in- and outflow boundaries can be of both types, this thesis restricts these combinations. It uses Dirichlet type boundaries for the inflow and Neumann type boundaries for the outflow [67]. To increase the stability, the finite-difference velocity gradient method, which is an adaptation of [87], is used [85].

For the Dirichlet type inflow boundary, the velocity $\boldsymbol{u}$ at the boundary is user-defined while the boundary condition adapts the density ρ [67, 85]:

$$\rho^{Wall} = \frac{1}{1 + u_n}(2\rho^+ + \rho^0)$$

The formula uses ρ^+ calculated from populations moving out of the domain and ρ^0 calculated from populations moving tangentially to the boundary plus $\hat{f}_0$.

In contrast, the Neumann type outflow boundary uses the user-defined density ρ and adapts the velocity normal to the wall u_n^{Wall} [67, 85]:

$$u_n^{Wall} = sign(n_n)\frac{2\rho^+ + \rho^0}{\rho}$$

This formula uses the normal component of the outward pointing normal vector n_n to determine the sign. The velocity $\boldsymbol{u}$ in the non-normal directions is calculated using 6.12.

In order to calculate the populations traveling from the outside into the domain $\hat{g}_i$, the stress tensor σ_{ij} is calculated from the strain rate using finite differences [85]:

$$\hat{g}_i = \hat{f}_i^{eq}(\rho, \boldsymbol{u}) - \frac{\rho w_i}{c_s^2 \omega}\boldsymbol{Q}_i : \boldsymbol{S}$$

This contracts the strain rate tensor $\boldsymbol{S}$ and the tensor $\boldsymbol{Q}_i$ which are defined as [85]:

$$\boldsymbol{S} = 0.5(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T)$$
$$\boldsymbol{Q}_i = \hat{c}_i \hat{c}_i - c_s^2 \boldsymbol{I}$$

With the defined wall and in- and outflow boundary conditions, the solver allows flow simulations around and inside objects. Nevertheless, using these boundary conditions on edges and corners of the domain requires special adaptations which are described in [67]. Furthermore,, the in- and outflow boundary conditions need to be placed at a sufficient distance from the object creating the wake in order to be stable. This results from the inability to pass a fluid state that contains significant stresses $\boldsymbol{\sigma}$ through the boundary. Furthermore, the outflow boundary can reflect part of the information that streams to the outside, thus interfering with the flow field on the inside [88].

Impedance boundary for non-reflective outlets

Placing boundaries at large distances from the rotor increases the computational expense per time step as more cells are necessary to fill the domain. The resulting computational expenses conflict with the firm real-time requirement that the LRM domain must fulfill. Therefore, outflow boundaries are required to pass information out of the domain without reflecting while being stable enough to be placed close to the rotor.

A boundary formulation that fulfills these requirements are impedance boundaries. This boundary type originates in acoustic simulations and allows free passage of information carried to the boundary by waves [88]. In order to do this, they adapt the acoustic impedance of the fluid to allow waves coming from inside to pass without reflection [88].

In order to prevent the reflection of an incoming wave with velocity u^{wave}, the momentum change in the cell has to compensate for the incoming momentum flux [88]:

$$\frac{\partial \hat{J}^{cell}}{\partial A \partial t} + \frac{\partial \hat{J}^{flux}}{\partial A \partial t} = \pm \hat{d}\rho c_s^2 \pm 0.5\rho(\hat{d}u^{wave})^2 + \hat{d}u^{wave}\rho c_s = 0 \tag{6.15}$$

Which uses the discrete differential $\hat{d}\star = \star(t) - \star(t - \Delta t)$. The sign of the outward pointing normal determines the $\pm$. For $sign(n_n) > 0$ the $\pm \rightarrow -$ [88]. Furthermore, the differential wave velocity $\hat{d}u^{wave} = sign(\hat{d}u_n)|\hat{d}\boldsymbol{u}|$ is defined.

This corresponds to setting the acoustic impedance z of the fluid to [88]:

$$z = \frac{\hat{d}p}{\hat{d}u^{wave}} = \pm \frac{c_s \hat{d}\rho}{\sqrt{1 - 2\hat{d}\rho/\rho - 1}}$$

Nevertheless, only the populations traveling from inside the domain towards the boundary are known. To address this, the approach of Zou and He [89] is used to calculate the density ρ at the wall [88]. It computes the density ρ by subtracting equation 6.12 from 6.11:

$$\rho = \frac{\sum_i \hat{f}_i - \sum_i \hat{c}_{in} \hat{f}_i}{1 - u_n}$$

Inserting this in equation 6.15 gives the following equation[88]:

$$\pm \hat{\rho} c_s^2 \pm 0.5\rho(\hat{d}u^{wave})^2 + \hat{d}u^{wave} \rho c_s = 0$$

This non-linear equation can be solved using an iterative Newton-Raphson solver [88]. With ρ and u^{wave} defined, the boundary has the correct impedance z to let information pass from the inside of the domain to the outside. Nevertheless, some populations are still undefined. The used solution bounces back the non-equilibrium part of these populations as described in [89, 88]. Furthermore, as previously, imposing the boundary conditions on edges and corners of the domain requires special adaptations given in [67].

Friedmann [52] successfully used this boundary condition to simulate a rotor in a cube with four rotor radii edge lengths. However, these simulations focused on static rotor forces. Nevertheless, the boundary proved its stability in investigations on the dynamic response to a step in rotor thrust [1].

6.1.4 Arbitrary Lagrangian-Eulerian formulation for free grid motion

As the LRM domain is fixed to the rotorcraft's center of gravity, it moves with the rotorcraft during flight. Therefore, the LBM requires the solver to support moving grids.

An Arbitrary Lagrangian-Eulerian (ALE) domain allows the LRM to realize this requirement. This method allows the movement of the domain containing the grid of cells independent of the solution calculated inside the domain [90]. Therefore, it allows the movement of the domain with the rotorcraft in each time step while capturing the effect of the motion on the solution inside the domain [90]. ALE methods define the projection function $\boldsymbol{\varphi}$ to map information from an Eulerian description in $R^{\boldsymbol{X}}$ to a Lagrangian description in $R^{\boldsymbol{x}}$ [90]. Additionally, the functions $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$ map from the domain used for the computation in $R^{\boldsymbol{\chi}}$ to the Eulerian and Lagrangian description [90]. Therefore, the physical equations can be defined in Eulerian or Lagrangian notation

and mapped to the computational domain. Thus, the description of the equations becomes independent of the grid used to represent the domain in $R^{\boldsymbol{\chi}}$ [90]. Figure 6.2 depicts the descriptions and the respective mapping functions.

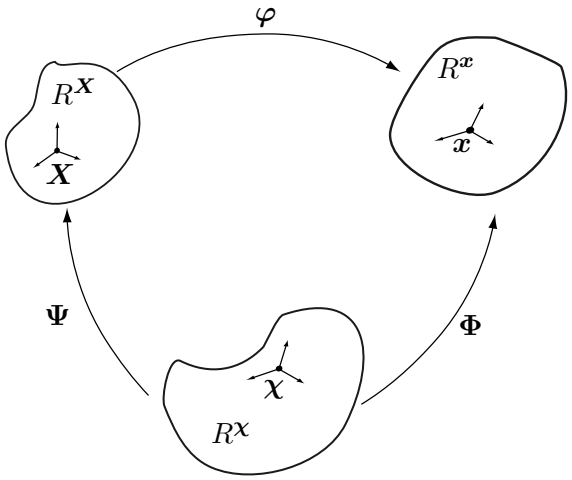


Figure 6.2: ALE domain in $R^{\boldsymbol{\chi}}$ is independent of the mapping $\boldsymbol{\phi}$ from Eulerian description in $R^{\boldsymbol{X}}$ to Lagrangian description in $R^{\boldsymbol{x}}$. $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$ map from $R^{\boldsymbol{\chi}}$ to $R^{\boldsymbol{X}}$ and $R^{\boldsymbol{x}}$ respectively.

The mapping functions for the ALE depend on the translation, rotation, and deformation the reference grid defined in $R^{\boldsymbol{\chi}}$ undergoes [90]. Nevertheless, in the context of this thesis, the cells in the fluid domains are not deformed, just translated and rotated. Furthermore, the fluid state in the domain describes the absolute values in its domain-specific coordinate system $\boldsymbol{\chi}$. Therefore, interpolating and rotating the values suffices to describe the mapping between the domains.

This affine transformation simplifies the ALE algorithm, as it does not require a transformation into a Lagrangian state. After computing the collision and streaming step in the domain in $R^{\boldsymbol{\chi}}$ at time t, the resulting absolute fluid quantities can be projected with $\boldsymbol{\Psi}$ into Eulerian space $R^{\boldsymbol{X}}$ by a rotation. In the next time step of the simulation at $t + \Delta t$, the domain in $R^{\boldsymbol{\chi}}$ can be filled by projecting the values in Eulerian space $R^{\boldsymbol{X}}$ via $\boldsymbol{\Psi}^{-1}$ into $R^{\boldsymbol{\chi}}$. Cells outside $R^{\boldsymbol{X}}$ at t receive their values from the user-defined surrounding state. Nevertheless, this requires storing the last solution at t and providing storage for the domain $R^{\boldsymbol{\chi}}$ to project into. Therefore, this projection effectively doubles the storage requirement of the used domain.

The fluid state in the incompressible Navier-Stokes equations consists of the density ρ, velocity $\boldsymbol{u}$, and the stresses $\boldsymbol{\sigma}$. These quantities result from equations 6.11 to 6.13 for every cell. For the projection $\boldsymbol{\Psi}$ and its inverse $\boldsymbol{\Psi}^{-1}$, the density ρ is interpolated while velocity $\boldsymbol{u}$ and stresses $\boldsymbol{\sigma}$ are tri-linearly interpolated and rotated [91]. Allowing arbitrary points to represent the center of rotation requires transforming the translation and the rotation. This step consists of computing the values with respect to the domain's local coordinate system. This transformation allows to enter the translation and rotation with respect to the center of gravity of the NatoGD or MBB Bo 105.

After the projection of ρ, $\boldsymbol{u}$, and $\boldsymbol{\sigma}$ into R^{χ} at $t + \Delta t$, the populations are initialized from this macroscopic fluid state [91]:

$$\hat{f}_i = \hat{f}_i^{eq} + \hat{f}_i^{neq} \tag{6.16}$$

Which uses 6.8 to calculate $\hat{f}_i^{eq}$. The non equilibrium part $\hat{f}_i^{neq}$ can be calculated from the local stress tensor $\boldsymbol{\sigma}$ [67]:

$$\hat{f}_i^{neq} = \frac{w_i}{2c_s^4} \frac{\hat{c}_{ia}\hat{c}_{ib} - c_s^2\delta_{ab}}{\sigma_{ab}}$$

Although computing, interpolating, and rotating the macroscopic fluid values guarantees the scheme to match the Navier-Stokes equations [91], the computational load is high. To reduce this, the rotation of the domain from t to $t + \Delta t$ can be neglected. This simplification eliminates all rotations of $\boldsymbol{u}$ and $\boldsymbol{\sigma}$, thus allowing to interpolate the populations $\hat{f}_i$ directly. Neglecting the rotation saves the computations necessary for equations 6.11 to 6.13, rotating $\boldsymbol{u}$ and $\boldsymbol{\sigma}$ and reconstructing the populations from 6.16. Nevertheless, it introduces a directional error of $O(\alpha)$ in $\boldsymbol{u}$ and $\boldsymbol{\sigma}$ for the domain rotating a total angle α while moving from t to $t + \Delta t$.

Despite this error, the algorithm was used in previous work [52, 1, 2, 3, 53]. As the time step Δt of the LBM is small compared to the time scale the rotorcraft motion requires, the introduced error was considered negligible. Nevertheless, the used ALE algorithm in this thesis varies for different domains. The respective domain description lists the used version.

6.1.5 Overset technique

In order to simulate moving geometries inside an LBM domain, immersed boundary conditions are commonly used [92, 93]. Especially, the NatoGD was simulated using LBM and immersed boundary conditions [34]. The immersed boundary conditions eliminate the need for an ALE grid. Despite this advantage, the immersed wall boundaries require additional computational effort compared to simple bounce-back wall boundaries.

An ALE grid simulates moving geometries by defining bounce-back walls inside a domain that moves with the geometry [1]. Nevertheless, this only works with multiple geometries if they move along the same path inside the domain. Furthermore, different Reynolds numbers of flow around the objects require different cell sizes in the vicinity of the objects. Using local grid refinement in combination with immersed boundaries [34] solves this problem. Alternatively, multiple ALE grids around the objects and combining them with the overset technique [94, 95], which is also referred to as Chimera technique [96, 97] or composite-grid technique [98] can be used.

This thesis follows the approach in [63] to implement an overset mesh in LBM. Nevertheless, it extends the method to three dimensions and combines it with the ALE mesh described in section 6.1.4. The overset technique allows combining any number of grids with different resolutions moving along different paths while being distributed onto any number of GPGPUs.

This chapter explains the overset technique with just two grids for simplicity. One larger domain, referred to as the outer domain, encloses the smaller domain, referred to as the inner domain. In addition to its regular cells, called core cells, the inner domain has ghost layer cells on the outside. Figure 6.3 shows the outer domain with its cells in (—), the core cells of the inner domain in (—) and the cells forming the ghost layer of the inner domain in (—).

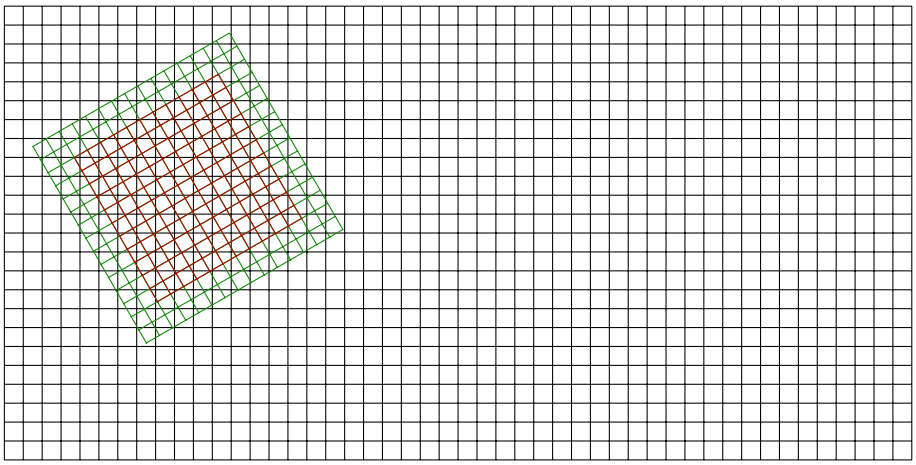


Figure 6.3: Overset mesh with outer grid (—), inner grid (—), and ghost layer of inner grid (—).

Although the description given in this chapter does not explicitly describe a complete simulation step of the LBM, extending the simulation algorithm of the LBM with the interface step used for the overset technique is trivial [63]. Nevertheless, this section briefly describes generalizing and integrating the overset technique into a simulation with an arbitrary number of domains at arbitrary positions at the end.

Apart from using only two grids, the algorithm described in the following uses the same time step for both the inner and outer grids. Thus, no interpolation in time is required. Nevertheless, this is not a requirement of the overset technique. Furthermore, the values stored in the populations $\hat{f}_i$ inside the grids are absolute in the coordinate system of their domain, as defined in section 6.1.4. Additionally, the values are in non-dimensionalized form, which depends on the parameters the containing domain was set up with, as defined in section 6.1.2. The outer grid completely contains the inner grid. Therefore, the inner domain does not need boundary conditions, just the ghost layer for the overset interface.

The overset technique adds a new interface step the algorithm executes before the LBM starts colliding and streaming the populations. During this interface step, the populations $\hat{f}_i$ inside the ghost layer of the inner grid (—) are set using ρ, $\boldsymbol{u}$, and $\boldsymbol{\sigma}$ of the outer domain (—) at the cell position of the inner domain. Figure 6.4 describes the algorithm that every cell in the ghost layer of the inner domain (—) executes. This step refers to the outer domain (—) as the “other” domain.

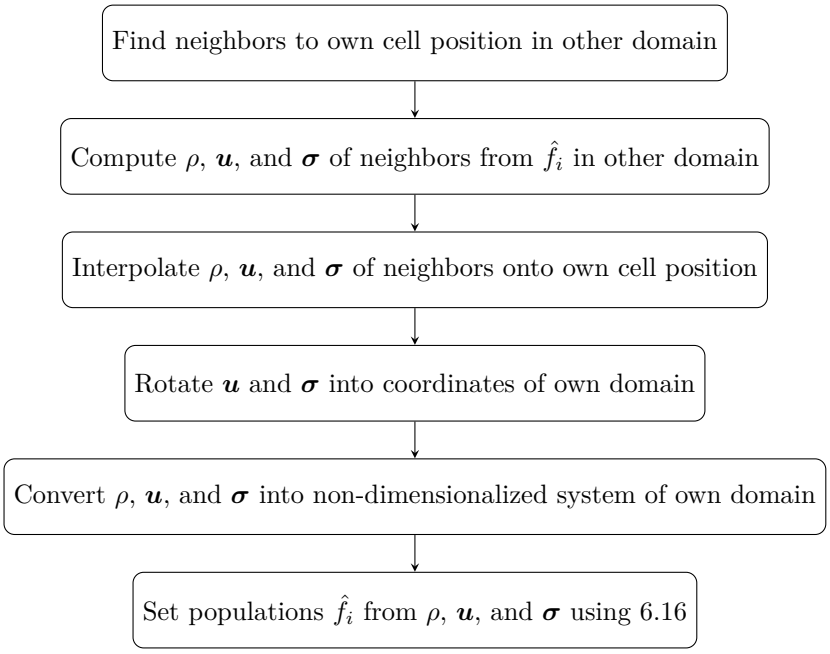


Figure 6.4: Schematic of the algorithmic steps at the interface of the overset mesh.

After this completes, the cells of the outer domain (—) that are inside the core of the inner domain (—) execute the same algorithm 6.4 to set their populations. In this case, the core of the inner domain (—) is referred to as the "other" domain.

With the overset technique completed after this interface step, the LBM algorithm executes the collision and streaming step and moves the domains according to the ALE formulation in 6.1.4. Multiple GPGPUs calculating different sections of the used domains does not change the interface step of the overset technique. Nevertheless, the mapping of domains to GPGPU has to be incorporated. Furthermore, if the time step Δt differs between the domains, the interface step must be extended for an interpolation step in time. As the ALE requires the storage of the solution at t and $t + \Delta t$, linear interpolation in time is possible without additional storage requirements. Nevertheless, as the LBM is a second-order scheme in time and space [48, 98], second-order interpolation in time is required to preserve the mathematical order.

Furthermore, to be consistent with the mathematical order of the LBM scheme, the spatial interpolation in algorithm 6.4 needs to be second-order or higher for ρ and $\boldsymbol{u}$ [98]. Additionally, the preservation of the entire fluid state the LBM computes requires interpolating the higher order moments like $\boldsymbol{\sigma}$ with first-order [98]. Nevertheless, as the ALE scheme uses first-order tri-linear interpolation to move the domain while neglecting all moments except ρ, $\boldsymbol{u}$, and $\boldsymbol{\sigma}$, the same order is used for the interpolation in the overset technique in this thesis.

In order to allow the interpolation algorithm to efficiently read the fluid data at the neighbors during the interface step of the outer domain (—) with the core of the inner domain (—), the ghost layer (—) needs to be as many cells wide as the stencil

of the interpolation function requires. Finally, if the neighbor a cell requires during the interface step is outside the respective domain, the user-defined surrounding state prescribes the values.

This overset technique, in combination with ALE domains, allows the adaptation of the resolution of the domains to the respective physical requirements inside while allowing relative motion. Furthermore, not requiring in-, outflow, or impedance boundary conditions on the inner domain eliminates their influence on the solution. Therefore, investigations into the effect of these boundary conditions are possible.

6.2 Domain and parameters for simulating the air wake of a moving NatoGD

This thesis uses the previously presented LBM to compute a representation of the air wake of the NatoGD. Therefore, the NatoGD geometry shown in 5.2 is voxelized from the step-file provided in [12]. The voxelization allows the LBM to represent the NatoGD using non-slip wall boundary conditions.

Furthermore, using an ALE domain allows capturing the influence of the ship motion on the wake of the ship, which is significant as shown in [9, 34]. Furthermore, the dataset includes ship motion profiles [12]. The used ALE domain containing the ship interpolates ρ, $\boldsymbol{u}$, and $\boldsymbol{\sigma}$ as described by Meldi [91]. Fixing the NatoGD to the ALE domain allows the wall boundary conditions that form the ship to be voxelized only once.

The rectangular domain spans $4l^{NatoGD}$ in x-direction, $1.5l^{NatoGD}$ in y-direction, and $0.88l^{NatoGD}$ in z-direction. As reference length, the length over all of the NatoGD, l^{NatoGD} in x-direction, shown in Fig. 5.2, is used. The domain coordinate system follows the definition of the cg system for the rotorcraft shown in Fig. 5.4. Therefore, x points along the ship from stern to bow, y points towards the right, and z points down. Figure 6.5 shows the domain, the used coordinate system, and the dimensions. Furthermore, it shows the voxelized NatoGD and the sea surface in grey.

Inside the domain, the crossing of waterline and stem, as indicated by the 1-coordinate system in Fig. 5.2, is placed at $x = (4 - 0.75)l^{NatoGD}$. While the ship is centered in y-direction, the waterline is placed at $z = (0.88 - 0.088)l^{NatoGD}$. Non-slip wall boundary conditions form the solid hull of the ship and the sea surface. Nevertheless, the cells forming the sea surface vary depending on the ship's motion as the grid moves with the hull. Both sides of the domain (in y-direction) and the top of the domain at $z = 0$ use slip wall boundaries. This minimizes the influence on the wake and avoids impedance boundaries that are unstable in this simulation. These wall boundaries span the whole range in x. Thus, the inflow at $x = 4l^{NatoGD}$ and the outflow at $x = 0$ do not span the corresponding plane but the insert left by the wall boundary conditions. The number of cells per ship length refers to the resolution of this domain N^{NatoGD}. Furthermore, the Smagorinsky turbulence model with $s = 0.025$ stabilizes the turbulent simulation.

The solver sets the values of $\boldsymbol{u}$ in the cells that form the NatoGD to their respective velocities resulting from the ship motion in every time step. Therefore, the ALE

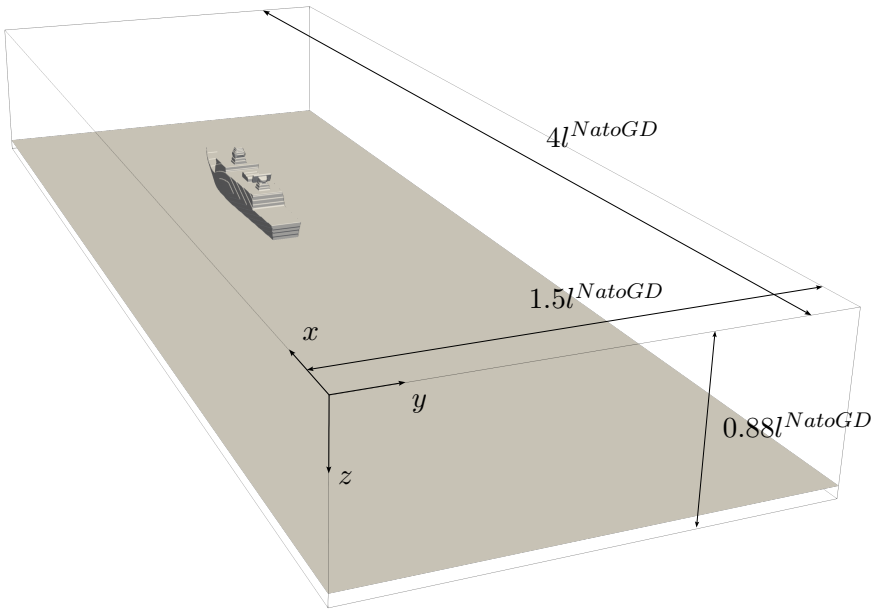


Figure 6.5: Domain used to simulate the air wake of the NatoGD including domain coordinate system and dimensions. The domain is fixed to the NatoGD. The $x-y$-plane represents the surface of the sea.

interpolates the correct velocity in cells close to the ship geometry. Furthermore, it accounts for the ship hull displacing air while moving.

As the measurements in [12] use a $1/50th$-scale model, openLB constructs the non-dimensionalized system [67] using the reference values provided in Tab. 6.2.

$Variable$	$Value$
$u^{ref}_{lattice}$	$0.3\frac{1}{\sqrt{3}}$
$u^{ref}_{physical}$	$60\,[\frac{m}{s}]$
$l^{ref}_{physical}$	$l^{NatoGD} = 3\,[m]$
$\nu^{ref}_{physical}$	$0.000146072\,[\frac{m}{s^2}]$
$\rho^{ref}_{physical}$	$1.225\,[\frac{kg}{m^3}]$

Table 6.2: Values used to non-dimensionalize the $1/50th$-scale model simulations of the NatoGD. Used in combination with the unit converter in [67].

This table uses ref to indicate the values being reference values. Furthermore, it indicates if the given values are in the non-dimensional unit system of the LBM using *lattice* or in physical units using *physical*. For the full scale NatoGD calculations, the unit converter uses the values shown in Tab. 6.3.

$Variable$	$Value$
$u^{ref}_{lattice}$	$0.3\frac{1}{\sqrt{3}}$
$u^{ref}_{physical}$	$80\,[\frac{m}{s}]$
$l^{ref}_{physical}$	$l^{NatoGD} = 150\,[m]$
$\nu^{ref}_{physical}$	$0.000146072\,[\frac{m}{s^2}]$
$\rho^{ref}_{physical}$	$1.1934\,[\frac{kg}{m^3}]$

Table 6.3: Values used to non-dimensionalize the full-scale model simulations of the NatoGD. Used in combination with the unit converter in [67].

6.3 Tabulation and interpolation of ship wake

Imposing the fluid state from the precomputed ship wake as velocity boundary conditions onto the LRM domain requires an efficient way to access the exported data at the positions of the boundary. In this thesis, the fluid state, consisting of ρ, $\boldsymbol{u}$, and $\boldsymbol{\sigma}$, is written to disk within a subregion of the domain described in 6.2 for the NatoGD [3].

For the NatoGD, this subregion starts at $x = 0$ and ends at the crossing of waterline and stem (indicated by coordinate system 1 in Fig. 5.2). It spans $0.6l^{NatoGD}$ centered around the ship midline in y-direction, starts at $0.24l^{NatoGD}$, and ends at $z = 0.88l^{NatoGD}$ in z direction. This subregion captures the visible wake of the NatoGD.

The wake of the NatoGD develops in $20s$ start-up period before the simulation tabulates the data every $0.02s$. Because this thesis only simulates flight with durations $< 30s$, 1500 export steps are sufficient. The LRM uses the tabulated data to prescribe the fluid state at the front and sides of the domain containing the rotor. The fluid state is tri-linearly interpolated for every time step of the LRM from the data exported by the NatoGD simulation. There is no interpolation in time, as the simulated ship wakes did not show significant change during the $0.02s$.

The data is stored in raw binary data to prevent losses and allow fast in and output between disk and Random Access Memory (RAM). Furthermore, the hash function described in equation 6.14 indexes the data. The data format and hash function allow minimizing the computational expense of prescribing the values at the inflow boundary conditions of the LRM.

6.4 Simulation of the Local Recalculation Model domain using the Lattice-Boltzmann Method

To simulate the fluid flow near the rotorcraft, the LRM uses a cubic domain with the rotor at its center. In order to investigate the influence of the size of the LRM domain, the edge length of the cube is defined by the size factor $a > 1.0$ times the minimal domain edge $4R^{mr}$ [3]. As the model computes the flow inside the LRM domain, setting a allows adapting the distance at which the LRM model incorporates information. Therefore, a is referred to as reach-factor [3].

Inside the domain, the rotor is placed centered in all directions. Furthermore, the domain moves through space along the path of the main rotor hub [3]. Due to the firm real-time requirement of the LRM, it uses the ALE formulation that directly interpolates the $\hat{f}_i$. Additionally, to ease interfacing the motion with the rotorcraft flight dynamics simulation, the center of rotation for the ALE domain is set to the rotorcraft center of gravity as shown in Fig. 5.4. Furthermore, the coordinate system of the LRM domain is aligned parallel to the *cg*-system used by the rotorcraft flight dynamics [3]. Figure 6.6 depicts the used domain, including the solver coordinate system and the used edge length of the cube.

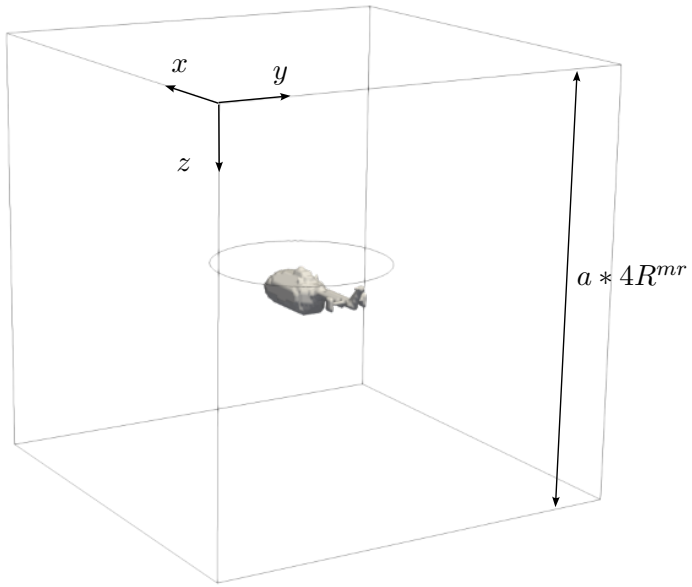


Figure 6.6: Domain used to simulate the LRM, including coordinate system and dimensions. The domain is fixed to the center of gravity of the MBB Bo 105. Grey solids represent the voxelized fuselage. The blade tip path shows the main rotor.

Non-slip wall boundary conditions represent the fuselage in the domain of the LRM. Thus, the geometry of a MBB Bo 105 fuselage is voxelized at the resolution N^{LRM}, which describes the number of cells per $4R^{mr}$ used. On the faces that impose information from the outside into the domain, shown in red in Fig. 5.3, the inflow boundary conditions described in section 6.1.3 are used. These interpolate the flow conditions of the NatoGD simulation described in section 6.2 from the tabulated data as described in section 6.3.

To allow the momentum the rotor adds to the fluid inside the domain to exit, the impedance boundary conditions described in section 6.1.3 are used on the faces marked in blue in Fig. 5.3. The impedance based formulation allows the outflow boundaries to be close to the rotor. With these boundary conditions and a Smagorinsky turbulence model at $s = 0.05$, simulations inside the flight envelope of the MBB Bo 105 are stable.

The reference values in Tab. 6.4 are used with the unit converter in [67] to non-dimensionalize the fluid quantities in the simulation. Furthermore, the LRM domain and the rotorcraft flight dynamics exchange the thrust and inflow of the main rotor, shown in green in Fig. 5.3. The next section describes this interface.

$Variable$	$Value$
$u^{ref}_{lattice}$	$0.3\frac{1}{\sqrt{3}}$
$u^{ref}_{physical}$	$30\,[\frac{m}{s}]$
$l^{ref}_{physical}$	$4R^{mr} = 4 * 4.92\,[m]$
$\nu^{ref}_{physical}$	$0.000146072\,[\frac{m}{s^2}]$
$\rho^{ref}_{physical}$	$1.1934\,[\frac{kg}{m^3}]$

Table 6.4: Values used to non-dimensionalize the simulations of the LRM domain. Used in combination with the unit converter in [67].

6.5 Discretization of rotor forces and inflow

In order for the rotorcraft and, ultimately, the pilot to feel the influence of the surroundings captured by the LRM domain, the forces and inflow velocities of the main rotor are coupled to the BEM [1, 3]. Fig. 5.3 shows this interface between the rotorcraft flight physics and the LRM in green. As highlighted in section 5.2, one part of the interface consists of imposing the rotor forces inside the rotor disk calculated by the rotorcraft flight dynamics model onto the rotor disk inside the LRM domain. The second part of the interface extracts the inflow velocities inside the rotor disk for use as inflow inside the rotorcraft flight dynamics model.

Generally, this interface's discretization of aerodynamic forces and inflow velocities are independent. The aerodynamic forces can be prescribed on individual blade levels for every blade element using different discretizations for the forces as shown by [99]. This thesis restricts the discretization of the rotor forces to a harmonic representation in cylindrical coordinates without representing individual blades [1] to maximize computational efficiency. The number of harmonics is restricted to $N^{harm} = 9$ which was able to capture 99.9% of the variability of the blade forces in ψ for the flight tests shown in chapter 8.2.2:

$$\boldsymbol{F}^{mr}_{mr}(\psi) = \sum_{k=0}^{N^{harm}} (\boldsymbol{cos}_k(k\psi) + \boldsymbol{sin}_k(k\psi))$$

In this, the zeroth cosine component $\boldsymbol{cos}_0$ represents the mean forces of the rotor disk [1]. The component vectors $\boldsymbol{cos}_k$ and $\boldsymbol{sin}_k$ are calculated from the forces $\boldsymbol{F}^i$ of

the $N^{blades} = 4$ blades discretized with $N^{\psi} = 72$ steps per revolution [1]:

$$\boldsymbol{cos}_k = \frac{2}{N^{\psi}} \sum_{i=1}^{N^{blades}} \sum_{j=1}^{N^{\psi}} \boldsymbol{F}^i(\psi_j) cos(k\psi_j)$$

$$\boldsymbol{sin}_k = \frac{2}{N^{\psi}} \sum_{i=1}^{N^{blades}} \sum_{j=1}^{N^{\psi}} \boldsymbol{F}^i(\psi_j) sin(k\psi_j)$$

This discretization allows force variations with ψ but not with the radial coordinate r. Nevertheless, this represents an adequate approximation for the linearly twisted rotor blades of the MBB Bo 105. The approximation can result in significant errors for rotors with different force distributions in the radial coordinate.

As the LRM neglects the flapping of the blades in the rotor interface, the rotor disk resides in one layer of cells of the domain. Nevertheless, the cubic cells inside the domain fail to represent the circular rotor disk. Therefore, to impose the forces of the main rotor $\boldsymbol{F}_{mr}^{mr}$, the rotor disk intersects with the grid of cells in the plane of the rotor disk. A convex polygon using 360 points discretizes the rotor disk for this process. Then a polygon intersection algorithm [100] calculates the area fraction inside each cell with a radial distance of the center point between $R^{outer} = R + 0.5\sqrt{2}\Delta x$ and $R^{inner} = R - 0.5\sqrt{2}\Delta x$. Dividing the area of the resulting overlapping polygon by the cell's area gives the area fraction $\epsilon^{inside} = A^{inside}/A^{cell}$. Figure 6.7 depicts this intersection in red. All cells with a radial distance smaller than R^{inner} have $\epsilon^{inside} = 1$, while all cells with radial distance greater than R^{outer} have $\epsilon^{inside} = 0$.

To impose the force $\boldsymbol{F}_{mr}^{mr}(\psi)$, it is non-dimensionalized with the unit converter of the LRM domain, divided by the total rotor area and multiplied with ϵ^{inside} of the respective cell. By placing the rotor disk at the center of the domain and aligning the domain coordinate system with the main rotor system mr, the calculated area fractions ϵ^{inside} of the cells are symmetric to the x- and y-axis. In combination with the rotational symmetry of the force $\boldsymbol{F}_{mr}^{mr}$, this guarantees that the total amount of force added to the LRM domain equals the mean values of $\boldsymbol{F}_{mr}^{mr}$.

With force imposed on the LRM, the domain can advance for at least one time step Δt before extracting the inflow to be used by the rotorcraft flight dynamics. To discretize the inflow $\boldsymbol{v}$, the mean values $\bar{\boldsymbol{v}}$ and coefficients linear in r and with the first harmonic in ψ are calculated [1]:

$$\boldsymbol{v}(r, \psi) = \bar{\boldsymbol{v}} + \boldsymbol{v}^{sine} \frac{r}{R} sin(\psi) + \boldsymbol{v}^{cosine} \frac{r}{R} cos(\psi)$$

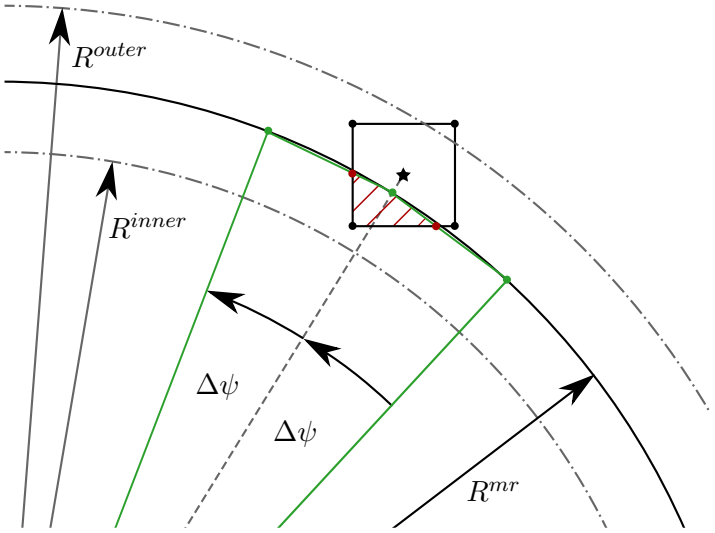


Figure 6.7: Schematic view of the intersection of a cell with the rotor disk. (—) represents a segment of the circular disk formed by the resolution N^{ψ}. Intersection-points and -area in (—). Adapted from [1].

The mean and coefficient vectors result from summing over all cells N^{mr} that lie inside R^{outer} [1]:

$$\bar{\boldsymbol{v}} = \frac{1}{A^{mr}_{lattice}} \sum_{i=1}^{N^{mr}} \epsilon^i \boldsymbol{u}(\boldsymbol{x}^i)$$

$$\boldsymbol{v}^{sine} = \frac{1}{A^{mr}_{lattice}} \sum_{i=1}^{N^{mr}} \epsilon^i \boldsymbol{u}(\boldsymbol{x}^i) \frac{r}{R} sin(\psi)$$

$$\boldsymbol{v}^{cosine} = \frac{1}{A^{mr}_{lattice}} \sum_{i=1}^{N^{mr}} \epsilon^i \boldsymbol{u}(\boldsymbol{x}^i) \frac{r}{R} cos(\psi)$$

The formulas use the area fraction of the cells ϵ^{cell} and the area of the rotor disk in the non-dimensionalized lattice unit system $A^{mr}_{lattice} = \sum_i^{N^{mr}} \epsilon^i$ to compute the average of the velocities inside the rotor disk. The discretization for the forces and the velocities span the complete rotor disk, ignoring a root cutout of the blades or the rotor hub at the rotor center. Nevertheless, the proven and widespread Pitt-Peters model [7] can predict the dynamic rotorcraft motion in free flight using this simplification. Furthermore, not imposing forces or wall boundary conditions in the cutout and hub region can lead to recirculation [99]. Choosing the exact discretization of the Pitt-Peters allows direct comparison of the LRM with the Pitt-Peters model.

Using this interface definition for rotor forces and inflow allows coupling the LRM domain with the rotorcraft flight dynamics. In order to prevent the need for interpolation in time in the interface, both the LRM and the rotorcraft flight physics use the

same time step Δt. The minimum time steps both algorithms require to be stable are used. Section 8.3.2 gives details on the time step size.

The Pitt-Peters inflow model and the ship rotorcraft dynamic interface model described in this thesis use the same interface to exchange inflow and aerodynamic forces. Thus, the inflow velocities $\boldsymbol{v}$ for given flight states and aerodynamic forces can be compared directly. Chapter 8 uses this extensively to verify the LRM.

6.6 Blade element theory for rotorcraft flight dynamics

This thesis uses a mechanical model of a rigid body fuselage connected via hinges to rigid blades to model the rotorcraft flight dynamics. Integrating the accelerations resulting from the external forces and moments of all rigid bodies in time gives the rotorcraft flight path. The rotorcraft flight dynamic model is a simulation software produced by Airbus Helicopters [57]. The software allows real-time simulation of the rotorcraft and full, six degrees of freedom trim [57].

This thesis uses the MBB Bo 105 C with the aircraft registration D-HDDP used by the Deutsches Zentrum für Luft- und Raumfahrt (DLR) for research purposes [101]. It features rectangular blades with NACA 23012 profiles and a linear twist of $-6.2deg$ in the profiled section. Table 6.5 shows the inertia-related coefficients of the rotorcraft.

$Variable$	$Value$
m	$2200\,[kg]$
I^{xx}	$1433\,[kgm^2]$
I^{yy}	$4973\,[kgm^2]$
I^{zz}	$4099\,[kgm^2]$
I^{xy}	$0\,[kgm^2]$
I^{xz}	$660\,[kgm^2]$
x_{mr}^{cg}	$0.03\,[m]$
y_{mr}^{cg}	$0.0\,[m]$
z_{mr}^{cg}	$1.49\,[m]$

Table 6.5: Inertial properties of the MBB Bo 105 C with registration D-HDDP of the DLR.

Furthermore, a stationary polar models the drag the fuselage produces in forward flight. The rigid blades are divided into blade elements as the BEM is used to model the rotors. Nevertheless, this thesis does not provide details about the modeling of the tail rotor which the LRM does not represent. The tail rotor discretization uses the defaults given in the rotorcraft flight dynamics software.

Eight blade positions divide every main rotor blade into seven elements to discretize the main rotor in the BEM. Figure 6.8 depicts a scalable schematic of the MBB Bo 105 blade, including the blade elements. As the force distribution varies more in the outer sections of the rotor blade, the blade elements get finer towards the blade's tip. Neglecting the forces produced by the root cutout and the rotor hub, the first blade element starts at 0.223.

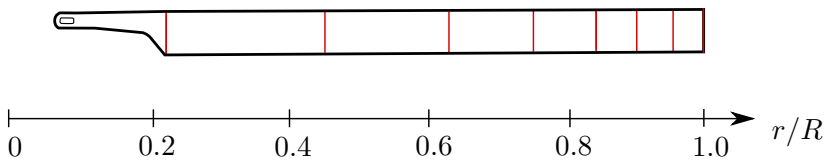


Figure 6.8: Schematic of the rotor blade in black with the blade positions used in the BEM code marked in red. Taken from [58].

At these blade elements, the produced forces result from the local angle of attack and prescribed airfoil polars of the NACA 23012 profile. The angle of attack results from the rotor inflow velocity, the local blade angles β, ζ, and ψ, and the absolute velocity of the blade element. With this information, the local lift-, drag-, and moment coefficients are interpolated from the airfoil polars and multiplied by the local cord length of the blade. Integrating the resulting forces and moments in the radial direction gives the blade forces and moments. Using a Forward-Euler method, the individual blade motion results from integrating the blade forces and moments in Newton's second law in time. Furthermore, summing over all four blades of the MBB Bo 105 calculates the total aerodynamic forces and moments the main rotor produces. Integrating the resulting rotor forces and moments using a 4th-order Runge-Kutta scheme calculates the fuselage motion in time.

Nevertheless, the rotor's net forces and moments consist of aerodynamic and inertial forces. As the LRM is simulating the fluid flow through and around the rotors, the interface described in section 6.5 only exchanges the aerodynamic forces [1]. This interface replaces the algebraic inflow calculation using the Pitt-Peters model [7] in the rotorcraft flight dynamics model. Furthermore, this reiterates that the inflow transports the interaction of the rotor wake with the surrounding objects to the rotors, where resulting forces at the blades move the rotorcraft and, ultimately, the pilot.

6.7 Overset simulation of NatoGD and LRM domain as numerical reference

The LRM developed in this thesis decouples the simulation of the NatoGD wake from the LRM domain surrounding the rotor for computational efficiency. Unfortunately, no measurement data for a MBB Bo 105 approach on a NatoGD is available to validate the results of the LRM. Therefore, the LRM can only be compared to another numerical solution to the ship-rotorcraft dynamic interface.

The overset technique described in section 6.1 allows the computing of the ship-rotorcraft interaction in one simultaneous fluid simulation. Therefore, the overset simulation can be used as a reference solution to compare the LRM results.

Therefore, the same full-scale simulation of the NatoGD, as described in section 6.2, is combined with a rotorcraft domain with the same shape, dimensions, resolution, and reach factor a as the LRM domain using the overset technique. Nevertheless, the rotorcraft domain differs from the LRM domain as it does not require the in- and outflow boundary conditions the LRM domain uses. The overset technique directly transports the fluid entering and exiting the rotorcraft domain to and from the NatoGD domain. Nevertheless, the rotorcraft domain uses the same interface to exchange forces and inflow at the rotor disk with the rotorcraft flight dynamics model. Furthermore, the rotorcraft flight dynamics model remains unchanged.

Therefore, comparing the solution using the overset mesh with the LRM solution allows assessing the error introduced by the LRM, which is the central model of this thesis. The overset technique eliminates the influence of the in- and outflow boundary conditions in the rotorcraft domain. It captures the entire interaction of the wake of the NatoGD and all influences of the rotorcraft. Thus, a comparison of an otherwise unchanged simulation setup allows to assess the errors introduced by modeling the entire ship rotorcraft interaction shown in Fig. 1.1, represented by the overset technique with the interaction shown in Fig. 4.2 representing the LRM.

7 Verification of the ship wake as Input Data

This chapter verifies that the requirements specified in section 2.2 get satisfied by the simulations of the NatoGD using the LBM solver with the domain setup described in section 6.2. Therefore, this chapter corresponds to the V-model's "Verification" stage for the ship wake simulation. In this thesis, verification represents proof of compliance with the requirements [102].

The concept of validation applies to all sub-components and their respective requirements of the developed model and to all software modified and extended during this process. As a complete description of all verification tests of the used LBM solver exceeds the scope of this document, this chapter focuses only on the verification of the top-level, abstract, physical requirements given in section 2.2. Nevertheless, Appendix III briefly describes the implemented unit tests to verify critical aspects of the developed LBM solver.

Although not listed in the requirements, section 7.1 assesses the grid convergence of the NatoGD simulation. As most numerical schemes discretize continuous equations, various errors are introduced [43, 48]. Nevertheless, the discrete equations equal the continuous ones at the limit of infinitely fine discretization. As computational resources are limited, most technical simulations need to balance computational expense and the accuracy of the results. Thus, grid convergence compares various simulations to identify the resolutions that meet this balance.

Nevertheless, convergent simulations can still fail to predict the correct physical behavior. Thus, this chapter compares the solutions to measurements of a 1/50th-scale model of the NatoGD in a wind tunnel. As the ship's motion influences the wake that develops, section 7.2 investigates this effect and compares it to measurement data. Based on this, the chapter verifies that the NatoGD simulations in this thesis fulfill the requirements in section 2.2.

7.1 Grid convergence of stationary NatoGD

In the experiments, the wake of the NatoGD is turbulent [12]. Therefore, the simulation of the NatoGD with LBM must be able to predict the turbulent nature of this wake. The LBM simulates the fluid at the level of a LES where turbulence requires a minimal resolution to develop [43]. Figure 7.1 shows an exemplary turbulent wake of the NatoGD simulation with $N^{NatoGD} = 364$. The figure proves that the LBM predicts a turbulent wake for the NatoGD in headwind with $\mathbf{u}^{\infty} = 9.77\frac{m}{s}$.

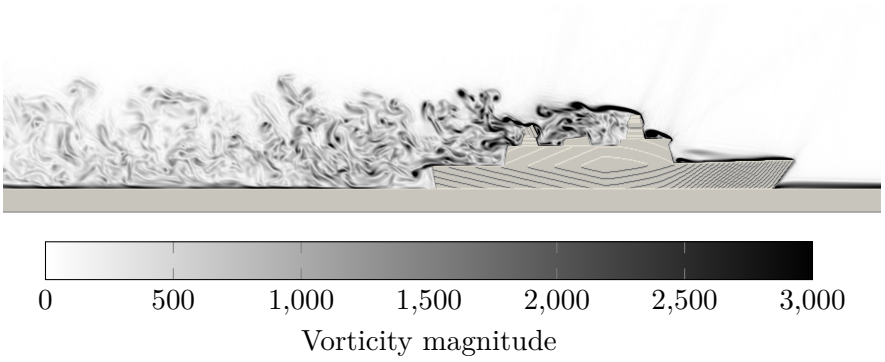


Figure 7.1: Turbulent wake highlighted by vorticity magnitude on ship centerslice. Computed using $N^{LRM} = 364$. Grey solids show the voxelized hull and ground.

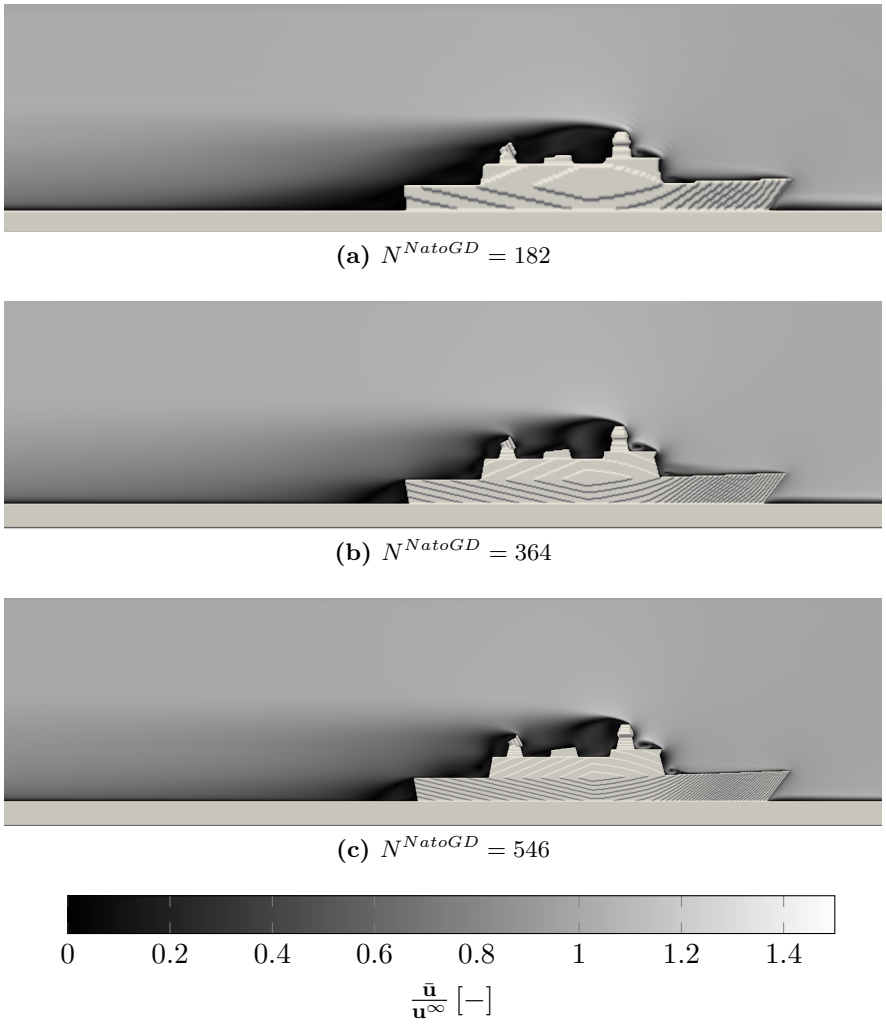
The vorticity indicates a turbulent wake developing behind the exhaust, radar, and hangar doors. Validating the turbulent wake of the NatoGD requires a convergence study and comparison to experimental data. Furthermore, simulating the flow around the NatoGD at different resolutions allows for comparing the mean and turbulent quantities and assessing the changes between the resolutions. The simulation reaches convergence if the evaluated quantities no longer change with the increase in resolution. Therefore, the 1/50-scale model NatoGD is simulated without ship motion at the resolutions $N^{NatoGD} = 182$, $N^{NatoGD} = 364$, and $N^{NatoGD} = 546$ cells per l^{NatoGD}.

Figure 7.2 shows the mean velocity $\bar{\mathbf{u}}$ divided by reference velocity $\mathbf{u}^{\infty} = 9.77\frac{m}{s}$ in the center slice of the NatoGD domain in greyscale. The subplots show increasing resolutions from top to bottom. $N^{NatoGD} = 182$ in (a), $N^{NatoGD} = 364$ in (b), and $N^{NatoGD} = 546$ in (c). $\bar{\mathbf{u}}$ represents the average of 10000 samples taken at $100Hz$ in $100s$ of simulation after $20s$ of start-up period. The velocity at the inflow boundary condition as measured in the case [12]:

`AirwakeData_Case01_CPship_x=+11p811_y=+0p000_z=+2p756_NoMotion_NRC433`

Comparing the mean velocity for the different resolutions shows a significant difference between $N^{NatoGD} = 182$ in (a), and $N^{NatoGD} = 364$ in (b). The flow separation areas behind any superstructure are shorter, as small mean velocities show. Furthermore, the separation areas are closer to the solid ship structure at higher resolution. Therefore, the resolution of $N^{NatoGD} = 182$ in (a) is considered too coarse to adequately capture the mean velocity of the NatoGD.

In contrast, the mean velocity shows only a minor difference between $N^{NatoGD} = 364$ in (b), and $N^{NatoGD} = 546$. The flow separation area behind the exhaust is slightly smaller, but the other separation areas remain unchanged when the resolution increases. Therefore, for the scope of this thesis, the mean velocity is considered converged at $N^{NatoGD} = 364$. Additionally, the simulation fulfills the requirement to correctly predict the separation areas behind the ship superstructure.



(a) $N^{NatoGD} = 182$

(b) $N^{NatoGD} = 364$

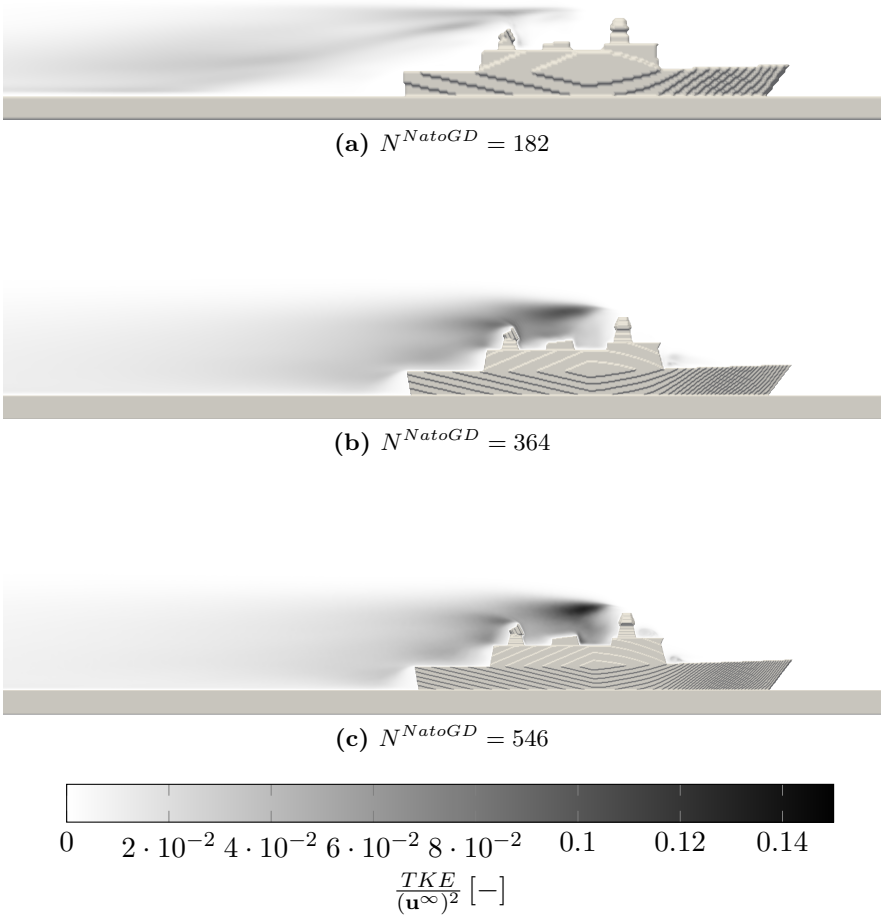
(c) $N^{NatoGD} = 546$

Figure 7.2: Mean velocity $\bar{\mathbf{u}}$ divided by reference velocity $\mathbf{u}^\infty$ for resolutions $N^{NatoGD} = 182$ in (a), $N^{NatoGD} = 364$ in (b), and $N^{NatoGD} = 546$ in (c). Grey solids show the voxelized hull and ground.

Nevertheless, the convergence of the mean velocity is insufficient to decide if a turbulent ship wake simulation is converged [30]. Therefore, the following investigates the convergence of the turbulent kinetic energy in the center slice of the NatoGD.

Figure 7.3 shows the turbulent kinetic energy TKE divided by square of reference velocity $(\mathbf{u}^\infty)^2$ at the center slice of the NatoGD domain for resolutions $N^{NatoGD} = 182$ in (a), $N^{NatoGD} = 364$ in (b), and $N^{NatoGD} = 546$ in (c). $\bar{\mathbf{u}}$ and TKE calulcated from the same case and samples.

As seen for the mean velocity, the simulation with $N^{NatoGD} = 182$ in (a) differs significantly from $N^{NatoGD} = 364$ in (b). The shear layers between the separation areas and the free flow around the ship produce the observed TKE. Nevertheless, as



(a) $N^{NatoGD} = 182$

(b) $N^{NatoGD} = 364$

(c) $N^{NatoGD} = 546$

Figure 7.3: Turbulent kinetic energy TKE divided by square of reference velocity $(\mathbf{u}^\infty)^2$ for resolutions $N^{NatoGD} = 182$ in (a), $N^{NatoGD} = 364$ in (b), and $N^{NatoGD} = 546$ in (c). Grey solids show the voxelized hull and ground.

the resolution $N^{NatoGD} = 182$ in (a) fails to predict the correct separation areas, the areas of TKE production do not get predicted well either. Therefore, the turbulent wake in the simulation with $N^{NatoGD} = 182$ in (a) does not fulfill the requirement to predict the turbulent nature of the NatoGD wake.

Comparing the distribution of TKE between $N^{NatoGD} = 364$ in (b), and $N^{NatoGD} = 546$ in (c) shows few variations. Apart from the higher TKE in the wake behind the exhaust in the simulation with $N^{NatoGD} = 546$ in (c), the TKE is similarly distributed in the simulation with the resolution $N^{NatoGD} = 364$ in (b). The TKE distribution is similar for the wake behind the hangar doors, radar, and landing deck. This region is particularly important for ship-rotorcraft dynamic interface models as it is close to the

landing spot of the rotorcraft. Unfortunately, no experimental flow measurements of the center plane of the NatoGD exist. Therefore, the absolute values of the TKE can not be verified. Although the TKE is not considered fully converged with a resolution of $N^{NatoGD} = 364$, it is considered sufficient to fulfill the requirement of capturing the turbulent nature of the ship wake for the scope of this thesis. An increase in resolution does not change the TKE present in the area around the landing spot and in the far wake of the ship, which are the areas the rotorcraft passes during take-off and landing. Therefore, increasing the resolution is considered to not improve the results for the ship wake.

Nevertheless, the turbulent characteristics can differ even if the TKE calculated in a simulation agrees with the TKE in an experiment. By definition, $\bar{\mathbf{u}}$ and TKE assess the mean and standard deviation of the fluid velocity. However, this neglects the shape of the velocity distribution curve, as different distributions can feature the same mean and standard deviation. As the ship wake develops, it transitions into isotropic turbulence and features normally distributed velocity [55]. Therefore, assessing mean and standard deviation are sufficient in the far wake. Nevertheless, in the regions close to the ship structure, the turbulence had no time to significantly decay into isotropic turbulence. Therefore, the velocity is not normally distributed in frequency and time domain as shown by the evaluation of experimental measurements of NatoGD wake in [103].

In order to assess the turbulence close to the ship structure in the region around the landing spot, the velocity at two points on the landing deck of the NatoGD are investigated in detail. These points are located at $(0.3, 0, 0.07)[m]$ and $(0.3, 0.06, 0.07)[m]$ in the coordinate system 2 of Fig. 5.2. As these points are close to the ship's deck and inside the separation area behind the hangar, the velocity distribution at these points exhibits substantial deviation from a normal distribution. Therefore, they allow assessing if the simulation of the NatoGD at the converged resolution shows the correct ship wake. Figure 7.4 shows a schematic of the landing deck section of the NatoGD with the location of the two points highlighted in red.

At these points, ship-fixed COBRA-probes [104] measure the velocity during wind tunnel experiments of the 1/50th-scale model of the NatoGD [12]. Although the COBRA-probes allow to measure highly frequent turbulent wakes [104], the measurement is restricted to a 45-degree cone [104, 103]. Thus, to compare the measurements to the simulation, it is advised to restrict the simulation data to the same 45-degree cone [103].

Dividing the velocity space into cubic bins with $0.1[m/s]$ edge length allows comparing the velocity distribution inside the measurement cones of the COBRA-probes using histogram plots. Sampling for the $100s$ measurement period after $20s$ of the start-up period with a frequency of $1000Hz$ resulted in 100000 points in velocity space at the positions of the probes. The probability distribution of the velocity is calculated by counting the number of times a velocity measurement falls within each bin and dividing by the total number of velocities measured.

Figure 7.5 shows the resulting velocity distributions for the experimental measurements [12] in the first row, and the three resolutions $N^{NatoGD} = 182$, $N^{NatoGD} = 364$,

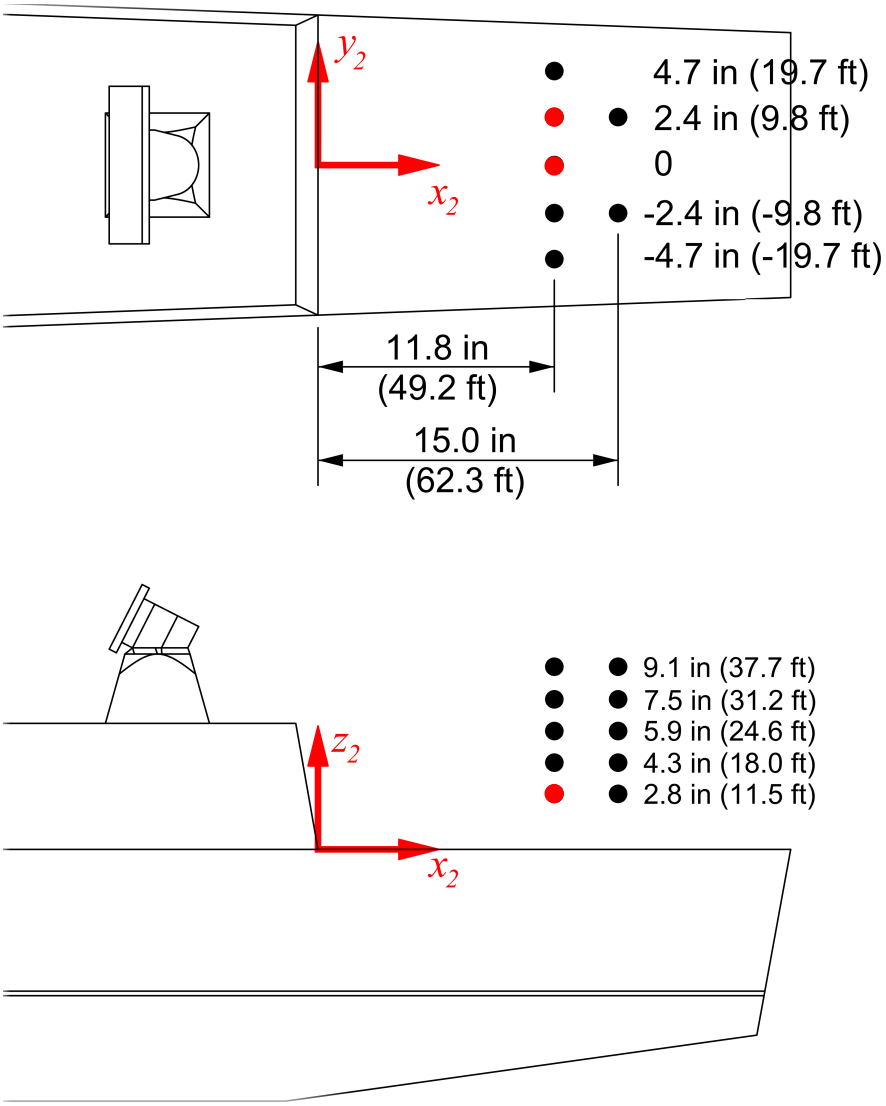


Figure 7.4: Schematic side and top view of the 1/50th-scale NatoGD model including the positions of COBRA-probes fixed to the ship represented as black dots. Red dots mark the points $(0.3, 0, 0.07)[m]$ and $(0.3, 0.06, 0.07)[m]$ investigated in detail in this thesis. Schematic adapted from [12].

and $N^{NatoGD} = 546$ in the second to last row. Red lines (- - -) indicate the 45-degree cone of the COBRA-probes. The mean of the velocity distribution is indicated by (○). Nevertheless, the marked mean values do not correspond to $\bar{\mathbf{u}}$ due to the cut-off of the probes.

The first row in Fig. 7.5 suggests that the measurement cone of the COBRA-probes misses a significant portion of the occurring velocities during the experiment. Furthermore, in the u_2 vs u_1 and u_3 vs u_1 plot, the probability of encountering velocities close

to **0** is small. As this pattern is persistent in all measurements investigated in this thesis, it suggests that the COBRA-probes are not suited for flows with velocities close to **0**.

The histograms also indicate that the negative u_3 and positive u_1 velocities tend to occur together, which the high probability within the lower half of the 45-degree cone in the u_3 vs u_1 plot indicates. Nevertheless, as most measurements are close to the cut-off of the cone, no statement about the flow can be inferred from the statistic.

The u_3 vs. u_2 plot shows a dominant pattern in the $u_3 < 0$ section, looking like an inverted V-shape. Although the plot's axis prevents drawing the measurement cone of the COBRA-probes, the registered values are still subject to the measurement range of the probes. Nevertheless, the registered values exhibit a substantial deviation from being normally distributed. Given the measurement values inside the 45-degree cone, normal distribution for the velocity at the point $(0.3, 0, 0.07)[m]$, located above the landing spot, seems unlikely. Unfortunately, the restriction of the COBRA-probes to the 45-degree cone prevents a complete statistical comparison.

Nevertheless, the measured data is sufficient to show that none of the simulations of the NatoGD captures the correct flow state at this point above the landing spot. In the simulation with $N^{NatoGD} = 182$ in the second row of Fig. 7.5, no velocity distribution of a turbulent flow is visible, as the resolution is insufficient to predict the correct flow separation. The failure to develop a turbulent wake was already indicated by the mean velocity in Fig. 7.2 and is confirmed here.

The resolutions $N^{NatoGD} = 364$, and $N^{NatoGD} = 546$, which showed converged values in $\bar{\mathbf{u}}$ and TKE for the center slice of the ship, also show converged behavior for the velocity distributions shown by the histograms in row three and four. Table 7.1 lists the velocity distributions' numerical mean, variance, and TKE.

$Variable$	Exp.	$N^{NatoGD} = 364$	$N^{NatoGD} = 546$
$\bar{u}_1\,[\frac{m}{s}]$	2.37	2.77	2.52
$\bar{u}_2\,[\frac{m}{s}]$	0.22	0.01	0.06
$\bar{u}_3\,[\frac{m}{s}]$	−1.10	−1.53	−1.33
$var(u_1)\,[\frac{m^2}{s^2}]$	1.20	1.18	1.01
$var(u_2)\,[\frac{m^2}{s^2}]$	1.39	1.69	1.40
$var(u_3)\,[\frac{m^2}{s^2}]$	1.33	1.48	1.45
$\frac{TKE}{(\boldsymbol{u}^\infty)^2}\,[-]$	0.021	0.023	0.020

Table 7.1: Mean, variance and TKE of the velocity distribution at point $(0.3, 0, 0.07)[m]$ for the experiment (Exp.) and simulations with the resolutions $N^{NatoGD} = 364$ and $N^{NatoGD} = 546$.

Comparing the numerical values for mean and variance indicates a good agreement of both resolutions $N^{NatoGD} = 364$ and $N^{NatoGD} = 546$. Nevertheless, the histogram plots in Fig. 7.5 indicate that the inverted V-shape in the u_3 vs. u_2 is not as prominent as in the experiment. Furthermore, no convergence of the velocity distribution towards the experiment is present in u_3 vs. u_2. The non-converging distributions indi-

cate that increasing resolution is not the best strategy to improve the agreement with experimental measurements. Increasing resolution towards a DNS or a wall model is expected to improve the agreement with the measured data. Nevertheless, a wall model is computationally less expensive [43]. Appendix I shows the evaluation for the second point at $(0.3, 0.06, 0.07)[m]$, which results in the same conclusions.

In general, the turbulent characteristics of the ship wake agree with the findings of Ashok [34]. Although their simulations used a shorter sampling period and neglected the velocity variation at the inflow boundary, the reported mean velocity and TKE are in reasonable agreement. Furthermore, their analysis indicates that their results converge for $N^{NatoGD} = 400$. The results in this thesis indicate that $N^{NatoGD} = 364$ is sufficient. Nevertheless, their simulation used local grid refinement around the NatoGD and immersed boundary conditions to represent the hull.

Concerning the requirements on the wake of a ship simulation set in section 2.2, the resolution $N^{NatoGD} = 364$ is sufficient for the thesis scope. It predicts the flow separation areas, and the turbulent nature of the wake. Furthermore, with a cell size $\Delta x = 150[m]/364 \approx 0.41m$ for the full-scale simulation, it features a small enough cell size to capture disturbances on the scales relevant for a manned, conventional rotorcraft.

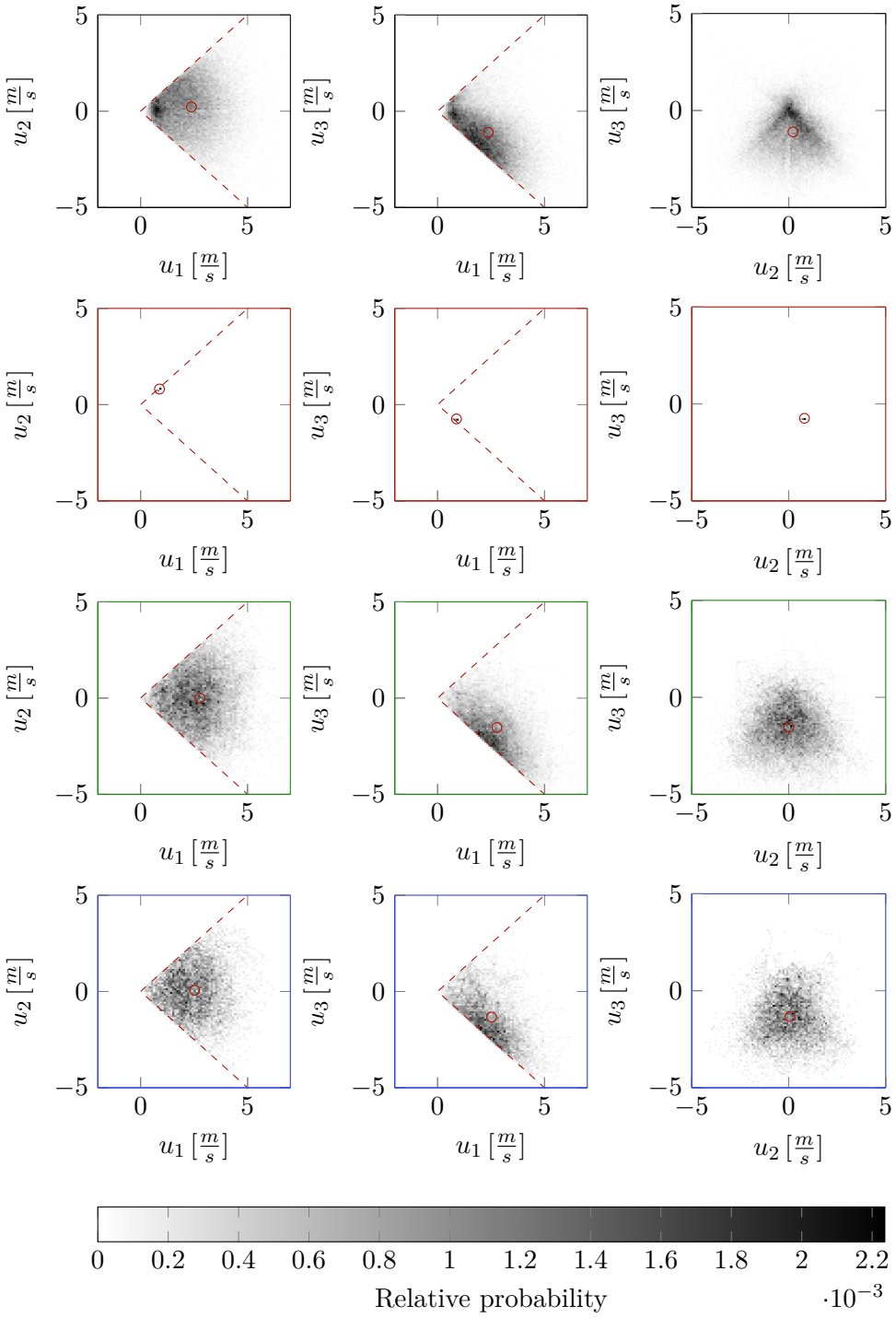


Figure 7.5: Relative probability in bins with a length of $0.1\frac{m}{s}$ for the point $(0.3, 0, 0.07)[m]$. Red lines (- - -) mark the cone the COBRA-probes are able to measure within [12, 103, 104]. The mean is indicated in (○) First row are measurements taken from [12], second to last row are calculations with resolutions $N^{NatoGD} = 182$, $N^{NatoGD} = 364$, and $N^{NatoGD} = 546$ from top to bottom.

7.2 NatoGD moving to sea state

The simulation should include the ship's motion because the ship moves at frequencies below $2Hz$, and the motion affects the turbulent spectra in the wake [34, 9]. Furthermore, this adds motion to the landing deck and hangar relative to the rotorcraft. This section compares the sinusoidal pitching given in [12] to measurements to verify the influence of the ship motion on the ship wake. Sinusoidal heaving is included in Appendix II as it does not provide further insights.

The simulation for the pitching 1/50th-scale NatoGD model uses the ALE formulation described in section 6.1.4 to allow the ship's motion. The ship hull is fixed to the domain described in section 6.2 and moved according to the prescribed data [12]. Furthermore, the resolution is set to $N^{NatoGD} = 364$ as the non-moving simulations found it to be converged. The velocities at the inflow boundary, the motion profile, and the measurement values for comparison result from the measurement [12]:

`AirwakeData_Case06_CPship_x=+11p811_y=+0p000_z=+2p756_PitchOnly_NRC436`

Ensemble averaging the results allows evaluating the wake at different points of the sinusoidal pitching motion with a $5deg$ amplitude. The ensemble average is required to statistically investigate the turbulence [55]. Therefore, during the $100s$ simulation time after $20s$ start-up, samples are taken with $1Hz$ at four points of the sine prescribing the motion. Thus, averaging these four sets of 100 samples at $1Hz$ with the same ship position and attitude forms four ensembles. The four points are $t = 0s$, $t = 0.25s$, $t = 0.5s$, and $t = 0.75s$ of the sinusoidal pitching. Therefore, the points correspond to the maximum hull pitch rate up at $t = 0s$, maximal hull pitch angle upwards at $t = 0.25s$, maximum hull pitch rate down at $t = 0.5s$, and maximal hull pitch angle downwards at $t = 0.75s$. Figure 7.6 shows the difference in mean velocity $\bar{\boldsymbol{u}}$ to the ensemble average of the non-moving case $\bar{\boldsymbol{u}}^{static}$ divided by reference velocity $\boldsymbol{u}^{\infty} = 9.77\frac{m}{s}$. $t = 0s$ is shown in 7.6a, $t = 0.25s$ is shown in 7.6b, $t = 0.5s$ is shown in 7.6c, and $t = 0.75s$ is shown in 7.6d.

As the plot shows $\bar{\boldsymbol{u}} - \bar{\boldsymbol{u}}^{static}$ in the center plane of the NatoGD domain in greyscale, lighter areas correspond to higher mean velocity compared to the non-moving case. In comparison, darker areas indicate regions with lower mean velocity. As the domain is fixed to the ship and moved via the ALE method, the white area underneath the waterline in front of the bow in Fig. 7.6b and behind the bow in Fig. 7.6d originates from $\bar{\boldsymbol{u}} = \boldsymbol{0}$ in the ground for the non-moving case. Inside these areas, the plot is invalid. Furthermore, the far wake behind the NatoGD features a fine-grained structure, which originates from the low sample count of 100 samples possible with the $120s$ long measurement period [12].

Starting at $t = 0s$ in 7.6a, the ship pitches up at maximum rotational speed. The pitch-up causes the wake behind the ship superstructure to slow down. This deceleration reaches the maximum as the ship reaches the maximum pitch angle of $5deg$ in Fig. 7.6b. The largest deceleration occurs in the shear layer behind the exhaust and at the stern part of the flight deck. Thus, a rotorcraft flying above the landing deck would expect slower wind speeds while the ship pitches upwards.

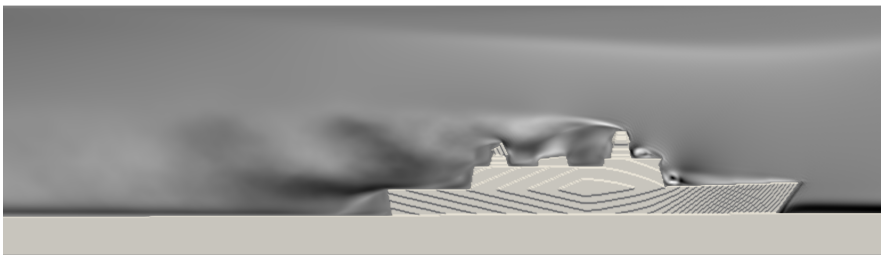
When pitching down after the maximum pitch angle, the flow at the bow starts to separate, as can be seen by the large area at low velocity at the bow in Fig. 7.6c. This separation continues to be present even at maximum pitch angle downwards in Fig. 7.6d. Nevertheless, in contrast to the decelerating velocity at the bow, the velocity at the stern section of the ship increases during the pitch down. Like the pitch-up phase, the largest velocity changes occur in the shear layer behind the exhaust. Thus, a rotorcraft hovering above the landing deck would expect faster wind speeds when the ship pitches downwards.

Overall, the pattern emerging in Fig. 7.6 indicates a variation of the ship wake with the frequency of the ship's motion. Both Owen [9] and Ashok [34] find that the $1Hz$ heaving and pitching motion add energy at $1Hz$ in the power spectral density in their simulations. Nevertheless, the general turbulent wake of the 1/50th-scale NatoGD remains largely unaffected by the ship motion [34].

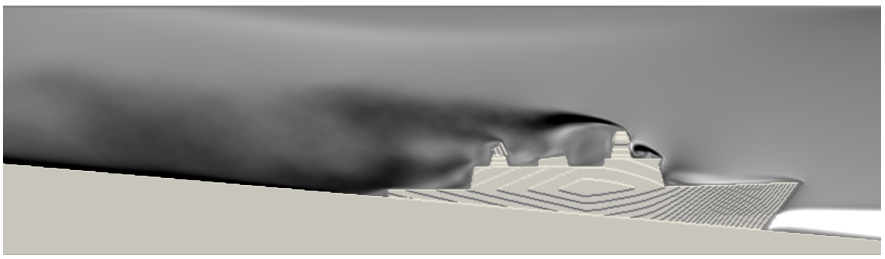
Comparing the TKE of the four ensembles to the TKE^{static} of the non-moving NatoGD simulation allows investigating the influence of the motion on the wake. Figure 7.7 depicts this difference in the center slice of the NatoGD domain. $t = 0s$ is shown in 7.7a, $t = 0.25s$ is shown in 7.7b, $t = 0.5s$ is shown in 7.7c, and $t = 0.75s$ is shown in 7.7d. Lighter grey represents a decrease in TKE, and darker grey represents an increase.

In general, the influence of the ship motion on the TKE is mainly limited to the area behind the exhaust and above the landing deck in the stern of the NatoGD. In the far wake behind the landing deck of the ship, the difference between the moving and the non-moving simulation quickly diminishes. The visible fine-grained structure in the far wake in Fig. 7.7 is attributed to the small sample size of 100 being too small. Furthermore, the main differences of TKE to the static case lie behind the exhaust. Therefore, they are irrelevant for ship-rotorcraft interaction as regular rotorcraft flight paths do not cross this section.

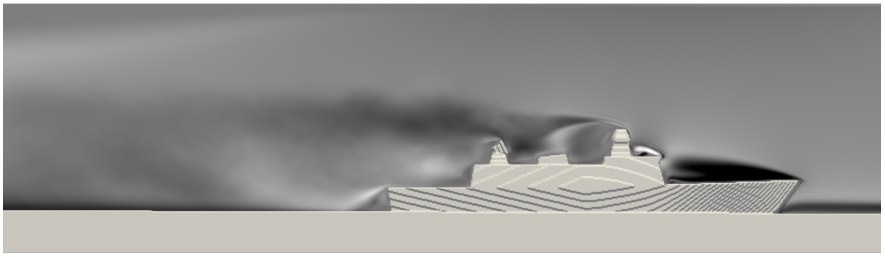
Nevertheless, differences in TKE due to the pitching motion of the ship exist. The contour plots indicate that the motion redistributes some TKE inside the area behind the exhaust and above the landing deck. However, the changes are not substantial enough to change the general distribution of turbulence in the wake of the NatoGD. Ashok formulated a similar postulate based on the power spectral density of the vertical velocity at a point $0.1904m$ above the landing deck of moving 1/50th-scale NatoGD simulations [34].



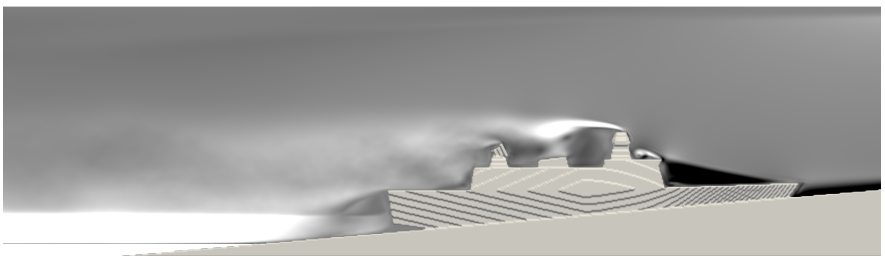
(a) $t = 0$



(b) $t = 0.25$



(c) $t = 0.5$



(d) $t = 0.75$

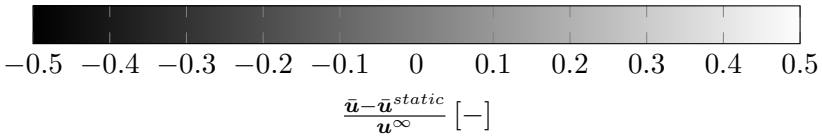
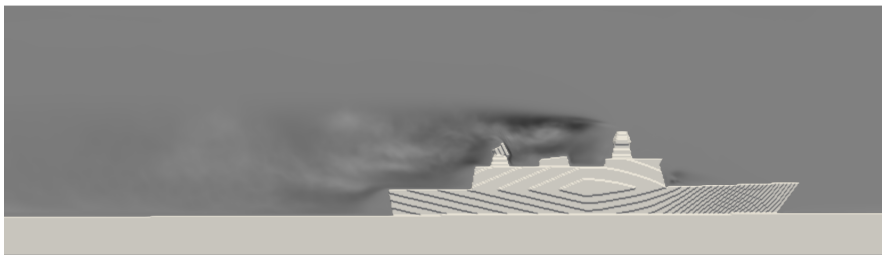
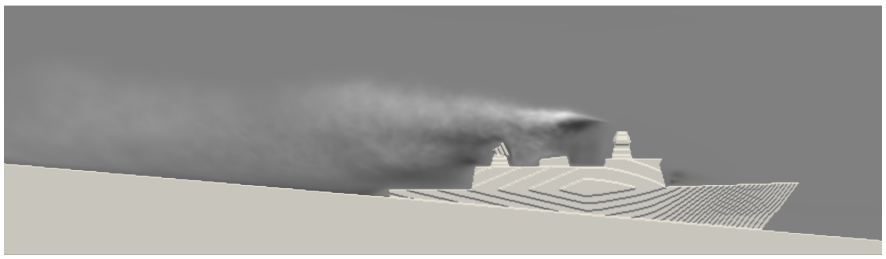


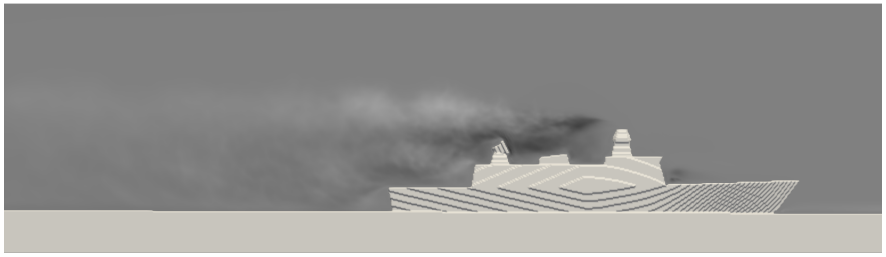
Figure 7.6: Difference of mean velocity $\bar{\boldsymbol{u}}$ to static case $\bar{\boldsymbol{u}}^{static}$ divided by reference velocity $\boldsymbol{u}^{\infty}$ for resolutions $N^{NatoGD} = 364$. 1/50th-scale model pitching at a sine wave with $1Hz$ and an amplitude of $5.0deg$ [12]. (a) shows ensemble average at $t = 0s$, (b) at $t = 0.25s$, (c) at $t = 0.5s$, and (d) at $t = 0.75s$. Grey solids represent the voxelized hull and ground.



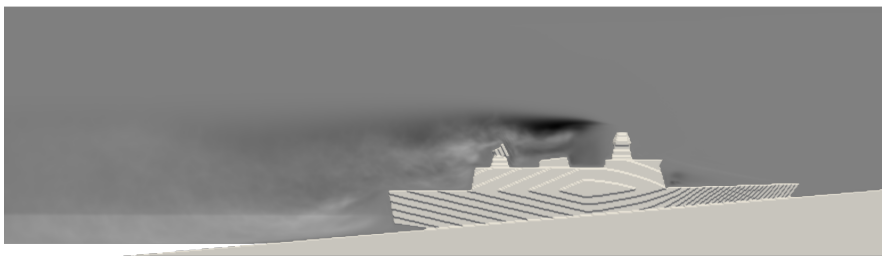
(a) $t = 0$



(b) $t = 0.25$



(c) $t = 0.5$



(d) $t = 0.75$

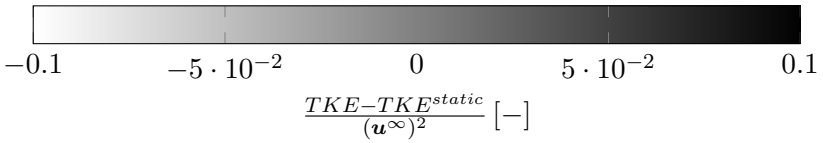


Figure 7.7: Difference of turbulent kinetic energy TKE to static case TKE^{static} divided by reference velocity $\boldsymbol{u}^{\infty 2}$ for resolutions $N^{ship} = 364$. 1/50 scale model pitching at a sine wave with $1Hz$ and an amplitude of $5.0deg$ [12]. (a) shows ensemble average at $t = 0s$, (b) at $t = 0.25s$, (c) at $t = 0.5s$, and (d) at $t = 0.75s$. Grey solids represent the voxelized hull and ground.

The velocity distributions are compared to experimental measurements to investigate the postulate further. Figure 7.8 shows the velocity distribution of the experiment in the first row and the second row of the NatoGD simulation with $N^{ship} = 364$. The distributions result from the velocities at point $(0.3, 0, 0.07)[m]$ and represent a sinusoidal pitching ship with a frequency of $1Hz$ and $5deg$ amplitude. Comparison of the experimental data in the first row with the non-moving simulation in 7.5 indicates that the pitching motion does not influence the shape of the velocity distribution. Nevertheless, comparing the mean, variance, and TKE shown in table 7.2 indicates minor differences in the velocity variances. Therefore, the postulate that the ship's motion does not significantly alter the turbulent characteristic of the wake is supported.

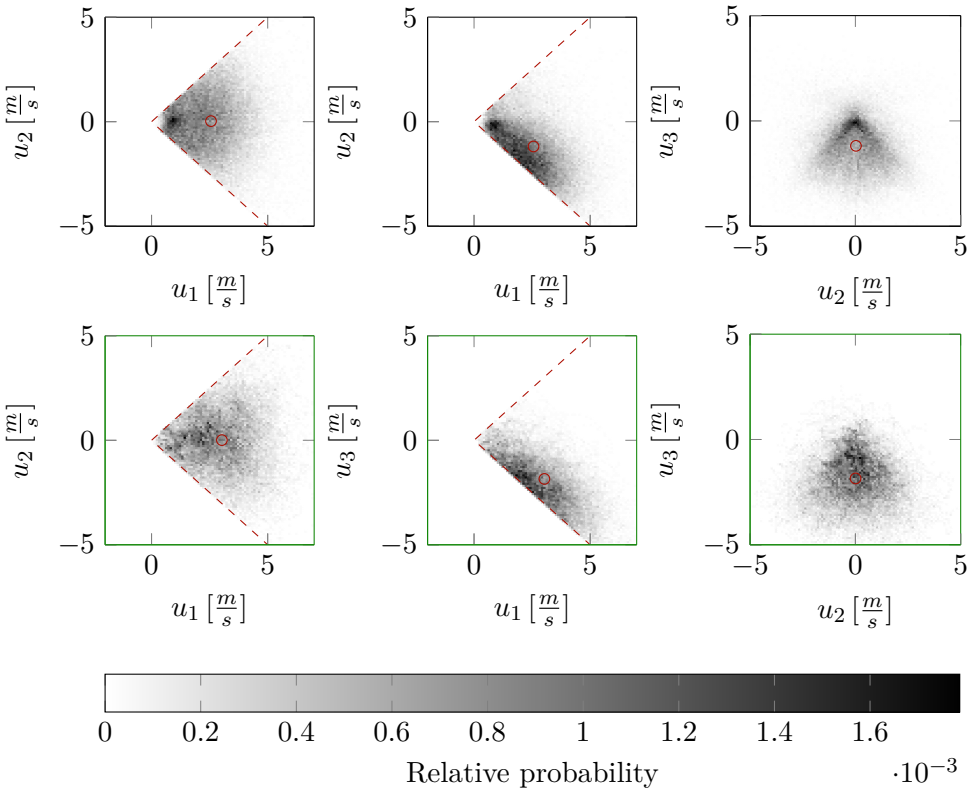


Figure 7.8: Relative probability in bins with a length of $0.1\frac{m}{s}$ for the point $(0.3, 0, 0.07)[m]$ for a 1/50th-scale model pitching at $1Hz$ with an amplitude of $5deg$. Red lines (- - -) mark the cone the COBRA-probes are able to measure within [12, 103, 104]. The mean is indicated in (○) First row are measurements taken from [12], second row shows calculations with resolutions $N^{NatoGD} = 364$.

As in the non-moving case, the velocity distributions of the simulation and the experiment are in good agreement. The same holds for the numerical values for mean and TKE. Furthermore, the velocity distribution at the point $(0.3, 0.06, 0.07)[m]$ shown in Appendix I and the simulations at sinusoidal heave shown in appendix II support the assessment.

$Variable$	Exp.	$N^{NatoGD} = 364$
$\bar{u}_1 \left[\frac{m}{s}\right]$	2.56	3.03
$\bar{u}_2 \left[\frac{m}{s}\right]$	0.02	0.08
$\bar{u}_3 \left[\frac{m}{s}\right]$	-1.18	-1.85
$var(u_1) \left[\frac{m^2}{s^2}\right]$	1.30	1.62
$var(u_2) \left[\frac{m^2}{s^2}\right]$	1.57	1.73
$var(u_3) \left[\frac{m^2}{s^2}\right]$	1.39	1.51
$\frac{TKE}{(\boldsymbol{u}^\infty)^2} [-]$	0.022	0.025

Table 7.2: Mean, variance and TKE of the velocity distribution at point $(0.3, 0, 0.07)[m]$ for the experiment (Exp.) and simulations with the resolutions $N^{NatoGD} = 364$ for a pitching motion with $1Hz$ and $5deg$ amplitude.

In total, the wake simulations of the NatoGD using LBM at a resolution of $N^{NatoGD} = 364$ fulfill the requirements listed in section 2.2. The solver predicts the flow separations behind the ship superstructure and captures the turbulent ship wake. As a cell size of $\Delta x = 0.41m$ in a full-scale NatoGD simulation is one order of magnitude smaller than the rotor radius of a MBB Bo 105, the solver should predict the turbulent spectra that induce disturbances of $2Hz$ and lower. By incorporating the ship motion, the LBM solver includes the low-frequency change in the mean velocity of the wake introduced by the ship motion. Nevertheless, the motion only slightly affects the turbulence in the wake. Due to the high inertia, rotorcraft should only be affected by a change in mean velocity and not by redistribution of TKE in the areas close to the ship.

8 Verification of the LRM Implementation

Analogous to chapter 7, this chapter verifies that the requirements specified in chapter 2 get fulfilled by the LRM domain set up as described in section 6.4. Therefore, this chapter describes the V-model's "Verification" step for the LRM simulation. Appendix III describes low-level source code verification tests.

First, section 8.1 investigates the convergence of the LBM solver used to simulate the LRM domain. Therefore, N^{LRM} and the reach factor a vary. With convergence established, section 8.2 verifies the physical requirements specified in section 2.1. Although previous chapters assess convergence, this section includes results for several resolutions N^{LRM}. Comparing the results at different resolutions verifies that convergence persists when the LRM domain interfaces to the rotorcraft flight dynamics model. Finally, section 8.3 verifies that the LRM fulfills the computational requirements given in section 2.3.

8.1 Grid convergence of isolated rotor in hover and forward flight

To investigate the convergence behavior of the fluid simulation inside the LRM domain, an isolated rotor is simulated over a range of forward flight speeds. Otherwise, the simulation uses the parameters specified in section 6.4. The rotor is flying with an angle of attack $\alpha = 0$ outside of ground effect in the velocity range of hover at $u^{mr}_{1,cg} = 0$ up to $u^{mr}_{1,cg} = 65[\frac{m}{s}]$. As the Pitt-Peters inflow model [7] calculates the flow through an isolated disk that applies a z-force to the fluid and is intensively validated [21], it acts as a reference for the convergence study.

The rotor interface, described in section 6.5, connects the LRM and rotorcraft flight dynamics simulation. As the inflow it discretizes transports the flow information from the surrounding into the rotor, the inflow velocities $\boldsymbol{v}$ for a given thrust distribution form a reasonable criterion to assess convergence.

Therefore, a uniform force of $F^{mr}_{mr,3} = 24525N$, which corresponds to the Maximum Takeoff Weight (MTOW) of a MBB Bo 105 CBS-5, is prescribed over the rotor in the LRM domain and to the Pitt-Peters inflow model. As the main direction of the rotor force is in z_{mr}-direction for a rotor flying at $\alpha = 0$, the main focus is on the inflow components v_3. Furthermore, the Pitt-Peters inflow model used as reference only computes the inflow v_3 in z_{mr}-direction [7].

Comparing $\bar{v}_3$ to the reference model assesses if the LRM can produce the proper inflow for a given thrust. Furthermore, for the LRM to be usable in piloted flight simulations, the cosine part v_3^{cosine} of the inflow is especially important. It has to be regulated with a control input by the pilot to maintain stationary forward flight in a conventional rotorcraft [6, 2]. Thus, the convergence of the LRM is assessed by $\bar{v}_3$ and v_3^{cosine}.

Figure 8.1 plots both $\bar{v}_3$ and v_3^{cosine} over forward flight speed $u_{1,cg}^{mr}$ for different resolutions N^{LRM} and with a constant reach factor $a = 1.25$. For comparability, both are non-dimensionalized using the mean induced z-velocity of the reference model $\bar{v}_3^{h,pp} \approx 11.62[\frac{m}{s}]$. Furthermore, Fig. 8.1 includes the corresponding inflow velocities of the Pitt-Peters.

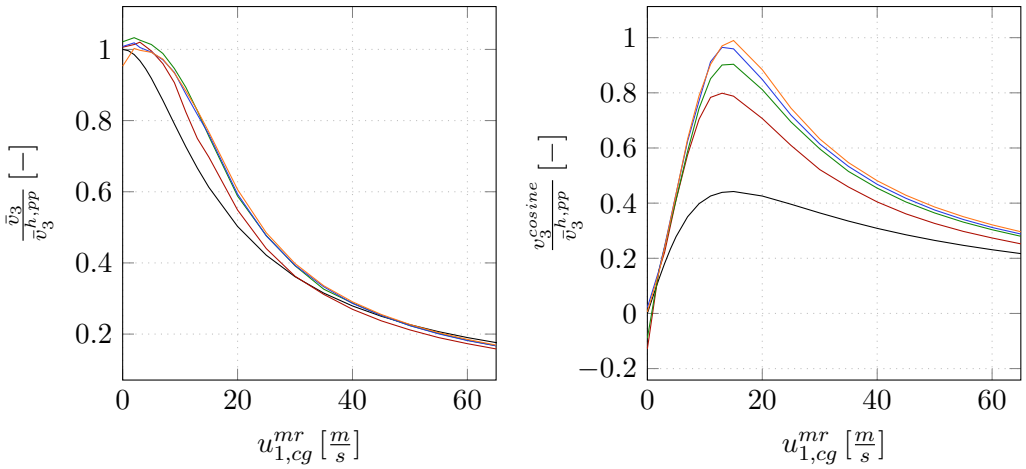


Figure 8.1: Influence of domain resolution on the mean $\bar{v}_3$ and cosine v_3^{cosine} component of the induced velocity. Calculated at a domain size of $a = 1.25$. Resolution $N^{LRM} = 32$ (—), $N^{LRM} = 64$ (—), $N^{LRM} = 128$ (—), and $N^{LRM} = 256$ (—). Pitt-Peters solution for comparison (—).

The left plot in Fig. 8.1 depicts $\bar{v}_3$ diminishing with forward flight speed. The decrease originates from the increased mass flow through the rotor in forward flight [20, 19]. The constant force results in less fluid acceleration for the larger mass flow passing through the rotor disk. All resolutions of the LRM correctly capture this effect. Nevertheless, the LRM calculates slightly higher $\bar{v}_3$ than the Pitt-Peters reference model at low speeds $u_{1,cg}^{mr} < 30[\frac{m}{s}]$. Furthermore, the LRM does not converge for speed lower than $u_{1,cg}^{mr} < 3[\frac{m}{s}]$. This is especially pronounced for the hover case at $u_{1,cg}^{mr} = 0[\frac{m}{s}]$. An enlarged version of the low-velocity region of the plot is included in Appendix IV.

In contrast, the cosine component of the inflow v_3^{cosine} in the right plot of Fig. 8.1 shows the same shape but large differences in value compared to the Pitt-Peters reference. This difference is greatest at the peak at $u_{1,cg}^{mr} \approx 15[\frac{m}{s}]$. Nevertheless, the position of the peak was in the expected velocity range below $2\bar{v}_3^h$ [26]. Generally, the v_3^{cosine} is larger than the reference. Nevertheless, the Pitt-Peters reference model modified the homogeneous force distribution over the disk. In comparison, flight tests gave better

results for an elliptical lift distribution in the radial coordinate [7] although not representing the actual thrust distribution. Although the assumption of lift distribution in the Pitt-Peters model does not explain the difference between the LRM and the reference model, prior versions of the LRM showed that a higher v_3^{cosine} in the LRM might give better correlations with flight test data for the LRM [2]. Nevertheless, this investigation used a different LBM fluid solver and compared the inflow velocities in trimmed forward flight of a complete MBB Bo 105 instead of an isolated rotor, which results in additional influencing factors like the resulting attitudes. The differences of v_3^{cosine} in free flight will be further discussed in section 8.2.1. Independent of the value of v_3^{cosine} the LRM converges for N^{LRM} above 128.

Furthermore, similar to $\bar{v}_3$ no convergence is found for $u_{1,cg}^{mr} < 3[\frac{m}{s}]$. Appendix IV includes an enlarged version of the low-velocity region. Especially in hover, the LRM computes negative v_3^{cosine}. The impedance boundary conditions described in section 6.1 might cause the negative v_3^{cosine}, as the boundary fails to handle large gradients. By accelerating the fluid, the rotor disk produces a free jet, which is fastest during hover. Furthermore, the free jet incites perpendicular to the impedance boundary below the rotor. Therefore, the fluid velocity gradient at the free jet's boundary is largest in hover. Forward flight reduces the incident angle onto the boundary.

In summary, the LRM shows convergent behavior with resolutions $N^{LRM} \geq 128$ for both $\bar{v}_3$ and v_3^{cosine}. Furthermore, the resolution $N^{LRM} = 64$ represents a reasonable choice concerning computational efficiency. On the one hand, the computational load of the LRM scales $\propto N^4$ due to the scaling behavior of the LBM solver [1]. On the other hand, the $N^{LRM} = 64$ converges to the v_3^{cosine} of higher resolutions except at the peak at $u_{1,cg}^{mr} \approx 15[\frac{m}{s}]$, thus representing a reasonable compromise.

The following investigates the convergence of the reach factor a. Figure 8.2 plots $\bar{v}_3$ and v_3^{cosine} over the forward flight velocity for $N^{LRM} = 64$ while varying the reach factor a. The plots include the reach factors $a = 1.0$, $a = 1.25$, $a = 1.5$, and $a = 2.0$.

The mean inflow velocity $\bar{v}_3$ shown in the left plot of Fig. 8.2 indicates that the LRM domain converges for all reach factors a at the resolution $N^{LRM} = 64$. The convergence of $\bar{v}_3$ concerning a indicates that the previously discovered convergence problems at low forward flight speed $u_{1,cg}^{mr} < 3[\frac{m}{s}]$ depend on the resolution. In contrast, v_3^{cosine} shows a convergence with increasing a for low forward flight speeds $u_{1,cg}^{mr} < 3[\frac{m}{s}]$. $a = 2$ even results in $v_3^{cosine} = 0$ in hover, which matches the Pitt-Peters reference, as seen from the enlarged plot in Appendix IV.

To this point, the LRM shows convergent behavior with a reach factor of $a = 2$. Nevertheless, as the computational expense scales $\propto a^3$, smaller reach factors represent reasonable choices due to the lower computational expense without introducing substantial errors.

As the LRM targets ship-rotorcraft interaction, the wake of the rotor inside the domain is essential. For a uniformly loaded rotor disk, this wake forms a similar shape to the wake of a fixed-wing but with increased thickness [105]. Figure 8.3 sketches this wake.

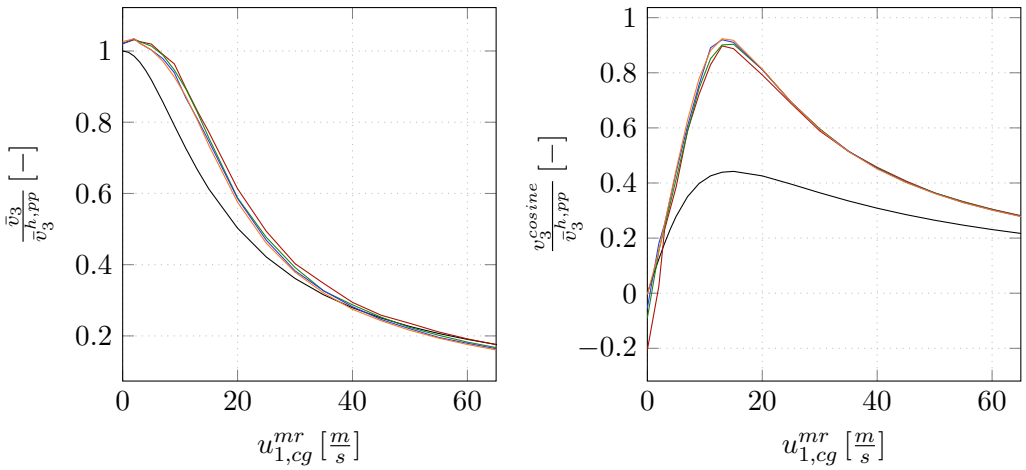


Figure 8.2: Influence of domain size on the mean $\bar{v}_3$ and cosine v_3^{cosine} component of the induced velocity. Calculated at a resolution of $N^{LRM} = 64$. Domain reach factors $a = 1$ (—), $a = 1.25$ (—), $a = 1.5$ (—), and $a = 2$ (—). Pitt-Peters solution for comparison (—).

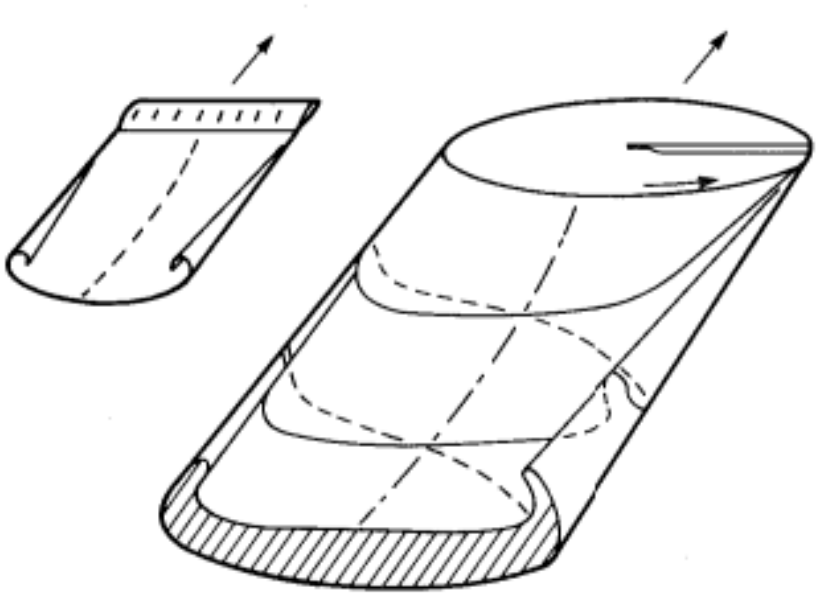


Figure 8.3: Schematic view of the wake of a rotor (right) and fixed-wing with a low aspect ratio (left), taken from [105].

In contrast, Fig. 8.4 displays the wake of an isolated rotor in the LRM domain. A vorticity isosurface indicates the wake. Comparing the schematic in Fig. 8.3 with the wake in the LRM domain in Fig. 8.4 indicates that the LRM domain can predict

the wake shape typical for homogeneously loaded rotor disks. Nevertheless, further verification with velocity data of rotor wakes in various flight conditions is necessary to assess the quality of the rotor wake prediction by the LRM.

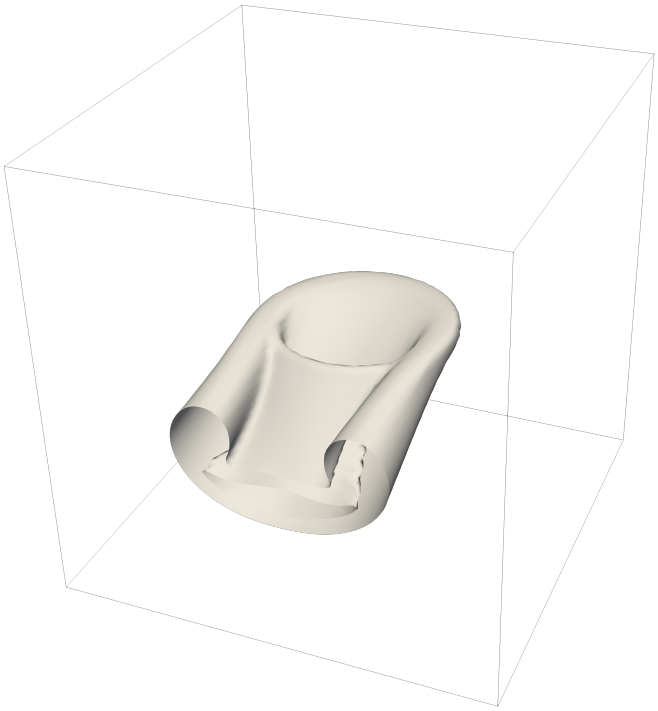


Figure 8.4: Schematic view of the wake of a rotor inside the LRM domain with $N^{LRM} = 64$ and $a = 1.25$. Rotor wake indicated by vorticity isosurface.

For the scope of this thesis, the LRM domain as described in 6.4 is sufficiently converged with a resolution $N^{LRM} = 64$ and a reach factor of $a = 1.25$. Although increasing the resolution or the reach factor improves convergence for low forward flight speeds, doing so increases the computational cost substantially. Furthermore, the prediction of v_3^{cosine}, which showed significant differences to the reference model, did not improve with an increase in either of the two parameters.

8.2 Physical Requirements

This section verifies that the LRM can represent the physical phenomena specified in the requirements section 2.1. On success, the LRM captures the relevant phenomena and thus qualifies as a model for the ship-rotorcraft dynamic interface. Thus, the LRM domain interfaces with the rotorcraft flight dynamics model as described in section 5. As section 8.1 did not find substantial influence, the reach factor a is set to 1.25 unless otherwise specified in the following.

The sections 8.2.1 and 8.2.2 are verifying the static and dynamic behavior of the MBB Bo 105 C of the DLR in still air to verify that the LRM can predict rotorcraft

flight outside of the wake of a ship. Therefore, section 8.2.1 compares the control inputs and power for the MBB Bo 105 C of the DLR in stationary forward flight. With the stationary values verified, section 8.2.2 assesses the model's capability to predict the dynamic reaction of the model to doublets input by the pilot.

Section 8.2.3 validates the ability of the LRM to represent the ground effect in hover and forward flight. Finally, section 8.2.4 verifies the remaining requirement regarding wall effects and recirculation of the rotor wake.

8.2.1 Control and Power in Stationary Forward Flight

In order to assess that the LRM coupled with the rotorcraft flight dynamics model predicts the correct flight performance, the main rotor power P^{mr}, attitudes, and control inputs for stationary forward flight are compared to flight test data provided by the DLR. The rotorcraft flight dynamics model uses the parameters given in section 6.6, which corresponds to the MBB Bo 105 configuration of the flight test.

To start the comparison, Fig. 8.5 plots the total rotor power P^{mr} over stationary, trimmed forward flight. It shows the flight test data and the power calculated by the rotorcraft flight dynamics model with either the LRM at various resolutions and $a = 1.25$ or the Pitt-Peters model to calculate the inflow. In order to compare the simulation data calculated at the density given in section 6.4 with the flight test data at various densities, a density correction is applied [106].

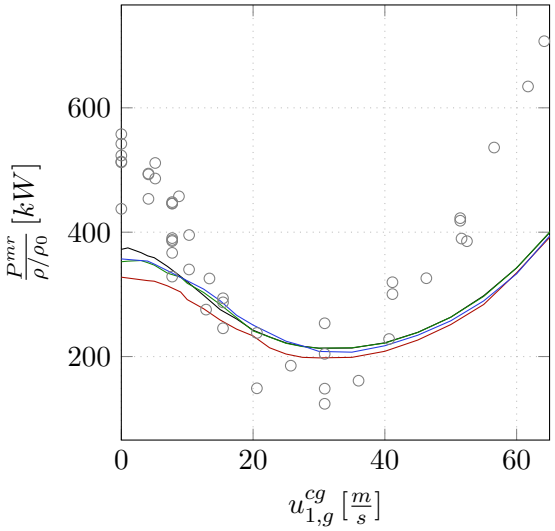


Figure 8.5: Total power required by main rotor over forward flight velocity, $u^{cg}_{1,g}$. Results obtained by the fully coupled LRM with reach factor $a = 1.25$ for $N^{LRM} = 32$ (—), $N^{LRM} = 64$ (—), and $N^{LRM} = 128$ (—). Pitt-Peters model (—), and flight test measurements (○) for comparison.

Independent of the correctness of the predicted power, the LRM model converges for $N^{LRM} \geq 64$. This resolution equals the identified convergence resolution for the

inflow velocity in section 8.1. Nevertheless, the inflow velocity did not converge for an isolated rotor in hover. With the fuselage in the LRM domain, the power is convergent for $N^{LRM} \geq 64$. As the impedance boundaries introduce fewer errors at lower gradients in the boundary, the presence of the fuselage explains this finding. The fuselage deflects the rotor wake and prevents contraction of the free jet. The deflection decreases the gradients that develop when the boundary of the free jet incites perpendicular to the impedance boundary condition at the bottom of the LRM domain. Nevertheless, the presence of a fuselage in previous versions of the LRM did not result in convergent power in hover [2]. However, various formulations of the used LBM solver changed since this study.

Nevertheless, for low forward flight speeds below $u_{1,g}^{cg} < 10[\frac{m}{s}]$, neither the LRM nor the Pitt-Peters reference inflow model predict P^{mr} of the flight test. While both give good predictions for the region $10 < u_{1,g}^{cg} < 40[\frac{m}{s}]$, both fail to predict the main rotor power for velocities larger than $40[\frac{m}{s}]$. For low speeds, this might originate from the inability of the disk-based models to account for non-ideal effects in the wake [19, 20]. At high forward flight speeds above $40[\frac{m}{s}]$ investigations performed by Johnson [107] indicate that the induced inflow velocity increases instead of decreasing monotonically with $u_{1,g}^{cg}$. Both rotor disk-based LRM and Pitt-Peters model do not capture this effect [2].

Furthermore, the polar to compute the aerodynamic forces on the fuselage of the MBB Bo 105 varies only concerning the pitch attitude Θ and is constant concerning $u_{1,g}^{cg}$ [2]. The missing dependence on $u_{1,g}^{cg}$ implies that the polar does not capture the effects of flow separation at the back of the fuselage, especially below the tail boom. Nevertheless, this effect would increase the power requirement of the MBB Bo 105.

In summary, the LRM predicted comparable main rotor power to the Pitt-Peters inflow model for the MBB Bo 105 in forward flight. The predicted power converges for $N^{LRM} \geq 64$. Furthermore, the absence of correction factors in the rotor disk models and a Θ dependent fuselage polar can explain the differences found between both LRM and Pitt-Peters and the flight test data. Nevertheless, the prediction of the P^{mr} can be improved by adding correction factors for the induced power and improved fuselage polars [2]. Thus, using algebraic correction factors, the main rotor power prediction of the LRM fulfills the requirement listed in section 2.1.

Next, the predicted attitudes Φ and Θ are compared with flight test measurements to validate that the LRM gives adequate predictions. As the attitudes influence the pilot's perspective and the attitude indicator inside the cockpit, correct prediction increases the realism of the simulation as perceived by the pilot. Figure 8.6 plots Θ on the right and Φ on the left over the forward flight speed $u_{1,g}^{cg}$ for various resolutions of the LRM and the Pitt-Peters reference.

Roll angles Φ and pitch angle Θ show convergence for $N^{LRM} \geq 64$ as expected from the results in section 8.1. Furthermore, no significant difference exists in the attitudes calculated using the LRM and the Pitt-Peters inflow. The calculated values of the pitch angle Θ agree with the flight test data, predicting the required nose-down with increasing forward flight speeds. Unfortunately, the flight test data for the roll angle Φ measured by the DLR shows high variance, which prevents a robust interpretation.

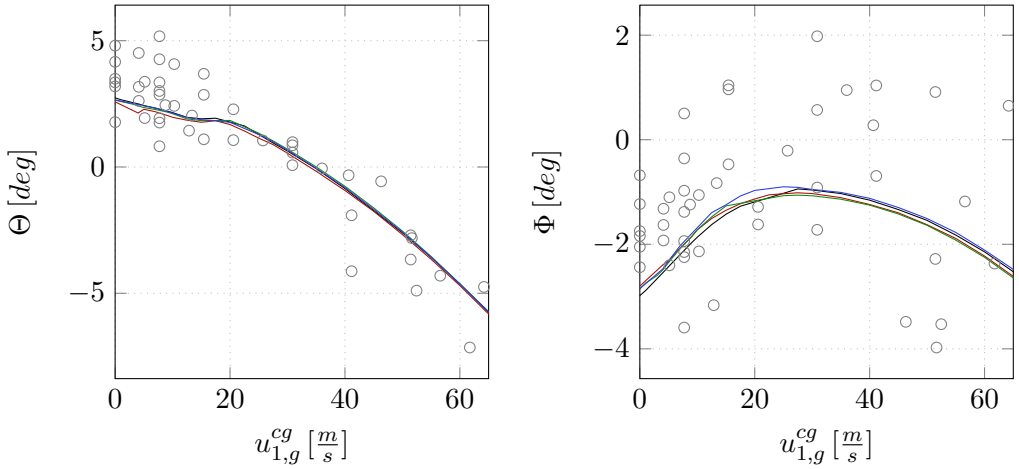


Figure 8.6: Roll angle, Φ, and pitch angle, Θ, over forward flight velocity, $u^{cg}_{1,g}$. Results obtained by the LRM using a reach factor $a = 1.25$ and resolutions $N^{LRM} = 32$ (—), $N^{LRM} = 64$ (—), and $N^{LRM} = 128$ (—). Pitt-Peters model (—), and flight test measurements (○) for comparison.

Nevertheless, this thesis considers the attitudes predicted by the LRM as validated concerning the requirements for a ship-rotorcraft dynamic interface model.

Apart from the visual input generated by the flight simulator for the pilot, a substantial part of a realistic flight simulator experience comes from the correct rotorcraft response to control inputs the pilot uses to maneuver. Therefore, section 2.1 lists the correct representation of control inputs as a requirement. To assess that the LRM fulfills this requirement, the control inputs for stationary, trimmed forward flight are plotted in Fig 8.7. It shows the four control inputs collective, pedal, and lateral and longitudinal cyclic for the LRM and Pitt-Peter inflow model over the forward flight envelope of the MBB Bo 105. Percentage of the total control movement of the corresponding sticks or pedals represents the unit to encode the controls. Nevertheless, as the origin for the measurement of the collective corresponds to the blade pitching angle at $0.75R$, the collective data of the DLR is corrected by 26.35% [2].

The LRM model generally converges for $N^{LRM} \geq 64$ in all inputs except lateral cyclic. The lateral cyclic shows significant differences between $N^{LRM} = 64$ and $N^{LRM} = 128$ for velocities $u^{cg}_{1,g} < 10[\frac{m}{s}]$. The lateral input required for stationary forward flight is related to the non-uniformity of the inflow expressed by v_3^{cosine} [6]. This lateral control input is the slowest converging factor section 8.1 identifies for the LRM simulating an isolated rotor. Furthermore, the Newton solver used to trim the rotorcraft shows difficulties finding stationary solutions at flight speeds $u^{cg}_{1,g} < 10[\frac{m}{s}]$ when used with the LRM. The increased number of iterations in the Newton solver suggests that the trim of the rotorcraft at these flight speeds is sensitive to small changes. The variance in the pilot-trimmed lateral cyclic in the flight test supports this suggestion.

Comparison of the predicted lateral inputs with flight test data suggests a constant offset over $u^{cg}_{1,g}$ between the flight test data and the predicted input. A different origin

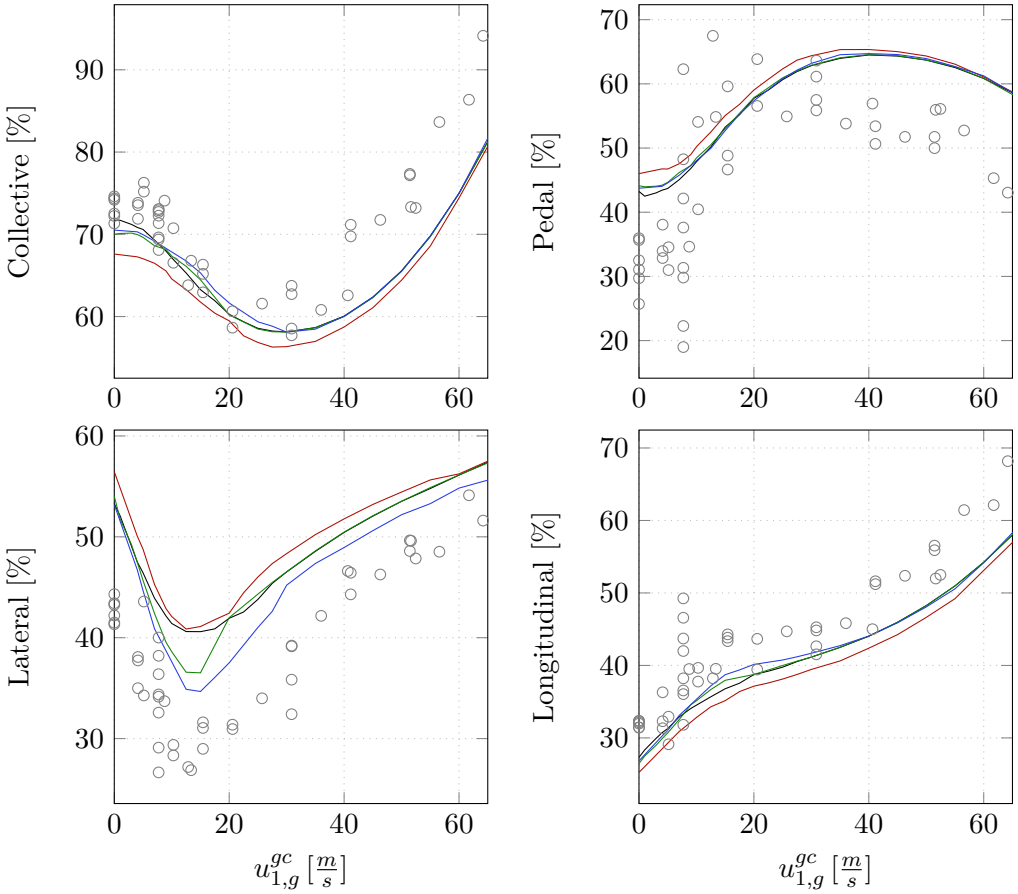


Figure 8.7: Collective, pedal, lateral, and longitudinal cyclic input in trimmed forward flight obtained with the LRM at reach factor $a = 1.25$ with resolution $N^{LRM} = 32$ (—), $N^{LRM} = 64$ (—), and $N^{LRM} = 128$ (—). Pitt-Peters model (—), and flight test measurements (○) for comparison.

could introduce this offset for the collective. Nevertheless, the author and the DLR could not identify a systematic cause for this offset [2].

Figure IX in Appendix V shows the lateral input but applies a suspected offset of 10% to the flight test data. Given this, a good agreement of the LRM using $N^{LRM} \geq 64$ and the flight test data is observed up to $u^{cg}_{1,g} < 30[\frac{m}{s}]$. At higher velocities, the LRM and Pitt-Peters reference underpredict the lateral input by a constant factor. Nevertheless, this agrees with the underprediction of the collective by a constant factor. As higher collective increases the main rotor force F^{mr}_3, the imbalance of the inflow, expressed by v^{cosine}_3, increases. This results from the force accelerating fluid downward through the rotor in forward flight. Nevertheless, the high variance in the flight test data below $u^{cg}_{1,g} < 10[\frac{m}{s}]$ prevents the assessing which of the models predicts the lateral control correctly.

Comparing the collective input for the LRM at converged solutions with the Pitt-Peters reference indicates that both models predict similar collective inputs with a shape following the main rotor power requirement in Fig. 8.5. In comparison to the flight test data, good agreement up to $u_{1,g}^{cg} = 30[\frac{m}{s}]$ is found. Both models underpredict the required collective at higher speeds by a constant amount. Nevertheless, the models agree with the trends in the flight test data for the collective input.

The predictions of both models for the pedal input show good agreement. Nevertheless, in comparison with the flight test data, the models overpredict the input at speeds $u_{1,g}^{cg} < 10[\frac{m}{s}]$ and above $u_{1,g}^{cg} > 30[\frac{m}{s}]$. Furthermore, the high variance of the pedal input in the flight tests makes an interpretation difficult. Nevertheless, the differences at $u_{1,g}^{cg} > 30[\frac{m}{s}]$ are suspected to be related to the underprediction of the collective at these velocities [2].

No significant differences between the LRM and the Pitt-Peters reference exist for the predictions of the longitudinal cyclic input. Compared to the flight test data, the models slightly underpredict the longitudinal but agree with the predicted trends.

To this point, the LRM computes power, attitudes, and controls in stationary forward flight in the same quality as the Pitt-Peters reference. Although the lateral input does not converge for $N^{LRM} \geq 64$ in the LRM, the requirement to correctly predict power, attitudes, and controls is defined as fulfilled for the scope of this thesis. Nevertheless, a piloted evaluation might require some algebraic factors to correct the power. Furthermore, a piloted evaluation can assess which model better predicts the lateral control input, which is especially important in the low flight speed regime [6].

8.2.2 Dynamic Reaction to Control Inputs

Section 2.1 specifies adequate prediction of the dynamic behavior of the rotorcraft and inflow as a requirement. This section verifies that the LRM, in combination with the rotorcraft flight dynamics model, meets this requirement. With the requirement fulfilled, the LRM can simulate flights that feature dynamic inputs from the pilot, like ship deck landing operations.

This capability is verified using flight test data from the DLR. In the flight tests, a doublet input by the pilot excites the MBB Bo 105 flying in trimmed flight while a data logger records the rotorcraft's response. Applying the same control inputs to the rotorcraft flight dynamics model using the LRM and Pitt-Peters as inflow model allows the verification of the dynamic rotorcraft behavior. Furthermore, as the inflow model is changed, this provides an opportunity to compare the dynamic behavior of the inflow between the LRM and Pitt-Peters reference.

The LRM targets the ship-rotorcraft dynamic interface, especially ship deck landing. Therefore, verifying the dynamic behavior uses flight test data at low forward flight speeds. Focusing the verification on the expected settings of later use cases increases the reliability of the verification of the LRM for the task.

Figure 8.8 shows the dynamic reaction of the MBB Bo 105 to a doublet-push to the right at the lateral cyclic at a forward flight speed of $u_{1,g}^{cg} = 13.37[\frac{m}{s}]$. As the flight test starts at trimmed forward flight, the control inputs, already validated in section 8.2.1,

presented as deltas to the trimmed state. A preceding Δ indicates this in the first row of Fig. 8.8. The collective, excluded in Fig. 8.8, is held constant during the flight test. The second row shows the roll, pitch, and yaw rate in reaction to the control input. These rates represent the temporal derivative of the attitudes shown in the third row. Nevertheless, this relation is not direct as the rates are given in absolute values in the *cg*-system while the attitudes represent Euler angles. The fourth row plots the rate of descent, the non-dimensionalized rotational speed of the main rotor, and the main rotor power P^{mr}.

The comparison generally indicates that the LRM converges for $N^{LRM} \geq 64$ with a reach factor of $a = 1.25$. This convergence is in agreement with previous findings.

The response in these variables is divided into two categories to ease the comparison of the roll, pitch, yaw rates, and attitudes with another flight test. The on-axis response describes the rotorcraft's response on the input axis. In Fig 8.8, the pilot introduces a roll input. Thus, the corresponding on-axis response is the roll rate p_{cg}^{cg}, and the roll attitude Φ. The other two axes represent off-axis responses. Figure 8.8 represents the three axes of the rotorcraft as columns in the plot matrix without the fourth row.

Starting with the on-axis response, the LRM compares well with the Pitt-Peters and the flight test data in the roll rate p_{cg}^{cg} for the first half of the doublet. In the second half, the LRM predicts a higher roll rate than the Pitt-Peters. Nevertheless, no rating is possible due to the high-frequent signal noise in the flight test. In contrast, the oscillations using the LRM and the Pitt-Peters inflow model originate from blade flapping in response to the sudden input.

After the doublet, the LRM predicts an increasing roll rate while the Pitt-Peters predicts a roll rate of $p_{cg}^{cg} \approx 0$. In contrast, the flight test measured a decreasing roll rate. Nevertheless, as Fig. 8.8 represents a time series, the models calculate the reaction by integrating in time. Therefore, errors in the models influence the development at times after they occur. Furthermore, as rotorcraft represents a non-linear system with non-stable eigenvalues [10], differences between the models or flight test data later in the time series can originate from minor errors in previous time steps.

The corresponding attitude Φ shown in the third row, first column predicted by the LRM agrees well with the flight test and the reference model for the first half of the doublet. In the second half, the LRM predicts a faster decrease than the reference model or the flight test data. The difference in Φ originates from the difference in the roll rate p_{cg}^{cg} in this half of the doublet.

The LRM and the Pitt-Peters inflow model predict different off-axis responses in the pitch-axis (second column). Furthermore, neither model agrees with the flight test data for q_{cg}^{cg} or Θ.

Comparing the off-axis response in the yaw-axis (third column), both models give almost identical predictions for r_{cg}^{cg} and consequently Ψ. Nevertheless, both underpredict the flight test measurements slightly.

The plot of the rate of descent, shown in the last row of Fig. 8.8, shows that the LRM and Pitt-Peters model agree in the predicted trend. Nevertheless, the LRM does not show the initial descent in the first half of the doublet. Furthermore, both models underpredict the rate of descent measured in the flight test. Although the flight test

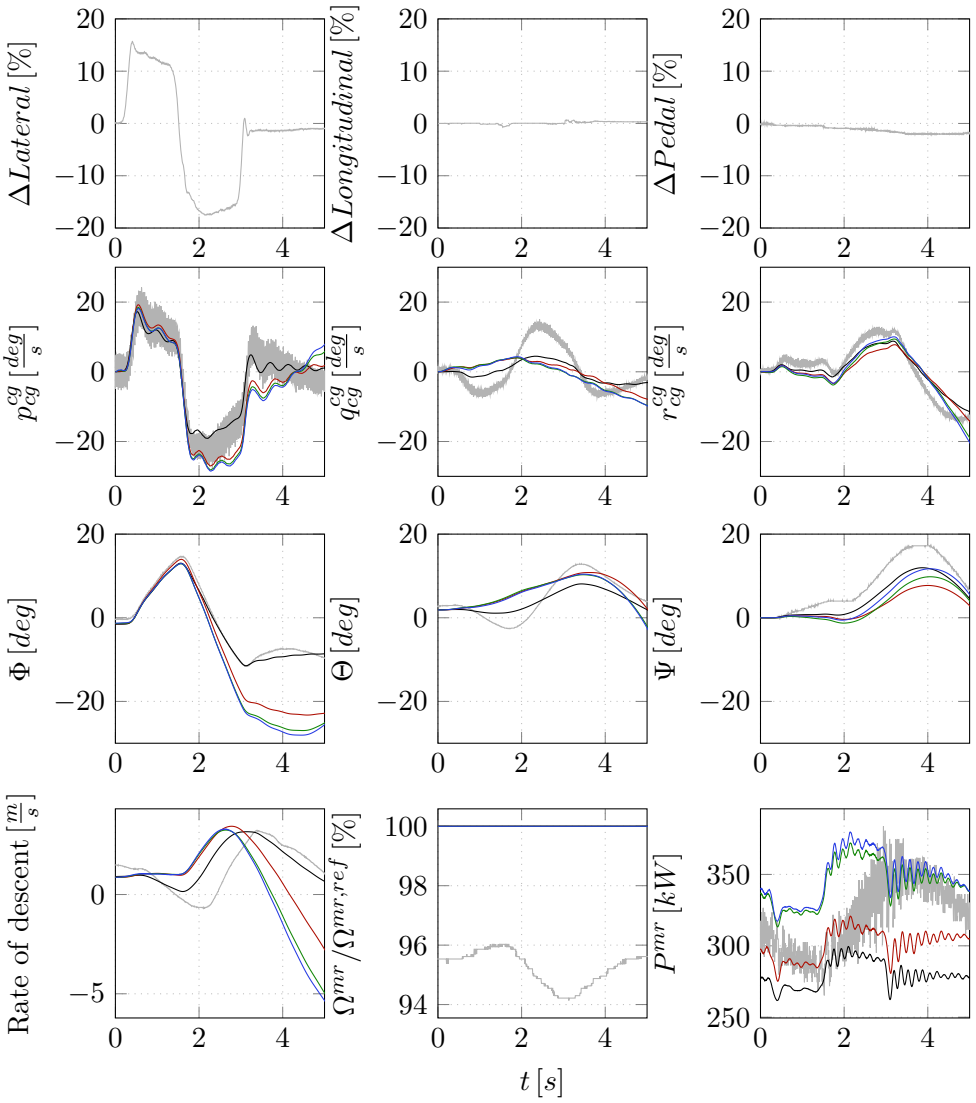


Figure 8.8: Reaction of the helicopter to a doublet right in the lateral cyclic at $u_{1,g}^{cg} = 13.37\,[\frac{m}{s}]$ using the LRM with $N^{LRM} = 32$ (—), $N^{LRM} = 64$ (—), $N^{LRM} = 128$ (—) at $a = 1.25$, and the Pitt-Peters model (—). Flight test measurements (—) for comparison. Pilot inputs at constant collective in the first row, angular velocities in the *cg*-system, $p_{cg}^{cg}, q_{cg}^{cg}, r_{cg}^{cg}$, in the second row, and attitudes, Φ, Θ, Ψ, in the third row. Rate of descent and main rotor relative rotational speed $\Omega^{mr}/\Omega^{mr,ref}$, together with the required main rotor power, P^{mr}, in the last row.

started in a trimmed forward flight, the initial rate of descent was non-zero. Because the fuselage flies at a non-zero pitch angle, the rate of descent, representing $u_{3,cg}^{cg}$, is non-zero. As the pitching angle Θ at $t = 0$ differs between the models and the flight test data, so does the initial rate of descent. Nevertheless, the nonsteady rate of descent

of the flight test data before the doublet indicates that the manual trim in the flight test was not perfect.

As the rotorcraft flight dynamics model did not model the MBB Bo 105 engines but used a generic engine that always supplies the required power, the relative rotational speed $\Omega^{mr}/\Omega^{mr,ref}$ in the model is always 100%. In contrast, during the flight test, the rotational speed of the main rotor varied according to the power deficit or surplus provided by the engines. Nevertheless, this difference is limited to $5 \pm 1\%$ during the doublet.

Both models predicted an identical trend regarding the main rotor power, although at different absolute values. These differences agree with the previously identified differences in power and inflow prediction.

To this point, the LRM predicts a general trend of the on-axis response to the doublet. Nevertheless, compared to the Pitt-Peters reference, the LRM prediction contains more deviation from the flight test data. The difference might be related to fitting the Pitt-Peters model to flight test data as described in section 8.1. Furthermore, the LRM uses all components of the aerodynamic rotor forces, whereas the Pitt-Peters reference only computes v_3. These additional components of the inflow velocity affect how the rotor wake is developing. Nevertheless, their overall influence is low as the values are generally small. Both models need to better predict the off-axis response in the pitch-axis. In contrast, the prediction of the off-axis response in yaw agrees well with the flight test data for both models. The preceding discussion also holds for the reaction of the rotorcraft to doublet inputs in the longitudinal cyclic and the collective. The corresponding plots, ordered like Fig. 8.8, are included with corresponding discussions in appendix VI to improve the readability of the thesis.

Despite the identified discrepancies between the LRM and the flight test data, the requirement on the correct prediction of the rotorcraft dynamic behavior, specified in section 2.1, is considered fulfilled for the scope of this thesis. The LRM gives responses comparable to the Pitt-Peters model for dynamic inputs. In order to improve the rotorcraft response using the LRM inflow model, piloted evaluation assessing realism would be necessary.

8.2.3 Ground Effect in Hover and Forward Flight

This section assesses if the LRM adequately predicts the ground effect in hover and forward flight. This requirement, specified in section 2.1, originates from the necessity to capture the effect of flying above the ship's deck during the landing procedure. For simplicity, a simulation of an isolated MBB Bo 105 rotor with $\alpha = 0$ verifies the LRM's ability to predict the ground effect. $\alpha = 0$ equals the rotor-axis being perpendicular to the ground. The rotor is loaded with $F^{mr}_{mr,3} = 24525N$ in uniform distribution over the disk as in the convergence tests in section 8.1. The rest of the LRM domain follows the description in section 6.4. Therefore, the front and sides of the LRM domain are velocity boundary conditions that block the outflow of fluid through these faces of the cubical domain.

Ground effect in hover

To verify if the LRM predicts ground effect in hover, Fig. 8.9 compares the non-dimensionalized mean inflow velocity of the LRM at various heights over the ground with the algebraic ground effect models of Cheeseman and Benett [108], and Hayden [109]. To non-dimensionalize the plot, the inflow velocity $\bar{v}_3$ is divided by its value out of ground effect $\bar{v}_3^{oge}$. In contrast, the height above ground x_3^{mr} is divided by the main rotor radius R^{mr}.

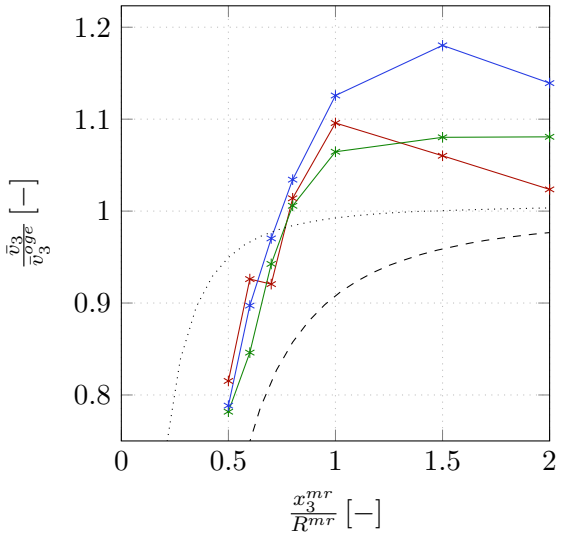


Figure 8.9: Influence of the height above ground on the mean $\bar{v}_3$ component of the induced velocity for an isolated MBB Bo 105 rotor. Calculated with the LRM at a domain size of $a = 1.25$ and resolutions $N^{LRM} = 32$ (—✱—), $N^{LRM} = 64$ (—✱—), and $N^{LRM} = 128$ (—✱—). Models of Cheeseman and Benett [108] (- - -) and Hayden [109] (·····) for comparison.

The LRM manages to predict a reduction in $\bar{v}_3$ that is in the range spanned by the two algebraic reference models for $0.5 < x_3^{mr}/R^{mr} < 0.6$ with resolutions $N^{LRM} \geq 64$. The resolution $N^{LRM} = 32$ fails to predict a monotonically increasing $\bar{v}_3$ required by the physical effects driving the ground effect. Additionally, the LRM fails to predict the ground effect for heights above ground $x_3^{mr}/R^{mr} > 0.6$ independent of the used resolution and predicts $\bar{v}_3$ higher than $\bar{v}_3^{oge}$ for $0.8 < x_3^{mr}/R^{mr}$. Although this error reduces with increasing x_3^{mr}/R^{mr}, the physics of ground effect prohibit any $\bar{v}_3/\bar{v}_3^{oge} > 1$. Additionally, Fig. 8.9 indicates that the LRM reaches no convergence at increasing resolutions in hover while the ground effect is present.

These findings conflict with previously published results by the author [1], which indicated that a resolution of N^{LRM} with a reach factor $a = 1$ is sufficient to capture the ground effect. The publication used an older state of the LBM solver, and the domain featured impedance boundaries on all faces of the cube. Nevertheless, this indicates that the velocity boundary at the front and sides of the LRM domain could cause the observed errors. As the concept of the LRM depends on this boundary

condition being present, other factors to improve the prediction of ground effect in hover are necessary.

One prominent variable that is suspected to improve the ground effect predictions of the LRM is the reach factor a. To investigate this, Fig. 8.10 plots the non-dimensionalized mean inflow velocity at various heights over the ground for reach factors from $a = 1.25$ up to $a = 2$. As previously, Fig. 8.10 includes the algebraic models of Cheeseman and Benett [108] and Hayden [109]. The inflow velocities and the height above ground are non-dimensionalized as in Fig. 8.9.

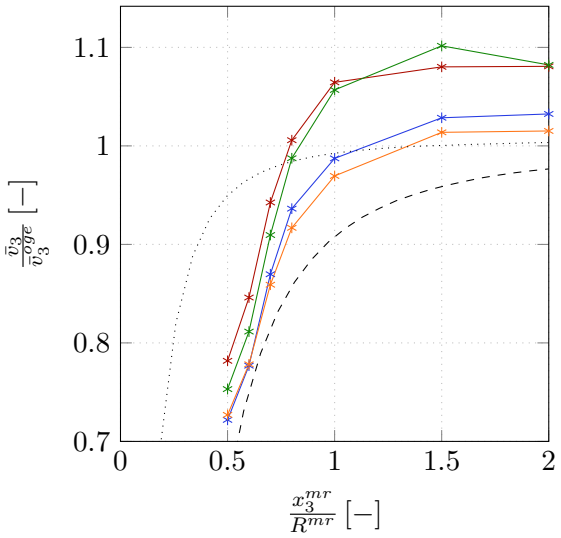


Figure 8.10: Influence of the height above ground on the mean $\bar{v}_3$ component of the induced velocity for an isolated MBB Bo 105 rotor. Calculated with the LRM at a resolution $N^{LRM} = 64$ for reach factors $a = 1.25$ (—✱—), $a = 1.5$ (—✱—), $a = 1.75$ (—✱—), and $a = 2$ (—✱—). Models of Cheeseman and Benett [108] (- - -) and Hayden [109] (·····) for comparison.

The reach factor a mainly influences the prediction of ground effect in hover using the LRM. With increasing a, the predicted ground effect converges towards the range the two algebraic models spanned. In general, the predicted strength of the ground effect increases with increasing a. Nevertheless, even with $a = 2$, the LRM predicts $\bar{v}_3$ slightly higher than the inflow velocity out of ground effect $\bar{v}_3^{oge}$. These findings amount to the LRM failing to fulfill the requirement to correctly predict the ground effect in hover as specified in section 2.1. Nevertheless, with reach factors of 2 and larger, the LRM might be used for piloted simulations in ship-rotorcraft interaction. Despite the non-physical inflow velocities at $x_3^{mr}/R^{mr} > 1.5$, piloted flight evaluation would be necessary to conclude if the error in $\bar{v}_3$ degrades the pilot experience in realistic flight situations.

Ground effect in forward flight

As the rotorcraft ship deck operations feature windy ambient conditions, the requirements specify that the ground effect in forward flight has to be adequately predicted by ship-rotorcraft dynamic interface models. Therefore, this chapter assesses the capability of the LRM to predict ground effect in forward flight. For the verification, the mean inflow velocity $\bar{v}_3$ of an isolated MBB Bo 105 rotor inside the LRM domain in constant forward flight at different heights over ground is compared to the extension to the ground effect model by Cheeseman and Benett [108] to forward flight [107]. Figure 8.11 plots $\bar{v}_3$ over the forward flight velocity $u_{1,g}^{cg}$ for various heights above ground. The left side of Fig. 8.11 uses $a = 1.25$ while the right side shows the results with $a = 2$. The non-dimensionalized heights above ground x_3^{mr} are encoded in the colors of the plotted lines. The LRM predictions use $N^{LRM} = 64$. Furthermore, the figure includes the model of Cheeseman and Benett [108].

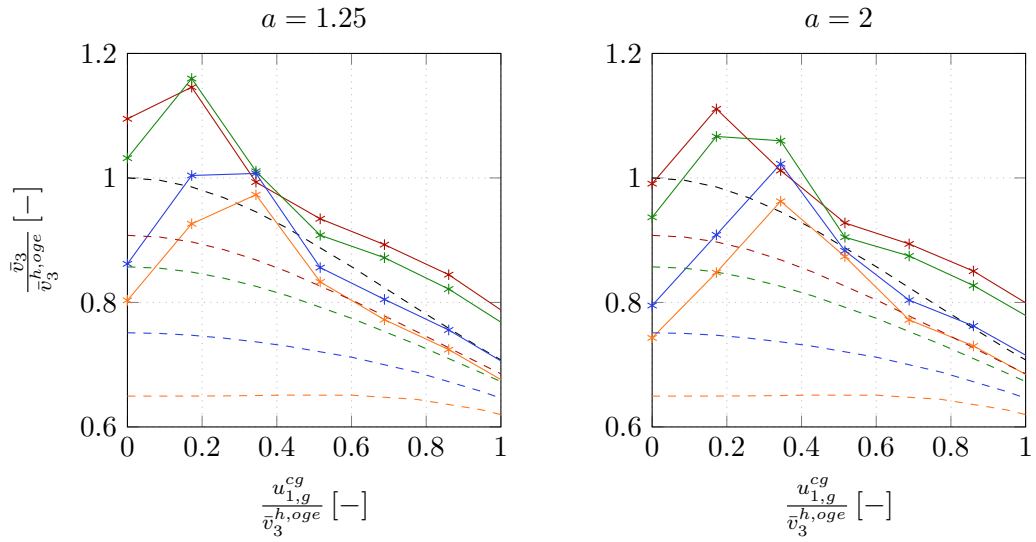


Figure 8.11: Influence of the height above ground on the mean $\bar{v}_3$ component of the induced velocity for an isolated MBB Bo 105 rotor in forward flight for $a = 1.25$ on the left and $a = 2$ on the right. LRM at a domain size of $a = 1.25$ and resolution $N^{LRM} = 64$ in solid lines (—✱—), and Pitt-Peters with Cheeseman and Benett correction for ground effect in forward flight [108, 107] in dashed lines (- - -). Colors encode height above ground, $x_3^{mr}/R^{mr} = \infty$ in (——), $x_3^{mr}/R^{mr} = 1.0$ in (——), $x_3^{mr}/R^{mr} = 0.8$ in (——), $x_3^{mr}/R^{mr} = 0.6$ in (——), and $x_3^{mr}/R^{mr} = 0.5$ in (——).

Comparing the left plot at $a = 1.25$ with the right plot at $a = 2$ indicates that increasing a does not change the LRM's prediction of ground effect in forward flight. Although the results converge concerning a, the differences in the algebraic reference model of Cheeseman and Benett are significant. The LRM fails to predict the reduction in $\bar{v}_3$ due to the ground effect independent of the height above ground x_3^{mr}/R^{mr} which is encoded in colors in Fig. 8.11. The Misrepresentation of the ground effect in forward

flight conflicts with previous studies by the author [1]. Nevertheless, the previous studies used impedance boundary conditions on all faces of the domain around the rotor. The difference in results due to different boundary conditions indicates that the velocity boundary conditions at the front and sides of the LRM domain prevent the LRM from correctly predicting the ground effect in forward flight. Unfortunately, the velocity boundary conditions are vital for transporting the ship wake into the LRM domain.

These findings amount to the LRM failing to fulfill the requirement to correctly predict the ground effect as specified in section 2.1. Nevertheless, the ship's speeds given by Wall for the full-scale NatoGD are approximate $u_{1,g}^{NatoGD} \approx 7.7[\frac{m}{s}]$ [12]. Therefore, a MBB Bo 105 approaching the NatoGD ship deck flies at $u_{1,g}^{cg}/\bar{v}_3^{h,oge} \approx 0.7$ which reduces the overall effect of the ground effect during the approach. Nevertheless, the algebraic model of Cheeseman and Benett predicts a significant reduction in induced velocity even for $x_3^{mr} = 1$ at this forward flight speed.

Although the LRM fails to predict the ground effect in an isolated setting, it might still capture the interaction of the rotorcraft with the ship deck. The evaluation of the mean fluid velocity of the NatoGD in a headwind, shown in chapter 7, indicates that the flow behind the hangar doors is very turbulent. Therefore, the author expects the interaction of an approaching MBB Bo 105 with the ship deck to differ from the isolated ground effect in forward flight.

Generally, the results in section 8.2.3 only show the LRM's capability to predict the stationary ground effect. Nevertheless, the requirements in section 2.1 specify that a ship-rotorcraft dynamic interface model must also predict partial and moving ground effect. However, given the LRM's poor performance for the stationary ground effect, verification for partial and moving ground effect is outside this thesis's scope.

8.2.4 Wall Effects and Recirculation

In the last phase of the approach, the rotorcraft flies close to the ship deck and superstructure. Therefore, the wake of the rotors can be deflected to the front by the ship deck and upwards by the hangar doors. Additionally, if the rotor is close enough to the hangar doors, it can draw in the deflected wake. This recirculation represents a significant change in the inflow velocities. Furthermore, the ship's turbulent wake influences the recirculation and where it incides into the rotor.

In addition to these aerodynamic effects, recirculation requires time to develop. It depends on the ship deck and hangar being at specific distances from the rotor. With the ship and rotorcraft moving relative to each other, the conditions for stable recirculations constantly change. Therefore, the recirculation can spontaneously collapse and reappear. Furthermore, both creation and collapse require time due to the inertia of the fluid, which increases the pilot's workload in controlling the rotorcraft.

Verifying the LRM's capability to predict recirculation requires a verified wall and ground effect prediction. Nevertheless, section 8.2.3 shows the difficulties of the LRM in predicting the ground effect. Therefore, a detailed investigation of wall effects and recirculation is outside this thesis's scope. However, Fig. 8.12 shows the LRM's ca-

pability to develop recirculations with rotors close to the hangar of a NatoGD. The streamlines in the figure show how the ground deflects the fluid drawn in from the upper deck towards the hangar. At the hangar doors, the fluid is deflected upwards and to the side from where the rotor draws it in.

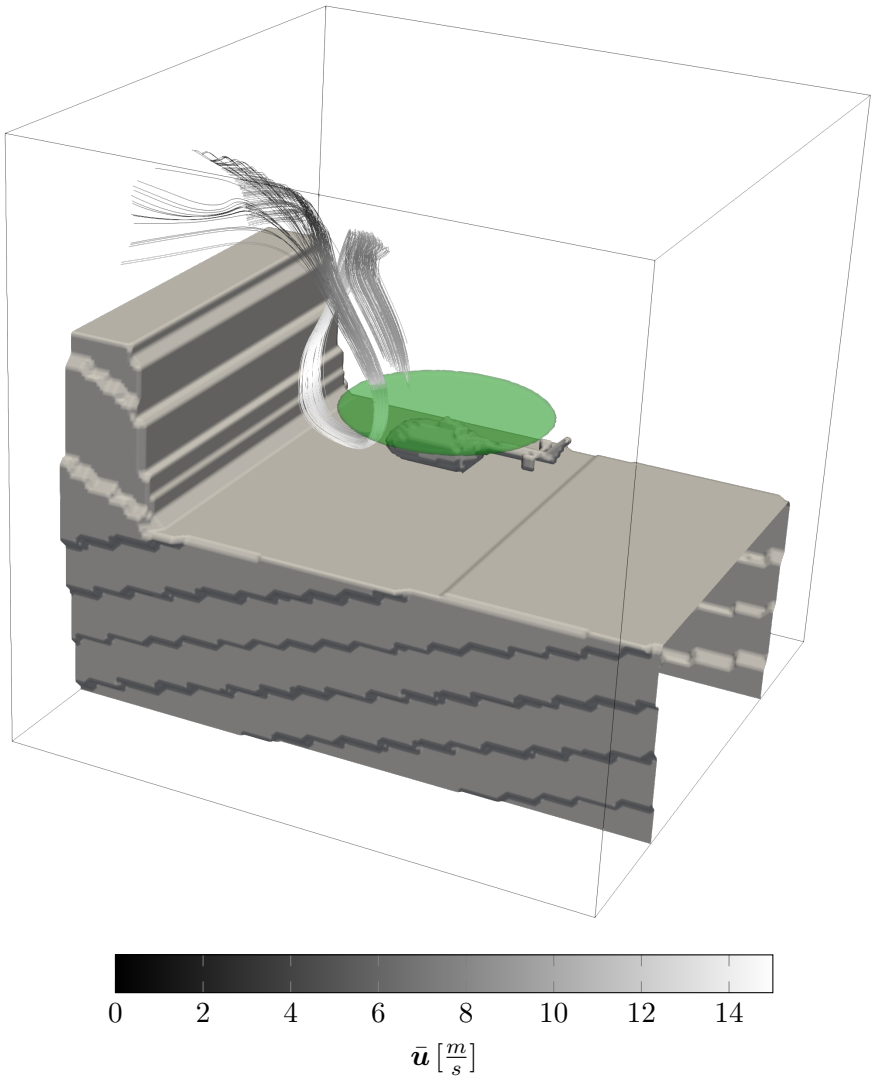


Figure 8.12: MBB Bo 105 hovering above the landing deck of a NatoGD. Streamlines through a section of the rotor, colored by $\bar{\boldsymbol{u}}$, show a complex recirculation. The simulation includes the ship wake inside the LRM domain. Rotor disk represented in transparent green, ship geometry and rotorcraft fuselage in grey.

Figure 8.12 highlights that the LRM can represent recirculations occurring within the domain. Furthermore, the deviation of the streamlines to the side while traveling along the hangar doors indicates a complex interaction between rotor wake and ship geometry.

The complex interaction requires verification by comparison with measurement data. Nevertheless, this is outside the scope of this thesis. Fig. 8.12 proves the LRM's capability to capture wall effects and recirculations within the domain. Therefore, the LRM fulfills the requirement on wall effects and recirculations for the scope of this thesis.

8.3 Computational Requirements

The LRM, designed in this thesis, aims to provide a solution to the ship-rotorcraft dynamic interface that is usable in piloted flight simulation. Therefore, the LRM must be able to run on contemporary hardware in firm real-time as specified in section 2.3. Nevertheless, this thesis considers the requirement of running on contemporary hardware fulfilled as the LRM runs on NVIDIA Tesla A100 GPGPUs interconnected with NVLink II. At the time of writing, these processors are in widespread use.

Section 8.3.1 verifies the real-time capability of the LRM. Furthermore, section 8.3.2 lists the resulting time steps and relates them to the required reaction time of simulators to pilot inputs.

8.3.1 Firm real-time simulations

In the LBM, the number of cells mainly defines the computational effort necessary. Thus, the imperative real-time requirement sets the available number of LBM cells that form the LRM domain. Therefore, the real-time requirement defines the resolution N^{LRM}, which allows the LRM domain to capture physical phenomena.

Unfortunately, due to the LBM linking time step and cell size, the computational cost of the cubic LRM domain scales $\propto N^4$. Therefore, increasing the resolution of the LRM domain is computationally expensive. In contrast, increasing the LRM domain reach factor a at constant N^{LRM} scales $\propto N^3$ as increasing the domain leaves the cell size and thus the time step unchanged.

Nevertheless, the LBM solver scales well on GPGPUs. Unfortunately, the required communication to distribute a domain over multiple GPGPUs adds computational effort. Therefore, the benefit of this level of parallelization depends on the specific resolution N^{LRM} and reach factor a of the LRM domain.

In the following, N^{LRM} and a are determined in two steps in order of decreasing computational cost. First, the resolution N^{LRM} that allows real-time capability at $a = 1$ is identified. The second step increases a at the found N^{LRM} to find the real-time capable maximum. This thesis measures two times to compare the performance of simulations. The wall clock time t^{Clock} represents the real-world time passed during a simulation. In contrast, the simulation time t^{Sim} represents the time interval the simulation advanced. This thesis considers simulations with $t^{Clock}/t^{Sim} < 1$ to be real-time capable.

Fig. 8.13 shows the scaling of the resolution N^{LRM} concerning the number of used GPGPUs. It plots the wall clock time divided by the simulation time of the LRM domain for multiple resolutions. The results represent the average of a $20s$ simulation

without any in- or output of files to/from disk. The used NVIDIA Tesla A100 processors GPGPUs interconnect with NVLink II.

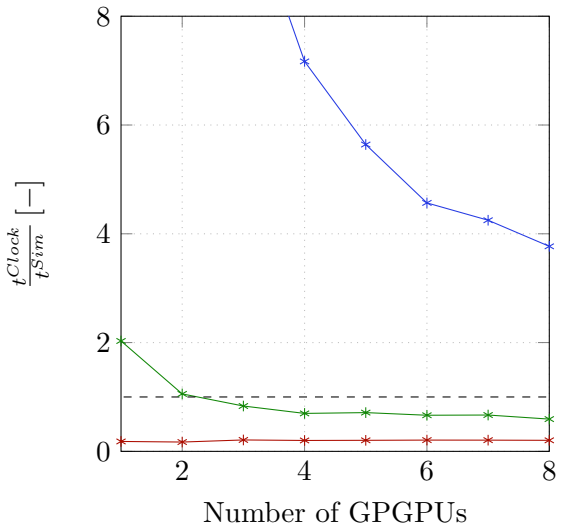


Figure 8.13: Wall clock time t^{Clock} divided by simulation time t^{Sim} of an LRM domain with reach factor $a = 1$. Computed with a varying number of GPGPUs. Resolution $N^{LRM} = 32$ in (✱), $N^{LRM} = 64$ in (✱), and $N^{LRM} = 128$ in (✱). Real-time limit of $t^{Clock}/t^{Sim} = 1$ in (- - -).

The $N^{LRM} = 32$ result, (✱) in Fig. 8.13, indicates that this resolution fulfills the real-time capability. As the computational load of a LRM domain at this resolution translates into little computational work, using multiple GPGPUs adds no performance benefit. The time to compute the workload per GPGPU is already smaller than the time required to communicate between the GPGPUs. Furthermore, previous investigations found the resolution $N^{LRM} = 32$ too coarse to give good predictions for the rotorcraft in stationary flight [2]. The results in section 8.2 further support this assessment.

In comparison, the resolution $N^{LRM} = 128$ requires too many calculations to fulfill the real-time requirement. Although using more than 7 GPGPUs still speeds up simulations at this resolution due to the high computational workload, the required wall clock time using eight GPGPUs is four times the real-time limit of $t^{Clock}/t^{Sim} = 1$. Therefore, $N^{LRM} = 128$ does not fulfill the real-time requirement with the eight NVIDIA Tesla A100 processors available.

In contrast, the resolution $N^{LRM} = 64$, (✱) in Fig. 8.13, is real-time capable by using more than three GPGPUs. However, the performance does not improve with more than four GPGPUs, which indicates that at this point, the added time for communication dominates the computational load and, thus, the wall clock time. Nevertheless, previous findings [2] and section 8.2 indicate that this resolution predicts many of the physical phenomena specified in section 2.1.

The results for $N^{LRM} = 64$ in Fig. 8.13 indicate, that the use of multiple GPGPUs increases performance. This available performance can increase the reach factor a of

the LRM domain. The maximum a that is real-time capable results from measuring the performance of simulations of the LRM domain at $N^{LRM} = 64$.

Figure 8.14 plots t^{Clock}/t^{Sim} over the number of GPGPUs for simulations with increasing reach factor a. The simulations use $N^{LRM} = 64$ for all results. The results represent the average of a $20s$ simulation.

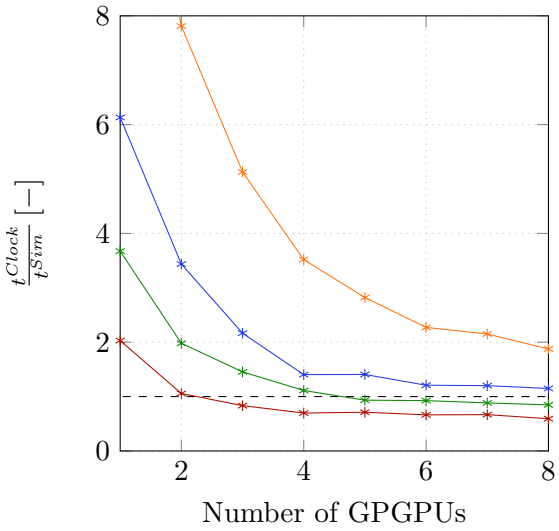


Figure 8.14: Wall clock time t^{Clock} divided by simulation time t^{Sim} of an LRM domain with $N^{LRM} = 64$. Computed with a varying number of GPGPUs. Reach factor $a = 1$ in (✱), $a = 1.25$ in (✱), $a = 1.5$ in (✱) and $a = 2$ in (✱). Real-time limit of $t^{Clock}/t^{Sim} = 1$ in (- - -).

The results indicate that the simulations with reach factors $a = 1$ in (✱) and $a = 1.25$ in (✱) are real-time capable with more than five A100 GPGPUs. In contrast, both $a = 1.5$ in (✱) and $a = 2$ in (✱) do not reach real-time capability even with the available eight GPGPUs. The results highlight that using contemporary hardware, the LRM fulfills the real-time requirement with $N^{LRM} = 64$ and $a = 1.25$. The eight GPGPUs require $0.8475s$ wall clock time to advance the LRM domain one second of simulation time.

The rotorcraft flight dynamics require computational time to compute the rotorcraft reaction using the inflow provided by the LRM domain. Nevertheless, the rotorcraft flight dynamics model outperforms the fluid simulation by orders of magnitude. The rotorcraft flight dynamics model requires $0.0423s$ wall clock time to advance the rotorcraft one second of simulation time. This performance represents the average of a simulation with a $20s$ duration on an Intel i7-5820K processor.

Therefore, the combined LRM and rotorcraft flight dynamics model fulfill the real-time requirement specified in section 2.3. Nevertheless, the quality of the real-time capability remains untested. The available computers do not use operating systems that guarantee a computational resource to an individual process. Therefore, the available systems prevent verification of the firm attribute to the real-time requirement given

in section 2.3. Nevertheless, unless interrupted by the user, the simulations performed in this thesis fulfilled the firm attribute. Additionally, the performance measuring excludes the loading of the NatoGD wake data from disk into the RAM. The thesis assumes that either a disk loads the ship wake data fast enough or the simulation loads the ship wake data into RAM before starting.

8.3.2 Resulting Time Steps

The FAA requires level D rotorcraft simulators to react to the pilot's control inputs in less than $100ms$ [18]. Therefore, section 2.3 lists the reaction time of the model $< 100ms$ as a requirement.

In general, the glslbm solver that simulates the LRM domain uses small time steps compared to the rotorcraft flight dynamics model. Therefore, the time step of the LRM domain is set for the rotorcraft flight dynamics model to prevent costly interpolation in time. With the time step $\Delta t = 1.80352ms$, the LRM fulfills the required reaction time of $< 100ms$. Nevertheless, this limit on the reaction time holds for the flight simulator as a complete system. In a flight simulator, other processes, like updating the visualization of the pilot's view or sampling control inputs, can require significant time. Thus, the performance of the LRM can pose problems, depending on the flight simulator the LRM is integrated into.

9 Validation of the LRM for Ship-Rotorcraft Interactions

This chapter validates that the developed LRM, the central model of this thesis, serves the intended purpose. Therefore, a complete approach of a MBB Bo 105 onto a NatoGD, including ship motion, is simulated. Therefore, the chapter corresponds to the V-model's "Validation" step. Section 9.1 describes the flight path during the approach, while section 9.2 compares the LRM to a simulation of ship and rotorcraft using the overset technique. The turbulent nature of the ship-rotorcraft interaction prevents direct comparison of the time series of rotor forces and moments, so the resulting frequency distributions are used. These distributions show the LRM's capability to capture the components of the ship-rotorcraft interaction that trigger a pilot response. The overset simulation acts as a reference model for the LRM as no measurement data for an approach of a rotorcraft on a ship is available to the author.

9.1 Definition of the investigated approach

The validation uses a MBB Bo 105 approaching a full-scale NatoGD. Therefore, the LBM simulates the NatoGD at a resolution of $N^{NatoGD} = 364$ with the domain description in section 6.2. The velocity at the inflow of the NatoGD domain corresponds to the $u_{1,g}^{\infty} = 20.58\frac{m}{s}$. Furthermore, the ship is heaving and pitching with a sinusoidal motion. The simulation corresponds to the following case [12]:

`RegularShipMotion_Hs2_Tp15_Vs_15_HG0`

The NatoGD domain simulates a $20s$ start-up period to guarantee a fully developed turbulent ship wake. Then, the NatoGD tabulates data to disk as described in section 6.3 for use with the LRM. The same approach is simulated with the overset technique to provide a reference solution to compare with the results of the LRM. Section 6.7 describes the setup of this overset simulation.

The rotorcraft is moved along a predefined path during the approach for comparability. Furthermore, the control inputs are in fixed position. This setup corresponds to a whirltower with a fuselage approaching the ship. The control inputs correspond to the controls for the MBB Bo 105 in trimmed forward flight at $u_{1,g}^{cg} = 25.58\frac{m}{s}$ using the Pitt-Peters inflow model. Using the same control inputs in all approach simulations eliminates the different trim states of the models from the simulation. Therefore, the validation can directly compare the rotor forces and moments of the LRM and the overset simulation.

For the approach, the rotorcraft starts at $x^{cg}_{1,hangar} = 97m$ behind the landing spot ls described in Fig. 5.2. In the y-direction, it is centered over the landing deck and flies at $x^{cg}_{3,hangar} = 8m$ above the landing deck. The used coordinate system *hangar* corresponds to the coordinate system centered at the bottom of the hangar doors of the NatoGD for a non-moving ship. Figure 5.2 and 7.4 list it with the subscript 2. Due to the ship movement, the height of the MBB Bo 105 above deck varies during the approach. The rotorcraft flies with a velocity of $u^{cg}_{1,g} = 5\frac{m}{s}$ for $t = 17s$ and linearly decreases the velocity until it starts hovering at $t = 18s$. It keeps hovering in place until $t = 30s$.

Figure 9.1 depicts the distance to the landing spot and the velocity of the rotorcraft. The period from 0 to $5s$ highlighted in red indicates the start-up period. At the start of this period, the LRM and overset simulations start exchanging information. Thus, the $5s$ period allows the simulations to flush the domain around the rotor with information extracted from the ship wake.

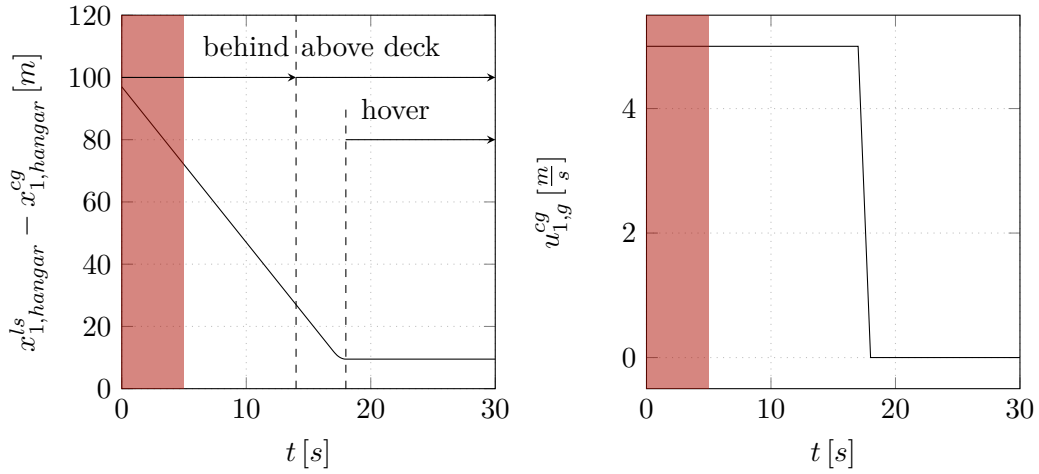


Figure 9.1: Approach velocity and path of a MBB Bo 105 onto the flight deck of a NatoGD. Rotorcraft approaching from behind the ship at constant height above sea level, centered on the flight deck in y-axis. Red area marks the start-up period of the simulation. Markers describing the flight state in (- - -)

The approach features the period $5s < t < 14s$, during which the rotorcraft flies through the ship wake with a significant distance to the ship deck. Between $14s < t < 18s$, the rotorcraft flies over the landing deck of the moving NatoGD and decelerates into a hover. "Above deck" corresponds to the center of gravity of the MBB Bo 105 above the flight deck. The rotorcraft hovers at its position until the simulation ends at $t = 30s$.

Figure 9.2 shows the geometry of the MBB Bo 105 and NatoGD in the center plane of the ship using grey solids. It shows the position of the MBB Bo 105 at $t = 5s$, which represents the start of the simulation after the start-up period in (a) and the hover

position of the MBB Bo 105 above the landing deck at $t = 20s$ in (b). The figure is true to scale.

(a) $t = 5s$

(b) $t = 20s$

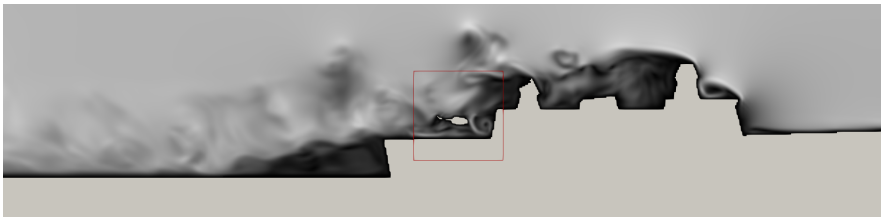
Figure 9.2: Center plane of NatoGD domain including the the MBB Bo 105 fuselage during the approach. Time $t = 5s$ in (a), $t = 20s$ in (b). Grey solids represent the voxelized hulls and ground. True to scale.

9.2 Comparison of the LRM to the overset technique

As no measurement data of a MBB Bo 105 approaching a NatoGD is available, this thesis validates the LRM by comparing it to the overset technique. This comparison uses the same rotorcraft flight dynamics model for both simulations. Because the simulations only differ in the inflow, the comparison validates the LRM as an inflow model for ship-rotorcraft dynamic interaction compared to an overset simulation. The overset simulation eliminates the boundary conditions used in the LRM domain and allows the rotor to influence the complete NatoGD domain. The distance between the rotor and boundaries is significant because the overset domain only uses boundaries at the faces of the NatoGD domain. Therefore, the overset simulation represents the reference of the ship-rotorcraft dynamic interface model with the used LBM solver at the given resolution N.

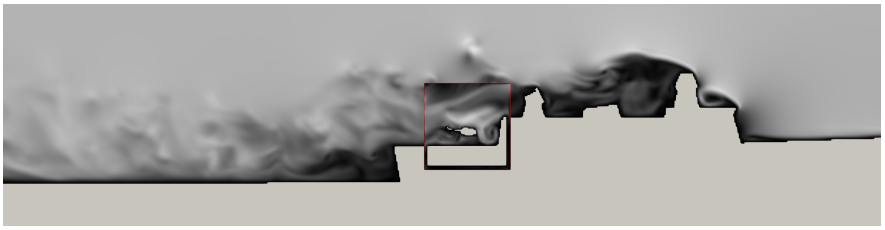
Figures 9.3 (a)-(c) compare the instantaneous flow field at $t = 20s$ for the overset, the NatoGD, and the LRM simulation. The center slices of the domains show the mean velocity $\bar{\boldsymbol{u}}$ and the voxelized ship and rotorcraft geometries in grey.

Comparing the overset simulation (a) with the NatoGD simulation (b) reveals the influence the rotorcraft exerts on the ship wake. Its presence increases the separation



(a) Overset simulation

(b) NatoGD simulation



(c) NatoGD and LRM simulation

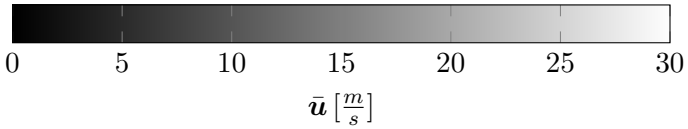


Figure 9.3: Mean velocity $\bar{\boldsymbol{u}}$ in center plane of overset, stand alone NatoGD, and LRM simulation of the approach at $t = 20s$. The resolutions are $N^{NatoGD} = 364$ and $N^{LRM} = 64$ with a reach factor $a = 1.25$. Grey solids represent the voxelized hulls and ground. Boundary of the inner domain or LRM domain respectively in red.

area downstream of the landing deck. Furthermore, it influences the wake upstream, as the differences around the radar tower (just upstream of the hangar doors) indicate.

(c) in Fig. 9.3 shows the mean velocity at the center plane of the NatoGD domain and the velocity in the center plane of the LRM domain. The red square marks the boundary of the LRM domain. Furthermore, the two-cell-wide gap between wall boundary conditions representing the ship and the in and outflow boundaries of the LRM is visible. This gap, described in section 6.4, is filled with fluid at zero velocity.

The LRM uses tabulated data from the NatoGD simulation to prescribe the fluid state at the front and sides of the domain as described in section 6.4. Nevertheless, while recalculating the fluid flow inside the LRM domain, the boundaries influence the

development of the flow field. Therefore, the fluid state at the velocity boundaries of the LRM does not equal the fluid state extracted from the NatoGD simulation at this position.

The comparison of the LRM simulation in (c) and the overset simulation in (a) focuses on the differences inside the domains containing the rotor. Red lines mark the LRM and inner domains, respectively. Both domains feature a flow pattern with a recirculation in front of the rotorcraft. Nevertheless, the flow velocity at the top of the LRM domain is smaller than at the same position in the overset simulation. The LRM also features higher flow velocity above and in front of the rotor.

This thesis attributes the differences between the LRM and the overset simulation to the simplifications the LRM introduces to reduce the computational load. Of these simplifications, the reduction in domain size represents the most profound one. Depending on the reach factor, the boundary conditions are close to the rotor and each other, increasing the mutual influence. Furthermore, the reach factor sets the distance the LRM captures the interaction of ship and rotorcraft. Nevertheless, increasing the reach factor is difficult due to the real-time requirement and limitations of current hardware.

Despite the differences between the LRM and the reference overset simulation, the LRM might predict the ship-rotorcraft interaction. Whether the reproduced interaction is sufficient for piloted flight simulation remains to be assessed.

As the rotor forces and moments drive the rotorcraft's reaction to the pilot's inputs, the main rotor's aerodynamic thrust and moment coefficients are the parameters for comparing the LRM and overset simulation. Fortunately, the setup of the approach described in section 9.1 allows a direct comparison of the forces and moments of the LRM and the overset simulation. For consistency with rotorcraft literature, the following investigations use the non-dimensionalized thrust $C^{T,aer}$ and moment coefficients $\Delta C^{Mx,aer}$ and $\Delta C^{My,aer}$.

Figure 9.4 plots the time series of $\Delta C^{T,aer}$, $\Delta C^{Mx,aer}$, and $\Delta C^{My,aer}$ during the approach. The figure plots values for the overset simulation and the LRM with the reach factors $a = 1.25$, $a = 1.5$, and $a = 2$ at $N^{LRM} = 64$. The reference points for the Δ are the respective values at the end of the start-up period at $t = 5s$. The simulations use the same fixed control inputs, resulting in different forces and moments of the rotor. Using the delta of the thrust and moment coefficients $C^{T,aer}$, $C^{Mx,aer}$, and $C^{My,aer}$ allows focusing on the variations during the approach. Section 8.2.1 compares the stationary values.

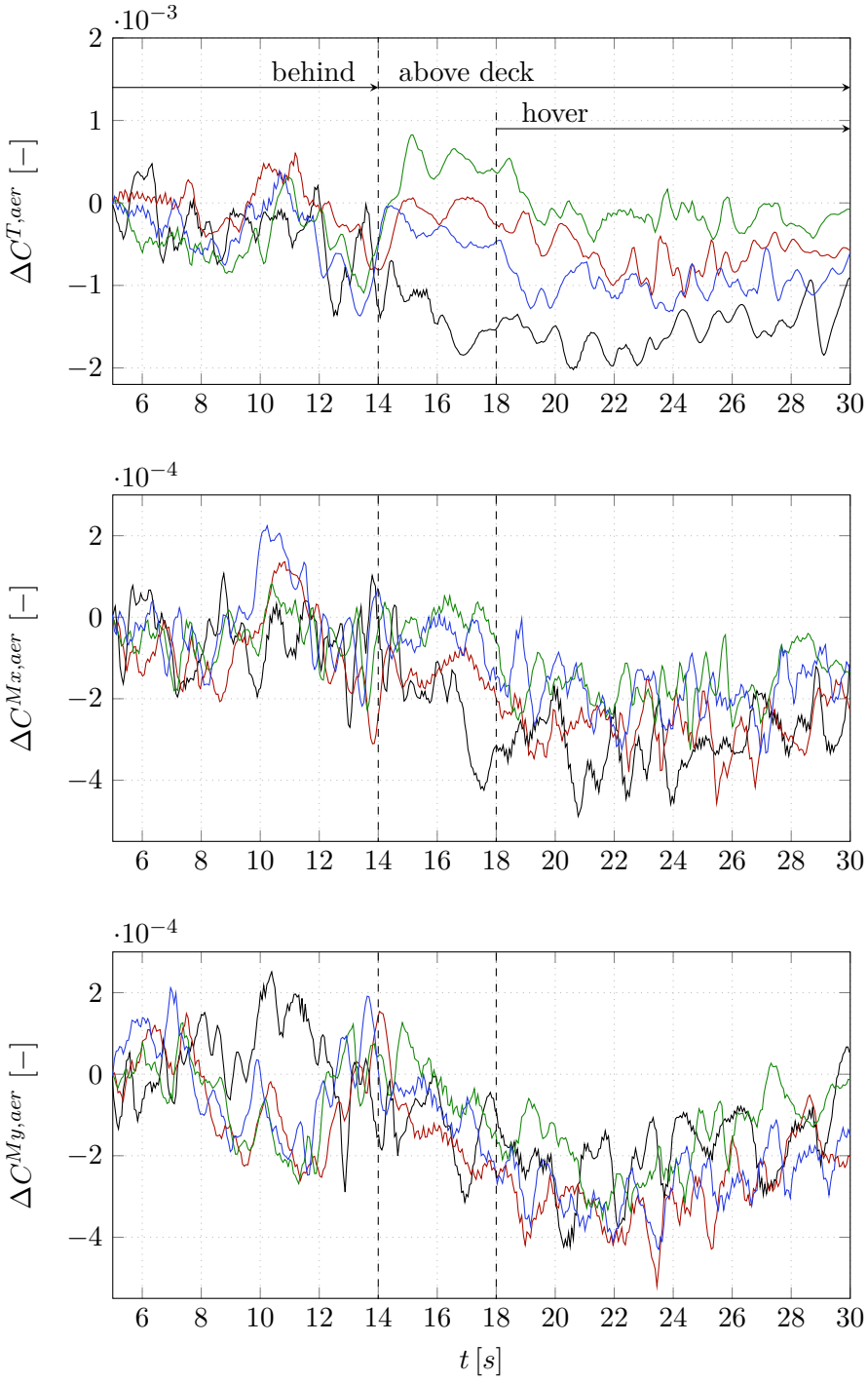


Figure 9.4: Time series of $\Delta C^{T,aer}$, $\Delta C^{Mx,aer}$, and $\Delta C^{My,aer}$ during the approach. The reference points for the Δ are the respective values at $t = 5s$. Results of the overset simulation in (—), LRM with $a = 1.25$ in (—), $a = 1.5$ in (—), and $a = 2$ in (—).

In general, the LRM does not show convergence with increasing reach factor a for the approach. Nevertheless, due to the influence of the boundary conditions at different distances, convergence of the time series is not expected. In the phase starting at $t = 5s$ until approximately $t \approx 14s$, the time series of $\Delta C^{T,aer}$, $\Delta C^{Mx,aer}$, and $\Delta C^{My,aer}$ all show high-frequency fluctuations with comparable amplitudes. The fluctuations in $\Delta C^{Mx,aer}$ and $\Delta C^{My,aer}$ continue for both the LRM and the overset simulation until the simulation ends at $t = 30s$. In contrast, the $\Delta C^{T,aer}$ fluctuations decrease for $t > 14s$ for all simulations. The reduction of the fluctuations in $\Delta C^{T,aer}$ is especially prominent between $t = 14s$ and $t = 20s$. At times $t > 14s$, the rotorcraft's center of gravity flies above the landing deck. Because of the temporal coincidence, the suspected cause is the deflection of the rotor wake by the ship geometry. At times $t > 18s$, the rotorcraft hovers above the landing deck of the NatoGD. In this phase, ΔC_T^{aer} shows fluctuations with smaller amplitudes and lower frequencies than for $t < 14s$. In general, the LRM does not predict the decrease in $C^{T,aer}$ for $t > 14s$ that the overset simulation predicts. Furthermore, increasing the reach factor a does not show converging behavior in $C^{T,aer}$ towards the overset simulation during this period. Nevertheless, all studied reach factors a show similar fluctuations in $C^{T,aer}$.

Direct comparison of time series of turbulent simulations is complex. Instead, a wavelet analysis of the calculated $C^{T,aer}$, $C^{Mx,aer}$, and $C^{My,aer}$ decomposes the time series into frequency distributions over time. The continuous wavelet analysis uses complex Morlet wavelets with a center frequency and bandwidth of $1Hz$. Using these parameters, the wavelet analysis produced minimal distortions in the 0.02 to $2Hz$ band. Outside the given time series, the wavelet analysis assumes a smooth continuation of the signals. As the wavelet analysis decomposes a input time series into a frequency distribution over time. Therefore the results show which frequencies are present in the input time series at which times. As $C^{T,aer}$, $C^{Mx,aer}$, and $C^{My,aer}$ are analyzed, the results represent force and moment fluctuations acting on the rotor during the approach. Fluctuations in the 0.02 to $2Hz$ band are of special importance as they can lead to pilot-induced oscillations as described in section 2.2. Fluctuations of the forces of frequencies higher than $2Hz$ produce vibrations but do not require a pilot response. Fluctuations below $0.02Hz$ might require a pilot response depending on the flight path but do not increase the danger of pilot-induced oscillations. Therefore, correct frequency distributions of the rotor forces and moments in ship-rotorcraft interaction scenarios is important for the realism of the LRM.

Figure 9.5 shows the time series of frequency components in $C^{T,aer}$, $C^{Mx,aer}$, and $C^{My,aer}$ during the approach resulting from the wavelet analysis. The first row shows the results of the overset simulation. The second, third, and fourth rows show the results of the LRM with increasing reach factor a from top to bottom. $C^{T,aer}$, $C^{Mx,aer}$, and $C^{My,aer}$ form the columns in Fig. 9.5. Red lines (- - -) mark the 0.02 to $2Hz$ band.

The first column of Fig. 9.5 indicates that the predicted $C^{T,aer}$ of the LRM with different reach factors a agree with the reference overset simulation. The time series of the frequencies that form $C^{T,aer}$ shows little contribution of frequencies higher than $2Hz$ in all models. Within the band of $0.2Hz$ to $2Hz$, the LRM and the overset simulations show some contributions for $t < 10$ and above $t > 25s$. Nevertheless, the

scaling of the figure prevents interpretation. Thus, Appendix VII shows the results of the wavelet analysis in the $0.2Hz$ to $2Hz$ band using a finer color scale. With the finer color scale, the figure shows a dotted pattern in the respective frequency band. This pattern represents the $C^{T,aer}$ fluctuations identified in the time series in Fig. 9.4. Although the pattern does not feature the dots at the same points in time and frequency, the overall pattern emerges in the LRM simulations independently of a and in the overset simulation. As the time series indicates, the $C^{T,aer}$ values predicted by the models are similar but not directly comparable—nevertheless, the predicted patterns in frequency space match. Thus, the LRM and overset simulation predict a similar frequency distribution of the fluctuations in $C^{T,aer}$ in the $0.2Hz$ to $2Hz$ band during the approach. Furthermore, the turbulence in the wake of the LRM creates randomness. This randomness and the inability to see the airflow prevents pilots from assessing when a vortex separates from the ship structure and gets drawn into the rotor. Therefore, predicting the correct pattern in frequency space is considered sufficient for the scope of this thesis.

For amplitudes of $C^{T,aer}$ in frequencies below $0.2Hz$, the LRM and overset simulation give similar distributions and time series. The amplitudes of these frequencies, which are important for the pilot as highlighted in section 2.2, seem well predicted by the LRM.

In the distribution of $C^{Mx,aer}$ and $C^{My,aer}$ over frequencies during the approach, the LRM for different a and the overset method predict similar patterns. Nevertheless, around $0.2Hz$ at $t = 10s$ to $t = 16s$ Fig. 9.5 indicates a significant difference. Figure XII in Appendix VII displays a detailed view of the difference. This prediction discrepancy can not be pinpointed to specific physical or numerical phenomena in the simulations using the LRM.

Apart from the difference around $0.2Hz$ in the approach, the LRM predicted a similar pattern in the distributions of $C^{Mx,aer}$ and $C^{My,aer}$ over time. Nevertheless, the LRM shows no distinct convergence with a towards the overset simulation for the distributions of $C^{T,aer}$, $C^{Mx,aer}$ and $C^{My,aer}$ in frequency space during the approach.

The performance measurements in section 8.3.1 found the LRM real-time capable with $N^{LRM} = 64$ and $a = 1.25$. Therefore, the error in the frequency distributions of this simulation is of particular interest. Figure 9.6 prints the difference of the real-time capable LRM simulation minus the overset simulation reference. This subtraction gives the relative error in the amplitude distribution of $C^{T,aer}$, $C^{Mx,aer}$, and $C^{My,aer}$ over the frequency domain.

Figure 9.6 indicates that the difference of the LRM at N^{LRM} and $a = 1.25$ and the overset simulation is slight in the frequency band between $0.2Hz$ and $2Hz$. Nevertheless, in the frequency band around $0.05Hz$, the difference in $C^{T,aer}$ is approximately 10%.

In contrast, in $C^{Mx,aer}$ and $C^{My,aer}$, the LRM at $N^{LRM} = 64$ and $a = 1.25$ shows significant differences to the overset simulation, which represents the reference. In both non-dimensionalized aerodynamic moment coefficients, errors of up to 20% occur at frequencies below $0.2Hz$. The previously noticed dotted pattern emerges in the $0.2Hz$ to $2Hz$ frequency band. The repetition of the dotted pattern results from the differ-

ent positions of the dots in the LRM and overset simulation in this frequency band. Furthermore, the previously described area around $0.2Hz$ between $10s$ and $16s$ features a significant error in the $C^{Mx,aer}$ predicted by the LRM. Additional discrepancies between the LRM and the overset simulation exist below $0.2Hz$. Unfortunately, the randomness of the fluid flow prevents an interpretation of these discrepancies.

In total, the LRM predicts $C^{T,aer}$, $C^{Mx,aer}$, and $C^{My,aer}$ comparable to the overset simulation for the simulated approach of a MBB Bo 105 onto a NatoGD. Nevertheless, the LRM does not converge towards the overset simulation with increasing reach factor a. Furthermore, significant discrepancies in $C^{T,aer}$ for the flight over the ship's deck exist. Additionally, the $C^{Mx,aer}$ and $C^{My,aer}$ show significant errors compared to the overset reference.

Despite these differences to the overset simulation, the LRM predicts the dotted pattern in $C^{T,aer}$, $C^{Mx,aer}$, and $C^{My,aer}$ in the $0.2Hz$ to $2Hz$ band. Additionally, the time series in Fig. 9.4 indicates that the LRM captures the essential influence of the ship onto the MBB Bo 105's main rotor. These findings indicate the LRM's capability to capture the ship-rotorcraft interaction. To highlight the influence of wake and ship geometry, Fig. XIII in Appendix VIII compares the rotor's forces and moments of an approach with and without the ship wake present. The comparison indicates that without the wake, the LRM predicts larger forces and moments created by the interaction of the rotorcraft with the ship geometry. Appendix VIII includes a detailed description. Besides the LRM's capability to predict the interaction, the comparison of the wall clock time of the overset reference and the real-time capable LRM simulation on eight Nvidia A100 GPGPUs indicates that the LRM is approximately 75 times faster than the overset simulation.

Prior to its application in a training simulator, a piloted evaluation is necessary to validate the LRM. Collecting the feedback of naval rotorcraft pilots on the realism of the ship-rotorcraft interaction predicted by the LRM allows assessing if the LRM accomplishes the intended purpose in the intended environment. However, a corresponding piloted simulator study is outside this thesis's scope.

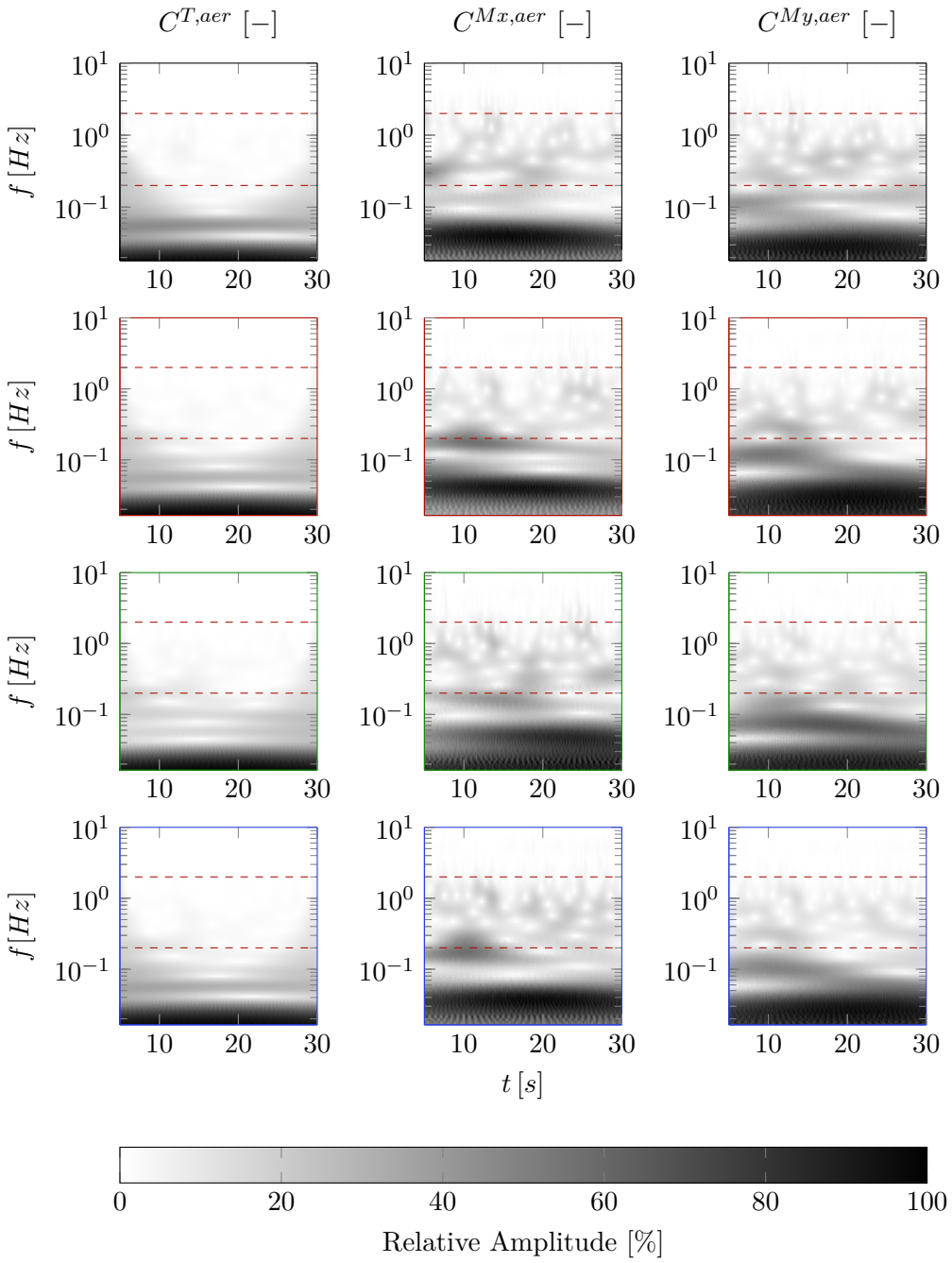


Figure 9.5: Amplitude of different frequencies of $C^{T,aer}$, $C^{Mx,aer}$, and $C^{My,aer}$ during the approach. Calculated by a continuous wavelet transform using complex Morlet wavelets with a center frequency and bandwidth of $1Hz$. Red lines (- - -) mark the 0.02 to $2Hz$ band. First row shows the results of the overset simulation in the (—) axis, second to last row show calculations with varying reach factor at $N^{LRM} = 64$. $a = 1.25$ in the (—) axis, $a = 1.5$ in the (—) axis, and $a = 2$ in the (—) axis from top to bottom.

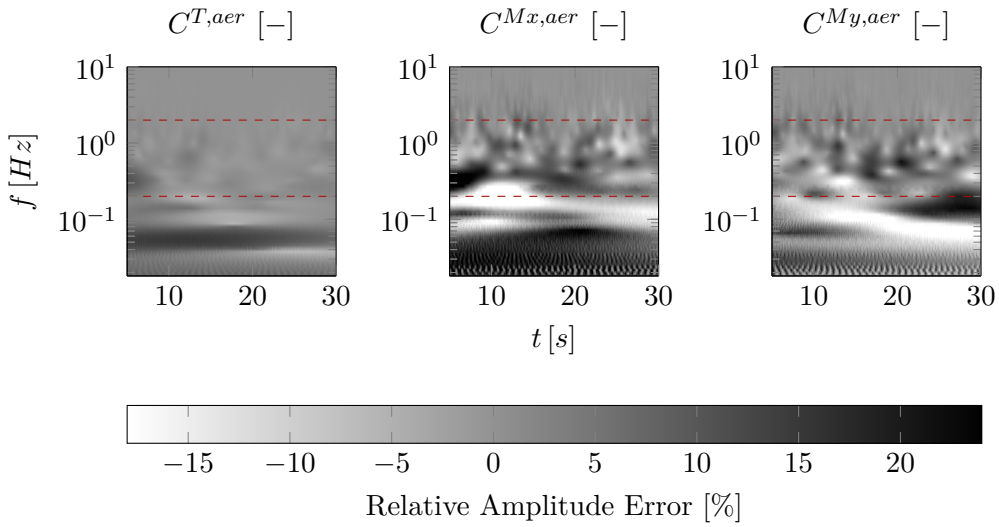


Figure 9.6: Relative error in amplitude of different frequencies of $C^{T,aer}$, $C^{Mx,aer}$, and $C^{My,aer}$ during the approach. Calculated by subtracting the amplitudes of the LRM with $N^{LRM} = 64$ and $a = 1.25$ from the amplitudes of the overset simulation. All amplitudes calculated by a continuous wavelet transform using complex Morlet wavelets with a center frequency and bandwidth of $1Hz$. Red lines (- - -) mark the 0.02 to $2Hz$ band.

10 Summary and Conclusions

Summary

This thesis describes, verifies, and validates a novel approach to model the ship-rotorcraft dynamic interface. The Local Recalculation Model uses the LBM to recalculate the ship's and rotorcraft's combined wake close to the rotor. As the model focuses on piloted flight simulation, it must have firm real-time capability while predicting the physical phenomena of ship-rotorcraft interaction to a degree sufficient for pilot training.

- *Model overview*: This thesis splits the ship-rotorcraft dynamic interface into multiple domains coupled by interfaces. This decomposition allows the LRM to use a combination of models and solvers to predict the ship-rotorcraft interaction. Central to the LRM is a cubic domain with the rotorcraft at the center. Inside the domain, wall boundary conditions represent the geometry of the fuselage and the ship. Prescribing the precomputed wake of the ship at the front and side faces of the domain transports the information from the ship wake upstream of the rotor into the LRM domain. The remaining sides allow free in- and outflow of fluid using impedance boundary conditions. The geometry and information of the upstream wake allow the LBM solver to compute the fluid flow through the rotor flying in the ship wake. A BEM based rotorcraft flight dynamics model calculates the rotor forces and moments based on the rotor inflow extracted from the LRM domain. The LRM imposes the rotor forces and moments onto the cells that form the rotor disk in its domain. This exchange closes the loop of interactions between the LRM and rotorcraft flight dynamics model. Therefore, the LRM captures both directions of interaction between the ship and rotorcraft. Thus, it is considered a two-way coupled ship-rotorcraft dynamic interface model. Nevertheless, the size of the LRM domain restricts the distance over which the model captures the interaction.

- *Ship wake*: Simulating a stand-alone NatoGD air wake using LBM provides a suitable ship wake for the LRM to extract the fluid state imposed at the front and side faces of the domain. The used LBM captures the turbulent nature of the ship wake and converges for resolutions above $N^{NatoGD} > 364$. Nevertheless, comparing velocity distributions at selected points above the landing deck with measurement data shows different distributions even with resolutions $N^{NatoGD} > 364$. However, this thesis attributes these differences to the LBM simulating the ship wake on the detail level of a LES but without a wall model that captures the

fine-scaled development of turbulence in boundary layers. In general, the results verify that the used LBM solver in combination with an ALE formulation gives adequate predictions of the NatoGD's wake for stationary and moving ships.

- *Rotorcraft free flight*: A series of performed tests verify the LRM's compliance with the requirements before assessing the LRM's validity. The comparison of the predicted inflow of the LRM for an isolated rotor with the Pitt-Peters model as a reference reveals a higher cosine component of the inflow predicted by the LRM. Therefore, a higher lateral input is required for stationary forward flight, which has to be evaluated by piloted for realism. Furthermore, the LRM does not improve with resolution or domain size at small forward flight speeds $u_{1,g}^{mr} < 3\frac{m}{s}$. Nevertheless, the resulting inflow velocities at these flight speeds are similar to the Pitt-Peters reference. An evaluation of the predicted power, controls, and attitudes to flight test data assesses the inflow velocities in the context of rotorcraft flight. Therefore, this thesis compares the trim states and reaction to doublet inputs of the LRM, Pitt-Peters, and flight test measurements. In this investigation, the LRM gives predictions comparable to the proven Pitt-Peters inflow model. Furthermore, the predictions are in reasonable agreement with the flight test data.

- *Ground and wall effects*: In contrast to the prediction of free flight, the LRM fails to predict the ground effect in hover and forward flight. In hover, the model predicts the ground effect in agreement with algebraic ground effect models for low heights over the ground but fails for distances above 0.8 rotor radii. For heights above the ground greater than one rotor radius, the LRM predicts mean inflow velocities higher than for hover out of ground effect. This thesis attributes the errors in the ground effect in hover to the small distance between the boundary conditions in the LRM domain and the rotor. The convergence of the predicted mean inflow velocities towards the analytical models for high reach factors a supports this assessment. In contrast, the LRM fails to predict the ground effect in forward flight independent of the reach factor a. Furthermore, the LRM does not converge with the reach factor a in this scenario. Therefore, the LRM does not fulfill the requirement to predict the ground effect. Nevertheless, different outflow boundary conditions in the LRM might improve the prediction of the ground effect.

  Despite the failure to predict the ground effect, the LRM proves able to predict recirculations of the rotor wake behind the backward-facing step formed by the hangar. Nevertheless, a detailed analysis or comparison with measurement data is outside of this thesis's scope.

- *Computational performance*: Performance measurements of the LRM on Nvidia A100 GPGPUs prove that the model is real-time capable with a resolution of $N^{LRM} = 64$ and a reach factor of $a = 1.25$. Nevertheless, the scaling indicates that increasing the real-time capable resolution requires a significant performance increase. In contrast, incremental performance improvements allow increasing the

reach factor a of the LRM while preserving real-time capability. The resulting time steps, dominated by the LBM solver, fulfill the requirements for level D flight simulators.

- *Ship-rotorcraft interaction*: To validate LRM for ship-rotorcraft interaction, an approach of a MBB Bo 105 onto a moving, full-scale NatoGD is investigated. The overset technique is used as a reference simulation to compensate for the lack of measurement data for the approach. As the overset simulation uses the identical domain setups as the LRM and NatoGD simulation, a direct comparison reveals the influence of the simplifications introduced by the LRM. In general, the LRM manages to predict the fluctuations of the rotor forces and moments caused by the interaction of rotorcraft and ship. Although the LRM predicts comparable patterns in the frequency distribution of the forces and moments compared to the overset reference, the time series of the forces and moments of the LRM does not match the corresponding values of the overset simulation. Nevertheless, the turbulent nature of the ship wake introduces randomness. Therefore, agreement in the frequency distribution patterns of the rotor forces and moments can be sufficient to provide a realistic feel of the interaction to the pilot. Further validating the LRM for ship-rotorcraft interaction requires piloted simulator studies.

Conclusions

This thesis investigates the LRM as a ship-rotorcraft dynamic interface model. The LRM uses local recalculation of the combined ship's and rotorcraft's wake within a small domain around the rotor to model the entire interaction with computational efficiency. The verification and validation of the LRM permits the following conclusions:

- *Real-time capability*: The computational efficiency gained by using local recalculation within a small domain around the rotor, in contrast to simulating the entire interaction in a large domain, resulted in a speedup factor of approximately 75. As a result, the performance measurements on eight Nvidia A100 GPGPUs indicate that the LRM is real-time capable with a resolution $N^{LRM} = 64$ and a reach factor of $= 1.25$. Furthermore, the LRM, representing a two-way coupled ship-rotorcraft dynamic interface model, fulfills the computational requirements for piloted simulator training.

- *Realism of the ship wake*: The LBM solver used for the LRM gives adequate predictions of the wake of a NatoGD in a headwind. Tabulation of the wake allows the LRM to combine the ship wake with the rotor wake to model the ship-rotorcraft interactions.

- *Realism of the rotorcraft flight dynamics*: The real-time capable LRM setup predicts the stationary and dynamic behavior of the investigated MBB Bo 105 in free flight. Furthermore, the LRM captures physical phenomena like recirculation and wall effects central to the ship-rotorcraft interaction. In its current version,

the LRM should not be used to predict the ground effect in hover and forward flight, as deviations from established models are high.

- *Realism of the ship-rotorcraft interactions*: In an approach of a MBB Bo 105 onto a moving NatoGD, the LRM predicted similar frequency distribution of fluctuations in the rotor forces and moments as an overset simulation. Because of the turbulent ship wake, the frequency distribution might be sufficient to provide a realistic feel to the pilot.

To this point, the results affirm both central questions formulated in the introduction of this thesis. The physics-based LRM developed in this thesis captures the ship-rotorcraft interactions and runs in real-time on contemporary hardware. However, piloted evaluation in flight simulators is necessary to assert that the LRM adds the realism the pilots require in simulations of rotorcraft ship deck operations.

11 Outlook

One apparent next step is the integration of the LRM into flight simulators. A successful integration allows further investigations of the LRM's capability to predict ship-rotorcraft interactions relevant to pilots. Integration into simulators and piloted studies allows the transition into the V-model's "Operation" step, representing the system's use in the intended environment with the intended purpose. Additionally, this capability would allow assessing the impact of the LRM on pilot training for naval rotorcraft operations and prove the LRM's capability to investigate ship rotorcraft operation limits. Furthermore, the flow field of the LRM can improve the visual representation of brown- and white-out or vegetation moving in the rotor wake for flight simulation as envisioned by Friedmann [52].

Despite the promising results, investigating the LRM's deficiencies in predicting the ground effect is advised. A comparison to experimental data for wall and recirculation effects can reveal the LRM's performance in the presence of these effects. Furthermore, the atmospheric boundary layer of the wind significantly alters the ship wake [110, 32]. These investigations indicate that including the atmospheric boundary layer in the ship wake calculation can improve the prediction of ship-rotorcraft interactions.

Additionally, the discretization of the rotor forces and moments used to exchange information between the LRM and the rotorcraft flight dynamics model limits the distributions to ones it can represent. Therefore, it filters the inhomogeneity of the distributions of thrust and inflow. Horvat investigates the influence of different discretizations for the rotor forces and inflow [111].

Furthermore, the capabilities of the LRM to predict the interaction of the rotor with its surroundings are not limited to ships. The LRM already supports an unlimited number of rotors within its domain [3]. Therefore, the LRM allows investigations of rotor-rotor interactions, as done by Ashok [112]. The multi-rotor capability enables the LRM to investigate new Vertical Take-off and Landing (VTOL) configurations used for drones or urban air mobility. The LRM can also be applied to investigate the interactions between rotors and mountains or urban structures. Given success, the LRM could support pilot training in search and rescue missions or special operations that feature landing in confined spaces like inside cities or on helipads.

Bibliography

[1] J. Bludau, J. Rauleder, L. Friedmann, and M. Hajek. Real-time simulation of dynamic inflow using rotorcraft flight dynamics coupled with a lattice-boltzmann based fluid simulation. In *55th AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, January 2017. doi:10.2514/6.2017-0050.

[2] J. Bludau, J. Rauleder, and M. Hajek. Validation of a dynamic inflow model based on a flight dynamics model and a lattice-boltzmann fluid solver using flight test data. In *43rd European Rotorcraft Forum*. Associazione Italiana di Aeronautica e Astronautica (AIDAA), 2017.

[3] J. Bludau, J. Rauleder, and M. Hajek. Solving the ship-rotorcraft dynamic interface problem using lattice-boltzmann aerodynamics two-way coupled with blade element based flight dynamics. In *Proceedings of the Vertical Flight Society 77th Annual Forum*. The Vertical Flight Society, May 2021. doi:10.4050/f-0077-2021-16725.

[4] L. Osborne, J. Brummond, R. Hart, and M. M. Zarean. Clarus: Concept of operations. Technical Report FHWA-JPO-05-072, Iteris Inc. for the Federal Highway Administration, 2005.

[5] J. Bludau, J. Rauleder, L. Friedmann, and M. Hajek. Real-time simulation of rotor inflow using a coupled flight dynamics and fluid mechanics simulation. In *Deutscher Luft- und Raumfahrtkongress 2016, Braunschweig*. Deutsche Gesellschaft für Luft- und Raumfahrt, 2016.

[6] W. D. Lewis. An aeroelastic model structure investigation for a manned real-time rotorcraft simulation. In *Proceedings of the American Helicopter Society 49th Annual Forum*. American Helicopter Society, 1993.

[7] D. M. Pitt and D. A. Peters. Theoretical predictions of dynamic-inflow derivatives. *Vertica*, 5:21–34, 1981.

[8] D. M. Roper, I. Owen, G. D. Padfield, and S. J. Hodge. Integrating cfd and piloted simulation to quantify ship-helicopter operating limits. *The Aeronautical Journal*, 110(1109):419–428, July 2006. doi:10.1017/s0001924000001329.

[9] I. Owen, R. Lee, A. Wall, and N. Fernandez. The nato generic destroyer – a shared geometry for collaborative research into modelling and simulation of shipboard helicopter launch and recovery. *Ocean Engineering*, 228:108428, May 2021. doi:10.1016/j.oceaneng.2020.108428.

[10] G. D. Padfield. *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modelling*. Wiley, 2nd edition, January 2007. doi:10.1002/9780470691847.

[11] A. Sydney, J. Ramsey, and K. Kimmel. Experimental investigation of the turbulent aerodynamic environment produced by a generic ship. In *54th AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, January 2016. doi:10.2514/6.2016-1071.

[12] A. Wall, R. Lee, H. Barber, and K. McTaggart. The nato generic destroyer - a shared geometry for collaborative research into modelling and simulation of shipboard launch and recovery: source data posting on open science canada. Technical report, National Research Council Canada, Aerodynamics Laboratory, 2021.

[13] I. Oruc, J. F. Horn, J. Shipman, and S. Polsky. Towards real-time pilot-in-the-loop cfd simulations of helicopter/ship dynamic interface. *International Journal of Modeling, Simulation, and Scientific Computing*, 08(04), December 2017. doi:10.1142/s179396231743005x.

[14] D. T. McRuer. Interdisciplinary interactions and dynamic systems integration. *International Journal of Control*, 59(1):3–12, January 1994. doi:10.1080/00207179408923067.

[15] R. G. Lee and S. J. Zan. Wind tunnel testing of a helicopter fuselage and rotor in a ship airwake. *Journal of the American Helicopter Society*, 50(4):326–337, October 2005. doi:10.4050/1.3092869.

[16] K. Shin and P. Ramanathan. Real-time computing: a new discipline of computer science and engineering. *Proceedings of the IEEE*, 82(1):6–24, 1994. doi:10.1109/5.259423.

[17] H. Kopetz. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Real-Time Systems Series. Springer US, 2011.

[18] U.S Department of Transportation. Ac 120-63 - helicopter simulator qualification. Technical report, Federal Aviation Administration, 1994.

[19] J. G. Leishman. *Principles of Helicopter Aerodynamics*. Cambridge University Press, New York, 2nd edition, 2008.

[20] W. Johnson. *Helicopter Theory*. Dover Publications, New York, 2nd edition, 1994.

[21] D. A. Peters. How dynamic inflow survives in the competitive world of rotorcraft aerodynamics. *Journal of the American Helicopter Society*, 54(1):11001–1100115, January 2009. doi:10.4050/jahs.54.011001.

[22] W. Kinner. Die kreisförmige Tragfläche auf potentialtheorischer Grundlage. *Ingenier-Archive*, VIII:47–80, 1937.

[23] D. A. Peters, D. D. Boyd, and C. J. He. Finite-state induced-flow model for rotors in hover and forward flight. *Journal of the American Helicopter Society*, 34(4):5–17, October 1989. doi:10.4050/jahs.34.5.

[24] J. Zhao, J. V. R. Prasad, and D. A. Peters. Rotor wake distortion model for helicopter maneuvering flight. *Journal of the American Helicopter Society*, 49(4), October 2004.

[25] F. Cardito, M. Gennaretti, R. Gori, J. Serafini, and G. Bernardini. A space-time accurate finite-state inflow model for aeroelastic applications. In *Proceedings of the American Helicopter Society 72nd Annual Forum*. American Helicopter Society, 05 2016.

[26] R. T. N. Chen. A survey of nonuniform inflow models for rotorcraft flight dynamics and control applications. Technical Memorandum 102219, National Aeronaitics and Space Administration, 1989.

[27] S. J. Zan. On aerodynamic modelling and simulation of the dynamic interface. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 219(5):393–410, May 2005. doi:10.1243/095441005x30315.

[28] G. Syms. Simulation of simplified-frigate airwakes using a lattice-boltzmann method. *Journal of Wind Engineering and Industrial Aerodynamics*, 96(6–7):1197–1206, June 2008. doi:10.1016/j.jweia.2007.06.040.

[29] S. Zan. Surface flow topology for a simple frigate shape. *Canadian Aeronautics and Space Journal*, 47(1):33 – 43, 2001.

[30] E. Kurban, S. Ashok, and J. Rauleder. Computationally efficient ship airwake simulations for rotorcraft shipboard operations using a gpu-accelerated lattice-boltzmann solver. In *48th European Rotorcraft Forum*. CEAS, 2022.

[31] B. Cheney and S. Zan. Cfd code validation data and flow topology for the technical co-operation program aer-tp2 simple frigate shape. Technical report, Institute for Aerospace Research (Canada), 1999.

[32] D. Seth, J. G. Leishman, E. Gnanamanickam, and Z. Zhang. Time-resolved ship airwake measurements in a simulated atmospheric boundary layer. *Journal of Aircraft*, 58(3):624–649, May 2021. doi:10.2514/1.c035886.

[33] N. Zhu, Z. Zhang, E. Gnanamanickam, and J. Gordon Leishman. Space-time characterization of ship airwakes. *AIAA Journal*, 61(2):681–697, February 2023. doi:10.2514/1.j062093.

[34] S. Ashok and J. Rauleder. Nato generic destroyer moving-ship airwake validation and rotor-ship dynamic interface computations using immersed boundary lattice-boltzmann method. In *Proceedings of the Vertical Flight Society 79th Annual Forum*. The Vertical Flight Society, May 2023. doi:10.4050/f-0079-2023-18125.

[35] S. J. Hodge, S. J. Zan, D. M. Roper, G. D. Padfield, and I. Owen. Time-accurate ship airwake and unsteady aerodynamic loads modeling for maritime helicopter simulation. *Journal of the American Helicopter Society*, 54(2):22005–2200516, April 2009. doi:10.4050/jahs.54.022005.

[36] S. J. Hodge, J. S. Forrest, G. D. Padfield, and I. Owen. Simulating the environment at the helicopter-ship dynamic interface: research, development and application. *The Aeronautical Journal*, 116(1185):1155–1184, November 2012. doi:10.1017/s0001924000007545.

[37] I. Owen, M. D. White, G. D. Padfield, and S. J. Hodge. A virtual engineering approach to the ship-helicopter dynamic interface – a decade of modelling and simulation research at the university of liverpool. *The Aeronautical Journal*, 121(1246):1833–1857, September 2017. doi:10.1017/aer.2017.102.

[38] S. A. Polsky, C. Wilkinson, J. Nichols, D. Ayers, J. Mercado-Perez, and T. S. Davis. Development and application of the safedi tool for virtual dynamic interface ship airwake analysis. In *54th AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, January 2016. doi:10.2514/6.2016-1771.

[39] A. Sharma, J. Xu, A. K. Padthe, P. P. Friedmann, and K. Duraisamy. Simulation of maritime helicopter dynamics during approach to landing with time-accurate wind-over-deck. In *AIAA Scitech 2019 Forum*. American Institute of Aeronautics and Astronautics, January 2019. doi:10.2514/6.2019-0861.

[40] J. S. Forrest and I. Owen. An investigation of ship airwakes using detached-eddy simulation. *Computers & Fluids*, 39(4):656–673, April 2010. doi:10.1016/j.compfluid.2009.11.002.

[41] C. Crozon, R. Steijl, and G. N. Barakos. Numerical study of helicopter rotors in a ship airwake. *Journal of Aircraft*, 51(6):1813–1832, November 2014. doi:10.2514/1.c032535.

[42] W. Yuan, A. Wall, and R. Lee. Combined numerical and experimental simulations of unsteady ship airwakes. *Computers & Fluids*, 172:29–53, August 2018. doi:10.1016/j.compfluid.2018.06.006.

[43] P. Sagaut. *Large Eddy Simulation for Incompressible Flows - An Introduction*. Springer-Verlag, 3rd edition, 2005. doi:10.1007/b137536.

[44] G. Rubenstein, D. M. Moy, A. Sridharan, and I. Chopra. A python-based framework for real-time simulation using comprehensive analysis. In *Proceedings of the American Helicopter Society 72nd Annual Forum*. American Helicopter Society, 2016.

[45] C. He and J. Zhao. Modeling rotor wake dynamics with viscous vortex particle method. *AIAA Journal*, 47(4):902–915, April 2009. doi:10.2514/1.36466.

[46] J. Zhao and C. He. A viscous vortex particle model for rotor wake and interference analysis. *Journal of the American Helicopter Society*, 55(1):12007–1200714, January 2010. doi:10.4050/jahs.55.012007.

[47] U. Saetti. Real-time simulation of a shipborne rotor via linearized state-space free-vortex wake models. *Journal of Aircraft*, page 1–9, February 2024. doi:10.2514/1.c037389.

[48] D. Hänel. *Molekulare Gasdynamik*. Springer-Verlag, Berlin Heidelberg, 1st edition, 2004.

[49] D. A. Wolf-Gladrow. *Lattice-Gas Cellular Automata and Lattice Boltzmann Models - An Introduction*. Springer-Verlag, Berlin Heidelberg, 1st edition, 2005.

[50] M. C. Sukop and D. T. J. Thorne. *Lattice Boltzmann Modeling: An Introduction for Geoscientists and Engineers*. Springer Publishing Company Incorporated, 1st edition, 2007.

[51] L. Friedmann, P. Ohmer, and M. Hajek. Real-time simulation of rotorcraft downwash in proximity of complex obstacles using grid-based approaches. In *70th Annual Forum of the American Helicopter Society*. The American Helicopter Society, 2014.

[52] L. M. Friedmann. *Echtzeit-Simulation des Rotorabwinds von Hubschraubern und Kipprotor-Flugzeugen mit der Lattice-Boltzmann Methode*. Phd tesis, Technische Universität München, 2017.

[53] B. Horvat, M. Hajek, and J. Rauleder. Computational flight path analysis of a helicopter in an offshore wind farm using a lattice-boltzmann method. In *AIAA Scitech 2021 Forum*. American Institute of Aeronautics and Astronautics, January 2021. doi:10.2514/6.2021-1827.

[54] S. G. Ashok and J. Rauleder. Towards real-time coupled ship–rotorcraft interactional simulations using gpu-accelerated lattice-boltzmann method. In *Vertical Flight Society's 80th Annual Forum*. Vertical Flight Society, May 2024.

[55] S. B. Pope. *Turbulent Flows*. Cambridge University Press, August 2000. doi:10.1017/cbo9780511840531.

[56] P. P. Friedmann and K. Duraisamy. A surrogate based framework for helicopter/-ship dynamic interface. Technical report, Office of Naval Research, 2019.

[57] M. Dietz, C. Maucher, and D. Schimke. Addressing today's aeromechanic questions by industrial answers. In *American Helicopter Society Specialists' Conference on Aeromechanics, San Francisco, CA*, 2010.

[58] J. Bludau. *Two-Way Coupled Real-time Simulation of Rotorcraft Flight Dynamics Interacting with a Lattice-Boltzmann Based Flow Model*. Master thesis, Technical University of Munich, 2016.

[59] S. Simonis, M. Frank, and M. J. Krause. On relaxation systems and their relation to discrete velocity boltzmann models for scalar advection–diffusion equations. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 378(2175), June 2020. doi:10.1098/rsta.2019.0400.

[60] S. Simonis, M. Frank, and M. J. Krause. Constructing relaxation systems for lattice boltzmann methods. *Applied Mathematics Letters*, 137, March 2023. doi:10.1016/j.aml.2022.108484.

[61] M. Junk, A. Klar, and L.-S. Luo. Asymptotic analysis of the lattice boltzmann equation. *Journal of Computational Physics*, 210(2):676–704, December 2005. doi:10.1016/j.jcp.2005.05.003.

[62] M. Krause, S. Avis, H. Kusumaatmaja, D. Dapelo, M. Gaedtke, N. Hafen, M. Haußmann, J. Jeppener-Haltenhoff, L. Kronberg, A. Kummerländer, J. Marquardt, T. Pertzel, S. Simonis, R. Trunk, M. Wu, and A. Zarth. Openlb release 1.4: Open source lattice boltzmann code, 2020. doi:10.5281/ZENODO.4279263.

[63] P. Lallemand, L.-S. Luo, M. Krafczyk, and W.-A. Yong. The lattice boltzmann method for nearly incompressible flows. *Journal of Computational Physics*, 431, April 2021. doi:10.1016/j.jcp.2020.109713.

[64] J. Latt, C. Coreixas, J. Beny, and A. Parmigiani. Efficient supersonic flow simulations using lattice boltzmann methods based on numerical equilibria. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 378(2175), June 2020. doi:10.1098/rsta.2019.0559.

[65] W. Feng, B. He, A. Song, Y. Wang, M. Zhang, and W. Zhang. *Compressible Lattice Boltzmann Method and Applications*, page 17–26. Springer Berlin Heidelberg, 2010. doi:10.1007/978-3-642-11842-5_3.

[66] A. Hsu, T. Yang, I. Lopez, and A. Ecer. *A Review of Lattice Boltzmann Models for Compressible Flows*, page 19–28. Elsevier, 2004. doi:10.1016/b978-044451612-1/50003-2.

[67] M. Krause, A. Mink, R. Trunk, F. Klemens, M.-L. Maier, M. Mohrhard, A. Claro Barreto, M. Haußmann, M. Gaedtke, and J. Ross-Jones. Openlb release 1.2: Open source lattice boltzmann code, 2018. doi:10.5281/ZENODO.3625960.

[68] P. L. Bhatnagar, E. P. Gross, and M. Krook. A model for collision processes in gases. I. small amplitude processes in charged and neutral one-component systems. *Physical Review*, 94(3):511–525, May 1954. doi:10.1103/physrev.94.511.

[69] D. d'Humières. Multiple–relaxation–time lattice boltzmann models in three dimensions. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 360(1792):437–451, March 2002. doi:10.1098/rsta.2001.0955.

[70] S. Chapman and T. Cowling. *The Mathematical Theory of Non-uniform Gases: An Account of the Kinetic Theory of Viscosity, Thermal Conduction and Diffusion in Gases*. Cambridge Mathematical Library. Cambridge University Press, 1990.

[71] K. Mattila, T. Puurtinen, J. Hyväluoma, R. Surmas, M. Myllys, T. Turpeinen, F. Robertsén, J. Westerholm, and J. Timonen. A prospect for computing in porous materials research: Very large fluid flow simulations. *Journal of Computational Science*, 12:62–76, January 2016. doi:10.1016/j.jocs.2015.11.013.

[72] S. Lee, J. Gounley, A. Randles, and J. S. Vetter. Performance portability study for massively parallel computational fluid dynamics application on scalable heterogeneous architectures. *Journal of Parallel and Distributed Computing*, 129:1–13, July 2019. doi:10.1016/j.jpdc.2019.02.005.

[73] S. Hou, J. Sterling, S. Chen, and G. D. Doolen. A lattice boltzmann subgrid model for high reynolds number flow. *Pattern formation and lattice gas automata*, 6:151–166, 1996.

[74] A. N. Kolmogorov. Local structure of turbulence in an incompressible fluid at very high reynolds numbers. *Dokl. Akad. Nauk SSSR*, 31:99–101, 1941.

[75] J. Smagorinksy. General circulation experiments with the primitive equations: I. the basic experiment*. *Monthly Weather Review*, 91(3):99–164, March 1963. doi:10.1175/1520-0493(1963)091<0099:gcewtp>2.3.co;2.

[76] J. W. Deardorff. A numerical study of three-dimensional turbulent channel flow at large reynolds numbers. *Journal of Fluid Mechanics*, 41(2):453–480, April 1970. doi:10.1017/s0022112070000691.

[77] N. Thürey. *Physically based Animation of Free Surface Flows with the Lattice Boltzmann Method*. Phd tesis, Universität Erlangen-Nürnberg, 2007.

[78] Z. Guo, C. Zheng, and B. Shi. Discrete lattice effects on the forcing term in the lattice boltzmann method. *Physical Review E*, 65(4), April 2002. doi:10.1103/physreve.65.046308.

[79] M. Mohrhard, G. Thäter, J. Bludau, B. Horvat, and M. J. Krause. Auto-vectorization friendly parallel lattice boltzmann streaming scheme for direct

addressing. *Computers & Fluids*, 181:1–7, March 2019. doi:10.1016/j.compfluid.2019.01.001.

[80] M. Wittmann, T. Zeiser, G. Hager, and G. Wellein. Comparison of different propagation steps for lattice boltzmann methods. *Computers & Mathematics with Applications*, 65(6):924–935, March 2013. doi:10.1016/j.camwa.2012.05.002.

[81] P. Bailey, J. Myre, S. Walsh, D. Lilja, and M. Saar. Accelerating lattice boltzmann fluid flow simulations using graphics processors. In *2009 International Conference on Parallel Processing*. IEEE, September 2009. doi:10.1109/icpp.2009.38.

[82] M. Geier and M. Schönherr. Esoteric twist: An efficient in-place streaming algorithmus for the lattice boltzmann method on massively parallel hardware. *Computation*, 5(4):19, March 2017. doi:10.3390/computation5020019.

[83] O. Filippova and D. Hänel. Boundary-fitting and local grid refinement for lattice-bgk models. *International Journal of Modern Physics C*, 09(08):1271–1279, December 1998. doi:10.1142/s012918319800114x.

[84] R. Mei, L.-S. Luo, and W. Shyy. An accurate curved boundary treatment in the lattice-boltzmann method. *Journal of Computational Physics*, 155(2):307–330, 1999.

[85] J. Latt, B. Chopard, O. Malaspinas, M. Deville, and A. Michler. Straight velocity boundaries in the lattice boltzmann method. *Physical Review E*, 77(5), May 2008. doi:10.1103/physreve.77.056703.

[86] K. Hu, J. Meng, H. Zhang, X.-J. Gu, D. R. Emerson, and Y. Zhang. A comparative study of boundary conditions for lattice boltzmann simulations of high reynolds number flows. *Computers & Fluids*, 156:1–8, October 2017. doi:10.1016/j.compfluid.2017.06.008.

[87] P. A. Skordos. Initial and boundary conditions for the lattice boltzmann method. *Physical Review E*, 48(6):4823–4842, December 1993. doi:10.1103/physreve.48.4823.

[88] M. B. Schlaffer. *Non-Reflecting Boundary Conditions for the Lattice Boltzmann Method*. Phd thesis, Technical University of Munich, 2013.

[89] Q. Zou and X. He. On pressure and velocity boundary conditions for the lattice boltzmann bgk model. *Physics of Fluids*, 9(6):1591–1598, June 1997. doi:10.1063/1.869307.

[90] J. Donea, A. Huerta, J. Ponthot, and A. Rodríguez-Ferran. Arbitrary lagrangian–eulerian methods. *Encyclopedia of Computational Mechanics*, August 2004. doi:10.1002/0470091355.ecm009.

[91] M. Meldi, E. Vergnault, and P. Sagaut. An arbitrary lagrangian–eulerian approach for the simulation of immersed moving solids with lattice boltzmann method. *Journal of Computational Physics*, 235:182–198, February 2013. doi:10.1016/j.jcp.2012.10.014.

[92] Z.-G. Feng and E. E. Michaelides. The immersed boundary-lattice boltzmann method for solving fluid–particles interaction problems. *Journal of Computational Physics*, 195(2):602–628, April 2004. doi:10.1016/j.jcp.2003.10.013.

[93] C. Peng, O. M. Ayala, and L.-P. Wang. A comparative study of immersed boundary method and interpolated bounce-back scheme for no-slip boundary treatment in the lattice boltzmann method: Part i, laminar flows. *Computers & Fluids*, 192, October 2019. doi:10.1016/j.compfluid.2019.06.032.

[94] J. L. Steger and J. A. Benek. On the use of composite grid schemes in computational aerodynamics. *Computer Methods in Applied Mechanics and Engineering*, 64(1–3):301–320, October 1987. doi:10.1016/0045-7825(87)90045-4.

[95] H. Tang, S. Casey Jones, and F. Sotiropoulos. An overset-grid method for 3d unsteady incompressible flows. *Journal of Computational Physics*, 191(2):567–600, November 2003. doi:10.1016/s0021-9991(03)00331-0.

[96] Z. J. Wang and V. Parthasarathy. A fully automated chimera methodology for multiple moving body problems. *International Journal for Numerical Methods in Fluids*, 33(7):919–938, 2000. doi:10.1002/1097-0363(20000815)33:7<919::aid-fld944>3.0.co;2-g.

[97] Z. Wang, N. Hariharan, and R. Chen. Recent development on the conservation property of chimera. *International Journal of Computational Fluid Dynamics*, 15(4):265–278, November 2001. doi:10.1080/10618560108970033.

[98] P. Lallemand and L.-S. Luo. Lattice boltzmann equation with overset method for moving objects in two-dimensional flows. *Journal of Computational Physics*, 407, April 2020. doi:10.1016/j.jcp.2019.109223.

[99] B. Horvat and I. Yavrucuk. Validation of a mid-fidelity cfd/csd coupling using the lattice-boltzmann method. In *49th European Rotorcraft Forum*. Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V., 2023.

[100] J. O'Rourke, C. B. Chien, T. Olson, and D. Naddor. A new linear algorithm for intersecting convex polygons. *Computer Graphics and Image Processing*, 19:384–391, 1982.

[101] Eurocopter BO 105. https://www.dlr.de/de/forschung-und-transfer/forschungsinfrastruktur/dlr-forschungsflotte/eurocopter-bo-105-1. Accessed: 2023-03-05.

[102] NASA Systems Engineering Handbook. Technical Report NASA/SP-2016-6105 Rev2, National Aeronautics and Space Administration, 2017.

[103] A. Wall, R. Lee, and H. Barber. Analysis of fast-response cobra probe data in highly turbulent wake flows. Technical report, National Research Council of Canada, 2020. doi:10.4224/40002057.

[104] J. Hooper and A. Musgrove. Reynolds stress, mean velocity, and dynamic static pressure measurement by a four-hole pressure probe. *Experimental Thermal and Fluid Science*, 15(4):375–383, November 1997. doi:10.1016/s0894-1777(97)00005-8.

[105] A. Landgrebe and M. J. Cheney. Rotor wakes - key to performance prediction. In *AGARD Conference Proceedings No. 111*. Advisory Group for Aerospace and Development, 1973.

[106] A. K. Cooke and E. W. H. Fitzpatrick. *Helicopter Test and Evaluation*. Whiley, 1st edition, 2002.

[107] W. Johnson. *Rotorcraft Aeromechanics*. Cambridge University Press, April 2013. doi:10.1017/cbo9781139235655.

[108] I. Cheeseman and W. Bennett. The effect of the ground on a helicopter rotor in forward flight. *ARC R&M 3021*, 1955.

[109] J. S. Hayden. The effect of the ground on helicopter hovering power required. In *Proceedings of the American Helicopter Society 32nd Annual Forum*. The American Helicopter Society, 1976.

[110] R. Thedin, S. M. Murman, J. Horn, and S. Schmitz. Effects of atmospheric turbulence unsteadiness on ship airwakes and helicopter dynamics. *Journal of Aircraft*, 57(3):534–546, May 2020. doi:10.2514/1.c035643.

[111] B. Horvat. *Rotor Inflow Modeling using a Lattice-Boltzmann Method based Flow Solver*. Phd thesis under review, Technical University of Munich, 2024.

[112] S. G. Ashok, S. Hosseini, and J. Rauleder. Real-time dynamic inflow model for a coaxial helicopter configuration using the lattice–boltzmann method. In *AIAA SCITECH 2024 Forum*. American Institute of Aeronautics and Astronautics, January 2024. doi:10.2514/6.2024-1718.

Appendix

I Velocity distribution of point (0.3,0.06,0.07) on NatoGD landing deck

This section of the Appendix holds the velocity distribution for the non-moving NatoGD with $N^{NatoGD} = 364$ and $N^{NatoGD} = 546$ at the point $(0.3, 0.06, 0.07)[m]$. These were excluded from the main chapters to ease reading. Furthermore, the dawn conclusions are the same as for point $(0.3, 0, 0.07)[m]$. Therefore, the reader is indicated to section 7.1 for the interpretation.

Apart from the velocity distribution shown in figure I, table I lists the corresponding numerical values for mean, variance, and TKE.

$Variable$	Exp.	$N^{NatoGD} = 364$	$N^{NatoGD} = 546$
$\bar{u}_1\,[\frac{m}{s}]$	2.99	2.37	2.42
$\bar{u}_2\,[\frac{m}{s}]$	-0.26	-0.009	-0.20
$\bar{u}_3\,[\frac{m}{s}]$	-0.98	-0.77	-0.72
$var(u_1)\,[\frac{m^2}{s^2}]$	1.26	1.02	1.12
$var(u_2)\,[\frac{m^2}{s^2}]$	3.24	1.61	2.00
$var(u_3)\,[\frac{m^2}{s^2}]$	2.42	1.56	1.68
$\frac{TKE}{(\boldsymbol{u}^\infty)^2}\,[-]$	0.036	0.022	0.025

Table I: Mean, variance and TKE of the velocity distribution at point $(0.3, 0.06, 0.07)[m]$ for the experiment (Exp.) and simulations with the resolutions $N^{NatoGD} = 364$ and $N^{NatoGD} = 546$.

Furthermore, Fig. VI shows the velocity distribution for the NatoGD pitching at $1Hz$ with an amplitude of $5deg$. The corresponding numerical values are listed in table II.

$Variable$	Exp.	$N^{NatoGD} = 364$
$\bar{u}_1\,[\frac{m}{s}]$	3.02	2.81
$\bar{u}_2\,[\frac{m}{s}]$	-0.51	-0.15
$\bar{u}_3\,[\frac{m}{s}]$	-1.13	-1.35
$var(u_1)\,[\frac{m^2}{s^2}]$	1.33	1.71
$var(u_2)\,[\frac{m^2}{s^2}]$	2.74	1.76
$var(u_3)\,[\frac{m^2}{s^2}]$	2.09	1.72
$\frac{TKE}{(\boldsymbol{u}^\infty)^2}\,[-]$	0.032	0.027

Table II: Mean, variance and TKE of the velocity distribution at point $(0.3, 0.06, 0.07)[m]$ for the experiment (Exp.) and simulations with the resolutions $N^{NatoGD} = 364$ for a pitching motion with $1Hz$ and $5deg$ amplitude.

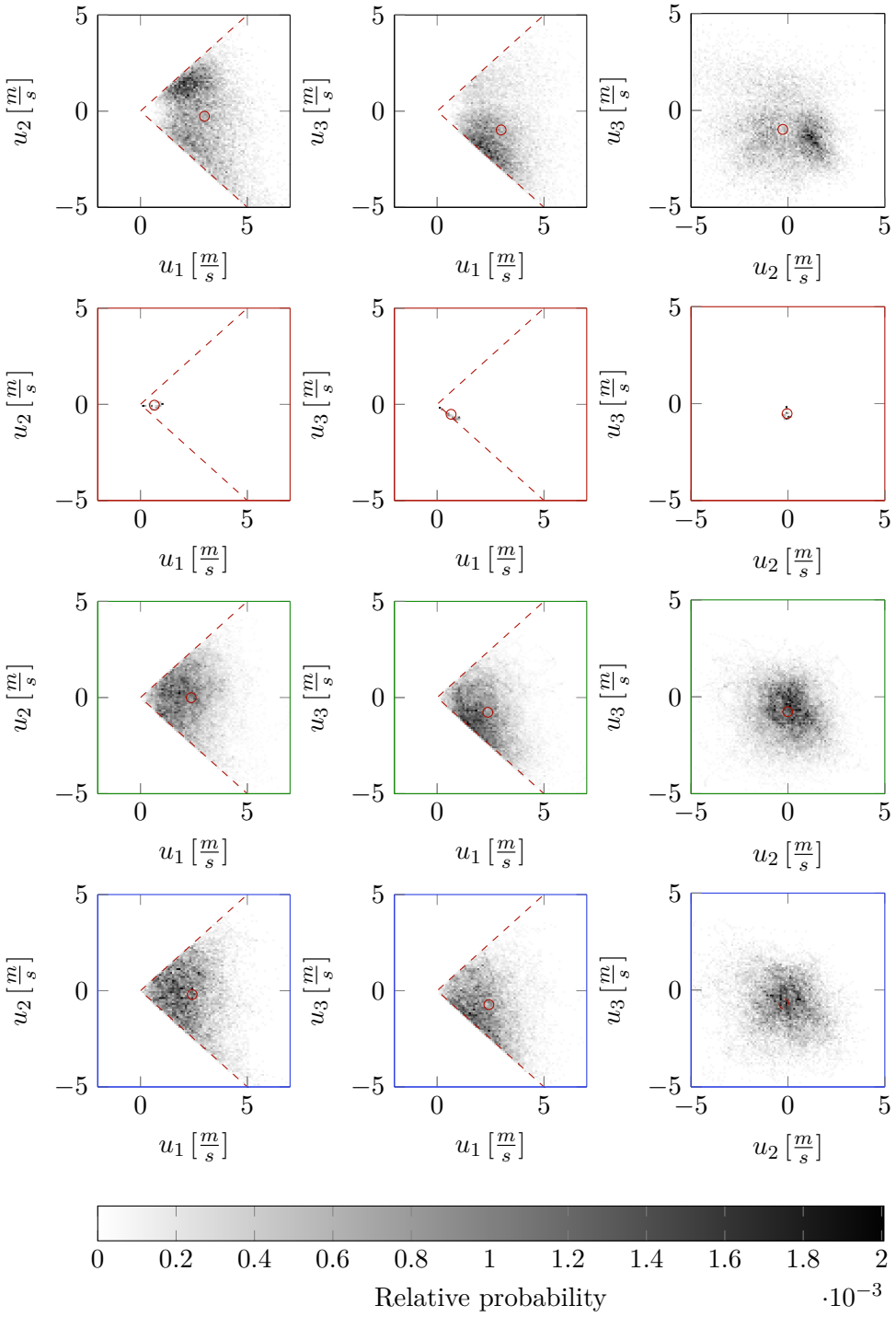


Figure I: Relative probability in bins with a length of $0.1\frac{m}{s}$ for the point $(0.3, 0.06, 0.07)[m]$. Red lines (- - -) mark the cone the COBRA-probes are able to measure within [12, 103, 104]. The mean is indicated in (○) First row are measurements taken from [12], second to last row are calculations with resolutions $N^{NatoGD} = 182$, $N^{NatoGD} = 364$, and $N^{NatoGD} = 546$ from top to bottom.

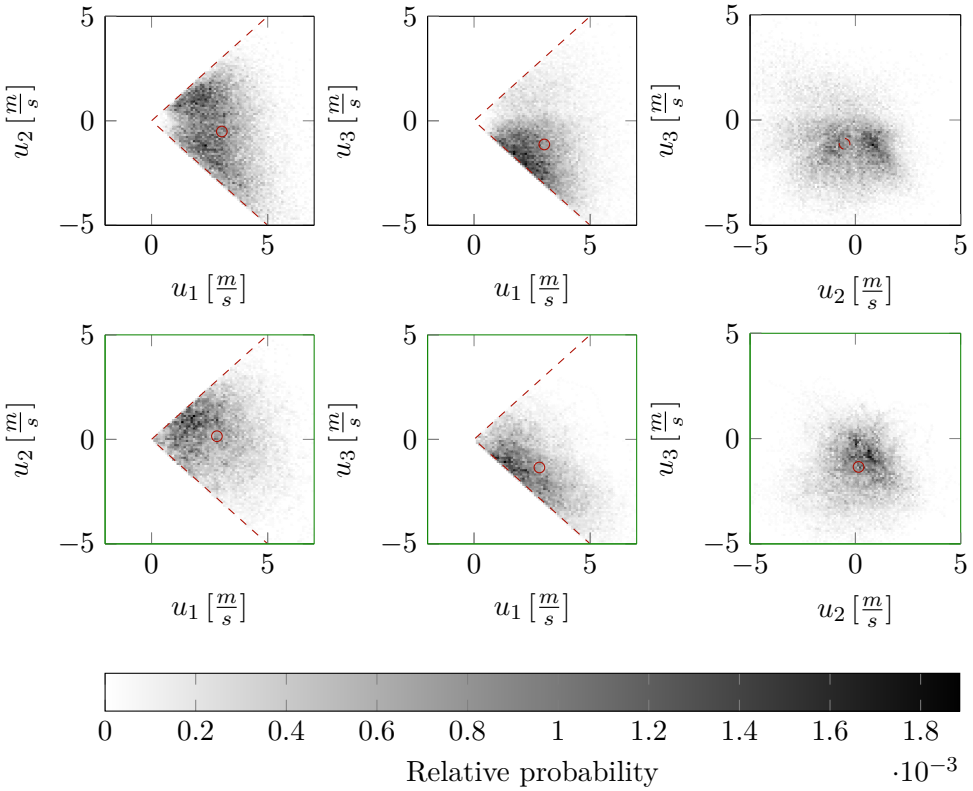


Figure II: Relative probability in bins with a length of $0.1\frac{m}{s}$ for the point $(0.3, 0.06, 0.07)[m]$ for a 1/50th-scale model pitching at $1Hz$ with an amplitude of $5deg$. Red lines (- - -) mark the cone the COBRA-probes are able to measure within[12, 103, 104]. The mean is indicated in (○) First row are measurements taken from [12], second row shows calculations with resolutions $N^{NatoGD} = 364$.

II NatoGD in harmonic heaving motion

This section holds the results of a 1/50th-scale NatoGD simulation including heaving motion at a resolution of $N^{NatoGD} = 364$. The heaving motion has an amplitude of $0.04m$ and a frequency of $1Hz$. Figure III depicts the difference of the mean velocity $\bar{\boldsymbol{u}}$ to the static case $\bar{\boldsymbol{u}}^{static}$ for ensembles at $t = 0s$ in IIIa, $t = 0.25s$ in IIIb, $t = 0.5s$ in IIIc, and $t = 0.75s$ in IIId. The corresponding difference of the TKE to the static case is shown in Fig. IV.

Furthermore, the velocity distribution at the points $(0.3, 0, 0.07)[m]$ and $(0.3, 0.06, 0.07)[m]$ are shown in Fig. 7.8 and VI respectively. The corresponding numerical values are given by the tables III and IV.

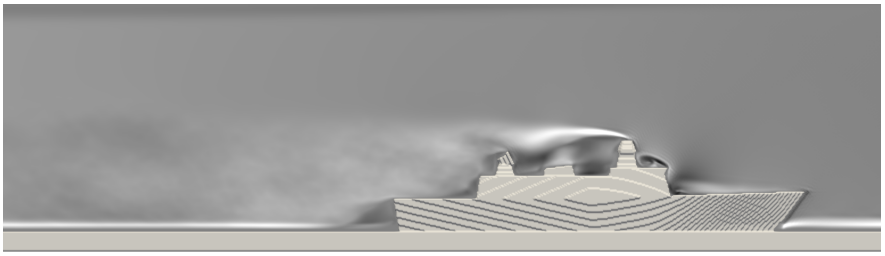
$Variable$	Exp.	$N^{NatoGD} = 364$
$\bar{u}_1\,[\frac{m}{s}]$	2.44	2.63
$\bar{u}_2\,[\frac{m}{s}]$	0.19	-0.005
$\bar{u}_3\,[\frac{m}{s}]$	-1.13	-1.70
$var(u_1)\,[\frac{m^2}{s^2}]$	1.19	1.40
$var(u_2)\,[\frac{m^2}{s^2}]$	1.43	1.47
$var(u_3)\,[\frac{m^2}{s^2}]$	1.36	1.30
$\frac{TKE}{(\boldsymbol{u}^\infty)^2}\,[-]$	0.021	0.022

Table III: Mean, variance and TKE of the velocity distribution at point $(0.3, 0, 0.07)[m]$ for the experiment (Exp.) and simulations with the resolutions $N^{NatoGD} = 364$ for a heaving motion with $1Hz$ and $0.04m$ amplitude.

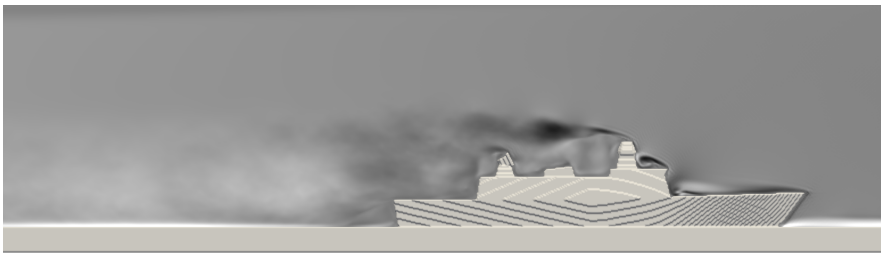
$Variable$	Exp.	$N^{NatoGD} = 364$
$\bar{u}_1\,[\frac{m}{s}]$	2.89	2.16
$\bar{u}_2\,[\frac{m}{s}]$	-0.28	0.29
$\bar{u}_3\,[\frac{m}{s}]$	-0.98	-1.05
$var(u_1)\,[\frac{m^2}{s^2}]$	1.30	1.02
$var(u_2)\,[\frac{m^2}{s^2}]$	3.04	1.08
$var(u_3)\,[\frac{m^2}{s^2}]$	2.16	1.19
$\frac{TKE}{(\boldsymbol{u}^\infty)^2}\,[-]$	0.034	0.017

Table IV: Mean, variance and TKE of the velocity distribution at point $(0.3, 0.06, 0.07)[m]$ for the experiment (Exp.) and simulations with the resolutions $N^{NatoGD} = 364$ for a heaving motion with $1Hz$ and $0.04m$ amplitude.

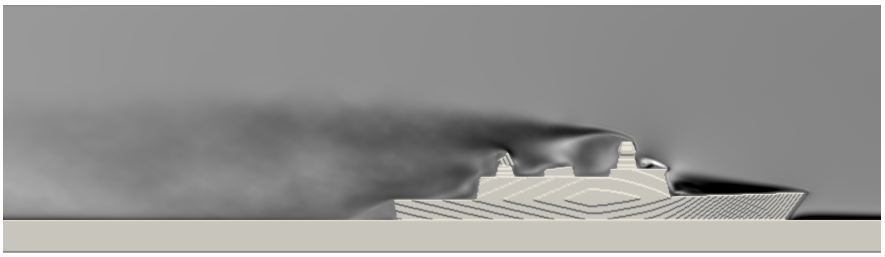
(a) $t = 0$



(b) $t = 0.25$



(c) $t = 0.5$



(d) $t = 0.75$

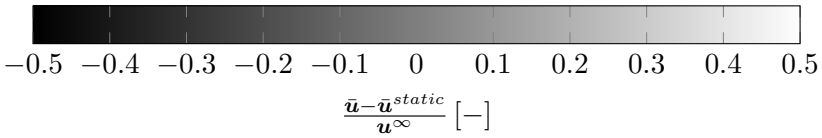
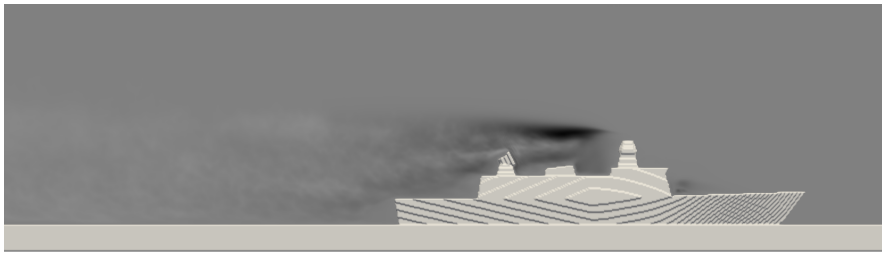
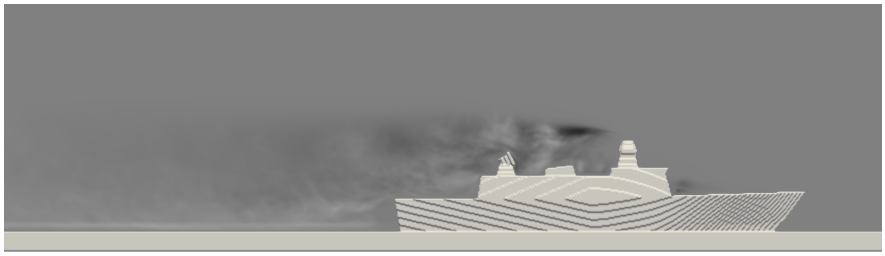


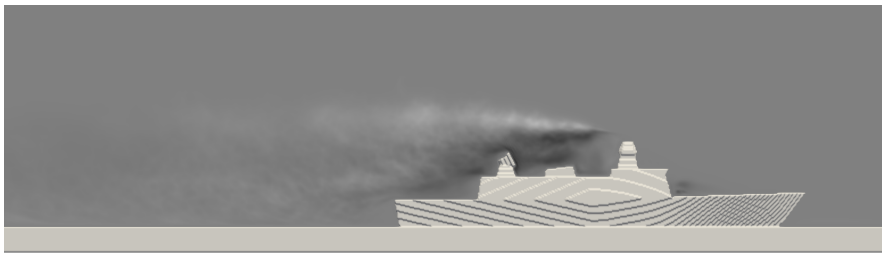
Figure III: Difference of mean velocity $\bar{\boldsymbol{u}}$ to static case $\bar{\boldsymbol{u}}^{static}$ divided by reference velocity $\boldsymbol{u}^{\infty}$ for resolutions $N^{NatoGD} = 364$. 1/50th-scale model heaving at a sine wave with $1Hz$ and an amplitude of $0.04m$ [12]. (a) shows ensemble average at $t = 0s$, (b) at $t = 0.25s$, (c) at $t = 0.5s$, and (d) at $t = 0.75s$. Hull and ground shown as voxelized solids in grey.



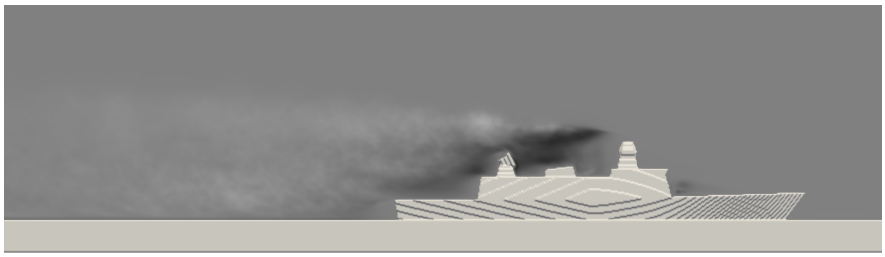
(a) $t = 0$



(b) $t = 0.25$



(c) $t = 0.5$



(d) $t = 0.75$

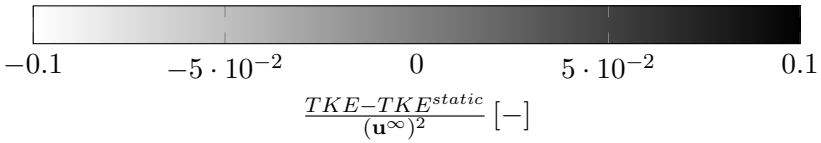


Figure IV: Difference of turbulent kinetic energy TKE to static case TKE^{static} divided by reference velocity $\boldsymbol{u}^{\infty 2}$ for resolutions $N^{NatoGD} = 364$. 1/50th-scale model heaving at a sine wave with $1Hz$ and an amplitude of $0.04m$ [12]. (a) shows ensemble average at $t = 0s$, (a) at $t = 0.25s$, (a) at $t = 0.5s$, and (a) at $t = 0.75s$. Hull and ground shown as voxelized solids in grey.

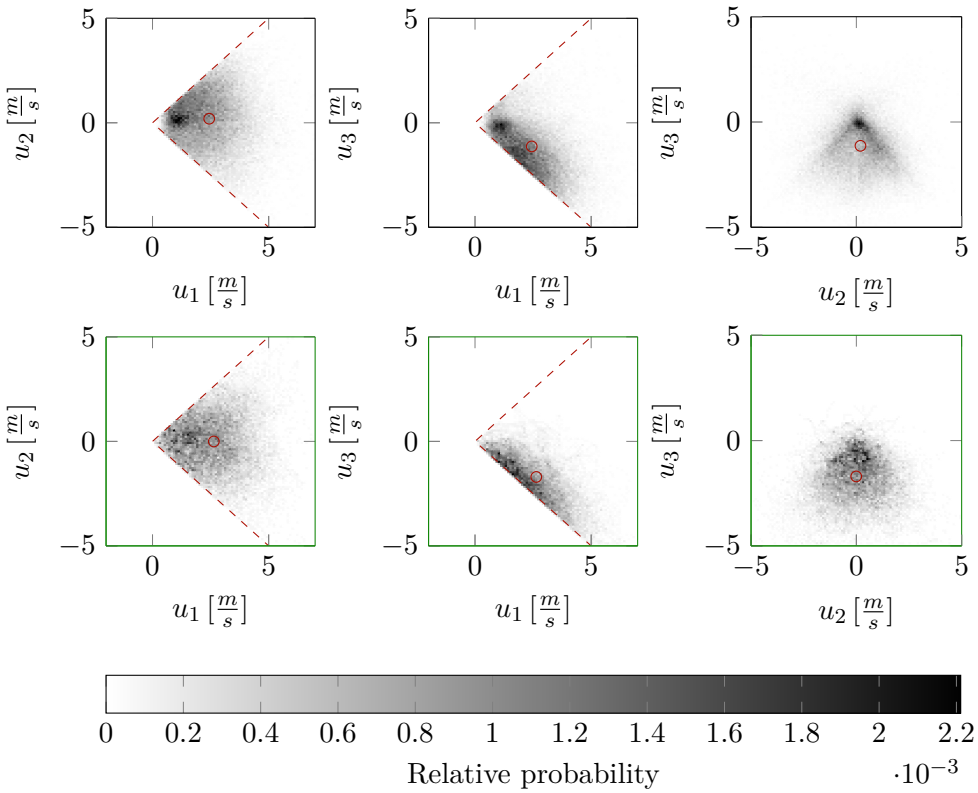


Figure V: Relative probability in bins with a length of $0.1\frac{m}{s}$ for the point $(0.3, 0, 0.07)[m]$ for a 1/50th-scale model heaving at $1Hz$ with an amplitude of $0.044m$. Red lines (- - -) mark the cone the COBRA-probes are able to measure within[12, 103, 104]. The mean is indicated in (○) First row are measurements taken from [12], second row shows calculations with resolutions $N^{NatoGD} = 364$.

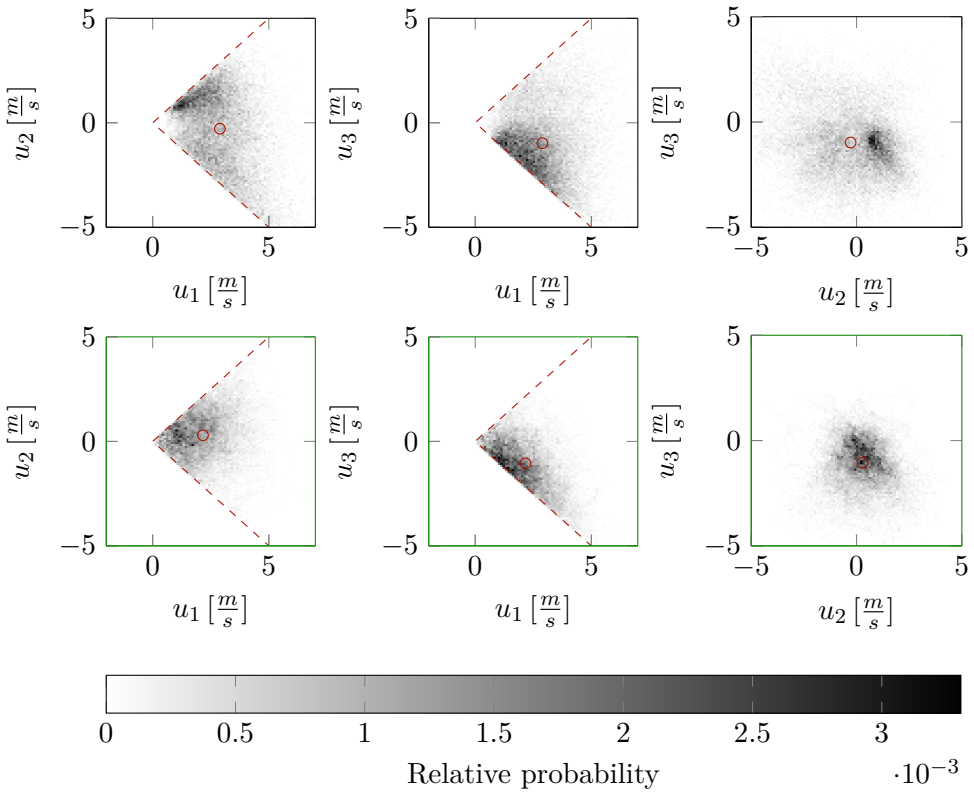


Figure VI: Relative probability in bins with a length of $0.1\frac{m}{s}$ for the point $(0.3, 0.06, 0.07)[m]$ for a 1/50th-scale model heaving at $1Hz$ with an amplitude of $0.044m$. Red lines (- - -) mark the cone the COBRA-probes are able to measure within [12, 103, 104]. The mean is indicated in (○) First row are measurements taken from [12], second row shows calculations with resolutions $N^{NatoGD} = 364$.

III Automated test testing of LBM code

Automated unit tests verify that the code written to extend OpenLB to multi GPGPUs, support ALE grids, and overset meshes. Approximately 40 unit tests check the implementation of the communication between GPGPUs, interpolation between general points in Lagrangian and Eulerian space, material maps, and Nvidia IPC access handles.

Nevertheless, the code imported from OpenLB did not come with unit tests. The implemented tests check the correctness of whole LBM grids in an attempt to implicitly verify the lower-level algorithms that form the collision and streaming step of the LRM. Therefore, a verified code base supports the simulations that use test-covered algorithms.

The unit tests reside in the institute-owned gitlab account in:

```
https://gitlab.lrz.de/HTMWTUM/olb-ripoff.git
branch: jb_dissertation
```

IV Induced velocity of an isolated rotor disk

This section shows the enlarged section between $u^{mr}_{1,cg} = 0[\frac{m}{s}]$ and $u^{mr}_{1,cg} = 10[\frac{m}{s}]$ of the Fig. 8.1 and Fig. 8.2 shown in section 8.1. The plots are provided to enhance readability but do not contain new information with respect to Fig. 8.1 and Fig. 8.2.

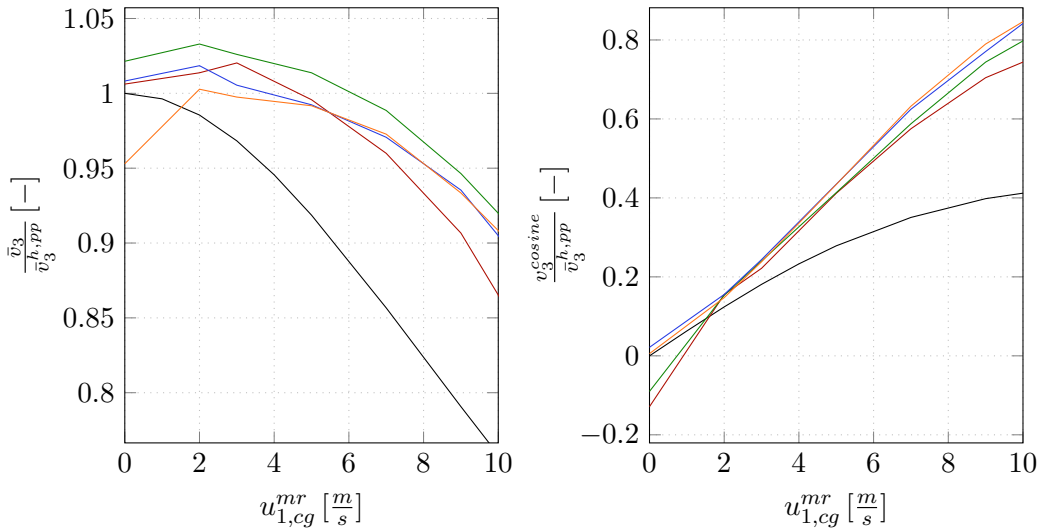


Figure VII: Influence of domain resolution on the mean $\bar{v}_3$ and cosine v_3^{cosine} component of the induced velocity. Calculated at a domain size of $a = 1.25$. Resolution $N^{LRM} = 32$ (—), $N^{LRM} = 64$ (—), $N^{LRM} = 128$ (—), and $N^{LRM} = 256$ (—). Pitt-Peters solution for comparison (—).

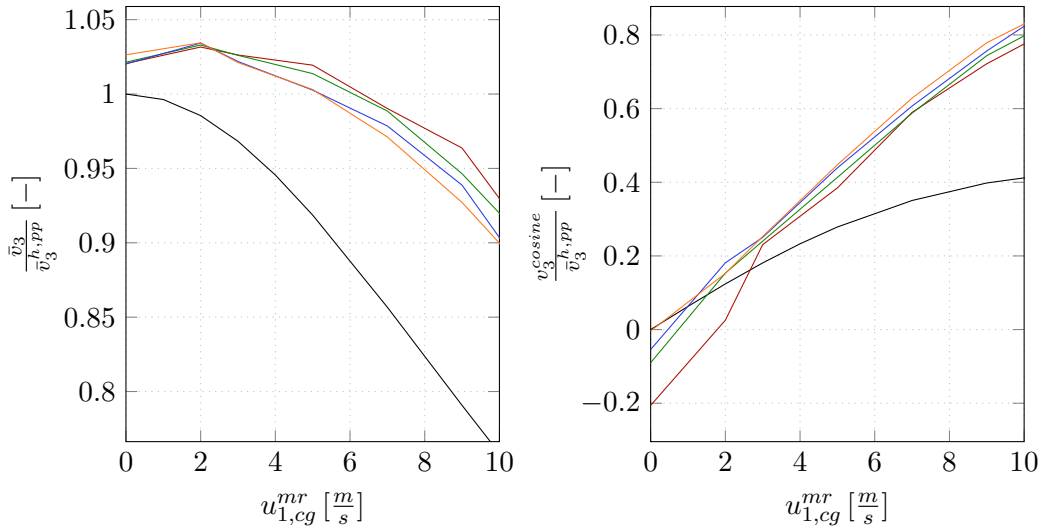


Figure VIII: Influence of domain size on the mean $\bar{v}_3$ and cosine v_3^{cosine} component of the induced velocity. Calculated at a resolution of $N^{LRM} = 64$. Domain reach factors $a = 1$ (—), $a = 1.25$ (—), $a = 1.5$ (—), and $a = 2$ (—). Pitt-Peters solution for comparison (—).

V Lateral control input with suspected correction

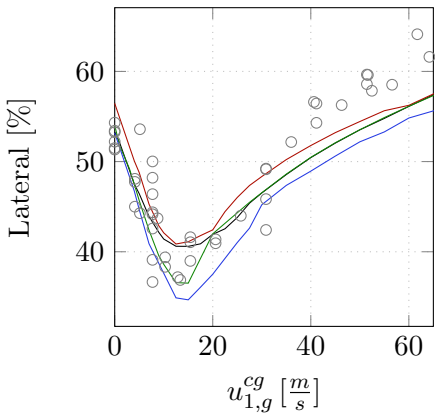


Figure IX: Lateral cyclic input in trimmed forward flight obtained with the LRM at reach factor $a = 1.25$ with resolution $N^{LRM} = 32$ (—), $N^{LRM} = 64$ (—), and $N^{LRM} = 128$ (—). A suspected 10% correction is applied to the flight test data. Pitt-Peters model (—), and flight test measurements (○) for comparison.

VI Dynamic response of MBB Bo 105 to step inputs in longitudinal cyclic and collective

Figure X shows the response of the rotorcraft flight dynamics model using the LRM and Pitt-Peters inflow model for a doublet push in the longitudinal cyclic at $u_{1,g}^{cg} = 12.86[\frac{m}{s}]$. The figure adheres to the ordering described in section 8.2.2. As the doublet is in the longitudinal cyclic, the pitch-axis represents the on-axis response while roll and yaw-axis represent the off-axis responses.

In the on-axis rate, q_{cg}^{cg} the LRM agrees well with the flight test data during the complete doublet. Nevertheless, the predictions using the reference Pitt-Peters model agree better with the flight test measurements. The same assessment can be transferred on the corresponding attitude Θ. Concerning the off-axis response in the roll-axis, the LRM roughly predicts the shape the Pitt-Peters reference predicts. Nevertheless, deviations in both halves of the doublet are present. Overall, neither of the two models is in good agreement with the flight test data. This also holds for the roll angle Φ as expected due to being related to p_{cg}^{cg} by integration. The off-axis response in the yaw-axis is predicted well by both the LRM and the Pitt-Peters reference. Nevertheless, above $t = 3s$, the yaw angle predicted by the LRM deviates from the reference and flight test data for resolutions $N^{LRM} \geq 64$.

The rate of descent of the LRM and the Pitt-Peters reference model shown in the fourth row of Fig. X agree well but fail to predict the values measured in the flight test. As the pitch angle Θ is well predicted by both models, this difference can not be attributed to errors in the pitch attitude. Furthermore, the constant rate of descent at the start of the test indicates that no false trim in vertical velocity was present in the flight test. To this point, the differences can not be attributed to specific causes. As in the comparison for the doublet input in the roll-axis, the double in pitch-axis creates a main rotor rotational speed difference of $5 \pm 1\%$. The required main rotor power during the double shows offsets in power as expected from the inflow and power comparison in section 8.2.1.

In general, the predictions of the LRM for the doublet push in the longitudinal cyclic agree with the flight test data comparable to the doublet push in the lateral cyclic shown in Fig. 8.8. They thus support the verification of the requirement as stated in section 8.2.2.

Figure XI plots the response of the rotorcraft flight dynamics model computed with the LRM and Pitt-Peters inflow model for a doublet push in the collective at $u_{1,g}^{cg} = 7.72[\frac{m}{s}]$. The figure adheres to the ordering described in section 8.2.2 except the pedal input is switched for the collective. During the double push, the pedal input is kept constant. As the doublet is in the collective, the yaw-axis represents the on-axis response while roll and pitch-axis represent the off-axis responses.

Comparing the on-axis response, both LRM and Pitt-Peters predict similar values for r_{cg}^{cg} in response to the doublet in collective, but neither predicts the flight test measurements. This might be caused by the yaw rate r_{cg}^{cg} not being trimmed to ≈ 0 at $t = 0$ in the flight test. Furthermore, the yaw angle Ψ is not trimmed to 0 at the start

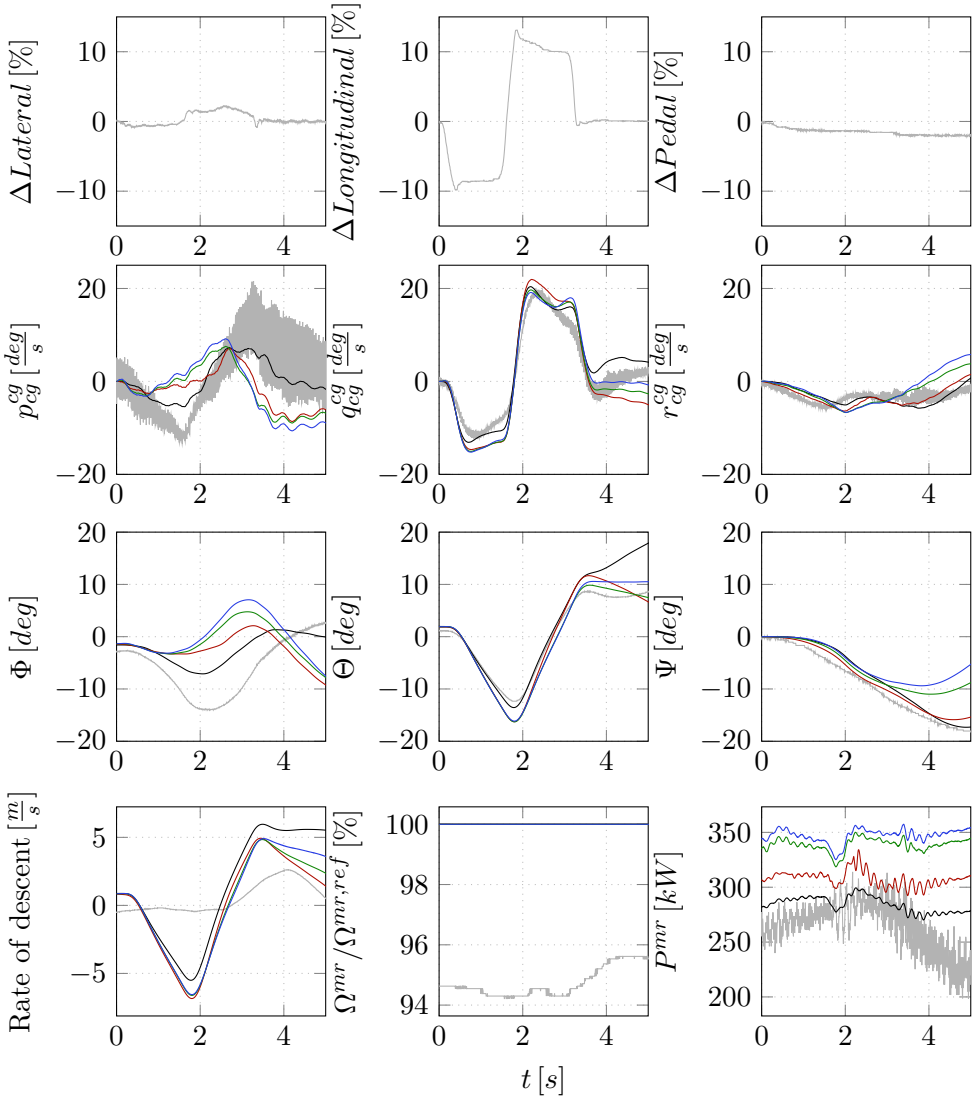


Figure X: Reaction of the helicopter to a doublet push in the longitudinal cyclic at $u_{1,g}^{cg} = 12.86\,[\frac{m}{s}]$ using the LRM with $N^{LRM} = 32$ (—), $N^{LRM} = 64$ (—), $N^{LRM} = 128$ (—) at $a = 1.25$, and the Pitt-Peters model (—). Flight test measurements (—) for comparison. Pilot inputs at constant collective in the first row, angular velocities in the *cg*-system, $p_{cg}^{cg}, q_{cg}^{cg}, r_{cg}^{cg}$, in the second row, and attitudes, Φ, Θ, Ψ, in the third row. Rate of descent and main rotor relative rotational speed $\Omega^{mr}/\Omega^{mr,ref}$, together with the required main rotor power, P^{mr}, in the last row.

of the flight test either. This means the MBB Bo 105 is yawing in flight which causes the relative velocity of rotorcraft to air to incide at a slight Ψ angle influencing, among other parameters, inflow and aerodynamic forces of the fuselage.

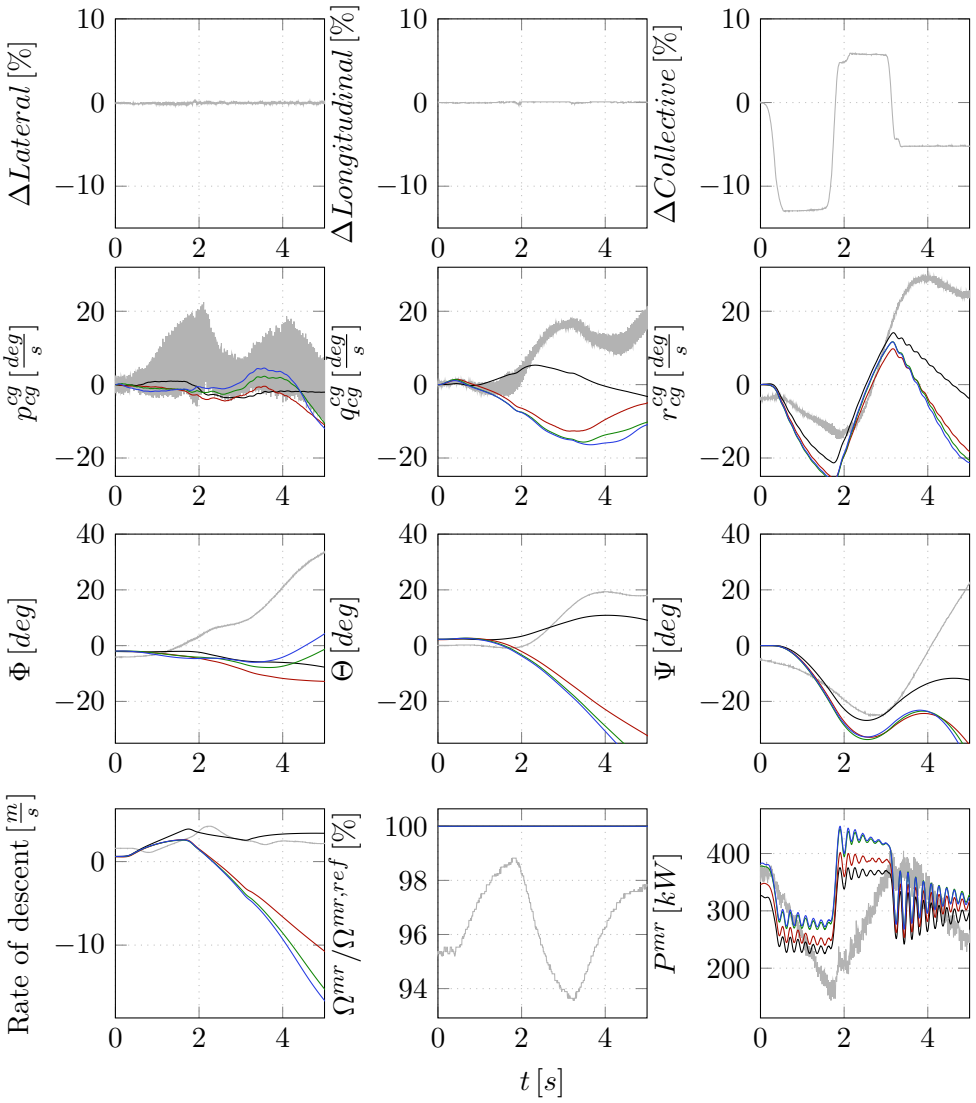


Figure XI: Reaction of the helicopter to a doublet push in the collective at $u_{1,g}^{cg} = 7.72\,[\frac{m}{s}]$ using the LRM with $N^{LRM} = 32$ (—), $N^{LRM} = 64$ (—), $N^{LRM} = 128$ (—) at $a = 1.25$, and the Pitt-Peters model with (—). Flight test measurements (—) for comparison. Pilot inputs at constant pedal in the first row, angular velocities in the *cg*-system, $p_{cg}^{cg}, q_{cg}^{cg}, r_{cg}^{cg}$, in the second row, attitudes, Φ, Θ, Ψ, in the third row. Rate of descent and main rotor relative rotational speed $\Omega^{mr}/\Omega^{mr,ref}$, together with the required main rotor power, P^{mr}, in the last row.

Both off-axis responses of the LRM do not agree with the flight test data. Nevertheless, this might be attributed to the previously described yaw of the MBB Bo 105 in this flight test. In the roll-axis, the LRM agrees well with the Pitt-Peters reference for

both rate p_{cg}^{cg} and attitude Φ. In contrast, for the off-axis response in the pitch-axis, the LRM predicts different rates q_{cg}^{cg} and attitudes Θ compared to the Pitt-Peters reference.

Interestingly, both LRM and Pitt-Peters predict rates of descent that agree with the flight test measurements for the first half of the doublet. In the second half, the LRM predicts a decreasing rate of descent while both Pitt-Peters and flight test data remain at low positive rates. This is attributed to the difference in pitch angle Θ between LRM and Pitt-Peters reference or flight test measurements in the second half of the doublet.

In comparison of the main rotor power and rotational speed, the flight test registered a large variation in rotational speed of $\approx 5\%$ that is in phase with the doublet. This results from the power required from the rotor varying according to the collective position. The P^{mr} is calculated from the rotor shaft moment and the rotational speed. Therefore, with the engines operating at constant power the difference between supplied and required power produces the variations in the main rotor speed Ω^{mr}. As the engine model used in the rotorcraft flight dynamics model always produces the required power to maintain Ω^{mr} at $\Omega^{mr,ref}$, it varies in accordance with the collective position.

In general, the reaction of the LRM to the doublet push in the collective is considered reasonably adequate in comparison to flight test data. Unfortunately, no comprehensive explanation for the differences in the pitch-axis can be identified. Nevertheless, it is suspected, that the incomplete trim in the flight test causes most of the observed differences. To this point, the results shown in Fig. X and XI support the verification of the requirement on predicting rotorcraft dynamic behavior as set in section 8.2.2.

VII Amplitude of wavelet analysis during the approach

Figure XII depicts the result of the wavelet transform that is shown in Fig. 9.5 but restricts itself to the frequency band of 0.02 to $2Hz$. Furthermore, the greyscale depicting the relative amplitude shows the range of $0 - 25\%$. Therefore, the dotted pattern in $C^{T,aer}$, mentioned in chapter 9.2, is visible.

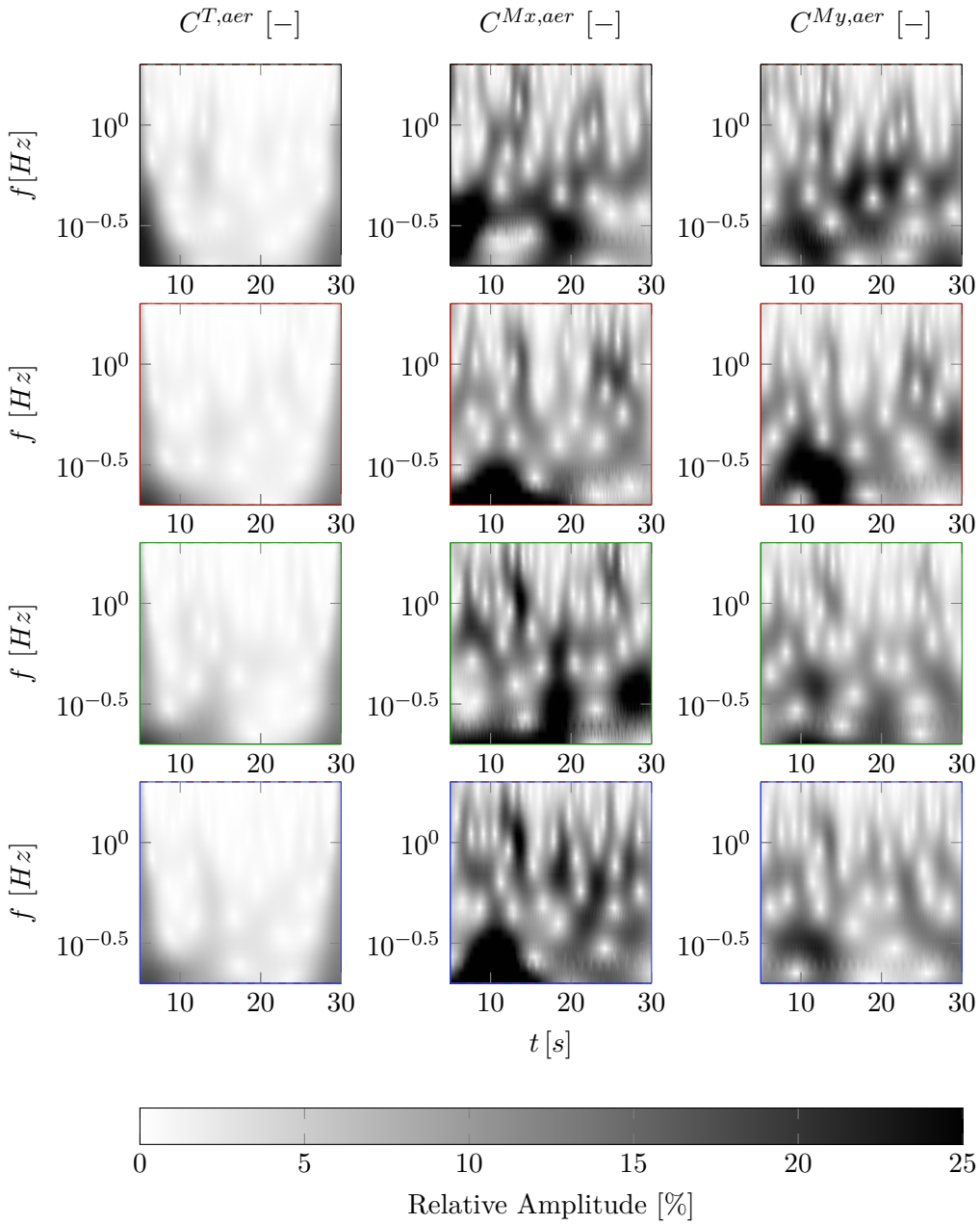


Figure XII: Amplitude of different frequencies of $C^{T,aer}$, $C^{Mx,aer}$, and $C^{My,aer}$ during the approach in the 0.02 to $2Hz$ band. Calculated by a continuous wavelet transform using complex Morlet wavelets with a center frequency and bandwidth of $1Hz$. First row shows the results of the overset simulation in the (——) axis, second to last row show calculations with varying reach factor at $N^{LRM} = 64$. $a = 1.25$ in the (——) axis, $a = 1.5$ in the (——) axis, and $a = 2$ in the (——) axis from top to bottom.

VIII Importance of ship wake and ship geometry for the ship-rotorcraft dynamic interface

Figure XIII depicts the time series of ΔC_T^{aer}, ΔC_x^{aer}, and ΔC_y^{aer} during the approach described in chapter 9.1 with and without the ship wake. Without the ship wake, the velocity is set to $\boldsymbol{u}^{\infty} = 20.58\frac{m}{s}$.

As long as the rotorcraft flies behind the ship's deck, the ship and rotorcraft do not interact. The interaction starts when the front of the rotor is above the ship deck for $t > 11s$ but is small compared to the case with the ship wake present. After the rotorcraft stops above the landing deck at $t = 18s$, the ship and rotorcraft interact strongly even if no ship wake is present. This results from the interaction of the ship deck with the rotor wake. Nevertheless, the generated fluctuations in C_T^{aer}, C_x^{aer}, and C_y^{aer} surpass the ones observed with the ship wake present.

This comparison highlights the importance of capturing the ship and rotorcraft wake and their interaction. An isolated but moving ship geometry creates different interactional loads on the rotor than a moving ship with a corresponding wake.

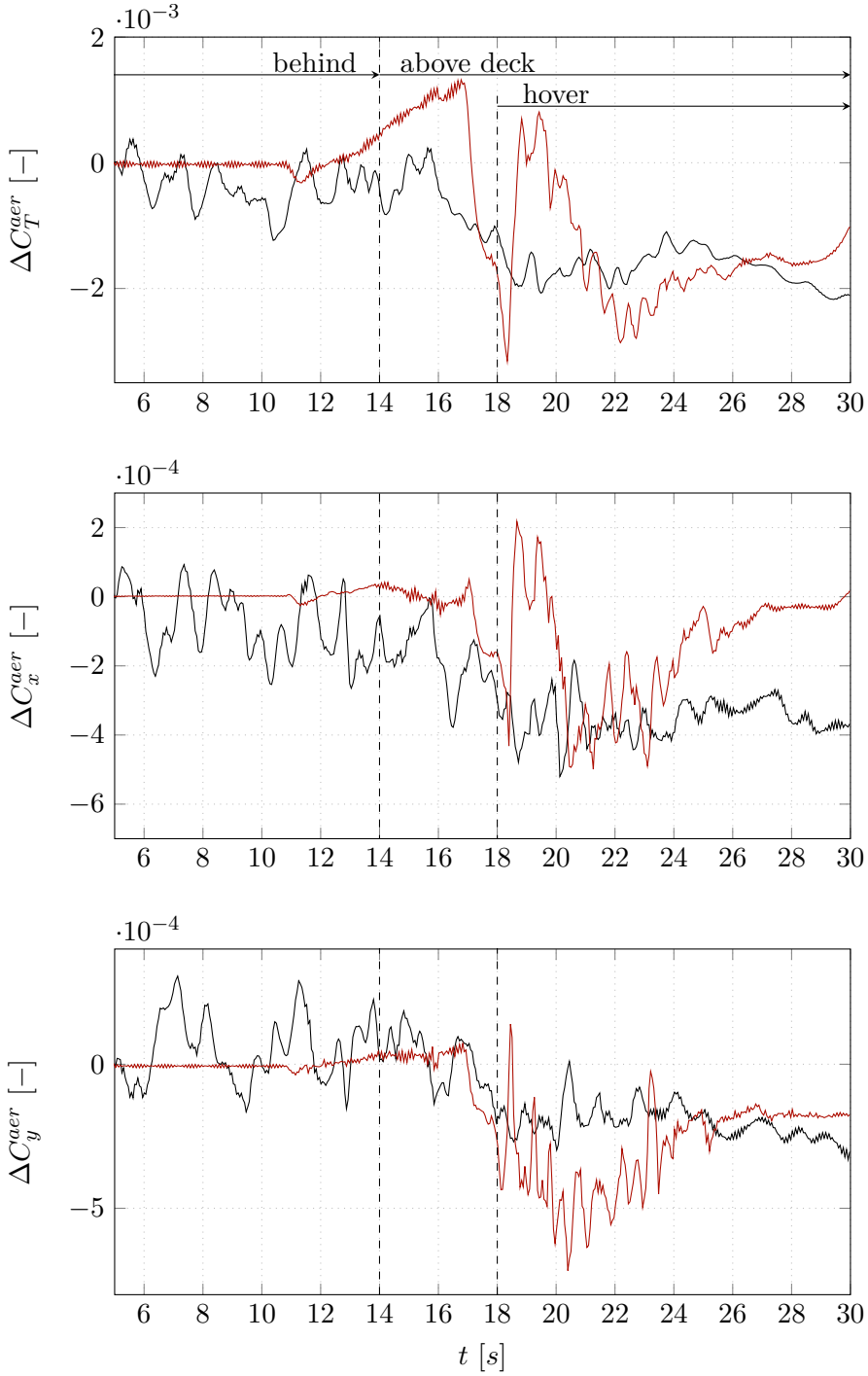


Figure XIII: Time series of ΔC_T^{aer}, ΔC_x^{aer}, and ΔC_y^{aer} during the approach. The reference points for the Δ are the respective values at $t = 5s$. Results of the LRM using $N^{LRM} = 64$, and $a = 1.25$ with ship wake and geometry in (—). (—) shows the LRM with the same resolution, reach factor and ship geometry but without the ship wake. Instead the velocity is set to $\boldsymbol{u}^{\infty} = 20.58\frac{m}{s}$ outside the LRM domain.